

EKONOMIKA INFORMATIKA

vedecký časopis FHI EU v Bratislave a SSHI

1

2025

Ročník XXIII.



- **hospodárska informatika**
- **účtovníctvo a auditorstvo**
- **ekonometria a operačný výskum**
- **aplikovaná štatistika**
- **aktuárstvo**

Vydavateľ

Fakulta hospodárskej informatiky Ekonomickej univerzity v Bratislave
a Slovenská spoločnosť pre hospodársku informatiku

IČO vydavateľa 00 399 957

Redakčná rada

Erik Šoltés - predseda

Ekonomická univerzita v Bratislave

Nenad Bjelić

University of Belgrade

Ivan Brezina

Ekonomická univerzita v Bratislave

Tatiana Čorejová

Žilinská univerzita v Žiline

Ferdinand Daňo

Ekonomická univerzita v Bratislave

Christopher D. Daykin

Government Actuary's Department, London

Dana Dluhošová

Vysoká škola báňská - Technická univerzita Ostrava

Richard Farkaš

KPMG Slovensko, spol. s r.o.

Richard Hindls

Vysoká škola ekonomická v Praze

Josef Jablonský

Vysoká škola ekonomická v Praze

Václav Janeček

Univerzita Hradec Králové

Luboš Marek

Vysoká škola ekonomická v Praze

Karol Matiaško

Žilinská univerzita v Žiline

Ladislav Mejzlík

Vysoká škola ekonomická v Praze

Józef Pociecha

Cracow University of Economics

Vincent Šoltés

Technická univerzita v Košiciach

Paweł Ulman

Cracow University of Economics

Gejza Wimmer

Univerzita Mateja Bela v Banskej Bystrici

Emin Zeytinoğlu

Kütahya Dumlupınar University

Marcela Žárová

Vysoká škola ekonomická v Praze

Výkonná rada

Petra Krišková - manažér

Ekonomická univerzita v Bratislave

Michaela Chocholatá

Ekonomická univerzita v Bratislave

Michal Páleš

Ekonomická univerzita v Bratislave

Marian Reiff

Ekonomická univerzita v Bratislave

Yuliia Serpeninova

Ekonomická univerzita v Bratislave

Peter Schmidt

Ekonomická univerzita v Bratislave

Mária Vojtková

Ekonomická univerzita v Bratislave

Redaktorka: Eva Čerteková

Adresa redakcie: Fakulta hospodárskej informatiky, Ekonomická univerzita v Bratislave

Dolnozemska cesta 1, 852 35 Bratislava

tel.: 02/6729 5723, e-mail: eva.certekova@euba.sk

Dátum vydania periodickej tlače: jún 2025

ISSN 1339-987X (online)

ISSN 1336-3514 (online vydanie)

OBSAH 1/2025

VEDECKÉ STATE A DISKUSIE

Nina Barčáková RÔZNE VÝPOČTY VÁH KRITÉRIÍ PRE METÓDU PROMETHEE PRI VÝBERE VHODNEJ DOVOLENKOVEJ DESTINÁCIE	5
Andrej Bednařík FUZZY LOGIKA V PREDSPRACOVANÍ ÚDAJOV A JEJ VPLYV NA VÝKONNOSŤ MODELU STROJOVÉHO UČENIA XGBOOST	22
Daniel Dudek DYNAMIC FINANCIAL WITHDRAWAL PLANNING FROM INTEREST- BEARING ASSETS	30
Ján Gogola OPTIMÁLNE ZAISTENIE POISŤOVATEĽA Z POHĽADU TEÓRIE RIZIKA	45
Slávka Jalčová, Jozef Koyš, Miroslav Hudec ANALÝZA KVAPÔČKOVÝCH ZAVLAŽOVACÍCH SYSTÉMOV NA PRINCÍPE FUZZY INFERENČNÉHO MECHANIZMU	56
Pavol Jurík, Natália Babjaková DETAILNÉ POROVNANIE VYBRANÝCH APLIKÁCIÍ NA PODPORU TÍMOVEJ KOMUNIKÁCIE VYUŽITEĽNÝCH V PRACOVNOM PROSTREDÍ	68
Lívia Krajčíková REGIONAL DISPARITIES IN HOUSEHOLD CONSUMPTION EXPENDITURE IN SLOVAKIA	79
Richard Martinus DETERMINING INVESTOR'S RISK GROUP USING FUZZY LOGIC UNDER THE UCITS	93
Richard Martinus MACHINE LEARNING METHODS APPLIED FOR INVESTMENT STRATEGIES	105
Peter Schmidt, Silvia Zelinová, Eva Rakovská POKROČILÉ MODELOVANIE RIZIKA V POVINNOM ZMLUVNOM POISTENÍ: VYUŽITIE ŠTATISTICKÝCH METÓD A STROJOVÉHO UČENIA	120

Pavol Sojka 131
**POROVNANIE EXEKUČNEJ EFEKTÍVNOSTI PROGRAMOVACÍCH
JAZYKOV PYTHON A JAVA PRI SPRACOVANÍ DÁT**

EXTERNÍ RECENZENTI 139

Rôzne výpočty váh kritérií pre metódu Promethee pri výbere vhodnej dovolenkovej destinácie

Different weight calculations for the Promethee method in selecting the suitable holiday destination

Nina Barčáková¹

Abstrakt

Tento článok sa zaoberá problematikou viackriteriálneho rozhodovania, ktoré je dôležitým nástrojom pre riešenie rozhodovacích problémov. V článku bude poskytnutý všeobecný prehľad metód na výpočty váh kritérií, všeobecná formulácia metódy Promethee a jej výpočtu. Konkrétne sa zameriame na výpočty váh v rôznych metódach, ktoré neskôr vo výpočtoch použijeme práve v metóde Promethee, ktorá nám zabezpečí výsledné odporúčania. Tento prístup bude potom aplikovaný na riešenie experimentálnych príkladov, kde je demonštrovaná jeho efektívnosť a flexibilita. Tento komplexný pohľad na problematiku viackriteriálneho rozhodovania v sebe nesie cieľ poskytnúť ucelené porozumenie rôznorodých metód a prístupov k riešeniu rozhodovacích problémov, pričom hlavným cieľom článku je ukážka aplikácií viackriteriálneho rozhodovania a výpočtov váh v odvetví hotelierstva alebo cestovného ruchu.

Kľúčové slová

Viackriteriálne rozhodovanie, Promethee, metóda smerodajnej odchýlky, metóda entropie, metóda CRITIC

Abstract

This paper deals with issue of multi-criteria decision-making, which is an important tool for solving decision-making problems. The paper provides a general overview of methods for calculating criterion weights, a general formulation of the Promethee method, and its calculations. Specifically, it focuses on the calculation of weights using various methods, which are later employed in the Promethee method to generate final recommendations. This approach is then applied to experimental examples to demonstrate its efficiency and flexibility. This comprehensive perspective on multi-criteria decision-making aims to provide an integrated understanding of diverse methods and approaches to solving decision-making problems, with the main goal of the article being to demonstrate the applications of multi-criteria decision-making and weight calculation in the hospitality or tourism.

Key words

Multi-criteria decision-making, Promethee, standard deviation method, entropy method, CRITIC method

JEL classification

C00, C6

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35 Bratislava, nina.barcakova@euba.sk.

1 Úvod

Viakriteriálne rozhodovanie je nástroj na výber najvhodnejšej alternatívy spomedzi viacerých, často využívaný v každodenných situáciách aj pracovnej sfére. Takýto výber dostaneme na základe metód viackriteriálneho rozhodovania (De Keyser, 1996).

Hodnotenie alternatív môžeme registrovať v každodenných situáciách, či už pri nakupovaní tovarov a služieb, zvolením najlepšej dovolenky alebo investovaním svojich peňažných prostriedkov do podielových fondov. V spomínaných prípadoch svoje rozhodnutia vykonávame práve na základe našej intuície a práve na základe nej vykonávame porovnania kritérií a prechádzame na výber najvhodnejšej alternatívy (Abdullah, 2018). V rozhodovaní sa používa intuícia, no v odborných prípadoch môžu nesprávne rozhodnutia spôsobiť negatívne dôsledky, odporúča sa využiť metódy ako Promethee, AHP a TOPSIS (Fiala, 1994).

Práca sa sústreďuje na metódu Promethee a výpočet váh pomocou metód entropie, smerodajnej odchýlky a CRITIC - „Criteria importance through inter-criteria“. Tieto metódy určujú váhy kritérií na základe rovnomernosti rozloženia údajov, rozptylu hodnôt či poradia a rozpätia hodnôt kritérií. Váhy sa následne aplikujú v metóde Promethee na hodnotenie alternatív. V závere sa hodnotí vplyv výpočtu váh na výsledky, porovnávajú sa metódy a navrhujú sa možnosti ich vylepšenia. Cieľom je zhodnotiť presnosť a efektívnosť metód pri riešení rozhodovacích problémov.

2 Viackriteriálny problém

Takmer všetky problémy, s ktorými sa každodenne stretávame, majú viackriteriálny charakter. Nie je rozumné rozhodovať sa iba na základe jediného kritéria, vhodnejšie je pozrieť sa na problém z viacerých uhlov pohľadu (Brans, 1998). Či už je to kúpa auta, pri ktorej zvažujeme nielen cenu, ale aj spotrebu, komfort, bezpečnosť, či ekologické parametre, alebo sa jedná o výber lokality pre výstavbu, kedy je dôležitá cena pozemku, rozloha, investičné náklady spojené s výstavbou, vplyv na životné prostredie a mnohé ďalšie aspekty. Prirodzeným očakávaním je nájdenie alternatívy, ktorá by bola optimálna vo všetkých kritériách, to je často nemožné, pretože takáto alternatíva nemusí existovať (Brans, 2005). Uvažujme nasledujúci viackriteriálny problém:

$$\max\{g_1(a), g_2(a), \dots, g_k(a) / a \in A\} \quad (1)$$

Kde A je konečná množina možných alternatív výberu $A = \{a_1, a_2, \dots, a_n\}$ a $\{g_1(\cdot), g_2(\cdot), \dots, g_k(\cdot)\}$ je množina hodnotiacich kritérií. Môžeme predpokladať, že niektoré kritériá chceme maximalizovať, kým iné minimalizovať (Brans, 2005). Základné dáta viackriteriálneho problému sa zapisujú do hodnotiacej tabuľky (Brans, 1998). Tab. 1 prezentuje dáta a ich uloženie v rámci tabuľky. Riešenie však nezávisí iba od dát, ktoré sa v tejto tabuľke nachádzajú, ale taktiež aj od toho, kto sa rozhoduje. Neexistuje riešenie, ktoré by vyhovovalo všetkým rovnako. Najlepšie kompromisné riešenie viackriteriálneho problému teda závisí aj od individuálnych preferencií (Brans, 2005).

Tab. 1: Hodnotiaca tabuľka

	$g_1(\cdot)$	$g_2(\cdot)$...	$g_j(\cdot)$...	$g_k(\cdot)$
a_1	$g_1(a_1)$	$g_2(a_1)$...	$g_j(a_1)$...	$g_k(a_1)$
a_2	$g_1(a_2)$	$g_2(a_2)$...	$g_j(a_2)$...	$g_k(a_2)$
:	:	:	:	:	:	:
a_i	$g_1(a_i)$	$g_2(a_i)$...	$g_j(a_i)$...	$g_k(a_i)$
:	:	:	:	:	:	:
a_n	$g_1(a_n)$	$g_2(a_n)$...	$g_j(a_n)$...	$g_k(a_n)$

Zdroj: Vlastné spracovanie

Alternatívu označujeme za preferovanú, pokiaľ je ohodnotená aspoň tak dobre ako iná vo všetkých kritériách. Nemožnosť porovnania dvoch alternatív pozorujeme, pokiaľ bez ďalších informácií nevieme rozhodnúť, ktorá z nich je výhodnejšia. Táto situácia nastane v prípade, ak je jedna z alternatív lepšia v jednom kritériu, druhá je lepšia v inom (Brans, 2005).

Väčšinou sa pri riešení viackriteriálneho problému stretávame s alternatívami, ktoré sú neporovnateľné, a preto sú pre nájdenie riešenia potrebné dodatočné informácie. Môžu byť poskytnuté napr. nasledujúcimi spôsobmi: váhy pridelujúce relatívnu dôležitosť každému kritériu, preferencie naviazané na každý pár alternatív v rámci každého kritéria, vytvorenie jendnokriteriálneho problému, pre ktorý existuje riešenie či hranice určujúce limity preferencie

Existuje množstvo metód viackriteriálnej analýzy. Všetky vychádzajú z hodnotiacej tabuľky, ale líšia sa dodatočnými informáciami, ktoré sú pre ich aplikáciu potrebné. Metódy triedy Promethee vyžadujú jasné a ľahko získateľné informácie (Brožová, 2003).

3 Metódy triedy Promethee

Problémy vyhodnocovania viacerých variantov riešia metódy triedy Promethee (Mlynarovič, 1998). Jej prvé prezentovanie bolo na konferencii na univerzite v Kanade. V tomto období obsahovala iba základné Promethee I – čiastočné usporiadanie alternatív, Promethee II – úplné usporiadanie alternatív. Existujú aj iné varianty triedy Promethee, o ktorých sa viac možno dozvedieť vo svetovej literatúre v rámci viackriteriálneho rozhodovania (Zapletal, 2023).

Pri využití danej metódy môžu byť stanovené rôzne ciele: identifikácia najlepšieho možného rozhodnutia; usporiadanie možností od najlepšej po najhoršiu; rozdelenie položiek do skupín, ktoré sme si stanovili, napr. nízka cena, stredná cena, vysoká cena; vizualizácia hodnotiaceho alebo rozhodovacieho problému. Európska škola viackriteriálnej analýzy navrhla základ daných metód, ktorý spočíva v analýze viackriteriálnej preferencie a zovšeobecnených kritérií. Daná analýza poskytuje dostatočné informácie avšak má i slabú stránku, ktorou je výrazný vplyv hodnotiteľ, ktorého osobné preferencie častokrát majú výrazný vplyv pri výbere najlepšieho z daných variantov.

Váhy kritérií, ktoré sú dôležité pri výbere najlepšej alternatívy sú zadávané hodnotiteľom. Z tohto dôvodu okrem použitej metódy zohráva pri danej metóde dôležitú úlohu ľudský faktor, ktorý určí dôležitosť jednotlivým kritériám, prostredníctvom čoho nakoniec vznikne konečné riešenie (Dubecová, 2009). Tieto metódy však zabezpečujú väčšiu transparentnosť, najmä pokiaľ hodnotiteľ nevie o konkrétnej ponuke ešte pred stanovením váh jednotlivých kritérií (Brans, 1985). Na výber alternatív zo všetkých, je potrebné disponovať dostatočným množstvom informácií. Práve to zohráva dôležitú úlohu aj pri metóde Promethee, kde sa stretávame s informáciami medzi kritériami a v rámci kritérií.

Informácie medzi kritériami

Množina ($w_j, j = 1, 2, \dots, k$) predstavuje váhy relatívnej dôležitosti, ktoré sú dopĺňané do hodnotiacej tabuľky. Každému kritériu odpovedá nejaká váha, kde vyššia hodnota znamená väčšiu dôležitosť kritéria, $w_j \geq 0$ a platí (Brans, 2005)

$$\sum_{j=1}^k w_j = 1 \quad (2)$$

Informácie v rámci kritérií

Metódy triedy Promethee sa zakladajú na postupnom párovom porovnávaní variantov z hľadiska všetkých kritérií. Výsledok je vyjadrenie intenzity preferencie medzi dvoma variantami, (Brans, 2005)

$$d_j(a, b) = g_j(a) - g_j(b) \quad (3)$$

kde

$g_j(a)$ – hodnota kritéria j pre alternatívu a

$g_j(b)$ - hodnota toho istého kritéria pre alternatívu b

$d_j(a, b)$ - o koľko je alternatíva a lepšia (alebo horšia) než b z pohľadu kritéria j

Na základe rozdielu vieme určiť preferenciu alternatív. Ak je rozdiel zanedbateľný, žiadna z alternatív nie je preferovaná. Ak rozdiel ukazuje výrazné rozdiely aj preferencia je vyššia. Je potrebné pre každé kritérium určiť **preferenčnú funkciu** $P_j(a, b)$ a funkciu F_j , ktorá rozdiel $d_j(a, b)$ premení na **preferenčné skóre** v rozsahu typicky medzi 0 a 1 (Brans, 2005).

$$P_j(a, b) = F_j[d_j(a, b)] \quad \text{pre } a, b \in A \quad (4)$$

Platí

$$0 \leq P_j(a, b) \leq 1 \quad (5)$$

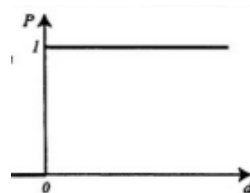
3.1 Typy preferenčných funkcií

Poznáme 6 typov preferenčných funkcií. Niektoré obsahujú parametre (Tzeng, 2011), ktoré potrebujeme poznať pri vybraných funkciách pre výpočty, ako napr. parameter q (prah indiferencie) - max hodnotu rozdielu medzi alternatívami. Parameter p (prah preferencie) - min hodnotu rozdielu medzi alternatívami. Parameter s – inflexný bod.

1. Obyčajná preferenčná funkcia (jednoduchá PF) – nie je za potreby zadávanie parametrov. Je vhodná hlavne vtedy, ak akýkoľvek rozdiel v hodnotení alternatív vedie k absolútnej preferencii (Zapletal, 2023).

$$P(d) = \begin{cases} 0 & d \leq 0 \\ 1 & d > 0 \end{cases} \quad (6)$$

Obr. 1 – Grafické znázornenie obvyčajnej preferenčnej funkcie

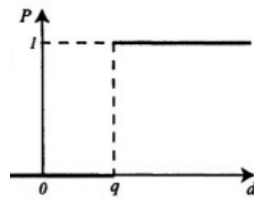


Zdroj: Brans, 2005

2. Kvázii preferenčná funkcia (U-jednoduchá PF) – obsahuje indiferenčnú oblasť. Táto funkcia sa používa vtedy, keď sú ohodnotenia vyjadrené v nejakej ordinálnej škále (napr. od 1-10) (Zapletal, 2023).

$$P(d) = \begin{cases} 0 & d \leq q \\ 1 & d > q \end{cases} \quad (7)$$

Obr. 2 – Grafické znázornenie kvázi preferenčnej funkcie

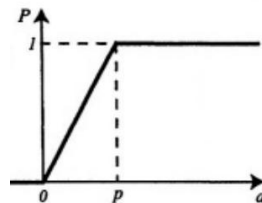


Zdroj: Brans, 2005

3. Lineárna preferenčná funkcia (V-lineárna PF) – je nutné poznať prah preferencie pre kritérium (Zapletal, 2023).

$$P(d) = \begin{cases} 0 & d \leq 0 \\ \frac{d}{p} & 0 \leq d \leq p \\ 1 & d > p \end{cases} \quad (8)$$

Obr. 3 – Grafické znázornenie lineárnej preferenčnej funkcie

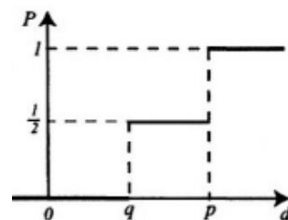


Zdroj: Brans, 2005

4. Úrovňová preferenčná funkcia (Stupňovitá PF) – je modifikáciou kvázi preferenčnej funkcie, okrem indierencie má hodnotu preferencie vo výške 0,5 (Zapletal, 2023).

$$P(d) = \begin{cases} 0 & d \leq 0 \\ \frac{1}{2} & q \leq d \leq p \\ 1 & d > p \end{cases} \quad (9)$$

Obr. 4 – Grafické znázornenie úrovňovej preferenčnej funkcie

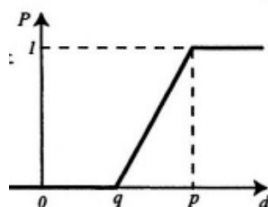


Zdroj: Brans, 2005

5. Lineárna preferenčná funkcia s indierenčnou oblasťou – je modifikácia lineárnej preferenčnej funkcie. Je potrebné poznať aj prah indierencie a aj prah preferencie (Zapletal, 2023).

$$P(d) = \begin{cases} 0 & d \leq 0 \\ \frac{d-q}{p-q} & q < d \leq p \\ 1 & d > p \end{cases} \quad (10)$$

Obr. 5– Grafické znázornenie lineárnej preferenčnej funkcie s indifferenčnou oblasťou

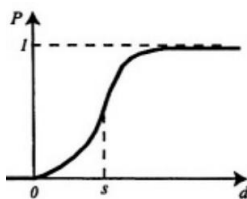


Zdroj: Brans, 2005

6. Gaussova preferenčná funkcia – obsahuje parameter s . Pri definovaní sa využíva smerodajná odchýlka ohodnotení alternatív v danom kritériu (Zapletal, 2023).

$$F(d) = \begin{cases} 0 & d \leq 0 \\ 1 - e^{-\frac{d^2}{2s^2}} & d > 0 \end{cases} \quad (11)$$

Obr. 6 – Grafické znázornenie Gaussovej preferenčnej funkcie



Zdroj: Brans, 2005

3.2 Viackriteriálny preferenčný index

Viackriteriálny preferenčný index $\pi(a, b)$ predstavuje stupeň uprednostňovania alternatív a pred alternatívou b s ohľadom na všetky kritériá. $\pi(b, a)$ vyjadruje ako je b preferovaná pred a (Brans, 2005).

$$\pi(a, b) = \sum_{j=1}^k P_j(a, b)w_j \quad (12)$$

$$\pi(b, a) = \sum_{j=1}^k P_j(b, a)w_j \quad (13)$$

Nasledujúce vlastnosti platia pre $\forall (a, b) \in A$

$$\pi(a, a) = 0 \quad (14)$$

$$0 \leq \pi(a, b) \leq 1 \quad (15)$$

$$0 \leq \pi(b, a) \leq 1 \quad (16)$$

$$0 \leq \pi(a, b) + \pi(b, a) \leq 1 \quad (17)$$

$$\pi(a, b) \sim 0 \text{ implikuje celkovú slabú preferenciu } a \text{ pred } b \quad (18)$$

$$\pi(a, b) \sim 1 \text{ implikuje celkovú silnú preferenciu } a \text{ pred } b \quad (19)$$

Každá alternatíva je porovnávaná s ostatnými $(n-1)$ alternatívami, preto uvádzame výstupný a vstupný tok, ktorého výpočet je nižšie (Brans, 2005):

$$\emptyset^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a, x) \quad (20)$$

$$\emptyset^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a) \quad (21)$$

Výstupný tok vyjadruje celkové uprednostňovanie alternatívy a pre ostatnými alternatívami. Čím je $\emptyset^+(a)$ väčšie, tým je alternatíva a hodnotená lepšie. Vstupný tok, naopak, vyjadruje ako sú ostatné alternatívy preferované pred alternatívou a . Čím je $\emptyset^-(a)$ menšie, tým je alternatíva a ohodnotená lepšie (Brans, 2005). Promethee I zodpovedá za čiastočné usporiadanie alternatív, ktoré získame z výstupného a vstupného toku. Oba toky vyjadrujú iné usporiadanie a Promethee I je ich prienikom. Metóda Promethee II poskytuje kompletne usporiadanie alternatív. Uvažujeme tzv. čistý tok, ktorý je rozdielom výstupného toku a vstupného toku. V tomto prípade možno tvrdiť, že čím je čistý tok vyšší, tým je aj alternatíva lepšia (Fiala, 1994).

$$\emptyset(a) = \emptyset^+(a) - \emptyset^-(a) \quad (22)$$

4 Výpočty váh – objektívne metódy

Pre výpočty sme si zvolili do metódy Promethee vložiť váhy určené na základe metód entropie, metód smerodajnej odchýlky a metódy CRITIC. Preto v nasledujúcom texte si bližšie popíšeme tieto metódy aby sme ich neskôr vedeli využiť pri našich výpočtoch a porovnaníach.

4.1 Metóda entropie

Prvá z objektívnych metód váhy kritérií určí na základe hodnôt rozhodovacej matice na základe miery, ktorá sa nazýva entropia. V literatúre sa môže táto miera označovať aj ako „miera neusporiadanosť“ hodnôt. Ako príklad možno uviesť, ak si dlhšie nebudeme upratovať miestnosť, tak miera jej entropie – neporiadku vzrastie. Problém je v tom, že termín neusporiadanosť je veľmi ťažko definovateľný (Zapletal, 2023).

Entropia meria množstvo informácií pri zistení výsledku náhodného pokusu. Čím je vyššia entropia kritéria, tým viac klesá hodnota jeho váhy, pretože sa varianty stávajú menej citlivé s ohľadom na toto kritérium (Zapletal, 2023). Výhodou tejto metódy je, že sa aplikuje aj na kvalitatívne dáta, čo napr. pre smerodajnú odchýlku neplatí. Algoritmus pre výpočet váh na základe danej metódy pozostáva z troch krokov (Zapletal, 2023).

V prvom kroku si najskôr normalizujeme hodnoty rozhodovacej matice. Rozlišujeme či ide o kritériá maximalizačné alebo minimalizačné. Kritérium j patrí do množiny K , ktorá reprezentuje všetky kritéria). Pre lepšiu prehľadnosť a vysvetlenie uvádzame oba vzorce (Zapletal, 2023):

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}, \text{ pre maximalizačné } j \in K \quad (23)$$

$$r_{ij} = \frac{\frac{1}{x_{ij}}}{\sum_{i=1}^n \frac{1}{x_{ij}}}, \text{ pre minimalizačné } j \in K \quad (24)$$

V druhom kroku prechádza na výpočet miery entropie kritéria j $e_j \in [0, 1]$:

$$e_j = -\frac{1}{\ln n} * \sum_{i=1}^n r_{ij} \ln r_{ij}, \text{ pre } j \in K \quad (25)$$

V poslednom kroku je normalizácia hodnôt entropie za kritéria tak, aby spĺňali podmienky, ktoré sme si pre váhy skôr stanovili a aby sa veľkosť váh nepriamoúmerne s rastúcou entropiou znižovala:

$$w_j = \frac{1 - e_j}{k - \sum_{j=1}^k e_j}, \text{ pre } j \in K \quad (26)$$

4.2 Metóda smerodajnej odchýlky

Smerodajná odchýlka je dobre známou mierou variability náhodnej premennej, ktorú možno vypočítať ako odmocninu z rozptylu. Toto vypovedá o tom ako sú hodnoty náhodnej premennej rozložené okolo strednej hodnoty (Zapletal, 2023). Čím je nižšia smerodajná odchýlka tým sú hodnoty premennej viac podobné a naopak.

Hlavnou myšlienkou metódy je, že čím smerodajná odchýlka hodnotenia variantov x pre zvolené kritérium j vyššia, tým má toto kritérium potenciálne vyšší vplyv na výsledné hodnotenie. Napríklad kúpa dovolenky - v množine variantov majú všetky dovolenky rovnakú cenu a dĺžka pobytu bude naopak veľmi rozdielna. V tomto prípade má význam uvažovať o tom, že cena dovolenky vďaka menšej hodnote smerodajnej odchýlky, by mala mať menšiu váhu (Zapletal, 2023).

Prvý krok je rovnaký ako aj pri metóde entropie a to, že si normalizujeme hodnoty x_{ij} rozhodovacej matice na r v závislosti od typu kritéria. Nasleduje výpočet smerodajnej odchýlky, ktorý nám ukazuje vzorec (27) kde r_{ij} s vlnkou predstavuje priemernú hodnotu normalizovaných hodnôt variant podľa kritéria j . Výsledné váhy sú potom vypočítané za pomoci vzorca (28), normalizovaných smerodajných odchýlok vypočítaných z r_{ij} :

$$\sigma_j \sqrt{\frac{\sum_{i=1}^n (r_{ij} - \bar{r}_j)^2}{n}} \quad (27)$$

$$w_j = \frac{\sigma_j}{\sum_{j=1}^k \sigma_j}, \text{ pre } j \in K \quad (28)$$

4.3 Metóda CRITIC

Poslednou metódou je metóda CRITIC. Rozdielnosť je v tom, že výsledná váha kritéria nezávisí iba na hodnotení variant z pohľadu tohto kritéria, ale aj na ich podobnosti hodnotení podľa ostatných kritérií. Predpokladá sa, že ak sú hodnoty variant z pohľadu kritéria j významne korelované s hodnotami podľa ostatných kritérií, kritérium j je považované za menej významné a jeho informačný prínos pre danú úlohu je oslabený a naopak. Vplyv korelácie medzi kritériami metóda navyše kombinuje tiež s použitím smerodajnej odchýlky (Zapletal, 2023). Závislosť medzi kritériami je meraná pomocou Pearsonovho korelačného koeficientu p_{ij} . Čím viac je jeho hodnota vzdialená od nuly, tým je závislosť medzi hodnotenými variantami podľa týchto kritérií silnejšia (Zapletal, 2023). Algoritmus metódy CRITIC pozostáva z troch krokov.

Prvý krok využíva normalizáciu pomocou rozpätia hodnôt s rozlíšením max a min kritérií. Z dôvodu, že pri výpočtoch využívame oba druhy vzorcov, pre lepšiu prehľadnosť ich uvádzame nižšie:

$$r_{ij} = \frac{x_{ij} - \min_i \{x_{ij}\}}{\max_i \{x_{ij}\} - \min_i \{x_{ij}\}}, \text{ pre maximalizačné } j \in K \quad (29)$$

$$r_{ij} = \frac{\max_i \{x_{ij}\} - x_{ij}}{\max_i \{x_{ij}\} - \min_i \{x_{ij}\}}, \text{ pre minimalizačné } j \in K \quad (30)$$

V druhom kroku vypočítame hodnoty Pearsonovho korelačného koeficientu, ktorý reprezentuje vzorec (31) (Zapletal, 2023). Strednú hodnotu normovaných hodnotení variant r_{ij} vyjadruje \bar{r} , za kritéria j a σ označuje smerodajnú odchýlku hodnotení. Pri výpočte váh najprv vypočítame nenormalizované hodnotenia, ktoré sú založené na korelačných koeficientoch a smerodajných odchýlok.

$$p_{ij} = \frac{\frac{\sum_{k=1}^n (r_{ki} - \bar{r}_i) \cdot \sum_{k=1}^n (r_{kj} - \bar{r}_j)}{n}}{\sigma_i \sigma_j} \quad (31)$$

$$c_j = \delta_j \sum_{l=1}^k (1 - p_{jl}), \text{pre } j \in K \quad (32)$$

Hodnota c_j udáva aká veľká časť informácie o variantoch je pokrytá j -tým kritériom. Treba si všimnúť, že čím je kritérium v priemere menej pozitívne alebo viac negatívne korelované s ostatnými, tým sa jeho informačný prínos a teda aj váha zvyšuje. V poslednom kroku si vypočítame váhy kritérií vzorcom (33):

$$w_j = \frac{c_j}{\sum_{i=1}^k c_i}, \text{pre } j \in K \quad (33)$$

Metóda CRITIC reflektuje pri hodnotení ako variabilitu hodnotenia, tak závislosti medzi kritériami, preto ju možno považovať v porovnaní so spomenutými za najdokonalejšiu, ale zároveň aj za najviac náročnú. Je dôležité si uvedomiť, že nie vždy koeficient korelácie zachytáva skutočnú funkčnú závislosť medzi porovnávanými prvkami (Zapletal, 2023).

5 Porovnanie výpočtov metódy Promethee II s použitím rôznych váh

Pri aplikovaní Promethee metódy je dôležitý výber preferenčných funkcií kritérií a ich parametrov, rovnako aj voľba váh. Tieto dodatočné informácie vyplývajú z preferencií investora a sú subjektívne, preto sme sa rozhodli využiť analýzu citlivosti. Cieľom analýzy je pozorovať zmeny v konečnom poradí získanom pomocou Promethee II vzhľadom na zmeny váh kritérií. Najdôležitejším krokom je pridelenie váh kritériám. Autori metódy Promethee neposkytujú návod na určenie váh. Odvolávajú sa na schopnosť toho, kto sa rozhoduje, poskytnúť tieto informácie. My sme v tejto práci váhy pridelovali tromi spôsobmi.

5.1 Experimentálny príklad

Na výpočet váh a ich použitie v Promethee metóde potrebuje poznať hodnotiacu tabuľku, v ktorej sa nachádzajú v riadkoch atribúty a stĺpce sú vyplnené kritériami. V nasledujúcej Tab. 2 práve tieto dôležité údaje prezentujeme. Náš experimentálny príklad je prispôbený práve na hotely v rámci dovolenkových destinácií a hľadáme najlepšiu z nich na základne požiadaviek na kritériá. Kritéria, ktoré sme si zvolili ako dôležité pre rozhodovanie sú nasledovné:

- **Cena v eurách (C1)** – v tomto príklade predstavuje cenu hotela vybranej destinácie
- **Počet hviezdíčiek (C2)** – hodnota hotela vzrastá podľa počtu hviezdíčiek. Dôležité kritérium hviezdíčky, a tým chceme zabezpečiť hotel na najvyššej úrovni.
- **Vzdialenosť (C3)** – predstavuje vzdialenosť hotela od letiska vyjadrenú v km.
- **Hodnotenie v bodoch (C4)** – hodnotenia ľudí na stránkach, toto hodnotenie je vyjadrené na bodovej škále od 0 do 110 bodov.
- **Druh stravy (C5)** – vyjadruje typ stravovania, ktoré si v hoteli zaplatíme. Medzi tieto druhy stravy možno zaradiť: 1 – raňajky, 2 – plná penzia a 3 – all inclusive

Tab. 2: Hodnotiaci tabuľka pre experimentálny príklad

	C1	C2	C3	C4	C5
(H1) Hotel A	290000	3	5,6	60	2
(H2) Hotel B	480000	5	6,2	80	1
(H3) Hotel C	510000	4	7,1	110	3
(H4) Hotel D	560000	3	5,7	90	2
(H5) Hotel E	610000	4	5,5	110	3
	MIN	MAX	MIN	MAX	MAX

Zdroj: Vlastné spracovanie

5.2 Výpočty váh

Na otestovanie metódy Promethee potrebuje v prvom rade pristúpiť k výpočtu váh kritérií, ktoré neskôr využijeme priamo pri výpočtoch a usporiadaní alternatív. Ako prvé si vypočítame váhy metódou entropie, neskôr svoju pozornosť budeme venovať metóde smerodajnej odchýlky a ako posledné otestujeme metódu CRITIC.

1. Metóda entropie

Prejdeme na výpočet, ktorý sa skladá z viacerých krokov. Normalizácia hodnôt v hodnotiacej matici. V tomto kroku si najskôr normalizujeme kritéria hotelov a následne vytvoríme celkový súčet normalizovaných údajov v Tab. 3. Výpočty potrebujeme do vzorcov, kde treba správne zvoliť či už pôjde o minimalizáciu alebo maximalizáciu hodnôt.

Tab. 3: Normalizácia hodnôt v hodnotiacej matici – čiastočné výpočty

	C1	C2	C3	C4	C5
(H1) Hotel A	3,44828E-06	3	0,178571429	60	2
(H2) Hotel B	2,08333E-06	5	0,161290323	80	1
(H3) Hotel C	1,96078E-06	4	0,14084507	110	3
(H4) Hotel D	1,78571E-06	3	0,175438596	90	2
(H5) Hotel E	1,63934E-06	4	0,181818182	110	3
SPOLU	1,09175E-05	19	0,8379636	450	11

Zdroj: Vlastné spracovanie

Tab. 4: Normalizácia hodnôt v hodnotiacej matici – celková normalizácia

	C1	C2	C3	C4	C5
(H1) Hotel A	0,315849874	0,157894737	0,213101653	0,133333333	0,181818182
(H2) Hotel B	0,190825966	0,263157895	0,192478913	0,177777778	0,090909091
(H3) Hotel C	0,179600909	0,210526316	0,168080177	0,244444444	0,272727273
(H4) Hotel D	0,163565114	0,157894737	0,209363028	0,2	0,181818182
(H5) Hotel E	0,150158137	0,210526316	0,216976229	0,244444444	0,272727273

Zdroj: Vlastné spracovanie

Výpočet entropie si rozdelíme na viac častí. V tab. 5 vidíme čiastočné výpočty atribútov $r_{ij} \ln r_{ij}$ a súčtový riadok týchto čiastkových výpočtov. Zostáva vypočítať $-\frac{1}{\ln n}$. Následne dostaneme sumárne výsledky vzorcu 25. Prejdeme na výpočet váh kritérií. Výpočty vykonáme podľa vzorca 26 v Tab. 6.

Tab. 5: Čiastočné výpočty a celková hodnota entropie

	C1	C2	C3	C4	C5
(H1) Hotel A	-0,36401327	-0,29144632	-0,329452169	-0,268653736	-0,309954199
(H2) Hotel B	-0,31608288	-0,35131607	-0,317160723	-0,307061502	-0,217990479
(H3) Hotel C	-0,308378	-0,328030446	-0,299739761	-0,34436532	-0,354349905
(H4) Hotel D	-0,29614185	-0,29144632	-0,327377943	-0,321887582	-0,309954199
(H5) Hotel E	-0,28470978	-0,328030446	-0,331532621	-0,34436532	-0,354349905
SPOLU	-1,56932579	-1,590269601	-1,605263217	-1,58633346	-1,546598686
e_j	0,975076938	0,988090059	0,997406116	0,985644396	0,960955793

Zdroj: Vlastné spracovanie

Tab. 6: Výsledné hodnoty váh kritérií pre metódu entropie

	C1	C2	C3	C4	C5
$1 - e_j$	0,024923062	0,011909941	0,002593884	0,014355604	0,039044207
w_j	0,268490236	0,128302975	0,027943298	0,154649512	0,420613978

Zdroj: Vlastné spracovanie

2. Metóda smerodajnej odchýlky

Pre výpočet si zoberieme údaje prvého kroku z metódy entropie z predchádzajúcej podkapitoly, ktoré sú totožné. Vykonáme výpočet smerodajnej odchýlky vo viacerých častiach. Za potreby je vypočítať priemer z r_{ij} , čo môžeme vidieť v Tab. 7. Tab. 8 disponuje výpočtami rozdielov, ich agregáciou a výpočtom rozptylu a smerodajnej odchýlky (27). Nasleduje posledný krok výpočet váh kritérií podľa vzorca (28), ako je v Tab. 9.

Tab. 7: Čiastočné výpočty metódy smerodajnej odchýlky

	C1	C2	C3	C4	C5
(H1) Hotel A	0,315849874	0,157894737	0,213101653	0,133333333	0,181818182
(H2) Hotel B	0,190825966	0,263157895	0,192478913	0,177777778	0,090909091
(H3) Hotel C	0,179600909	0,210526316	0,168080177	0,244444444	0,272727273
(H4) Hotel D	0,163565114	0,157894737	0,209363028	0,2	0,181818182
(H5) Hotel E	0,150158137	0,210526316	0,216976229	0,244444444	0,272727273
Priemer	0,2	0,2	0,2	0,2	0,2

Zdroj: Vlastné spracovanie

Tab. 8: Výpočet smerodajnej odchýlky pre výpočet váh kritérií

	C1	C2	C3	C4	C5
(H1) Hotel A	0,013421193	0,001772853	0,000171653	0,004444444	0,000330579
(H2) Hotel B	8,41629E-05	0,00398892	5,65668E-05	0,000493827	0,011900826
(H3) Hotel C	0,000416123	0,000110803	0,001018875	0,001975309	0,005289256
(H4) Hotel D	0,001327501	0,001772853	8,76663E-05	0	0,000330579
(H5) Hotel E	0,002484211	0,000110803	0,000288192	0,001975309	0,005289256
Spolu	0,017733191	0,007756233	0,001622954	0,008888889	0,023140496
Rozptyl	0,003546638	0,001551247	0,000324591	0,001777778	0,004628099
σ_j	0,059553659	0,039385867	0,018016402	0,042163702	0,068030134

Zdroj: Vlastné spracovanie

Tab. 9: Výsledné hodnoty váh kritérií pre metódu smerodajnej odchýlky

	C1	C2	C3	C4	C5
w_j	0,262177947	0,173391626	0,079315083	0,185620716	0,299494628

Zdroj: Vlastné spracovanie

3. Metóda CRITIC

Vykonanie normalizácie pomocou rozpätia hodnôt s dôležitým rozlíšením maximalizačných a minimalizačných kritérií, podľa vzorcov (29) a (30) v Tab. 10.

Tab. 10: Normalizácia hodnôt za pomoci metódy CRITIC

	C1	C2	C3	C4	C5
MIN	290000	3	5,5	60	1
MAX	610000	5	7,1	110	3
(H1) Hotel A	1	0	0,9375	0	0,5
(H2) Hotel B	0,40625	1	0,5625	0,4	0
(H3) Hotel C	0,3125	0,5	0	1	1
(H4) Hotel D	0,15625	0	0,875	0,6	0,5
(H5) Hotel E	0	0,5	1	1	1
σ_j	0,382094352	0,418330013	0,413162347	0,424264069	0,418330013

Zdroj: Vlastné spracovanie

V druhom kroku vypočítame hodnoty Pearsonovho korelačného koeficientu, podľa vzorca (31). Strednú hodnotu normovaných hodnotení variantov r_{ij} vyjadruje \bar{r} , za kritéria j a σ označuje smerodajnú odchýlku hodnotení.

Tab. 11: Hodnoty Pearsonového korelačného koeficientu

	C1	C2	C3	C4	C5
(C1) Cena v €	1	-0,293259195	0,058766834	-0,857833816	-0,366573994
(C2) Počet hviezdíčiek	-0,293259195	1	-0,415851333	0,281718085	-0,285714286
(C3) vzdialenosť	0,058766834	-0,415851333	1	-0,392207337	-0,171764681
(C4) Hodnotenie v bodoch	-0,857833816	0,281718085	-0,392207337	1	0,704295212
(C5) Druh stravy	-0,366573994	-0,285714286	-0,171764681	0,704295212	1

Zdroj: Vlastné spracovanie

Pri výpočte váh najprv vypočítame nenormované hodnotenia, ktoré sú založené na korelačných koeficientoch a smerodajných odchýlkach za pomoci vzorca (32). V poslednom kroku si cez vzorec 33 vypočítame váhy kritérií Tab. 13.

Tab. 12: Nenormované hodnotenia

	C1	C2	C3	C4	C5
(C1) Cena v €	0	1,293259195	0,941233166	1,857833816	1,366573994
(C2) Počet hviezdíčiek	1,293259195	0	1,415851333	0,718281915	1,285714286
(C3) vzdialenosť	0,941233166	1,415851333	0	1,392207337	1,171764681
(C4) Hodnotenie v bodoch	1,857833816	0,718281915	1,392207337	0	0,295704788
(C5) Druh stravy	1,366573994	1,285714286	1,171764681	0,295704788	0
$\sum_{l=1}^k (1 - p_{jl})$	5,458900171	4,713106729	4,921056517	4,264027856	4,119757748
c_j	2,085814923	1,971634	2,033195	1,809074	1,723418

Zdroj: Vlastné spracovanie

Tab. 13: Výsledné hodnoty váh kritérií pre metódu CRITIC

	C1	C2	C3	C4	C5
w_j	0,216750013	0,204885	0,211282	0,187992	0,179091

Zdroj: Vlastné spracovanie

5.3 Výpočet metódy Promethee

Na začiatok si musíme stanoviť aké typy funkcií priradíme ku kritériám, ktoré máme stanovené. Preto sme si zvolili nasledujúce typy funkcií

Tab. 14: Typy funkcií priradené ku kritériám

	Typ funkcie	q	r
(C1) Cena v €	Lineárna	10 000	320 000
(C2) Počet hviezdíček	U-jednoduchá	1	-
(C3) vzdialenosť	V-lineárna	-	1,6
(C4) Hodnotenie v bodoch	Lineárna	10	50
(C5) Druh stravy	U-lineárna	1	0

Zdroj: Vlastné spracovanie

Tab. 15: Experimentálny príklad

	C1	C2	C3	C4	C5
(H1) Hotel A	290000	3	5,6	60	2
(H2) Hotel B	480000	5	6,2	80	1
(H3) Hotel C	510000	4	7,1	110	3
(H4) Hotel D	560000	3	5,7	90	2
(H5) Hotel E	610000	4	5,5	110	3
	MIN	MAX	MIN	MAX	MAX
MIN	290000	3	5,5	60	1
MAX	610000	5	7,1	110	3

Zdroj: Vlastné spracovanie

Začneme postupným párovým porovnávaním variantov z hľadiska všetkých kritérií podľa vzorca (3). Výsledok je vyjadrenie intenzity preferencie medzi dvoma variantami.

Tab. 16: Postupné párové porovnávanie variantov

	C1	C2	C3	C4	C5
1,2	190000	-2	0,6	-20	1
1,3	220000	-1	1,5	-50	-1
1,4	270000	0	0,1	-30	0
1,5	320000	-1	-0,1	-50	-1
2,1	-190000	2	-0,6	20	-1
2,3	30000	1	0,9	-30	-2
2,4	80000	2	-0,5	-10	-1
2,5	130000	1	-0,7	-30	-2
3,1	-220000	1	-1,5	50	1
3,2	-30000	-1	-0,9	30	2
3,4	50000	1	-1,4	20	1
3,5	100000	0	-1,6	0	0
4,1	-270000	0	-0,1	30	0
4,2	-80000	-2	0,5	10	1
4,3	-50000	-1	1,4	-20	-1
4,5	50000	-1	-0,2	-20	-1
5,1	-320000	1	0,1	50	1
5,2	-130000	-1	0,7	30	2
5,3	-100000	0	1,6	0	0
5,4	-50000	1	0,2	20	1

Zdroj: Vlastné spracovanie

Viackriteriálny preferenčný index $\pi(a, b)$, vzorec (12) a (13), predstavuje stupeň uprednostňovania alternatívy a pred alternatívou b s ohľadom na všetky kritériá.

Tab. 17: Výpočet hodnôt podľa typu funkcií

C1	C2	C3	C4	C5	I
0,268490236	0,12830297	0,027943298	0,154649512	0,420613978	
0,580645161	0	0,375	0	0	0,166376293
0,677419355	0	0,9375	0	0	0,208077325
0,838709677	0	0,0625	0	0	0,226931816
1	0	0	0	0	0,268490236
0	1	0	0,25	0	0,166965353
0,064516129	0	0,5625	0	0	0,033040056
0,225806452	1	0	0	0	0,188929802
0,387096774	0	0	0	0	0,103931704
0	0	0	1	0	0,154649512
0	0	0	0,5	1	0,497938734
0,129032258	0	0	0,25	0	0,07330628
0,290322581	0	0	0	0	0,077948778
0	0	0	0,5	0	0,077324756
0	0	0,3125	0	0	0,008732281
0	0	0,875	0	0	0,024450386
0,129032258	0	0	0	0	0,034643901
0	0	0,0625	1	0	0,156395968
0	0	0,4375	0,5	1	0,510163927
0	0	1	0	0	0,027943298
0	0	0,125	0,25	0	0,04215529

Zdroj: Vlastné spracovanie

Každá alternatíva je porovnávaná s ostatnými $(n-1)$ alternatívami, preto uvádzame výstupný (20) a vstupný tok (21) v Tab. 18. Na dosiahnutie výsledku Promethee II, potrebujeme tzv. čistý tok cez vzorec (22).

Tab. 18: Výpočet hodnôt metódou entropie

METÓDA ENTROPIE							
	(H1) Hotel A	(H2) Hotel B	(H3) Hotel C	(H4) Hotel D	(H5) Hotel E	SPOLU	\emptyset^+
(H1) Hotel A	0	0,166376293	0,208077325	0,226931816	0,268490236	0,8699	0,217475
(H2) Hotel B	0,16696535	0	0,033040056	0,188929802	0,103931704	0,4929	0,123225
(H3) Hotel C	0,15464951	0,497938734	0	0,07330628	0,077948778	0,8038	0,20095
(H4) Hotel D	0,07732476	0,008732281	0,024450386	0	0,034643901	0,1452	0,0363
(H5) Hotel E	0,15639597	0,510163927	0,027943298	0,04215529	0	0,7367	0,184175
SPOLU	0,5553	1,1832	0,2935	0,5313	0,485		
\emptyset^-	0,138825	0,2958	0,073375	0,132825	0,12125		

Zdroj: Vlastné spracovanie

Tab. 19: Výpočet hodnôt metódou smerodajnej odchýlky

METÓDA SMERODAJNEJ ODCHÝLKY							
	(H1) Hotel A	(H2) Hotel B	(H3) Hotel C	(H4) Hotel D	(H5) Hotel E	SPOLU	ϕ^+
(H1) Hotel A	0	0,181975512	0,251962306	0,224848374	0,262177947	0,921	0,23025
(H2) Hotel B	0,21979681	0	0,06152944	0,232593098	0,101488237	0,6154	0,15385
(H3) Hotel C	0,18562072	0,392304986	0	0,080234592	0,076116178	0,7343	0,183575
(H4) Hotel D	0,09281036	0,024785963	0,069400698	0	0,033829412	0,2208	0,0552
(H5) Hotel E	0,19057791	0,427005335	0,079315083	0,056319564	0	0,7532	0,1883
SPOLU	0,6888	1,0261	0,4622	0,594	0,4736		
ϕ^-	0,1722	0,256525	0,11555	0,1485	0,1184		

Zdroj: Vlastné spracovanie

Tab. 20: Výpočet hodnôt metódou CRITIC

METÓDA CRITIC							
	(H1) Hotel A	(H2) Hotel B	(H3) Hotel C	(H4) Hotel D	(H5) Hotel E	SPOLU	ϕ^+
(H1) Hotel A	0	0,205085588	0,344907507	0,194995457	0,216750013	0,9617	0,240425
(H2) Hotel B	0,25188279	0	0,132829984	0,253828314	0,083903231	0,7224	0,1806
(H3) Hotel C	0,18799212	0,273087186	0	0,074965774	0,062927423	0,599	0,14975
(H4) Hotel D	0,09399606	0,066025618	0,18487173	0	0,027967744	0,3729	0,093225
(H5) Hotel E	0,20119724	0,365523051	0,211281977	0,073408277	0	0,8514	0,21285
SPOLU	0,7351	0,9097	0,8739	0,5972	0,3915		
ϕ^-	0,183775	0,227425	0,218475	0,1493	0,097875		

Zdroj: Vlastné spracovanie

Čisté toky Promethee II sú v Tab. 21. Čisté toky sú usporiadané a tým prezentujú výsledné poradie alternatív od tej najviac vyhovujúcej po tú najmenej.

Tab. 21: Výsledné hodnoty daných metód a určenie poradia čistých tokov

	METÓDA ENTROPIE		METÓDA SMERODAJNEJ		METÓDA CRITIC	
	ϕ	zoradenie	ϕ	zoradenie	ϕ	zoradenie
(H1) Hotel A	0,07865	2	0,05805	3	0,05665	2
(H2) Hotel B	-0,17258	5	-0,10268	5	-0,04683	3
(H3) Hotel C	0,127575	1	0,068025	2	-0,06873	5
(H4) Hotel D	-0,09653	4	-0,0933	4	-0,05608	4
(H5) Hotel E	0,062925	3	0,0699	1	0,114975	1

Zdroj: Vlastné spracovanie

6 Diskusia

Po vykonaní výpočtov sme sa dopracovali k zaujímavým výsledkom. Na základe váh, ktoré sme si stanovili prostredníctvom 3 rôznych metód na ich výpočet, sme získali tri rôzne usporiadania v metóde Promethee II. Ako prvé sa pozrieme na výsledky v rámci váh metódou entropie. V tomto prípade sme dostali nasledovné usporiadanie hotelov. Na prvom mieste sa umiestnil Hotel C, ktorý získal najvyššiu hodnotu. Hneď za ním vidíme Hotel A, potom Hotel E, Hotel D a posledný hotel B. Váhy metódy smerodajnej odchýlky neukazujú až také výrazné zmeny v usporiadaní hotelov. Výsledky sú pomerne rovnaké ako pri metóde entropie. To z dôvodu, že smerodajná odchýlka priamo meria rozptyl hodnôt, teda ako sú dáta rozložené okolo priemeru a metóda entropie meria mieru rovnomernosti rozdelenia hodnôt – ak sú hodnoty podobne rozložené, výsledky budú podobné ako pri smerodajnej odchýlke. Prvé tri

hotely sú Hotel E, Hotel C a Hotel A. Hotel D je rovnako ako aj v prvom prípade na pozícií 4 a ako posledný sa umiestnil Hotel B totožne s príkladom jedna.

Ako posledné svoju pozornosť venujeme práve metóde CRITIC, ktorá ako sme sa už dozvedeli v teoretickej časti pracuje na inom princípe, ako ostatné dve metódy, a práve preto sme vďaka nej dostali odlišné hodnoty. Berie do úvahy nielen variabilitu hodnôt kritérií, ale aj ich koreláciu s ostatnými kritériami, čo znamená, že kritériá s vysokou koreláciou budú mať nižšie váhy. Ako ďalšie zvyhodňuje kritériá, ktoré majú veľkú variabilitu a zároveň poskytujú nezávislé informácie. Na rozdiel od entropie alebo smerodajnej odchýlky nevyhodnocuje len rozptyl hodnôt, ale aj ich vzájomné vzťahy, čo môže viesť k rozdielnym váham oproti ostatným metódam, ako to bolo ukázané aj v našom prípade. Preto tieto faktory mohli vplývať na rozdielne výsledky v použitých metódach. V tejto metóde je usporiadanie hotelov nasledovné: na prvom mieste sa umiestnil práve Hotel E, na druhom Hotel A. Nečakane Hotel B, ktorý v predošlých dvoch príkladoch zostal na poslednom mieste tu je na mieste treťom. A ako posledné sa umiestnili Hotel D a Hotel C. Možno konštatovať, že pri výbere metód na výpočet váh je dôležité vybrať správne. Každá metóda sa zameriava na iný princíp a postup výpočtu a preto z nej vyplývajú aj iné výsledky a tým pádom aj iné usporiadania alternatív.

7 Záver

Na základe analýzy a hodnotenia použitia metódy Promethee II v tejto práci sme dospeli k nasledujúcim záverom. Metóda Promethee II je užitočným nástrojom pri rozhodovacom procese vo viacerých oblastiach, ako je napríklad výber optimálnej alternatívy medzi viacerými možnosťami. V našej práci sme sa zaoberali výpočtom váh kritérií pomocou troch rôznych metód - metódy entropie, metódy smerodajnej odchýlky a metódy CRITIC.

Prvým krokom výpočtu váh kritérií bol výber vhodnej metódy, ktorá nám umožní určiť dôležitosť jednotlivých kritérií. Následne sme použili metódu entropie na výpočet váh na základe rozptylu hodnôt v jednotlivých kritériách. Táto metóda nám poskytla informácie o množstve informácie, ktorú prináša každé kritérium v rozhodovacom procese. Ako ďalšiu metódu sme zvolili metódu smerodajnej odchýlky, ktorá nám umožnila zohľadniť aj variabilitu hodnôt kritérií. Pomocou tejto metódy sme určili váhy kritérií na základe rozptylu hodnôt v porovnaní s celkovým rozptylom. Na záver sme použili metódu CRITIC, ktorá využíva porovnanie medzi kritériami a ich vzťahom s výsledkom rozhodovacieho procesu. Táto metóda nám poskytla konečné váhy kritérií na základe ich významu pre konečné rozhodnutie.

Po výpočte váh kritérií sme mohli prejsť k výpočtu čistých tokov medzi jednotlivými alternatívami. Tieto čisté toky nám poskytnú kvantitatívne hodnoty o relatívnej výhodnosti jednotlivých alternatív vzhľadom na zvolené kritériá. Na základe vypočítaných čistých tokov sme určili poradie alternatív, pričom alternatíva s najvyšším čistým tokom bola vyhodnotená ako najlepšia. Tým sme dosiahli konečné riešenie v rámci rozhodovacieho procesu. Celkovo možno povedať, že metóda Promethee II a výpočet váh kritérií pomocou metódy entropie, metódy smerodajnej odchýlky a metódy CRITIC predstavujú užitočné nástroje pri rozhodovaní vo viacerých oblastiach a môžu pomôcť pri zdokonaľovaní rozhodovacieho procesu a presnejšom určovaní preferencii medzi alternatívami.

Literatúra

1. Abdullah, L., Chan, W., & Afshari, A. (2018). *Application of PROMETHEE method for green supplier selection: a comparative result based on preference functions*. *Journal of Industrial Engineering International*, 15(2), 271–285.
2. Brans, J. P., Macharis, C., & Mareschal, B. (1998). *The GDSS Promethee procedure*. *Journal of Decision Systems*, 7, 283-307.

3. Brans, J. P., Mareschal, B. (2005). Promethee Methods. In: *Multiple Criteria Decision Analysis: State of the Art Surveys*. International Series in Operations Research & Management Science, vol 78. Springer, New York, NY. https://doi.org/10.1007/0-387-23081-5_5
4. Brans, J. P., & Vincke, Ph. (1985). Note—a preference ranking organisation method. *Management Science*, 31(6), 647–656. <https://doi.org/10.1287/mnsc.31.6.647>
5. Brožová, H., Houška, M., & Šubrt, T. (2014 dotisk). *Modely pro vícekritériální rozhodování*. Česká zemědělská univerzita, Provozně ekonomická fakulta.
6. De Keyser, W., & Peeters, P. (1996). A note on the use of PROMETHEE multicriteria methods. *European Journal of Operational Research*, 89(3), 457–461.
7. Dubecová, I. (2009). *Komplexné vyhodnocovanie variantov prostredníctvom metód triedy Promethee: diplomová práca*. Fakulta hospodárskej informatiky EU.
8. Fiala, P., Jablonský, J., & Maňas, M. (1994). *Vícekritériální rozhodování*. Vysoká škola ekonomická [Praha].
9. Mlynarovič, V. (1998). *Modely a metódy viackritériálneho rozhodovania*. Vydavateľstvo Ekonóm.
10. Tzeng, G.-H., & Huang, J.-J. (2011). *Multiple Attribute Decision Making*. <https://doi.org/10.1201/b11032>
11. Zapletal, F. (2023). *Exaktní metody rozhodování*. Professional Publishing.

Fuzzy logika v predspracovaní údajov a jej vplyv na výkonnosť modelu strojového učenia XGBOOST

Fuzzy logic in data preprocessing and its impact on the performance of the XGBoost machine learning model

Andrej Bednařík¹

Abstrakt

Fuzzy logika poskytuje efektívny prístup k predspracovaniu číselných údajov v strojovom učení, najmä v regresii. Tento článok skúma vplyv fuzzyfikácie premenných, ako vek a BMI na presnosť predikcie nákladov na zdravotnú starostlivosť. Použitím fuzzy transformácie sme testovali výkon XGBoost regresora pri rôznych variantoch predspracovania datasetu. Výsledky naznačujú, že fuzzy logika môže v niektorých prípadoch zlepšiť presnosť predikcie (nižšie RMSE), najmä pri premenných s nejasnými hranicami. Diskutujeme tiež o situáciách, kde jej aplikácia neprináša zlepšenie, a identifikujeme scenáre, v ktorých je najvhodnejšia.

Kľúčové slová

Fuzzy, Regresia, XGBoost, Dáta

Abstract

Fuzzy logic provides an effective approach to preprocessing numerical data in machine learning, particularly in regression tasks. This paper explores the impact of fuzzification of variables such as age and BMI on the accuracy of healthcare cost prediction. By applying fuzzy transformation, we evaluated the performance of the XGBoost regressor across different dataset preprocessing variants. The results suggest that fuzzy logic can, in some cases, improve prediction accuracy (lower RMSE), especially for variables with unclear boundaries. We also discuss situations where its application does not lead to improvement and identify scenarios where it is most suitable.

Key words

Fuzzy, Regression, XGBoost, Data

JEL classification

C61, C89

1 Úvod

Fuzzy logika predstavuje prístup k spracovaniu neurčitých a nepresných informácií, ktorý je inšpirovaný spôsobom, akým ľudia uvažujú. Na rozdiel od klasickej binárnej logiky, kde hodnoty nadobúdajú len dve možnosti (pravda alebo nepravda, 0 alebo 1), fuzzy logika umožňuje hodnotám existovať v rozmedzí medzi týmito dvoma extrémami. Tento koncept bol prvýkrát predstavený Lotfi Zadehom v roku 1965 ako spôsob modelovania neurčitosti v reálnom svete (Zadeh, 1965). Základným princípom fuzzy logiky je, že premenné nie sú striktné

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra matematiky a aktuárstva, Dolnozemska cesta 1, 852 35 Bratislava, andrej.bednarik@euba.sk.

klasifikované do pevne daných kategórií, ale môžu patriť do viacerých kategórií s rôznou mierou príslušnosti. Napríklad teplota môže byť "nízka", "stredná" alebo "vysoká", pričom jedna konkrétna hodnota môže súčasne patriť do viacerých týchto skupín s rôznou intenzitou (Klir & Yuan, 1995). Táto flexibilita umožňuje efektívnejšie modelovanie systémov, ktoré pracujú s neurčitou, a nachádza široké uplatnenie v oblastiach ako riadiace systémy, rozpoznávanie vzorov, umelá inteligencia a spracovanie prirodzeného jazyka (Ross, 2010). V praxi sa fuzzy logika využíva na reprezentáciu neurčitých konceptov prostredníctvom tzv. funkcií členstva. Tieto funkcie určujú mieru príslušnosti konkrétnej hodnoty k danej fuzzy množine. Medzi najčastejšie používané patria trojuholníkové, lichobežníkové a gaussovské funkcie členstva. Napríklad, ak chceme definovať fuzzy množiny pre vek človeka, môžeme zaviesť kategórie "mladý", "stredný vek" a "starý" s plynulými prechodmi medzi nimi (Mendel, 2001). Fuzzy logika je tiež úzko spojená s fuzzy inferenciou, čo je proces odvodenia záverov na základe fuzzy pravidiel. Tieto pravidlá sú často formulované v podobe "Ak – Potom" (napr. "Ak je teplota vysoká, potom ventilátor beží na plný výkon"). Na spracovanie týchto pravidiel sa využívajú metódy ako Mamdaniho a Sugenov model inferencie, ktoré sa aplikujú v rôznych inžinierskych a vedeckých disciplínach (Ross, 2010). V oblasti strojového učenia môže fuzzy logika zlepšiť interpretovateľnosť a robustnosť modelov tým, že umožňuje efektívnejšie spracovanie číselných premenných s neurčitými hranicami. Napríklad, v predikčných modeloch môže byť vek alebo krvný tlak transformovaný do fuzzy premenných, čím sa lepšie zachytia ich skutočné vplyvy na cieľovú premennú (Mendel, 2001). Tento prístup sa osvedčil najmä v prípadoch, kde tradičné metódy spracovania údajov narážajú na problémy so striktným kategorizovaním premenných. Celkovo fuzzy logika predstavuje silný nástroj na modelovanie neurčitosti v rôznych oblastiach. V kombinácii so strojovým učením môže pomôcť pri efektívnejšom spracovaní údajov a zlepšení výkonnosti modelov, čo ju robí atraktívnou vo výskume aj priemyselnej praxi.

2 Fuzzy logika v predspracovaní dát

Predspracovanie dát je kľúčovým krokom v strojovom učení, ktorý ovplyvňuje presnosť a robustnosť modelov. Fuzzy logika ponúka alternatívu k tradičným metódam, najmä pri práci s kontinuálnymi premennými, ktoré nemajú presne definované hranice medzi kategóriami. Použitím fuzzyfikácie môžeme získať flexibilnejšiu reprezentáciu údajov a eliminovať skreslenia spôsobené ostrými prahmi kategorizácie. Dôvody prečo dáta fuzzyfikovať:

- Zachovanie kontinuity informácií: Tradičná kategorizácia (napr. vek rozdelený do skupín 20–30, 30–40) môže spôsobiť stratu informácií. Fuzzy reprezentácia umožňuje plynulý prechod medzi kategóriami.
- Redukcia šumu: Pri premenných, ako je krvný tlak alebo BMI, môže byť meranie ovplyvnené rôznymi faktormi. Fuzzy logika pomáha zmierniť vplyv náhodných odchýlok.

Zlepšenie interpretovateľnosti modelov: Fuzzy premenné môžu byť lepšie pochopiteľné pre analytikov a doménových expertov, najmä v medicíne alebo poisťovníctve. Príklady fuzzyfikácie premenných:

- Vek:
 - Klasický prístup: vek = 35 (jedna číselná hodnota)
 - Fuzzy prístup: vek môže byť súčasne "mladý" (0.2) a "stredný vek" (0.8), čo lepšie vystihuje realitu prechodu medzi skupinami.

- BMI:
 - Klasický prístup: BMI = 28 → kategória "nadváha".
 - Fuzzy prístup: BMI = 28 → príslušnosť k "normálna váha" (0.3) a "nadváha" (0.7), čím sa zachytí plynulý prechod medzi stavmi.

- Krvný tlak:
 - Klasický prístup: hodnota 139 mmHg sa môže považovať za normálnu, ale 140 mmHg už ako vysoký tlak.
 - Fuzzy prístup: Hodnota 139 mmHg môže mať členstvo 0.8 v "normálny" a 0.2 v "vysoký tlak", čím sa predíde ostrým hraniciam.

Fuzzyfikácia premenných sa realizuje pomocou funkcií členstva, ktoré definujú mieru príslušnosti danej hodnoty k jednotlivým fuzzy množinám. Bežne sa používajú:

- Trojuholníkové funkcie – jednoduchá reprezentácia pre plynulé prechody.
- Lichobežníkové funkcie – vhodné pre premenné s rozsiahlymi strednými hodnotami.
- Gaussovské funkcie – používajú sa v prípadoch, kde je potrebné modelovať postupné zmeny bez ostrých hraníc.

V ďalších častiach článku sa zameriame na praktickú implementáciu fuzzyfikácie pri analýze dát a hodnotenie jej vplyvu na výkonnosť predikčných modelov.

3 Metodika a experimentálny dizajn

V tejto časti skúmame vplyv fuzzy logiky na výkonnosť modelov strojového učenia pri predikcii nákladov na zdravotnú starostlivosť. Cieľom je overiť, či fuzzyfikácia niektorých numerických premenných môže viesť k zlepšeniu presnosti predikcie. Zameriavame sa predovšetkým na premenné, ktoré majú neostre alebo subjektívne definovateľné hranice – ako napríklad BMI či vek.

Obr. 1: Náhľad datasetu

age	sex	bmi	children	smoker	region	charges
18	male	33.77	1	no	southeast	1725.5523
19	male	24.6	1	no	southwest	1837.237
18	male	34.1	0	no	southeast	1137.011
18	female	26.315	0	no	northeast	2198.18985
19	female	28.6	5	no	southwest	1728.897
19	male	20.425	0	no	northwest	1625.43375
18	female	38.665	2	no	northeast	1728.897
18	female	35.625	0	no	northeast	2211.13075
19	female	28.9	0	no	southwest	1743.214
18	female	30.115	0	no	northeast	1728.897

Zdroj: Vlastné spracovanie

Metodika pozostáva z nasledujúcich krokov:

1. **Fuzzyfikácia vybraných numerických premenných** – pomocou preddefinovaných fuzzy množín transformujeme pôvodné hodnoty (napr. BMI) na fuzzy reprezentácie.

2. **Tréning modelov pomocou algoritmu XGBoost** – tento výkonný boostingový algoritmus aplikujeme na rôzne verzie datasetu, aby sme mohli porovnať vplyv fuzzy predspracovania.
3. **Porovnanie výsledkov** – hodnotíme výkonnosť modelov pomocou metriky RMSE (root mean squared error), pričom analyzujeme, v ktorých prípadoch fuzzy logika prispela k zlepšeniu a kedy nie.

Testujeme tri varianty modelov:

1. Model bez fuzzyfikácie – štandardný model pracujúci s pôvodnými numerickými hodnotami bez akéhokoľvek rozmazania hraníc medzi kategóriami.
2. Model s fuzzyfikáciou BMI – v tomto prípade je premenná BMI transformovaná do fuzzy priestoru pomocou jazykových hodnôt ako „nízky“, „normálny“, „vysoký“ a „veľmi vysoký“.

Obr. 2: Fuzzyfikácia premennej Bmi

```
def fuzzify_bmi(row):
    bmi = row['bmi']
    underweight, normal, overweight, obesity = 0, 0, 0, 0

    if bmi < 18.5:
        underweight = 1
    elif 18.5 <= bmi < 24.9:
        normal = (bmi - 18.5) / (24.9 - 18.5) # Increasing from 0 to 1
        underweight = 1 - normal # Decreasing from 1 to 0
    elif 24.9 <= bmi < 29.9:
        overweight = (bmi - 24.9) / (29.9 - 24.9) # Increasing from 0 to 1
        normal = 1 - overweight # Decreasing from 1 to 0
    elif bmi >= 30:
        obesity = 1
```

Zdroj: Vlastné spracovanie

3. Model s fuzzyfikáciou veku – kde vek bude transformovaný do fuzzy premenných

Obr. 3: Fuzzyfikácia premennej vek

```
def fuzzify_age(row):
    age = row['age']
    young, middle, old = 0, 0, 0

    if age <= 25:
        young = 1
    elif 25 < age <= 30:
        young = 1 - (age - 25) / 5 # Decreasing from 1 to 0
        middle = (age - 25) / 5 # Increasing from 0 to 1
    elif 30 < age <= 50:
        middle = 1
    elif 50 < age <= 55:
        middle = 1 - (age - 50) / 5 # Decreasing from 1 to 0
        old = (age - 50) / 5 # Increasing from 0 to 1
    elif age > 55:
        old = 1
```

Zdroj: Vlastné spracovanie

Použitý dataset obsahuje premenné ako vek, pohlavie, BMI, počet detí, fajčenie, región a cieľovú premennú charges (náklady na zdravotnú starostlivosť). Modely budú vyhodnotené pomocou metriky RMSE na testovacej množine. Na vytvorenie a tréning modelu použijeme XGBoost pre regresiu, ktorý umožňuje efektívne pracovať s nelineárnymi vzťahmi v dátach. Modely budú vyhodnotené pomocou metriky RMSE na testovacej množine. Okrem toho budeme

využívať K- Fold cross-validáciu s náhodným premiešavaním dát (shuffle), aby sme zabezpečili robustnejšie vyhodnotenie výkonu modelov a minimalizovali vplyv konkrétneho rozdelenia datasetu. Výsledky experimentov nám umožnia zhodnotiť, v ktorých situáciách fuzzy logika prispieva k lepšiemu modelovaniu reálnych, často neostro definovaných javov, ako sú zdravotné riziká spojené s nadváhou alebo vekom.

Nastavenie hyperparametrov XGBoost modelu:

- **n_estimators:** [5, 10, 20, 30, 40, 60, 70, 100] – Počet stromov v modeli, experimentálne testujeme viacero hodnôt.
- **max_depth:** 3 – Maximálna hĺbka stromov, zabraňuje nadmernému prispôsobeniu (overfittingu).
- **learning_rate:** 0.1 – Rýchlosť učenia, nižšia hodnota umožňuje postupnejšie optimalizovanie váh.
- **subsample:** 0.9 – Podiel náhodne vybraných vzoriek pri tréovaní každého stromu, zlepšuje generalizáciu.
- **random_state:** 42 – Zabezpečuje reprodukovateľnosť výsledkov.
- **colsample_bytree:** 1 – Použitie všetkých premenných pri konštrukcii jednotlivých stromov.

3 Experimentálne výsledky a hodnotenie modelov

Ako prvý sme natrénovali model bez fuzzyfikácie, ktorý pracoval s pôvodnými numerickými hodnotami všetkých premenných. Tento model slúžil ako referenčná základňa na porovnanie s ďalšími verziami, kde sme aplikovali fuzzy úpravy vybraných premenných. Následne sme vytvorili modely s fuzzyfikáciou veku a BMI, aby sme analyzovali ich vplyv na výkonnosť modelu a presnosť predikcií.

Obr. 4: Výsledky modelu bez fuzzyfikácie

```
Best number of estimators: 70  
Best RMSE (cross-validated): 1906.30
```

Zdroj: Vlastné spracovanie

Výsledky na obrázku 4 ukazujú, že optimálny počet stromov (`n_estimators`) je 70, čo znamená, že model dosiahol najlepšiu výkonnosť pri tréovaní s týmto počtom stromov. Menej stromov by mohlo viesť k underfittingu, zatiaľ čo viac stromov by nemuselo priniesť výrazné zlepšenie a mohlo by viesť k miernemu overfittingu. Hodnota Root Mean Squared Error (RMSE) = 1906.30 vyjadruje priemernú chybu predikcie v jednotkách cieľovej premennej, teda v tomto prípade v eurách. To znamená, že priemerná odchýlka medzi skutočnými a predikovanými nákladmi na zdravotnú starostlivosť je približne 1906 eur. Nižšia hodnota RMSE znamená presnejší model, preto je dôležité porovnať tento výsledok s modelom bez fuzzy logiky a s fuzzyfikáciou iných premenných, aby sme vyhodnotili prínos tohto prístupu.

Obr. 5: Náhľad datasetu s fuzzyfikáciou (BMI)

	age	sex	bmi	children	smoker	region	charges	underweight	normal	overweight	obesity
1	18	male	33.77	1	no	southeast	1725.5523	0.0	0.0	0.0	1.0
2	19	male	24.6	1	no	southwest	1837.237	0.046874999999999556	0.9531250000000004	0.0	0.0
3	18	male	34.1	0	no	southeast	1137.011	0.0	0.0	0.0	1.0
4	18	female	26.315	0	no	northeast	2198.18985	0.0	0.71699999999999994	0.28300000000000053	0.0
5	19	female	28.6	5	no	southwest	1728.897	0.0	0.25999999999999945	0.7400000000000005	0.0
5	19	male	20.425	0	no	northwest	1625.43375	0.69921874999999998	0.30078125000000017	0.0	0.0
7	18	female	38.665	2	no	northeast	1728.897	0.0	0.0	0.0	1.0
3	18	female	35.625	0	no	northeast	2211.13075	0.0	0.0	0.0	1.0
3	19	female	28.9	0	no	southwest	1743.214	0.0	0.19999999999999996	0.8	0.0
0	18	female	30.115	0	no	northeast	1728.897	0.0	0.0	0.0	1.0
1	19	female	28.4	1	no	southwest	2331.519	0.0	0.30000000000000004	0.7	0.0
2	18	male	23.75	0	no	northeast	1705.6245	0.17968749999999978	0.8203125000000002	0.0	0.0

Zdroj: Vlastné spracovanie

Po fuzzyfikácii datasetu otestujeme, či v modeli XGBoost fuzzyfikácia BMI zlepši presnosť predikcií. Skúmame dva prístupy: (1) úplné nahradenie pôvodnej premennej BMI fuzzy príslušnosťami alebo (2) zachovanie pôvodnej hodnoty BMI spolu s fuzzy reprezentáciou. Cieľom je zistiť, ktorá z týchto stratégií vedie k presnejšiemu modelu.

```
Best number of estimators: 70
Best RMSE (cross-validated): 2102.04
```

Zdroj: Vlastné spracovanie

Obr. 6: Výsledky experiment BMI prístup 1

Na základe uvedených výsledkov na obrázku 6 môžeme pozorovať, že počet optimálnych stromov (`n_estimators`) ostal rovnaký na hodnote 70, čo znamená, že zmena v predspracovaní údajov (fuzzyfikácia) neovplyvnila potrebu vyššej ani nižšej modelovej kapacity. Hodnota RMSE (cross-validated) sa však zvýšila o 195.74 bodov na 2102.04, čo predstavuje zhoršenie oproti pôvodnému modelu bez fuzzyfikácie, ktorý dosiahol RMSE 1906.30.

Obr. 7: Výsledky experiment BMI prístup 2

```
Best number of estimators: 70
Best RMSE (cross-validated): 1901.91
```

Zdroj: Vlastné spracovanie

Na základe výsledkov na obrázku 7 môžeme povedať, že počet optimálnych stromov (`n_estimators`) ostal rovnaký na hodnote 70. Hodnota RMSE (cross-validated) sa však v tomto prístupe znížila o 4.39 bodov na 1901.91, čo predstavuje zlepšenie predikcii oproti pôvodnému modelu bez fuzzyfikácie, ktorý dosiahol RMSE 1906.30.

Výsledky experimentu naznačujú, že prítomnosť pôvodného stĺpca spolu s jeho „fuzzy“ verziou modelu neškodí a zároveň jeho vynechanie nijako model výrazne nezlepšuje no naopak zhoršuje. Na základe tohto zistenia pre premennú vek vykonáme len experiment s prístupom 2.

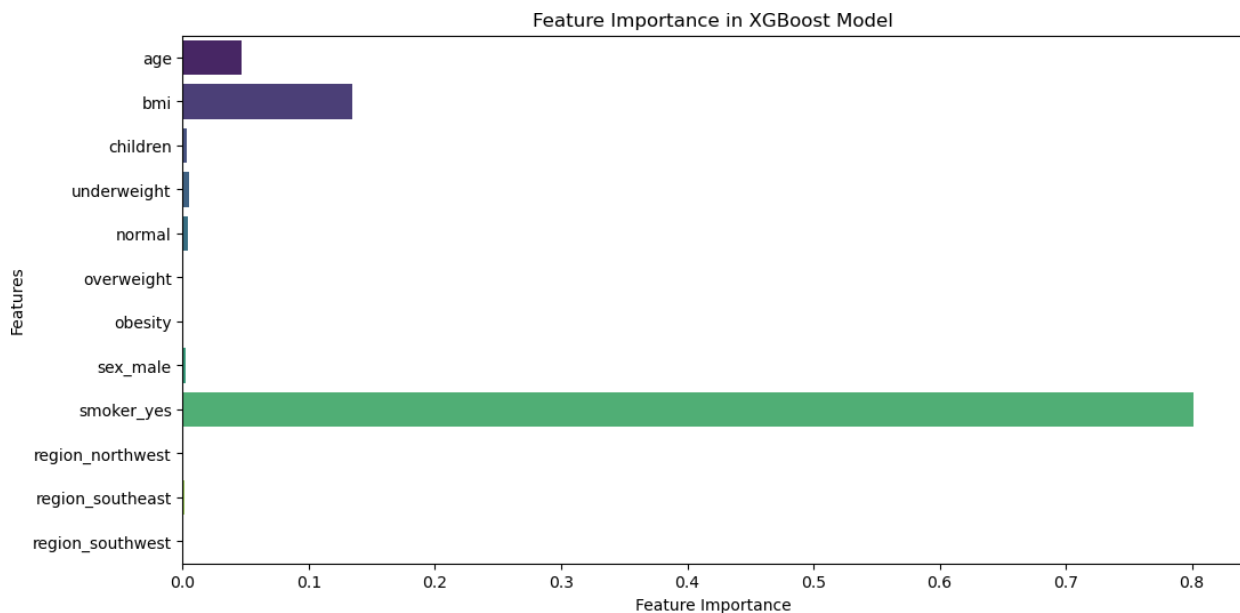
Obr. 8: Výsledky experiment Vek

```
Best number of estimators: 70
Best RMSE (cross-validated): 1908.01
```

Zdroj: Vlastné spracovanie

Na základe výsledkov na obrázku 8 môžeme povedať, že počet optimálnych stromov ($n_{\text{estimators}}$) ostal rovnaký na hodnote 70. Hodnota RMSE (cross-validated) sa však v tomto prístupe zvýšila o 1.71 bodu na 1908.01, čo predstavuje zhoršenie predikcii oproti pôvodnému modelu bez fuzzyfikácie, ktorý dosiahol RMSE 1906.30.

Na základe obrázku 9 môžeme povedať, že premenná "smoker_yes" má najvyššiu dôležitosť – to znamená, že najväčšiu časť variability v predikcii nákladov na zdravotnú starostlivosť vysvetľuje práve informácia o tom, či je osoba fajčiar alebo nie. V tomto prípade fuzzyfikácia iných premenných nemusí mať zásadný dopad, pretože hlavný prediktor je už jednoznačný a binárny (0/1). Premenná "BMI" je druhá najvýznamnejšia – keďže BMI má stále relatívne veľký vplyv na predikované hodnoty, fuzzyfikácia mohla pomôcť modelu lepšie zachytiť jeho vzťah k cieľovej premennej, čo viedlo k zlepšeniu RMSE. Premenná "age" má relatívne nízku dôležitosť – keďže vek je menej významný v porovnaní s BMI a fajčením, fuzzyfikácia veku by mohla priniesť len minimálne zlepšenie alebo zhoršenie ako aj vyplynulo z našich experimentov, pretože samotná premenná nie je kľúčovým faktorom v predikcii. Ostatné premenné (children, región, podkategórie BMI) majú veľmi nízku dôležitosť, čo naznačuje, že ich fuzzyfikácia by pravdepodobne nemala významný vplyv na zlepšenie modelu.



Zdroj: Vlastné spracovanie

Obr. 9: Feature importance

4 Záver

Výsledky experimentov ukázali, že fuzzyfikácia premenných môže v niektorých prípadoch prispieť k zlepšeniu presnosti modelu, avšak jej efektívnosť závisí od viacerých faktorov. V našej analýze sa ukázalo, že fuzzyfikácia veku neprinesla zlepšenie no naopak zhoršenie, zatiaľ čo fuzzyfikácia BMI v kombinácii s ponechaním pôvodnej hodnoty mierne znížila RMSE, čím sa zvýšila presnosť predikcií. Naopak, úplné odstránenie pôvodných hodnôt pri fuzzyfikácii spôsobilo výrazné zhoršenie výkonu modelu vytvoreného pomocou XGBOOST. Jedným z kľúčových poznatkov je, že významnosť premenných hrá zásadnú úlohu v tom, či fuzzyfikácia prinesie zlepšenie. V našom datasete bola najvýznamnejšou premennou binárna kategória „smoker_yes“, ktorá rozhodujúcim spôsobom ovplyvňovala predikované náklady na zdravotnú starostlivosť. To naznačuje, že v datasetoch, kde dominujú diskrétne alebo binárne kategórie, nemusí fuzzyfikácia číselných premenných výrazne pomôcť. Okrem presnosti modelu je dôležité zohľadniť aj jeho komplexnosť a výpočtovú náročnosť. Pridanie nových fuzzy premenných zväčšuje dimenzionalitu datasetu, čo môže viesť k vyššej výpočtovej záťaži a potenciálnym problémom s interpretáciou modelu. Preto je nutné vždy zvážiť, či mierne zvýšenie presnosti modelu kompenzuje túto dodatočnú zložitosť. Z uvedených zistení vyplýva, že fuzzyfikácia môže byť užitočná pri vhodných premenných, ale nie vždy je prínosná. V prípade datasetov, kde sú hlavné prediktory binárne alebo kategorické, môže byť jej efekt minimálny. Preto by sa mala fuzzyfikácia aplikovať selektívne, s ohľadom na významnosť premenných a celkový dopad na model. V ďalších výskumoch by bolo vhodné preskúmať, ako fuzzyfikácia ovplyvňuje modely pri iných typoch datasetov, kde by mohla mať väčší prínos – napríklad v prípadoch, kde sú všetky hlavné prediktory kontinuálne a majú nejednoznačné hranice medzi kategóriami. Najdôležitejším aspektom je správne rozdelenie fuzzy intervalov, pretože nesprávne zvolená fuzzy reprezentácia môže skresliť vzťahy medzi premennými a viesť k strate relevantných informácií. Optimálne nastavenie fuzzy intervalov je preto kľúčové pre dosiahnutie maximálneho prínosu tejto metódy.

Tento príspevok vznikol v rámci výskumného projektu

VEGA 1/0497/25, Implementácia inovatívnych prístupov v oblasti riadenia a modelovania rizík v rámci interných modelov poisťovní.

VEGA č. 1/0377/25 Inovatívne metódy Enterprise risk managementu a ich využitie v riadení a modelovaní rizík.

Literatúra

1. Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall.
2. Mendel, J. M. (2001). *Uncertain rule-based fuzzy logic systems: Introduction and new directions*. Prentice Hall.
3. Ross, T. J. (2010). *Fuzzy logic with engineering applications* (3rd ed.). John Wiley & Sons.
4. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)

Dynamic financial withdrawal planning from interest-bearing assets

Dynamické plánovanie výberu financií z úročených aktív

Daniel Dudek¹

Abstract

Dynamic programming offers tools to mathematically describe the effects of continuously executed processes. Policymakers or Process owners in companies could use dynamic programming to find the optimal use of such processes. This article aims to show the use of dynamic programming to develop a plan for the dynamic withdrawal of funds from interest-bearing assets.

Key words

Dynamic Programming, Plan Creation, Reproduction Model, Policy Making

Abstrakt

Dynamické programovanie ponúka nástroje na matematický opis účinkov nepretržite vykonávaných procesov. Tvorcovia politik alebo vlastníci procesov v podnikoch by mohli dynamické programovanie využívať na hľadanie optimálneho využitia takýchto procesov. Cieľom tohto článku je ukázať využitie dynamického programovania za účel vytvoreniu plánu pre dynamický výber financií z úročiach aktív.

Kľúčové slová

dynamické programovanie, tvorba plánu, reprodukčný model, tvorba politik

JEL classification

C61

1 Introduction

In this article, we will show the utilisation of Bellman Dynamic Programming on simple problems. The basic idea of Dynamic Programming is to solve a problem using a divide-and-conquer approach wherein the solutions of overlapping subproblems are reused to avoid recalculating solutions. Toth Horowitz and Sahni presented an improved dynamic programming algorithm for solving the knapsack problem (Chebil & Khemakhem, 2015). Generally, dynamic programming-based algorithms are efficient and easy to implement, particularly for small and medium-sized instances. We will show some applications of such matter in various problems across the world. In the end, we developed mathematical model inspired by reproduction model that create plan that will maximise the profit with highest liquidity form.

2 Literature review

The main idea of Dynamic Programming (DP) is to decompose the problem into more manageable subproblems. Computations are then carried out recursively, where the optimum solution of one subproblem is used as an input to the next subproblem. The optimum solution for the entire problem is at hand when the last subproblem is solved. How the recursive computations are carried out depends on how the original problem is decomposed. In particular,

¹ Bratislava University of Economics and Business, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemska cesta 1, 852 35 Bratislava, daniel.dudek@euba.sk.

the subproblems are usually linked by typical constraints. The feasibility of these common constraints is maintained at all iterations (Taha et al., 2017).

The central recursive equation expresses the shortest distance $f_i(x_i)$ at stage i as a function of the next node x_{i+1} . Here, i ranges over the finite set of stage indices $I=\{0,1,\dots,N\}$, so each value of i labels a specific decision stage in the process. In dynamic programming terminology x_i is the state at stage i . The state links successive stages in a way that allows optimal decisions at a future stage to be made independently of all choices at preceding stages. Defining the state in this manner leads to the following unifying framework for dynamic programming (Sieniutycz & Jeżowski, 2013).

Future decisions for all future stages constitute an optimal policy regardless of the policy adopted in all preceding stages. The principle of optimality does not address how a subproblem is optimised. The reason is the generic nature of the subproblem. It can be linear or nonlinear, and the number of alternatives can be finite or infinite. All the principle of optimality does is “break down” the original problem into more computationally tractable subproblems (Taha et al., 2017). Table 1 crystallises the standard finite-horizon dynamic-programming framework by cataloguing the temporal index, state and control manifolds, admissible boundary sets, cost functionals, and deterministic state-transition operator that underpin the subsequent analytical developments.

Tab. 1: Description of the elements of dynamic programming definition

Symbol	Meaning
T	The finite planning horizon, i.e., the number of stages (time-steps) in the decision process. Stages are indexed $t=0,1,\dots,T$. In a finite-horizon problem T is fixed; in an open (indefinite) horizon it may vary or be chosen optimally.
t	Continuous time variable or in other words stage, $t \in [0, T]$.
Q	The action (control) set – all decisions $q(t)$ that are admissible at any stage t .
X	The state space – all system states $x(t)$ that are allowed during the horizon.
$P \subseteq X$	The set of admissible initial states. At $t=0$ the process must start in one of these states.
$C \subseteq X$	The set of admissible terminal (goal) states. At the final stage $t=T$ the state must lie in this set.
$x(t)$	The state variable at stage t , taking values in X .
$q(t)$	The control (decision) variable at stage t , taking values in Q .
$f(x,q)$	The stage-cost (or reward) function incurred when the state is x and the action q is applied.
$f(t,x,q)$	Instantaneous cost (or reward) density incurred at time t when the state is x and the control is q .
$g(t,x,q)$	State-transition (dynamics) function giving (time derivative) $\dot{x}(t)=g(t,x(t),q(t))$

Source: Processed from (Lašćiak et al., 1983)

There are two types of optimisation methods based on dynamic programming that are Discrete dynamic optimisation and continuous dynamic optimisation.

- Discrete dynamic optimisation, which tries to find such policy from all the possible policies (Lašćiak et al., 1983).

The goal is to find a sequence of controls $\{q(t)\}_{t=0}^{T-1}$ such that

Controls are noted as:

$$q(t) \in Q, \quad \text{for } t = 0, 1, \dots, T - 1 \quad (1)$$

States are noted as:

$$x(t) \in X, \quad \text{for } t = 0, 1, \dots, T \quad (2)$$

Initial state is noted as:

$$x(0) \in P \quad (3)$$

Terminal state is noted as:

$$x(T) \in C \quad (4)$$

and the cumulative cost:

$$\sum_{t=0}^{T-1} f(x(t), q(t)) \quad (5)$$

is minimised (or maximised, depending on the formulation).

- Continuous dynamic optimisation problem, which tries to find such policy from all the possible policies, that finds a control trajectory $\{q(t)\}_{t \in [0, T]}$ where (Laščiak et al., 1983):

Controls are noted as:

$$q(t) \in Q, \quad t \in [0, T] \quad (6)$$

States are noted as:

$$x(t) \in X, \quad t \in [0, T] \quad (7)$$

Initial state is noted as:

$$x(0) \in P \quad (8)$$

Terminal state is noted as:

$$x(T) \in C \quad (9)$$

subject to the system dynamics

$$\dot{x}(t) = g(t, x(t), q(t)), \quad t \in [0, T] \quad (10)$$

and cumulative COST, that represents cost regarding the optimisation

$$COST = \int_0^T f(t, x(t), q(t)) dt \quad (11)$$

is minimised (or maximised, depending on the formulation).

There exists two horizon types (Carlson et. al., 1991):

- Finite-time horizon: T is prescribed and constant; optimisation is performed over the fixed interval $[0, T]$.
- Open (indefinite) horizon: T is not predetermined; it may depend on the policy or be itself a decision variable, leading to problems where the optimal stopping time is part of the solution.

Another characteristic of the dynamic programming approach is developing a recursive optimisation procedure, which builds to a solution of the overall N -stage problem by first solving a one-stage problem, sequentially including one stage at a time, and solving one-stage problems until the overall optimum has been found. This procedure can be based on a backward induction process, where the first stage to be analysed is the final stage of the problem, and problems are solved, moving back one stage at a time until all stages are included. Alternatively, the recursive procedure can be based on a forward induction process, where the first stage to be solved is the initial stage of the problem, and problems are solved moving forward one stage at a time until all stages are included. In specific problem settings, only one of these induction processes can be applied (e.g., only backward induction is allowed in most problems involving uncertainties). The basis of the recursive optimisation procedure is the so-called principle of optimality, which has already been stated: an optimal policy has the property that, whatever the current state and decision, the remaining decisions must constitute an optimal policy about the state resulting from the current decision (Massachusetts Institute of Technology, 2015). Those methods are:

1. Top-down method
2. Bottom-up method

The top-down method solves the overall problem before breaking it into subproblems. This process solves more significant problems by recursively finding the solution to subproblems, caching each result. This memorisation process helps avoid solving the problem repeatedly if it is called more than once. The top-down method, can return the result saved as it was solved in the context of the overall problem, thus storing the results of already solved problems. The most common related process is called forward induction (Jaffar et al., 2008).

The bottom-up—or tabulation—method works in the opposite direction. It evaluates **all** sub-problems in an order that guarantees every dependency is already known when needed, then stores those answers in a table. Because there may be many indices (stage, capacity, remaining time, etc), the table is typically n -dimensional, where n represents the number of independent indices that characterise a sub-problem, while ($n \geq 1$). Earlier we used the capital N for “the number of stages” in a discrete model. Here the lower-case n simply counts how many indices are required to label sub-problems; it is unrelated to N . Once every entry of the table is filled, the value of the original problem is read directly from the appropriate cell. Computing the table from the “end” of the decision process toward the “start” is known as backward induction (Wimmer et al., 2018).

Forward and backward induction always return the same optimal value. Nevertheless, most dynamic-programming textbooks and software libraries default to backward induction because, in many practical models, it stores fewer intermediate states and therefore runs faster (Taha et al., 2017).

Backward induction is determining a sequence of optimal choices of action by employing reasoning backwards in sequence, from the end of a problem or situation to its beginning, choice by choice. It proceeds by examining the last point at which a decision is to be made and then identifying the most optimal choice of action. Using this information, one can determine what to do at the second-to-last point of the decision. This process continues backwards until one has determined the best action for every possible point along the sequence (Matias et al., 2023).

However, the Bellman optimality Principle is the method that made dynamic programming a respected part of mathematics. Bellman's optimality principles are suitable for optimal conditions for inherently discrete processes. Nevertheless, under the differentiability assumption, the method only enables an easy passage to its limiting form for continuous systems. The application of the method is straightforward when it is applied in the optimisation of control systems without feedback. Dynamic programming (DP) is crucial for the optimal performance potentials discussed in this book and for deriving pertinent equations that describe these potentials. The DP method is based on Bellman's principle of optimality. It makes it possible to replace the simultaneous evaluation of all optimal controls with sequences of local evaluations at sequentially included stages for evolving subprocesses (Matias et al., 2023).

Description of the elements in Tab. 2 and in formulas below are completely separated from previous notations of elements and formulas.

Tab. 2: Description of the elements of Bellman's Expectation Equation in deterministic cases

Symbol	Definition
s	A state the agent can occupy.
a	An action chosen in state s .
s'	The next state reached after taking action a .
$V(s)$	The value (expected return) associated with state s .
$R(s,a)$	The immediate reward received when action a is executed in state s .
$\gamma \in [0,1)$	The discount factor that down-weights future rewards.
$\pi = \{\pi_1, \pi_2, \dots, \pi_I\}$	A finite collection of candidate policies (instruction sets) among which we search for the optimal one.
I	The number of policies in that set. We enumerate the options as $\pi_1, \pi_2, \dots, \pi_I$ to keep track of each distinct strategy.

Source: Processed from (Bellman, 1954)

The most basic Bellman's Expectation Equation for optimising value can be stated as (Bellman, 1954):

$$V(s) = \max_a (R(s, a) + \gamma V(s')) \quad (12)$$

During backward (bottom-up) induction the agent starts at the goal, assigns terminal values, and then works backward through the state space, filling in $V(s)$ by applying the Bellman equation. At each step it picks the action a that maximises the bracketed expression, thereby choosing the policy π_i that ultimately delivers the highest cumulative reward. In Tab.3, is depicted the exemplar solution where $\gamma = 0.9$:

Tab. 3: Example of space sweeping by agent

$V=0.81$ ($0+0.9*(0.9)$) $a^* = \rightarrow$	$V=0.9$ ($0+0.9*(1)$) $a^* = \rightarrow$	$V=1$ ($1+0.9*(0)$) $a^* = \rightarrow$	$R=+1$ (Goal)
$V=0.73$ ($0+0.9*(0.81)$) $a^* = \uparrow$	$R=-M$	$V=0.9$ ($0+0.9*(1)$) $a^* = \uparrow$	$R=-1$
$V=0.66$ ($0+0.9*(0.73)$) $a^* = \uparrow$ or \rightarrow (Start)	$V=0.73$ ($0+0.9*(0.81)$) $a^* = \rightarrow$	$V=0.81$ ($0+0.9*(0.9)$) $a^* = \uparrow$	$V=0.73$ ($0+0.9*(0.81)$) $a^* = \leftarrow$

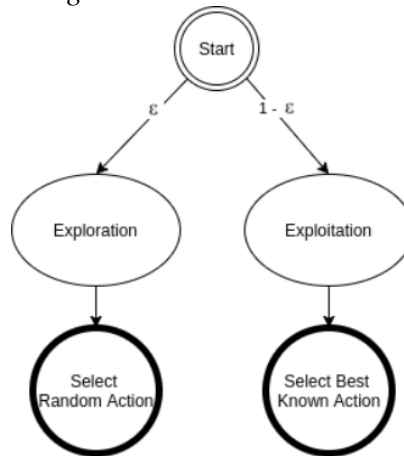
Source: Own Elaboration

When the agent calculates all the known $V(s)$ values, it can use a greedy algorithm to find the policy, leading him from the start to the goal. After the application of the greedy algorithm, the resulting optimal policy will be: $\pi_i = \uparrow\uparrow\rightarrow\rightarrow$ or $\rightarrow\rightarrow\uparrow\uparrow$

In dynamic programming, while we use the bellman method with the top-down method, we select a policy based on its reward. The agent always chooses the optimal action. Hence, it generates the maximum reward possible for the given state. In our problem, we will use this greedy algorithm, named the economic strategy (Baeldung, 2023).

If we used the tree diagram to calculate all the possibilities of the problem for the given number of iterations, we could use backward induction to find the optimal solution where we are exercising the reward (or payoffs) without the need for exploration. This effect is obtained because backward induction has the final values of all previous policies, so it can more easily determine the optimal action from all possibilities.

Fig. 1: Diagram depicting decisions between methods in DP of the agent



Source:(Baeldung, 2023)

In epsilon-greedy action selection, the agent uses both pathways, exploitation, to take advantage of prior knowledge and exploration to look for new options. Converging policy evaluation could achieve a similar effect.

We can use both Bellman equations to find a solution that will converge to the optimal solution by the policy and the value. This converging solution principle is called Generalized Policy Iteration (GPI), defined as any interaction of policy evaluation and policy improvement, independent of their granularity. In Fig 2, the GPI works as depicted in the conceptual model, that means the GPI is constantly switching between finding the optimal policy π^* (a decision

rule that maximises expected return from every state,) and the optimal state-value function v^* . Fig 3 depicts those two improving evaluations' convergence to the optimal solution.

Fig. 2: Depicment of Generalized Policy Iteration Mechanism

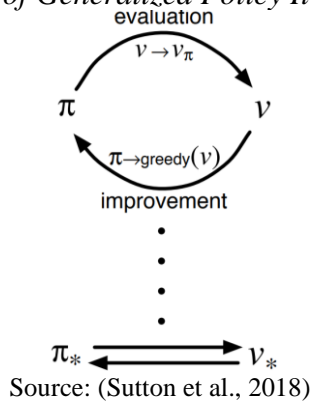
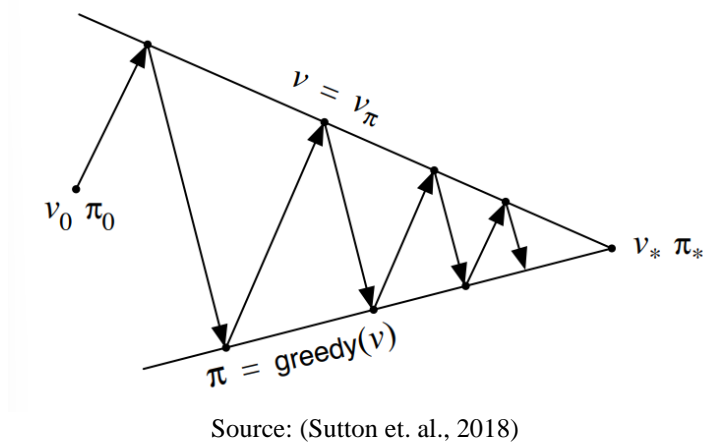


Fig. 3: Convergence to most effective solution possible

A geometric metaphor for convergence of GPI:



Dynamic programming is an excellent tool with wide application in various management optimisation problems. The first reason is that dynamic programming is centred around the effects of time. As we said before, ecological activities require some time to take effect, while this effect can be halted or boosted through additional activities in the time window. The second parameter, which could modify Bellman's optimisation equations, is stochastic values. In environment management, it is impossible to perfectly predict all outcomes and all the effects that could show up through the optimising process.

There are many models regarding dynamic programming. The most famous methods are:

- Workforce size model, where there could be a construction project that runs week by week. For every week the site manager knows the minimum number of workers that must be on the job to meet schedule targets. The company can raise the crew above that minimum by hiring extra people and can reduce it by letting workers go (So & Kek, 2020).

- Holding extra workers. Keeping more workers than the minimum costs money (overtime, idle time, salaries, benefits, and so on). The larger the surplus, the higher the weekly “holding” expense.
- Hiring new workers. Bringing additional people onto the crew from one week to the next triggers a separate “hiring” cost that covers recruiting, onboarding, and training.
- Firing workers. In this simplified version of the model, letting employees go is assumed to be free; no severance or rehiring penalty is counted.

The planner’s task is to decide, for each week of the project, how large the crew should be so that the total cost—holding extra staff plus any hiring charges—is as small as possible while never dropping below the required minimum. Dynamic-programming techniques can then be applied to find the cost-minimising schedule (Taha et al., 2017).

- Equipment-replacement model, that addresses that machines become more expensive to own the longer they remain in service: breakdowns grow more frequent, maintenance bills rise, and they earn less income. At some point it is cheaper to scrap an aging unit and buy a new one. The equipment-replacement model helps a manager decide exactly when that should happen over a planning horizon of several years. The life cycle of machines could be described in this model as (Lu & Wang, 2013) :
 - Operating year by year. At the start of every year, decision-maker must choose one of two actions:
 - Keep the current machine for another year, accepting its expected revenue and its operating and maintenance cost for that age
 - Replace it immediately with a brand-new unit, paying the purchase price and then earning the revenue and paying the costs associated with a first-year machine.
 - Age-dependent figures. For any given age of the machine the model tabulates three numbers:
 - the income the machine is expected to generate during that year,
 - the expense of running and maintaining it,
 - the amount that could be recover by selling or scrapping it at that moment (its salvage value).
 - Purchase cost. Buying a new machine always requires the same upfront investment, regardless of the calendar year in which it was done.

By comparing the stream of cash flows that results from “keep” versus “replace” decisions in every possible year, the model reveals the most economical schedule: how long to hold on to each machine before swapping it for a new one so that total profit over the entire planning period is maximised (or total cost is minimised). This approach follows the dynamic-programming treatment described by Taha et al. (2017).

- Investment-allocation model, that models situation where an investor plans to inject predetermined sums of cash at the beginning of each of the next several years. For every new deposit the investor can choose between banks, that offer different outcomes. To attract business, each bank also pays a bonus on fresh deposits. The bonus is calculated as a fixed percentage of that year’s new money, and the percentage can vary from year to year and from one bank to the other. Possible key rules of the scheme could be (Yu & Kuang, 2015):
 - Timing of bonuses. The bonus for a given deposit is credited at the **end** of the same year in which the deposit was made.

- Reinvestment options. At the start of the following year the investor may allocate that bonus—together with the next year’s scheduled cash contribution—to either bank, again earning interest and (if it counts as “new” money) another bonus.
- Lock-in period for principal. Once a deposit is placed in a bank, the principal must stay there until the final year of the multi-year horizon; it cannot be moved between banks or withdrawn early.

The decision problem is to work out, year by year, which bank should receive each scheduled deposit and each newly earned bonus so that the total wealth at the end of the planning horizon is maximised. This dynamic-programming formulation is adapted from the treatment in Taha et al. (2017).

- Inventory Models: DP has essential applications in inventory control.
- Reproduction Model
- Several algorithms emerged from the definition of dynamic programming, for example:
- Dijkstra algorithm: for the shortest path. Dijkstra’s algorithm, published in 1959 and named after its creator, Dutch computer scientist Edsger W. Dijkstra, can be applied to a weighted graph. The graph can be either directed or undirected. One stipulation to using the algorithm is that the graph needs to have a nonnegative weight on every edge (Abiy et al., 2017).

Dynamic programming is an excellent tool, even for more pressing problems that are now even more prominent than ever, such as environmental management. The first reason is that dynamic programming is centred around the effects of time (Munch & Brias, 2024). As we said before, ecological activities require some time to take effect, while this effect can be halted or boosted through additional activities in the time window. The second parameter, which could modify Bellman’s optimisation equations, is stochastic values (Davidsen et al., 2015). In environment management, it is impossible to perfectly predict all outcomes and all the effects that could show up through the optimising process.

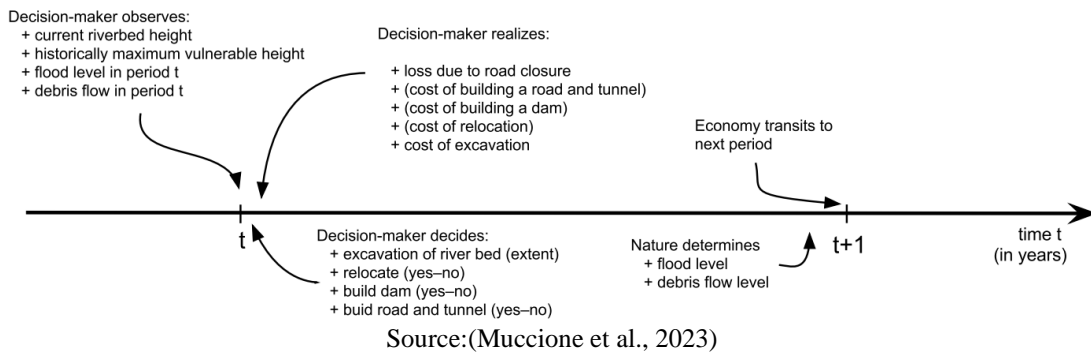
Several decision-support tools in environmental management exist to deal with different degrees of uncertainty. Cost-benefit analysis is a relatively common decision tool to inform adaptation employed for deterministic analysis, where variables such as costs and benefits of projects/programs are known and can be compared to justify interventions based on an efficient allocation of resources. At the other end of the spectrum, tools support decision-making under deep uncertainties (DMDU), where many plausible futures are possible, and a broad range of solutions or outcomes exist. DMDU tools such as robust decision-making, dynamic adaptive policy pathways and accurate options analysis can be handy in these cases (Muccione et al., 2023).

Uncertainties might not be profound but instead arise from a lack of information. In these cases, uncertainties can be dealt with by using iterative risk management approaches, which allow the integration of learning processes into decision-making cycles. Likewise, stochasticity or randomness can be appropriate representations for uncertainties in systems whose input parameters can be described probabilistically. Recent advances in computational economics have developed and applied methods of numerical dynamic programming that integrate stochastic or random uncertainties into the decision-making Natural Hazards processes at relatively low computational costs. Dynamic programming has been successfully employed in environmental decision-making to solve stochastic problems. It is a well-established approach in environmental management and the context of water resource management. Dynamic programming has been used to address stochasticity in water and food management and include ecological quality in the decision-maker’s utility function. Besides the direct integration of uncertainties into the decision-making process, dynamic programming has several other

advantages over alternative methods. One advantage is handling a long-time horizon (e.g., several decades with monthly intervals). A second advantage is that stochastic dynamic programming can use approximation methods to account for continuous decisions and state variables. In addition, stochastic dynamic programming is a flexible framework able to capture the optimal trade-offs and synergies in adaptation decision-making by modelling the risk preferences of a decision agent under uncertainties (Muccione et al., 2023).

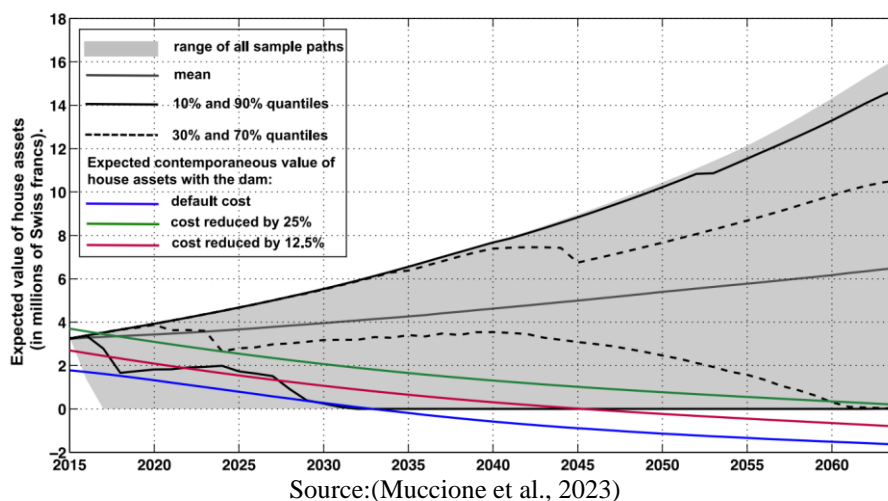
For example, Muccione et al. (2023) used dynamic programming to present a stylised application of stochastic dynamic programming for local adaptation decision-making for a small alpine community exposed to debris flows and floods. Fig 4 depicts a schematic representation of a single period in the dynamic decision framework. Note that at the beginning of each period, the decision maker observes a unique state of the world, i.e. time, riverbed height historically maximum vulnerable height of house asset and the realisations of the two shocks related to floods, and debris flow. Given that unique state of the world, the decision maker considers the possible future stochastic occurrence of floods and debris flow and chooses the optimal level of excavation and if the options of dam building or relocation and the building of an alternative road should be executed.

Fig 4: Time axis of decisions of example problem



This variable was transferred into the simulation, which simulated 10,000 randomised values. Then, calculations were conducted based on SDP (stochastic dynamic programming), and the results were statistics for the expected value of housing assets from 100,000 simulated paths and sensitivity regarding the cost of the dam. In addition, the expected value of the house assets is lower costs of building and maintaining the dam. All is shown in Fig.5 (Muccione et al., 2023), that is just for illustration of possible solution depiction.

Fig 5: Predicted development of the problem in future with the impact of decision maker



3 Results and discussion

We wanted to apply the methodology of dynamic programming in some sort of simple problem in financial management. This problem can be defined as the decision maker wants to maximise his profit from investing. The decision maker will make decisions is k years (total number of years), and he can make two decisions, he can withdraw some fraction of money or will let the stock money grow in value. There are however two main complications.

First is that the decision maker cannot withdraw all his money in the one go but he can withdraw just fraction of it. The second complication is that the investment project is risky and there is no guarantee that it is possible to withdraw money after all stages. We have chosen, substandard approach for financial management on how to calculate such problem, that is reproduction model (RM) that is widely used in environmental management.

In the RM literature, there are two models in which agents harvest a resource simultaneously. The first model focuses on situations where the resource users diminish the relative value per resource unit in the current period as their harvest level increases. However, the future value of the resource is undiminished. In contrast, in the dynamic RM, the resource's current users reduce the resource's level, thereby harming future users. Uncertainty of resource levels tends to promote over-harvesting, while resource scarcity induces greed. Thus, a more complete picture can be established by explicitly considering resource dynamics on the one hand and macroeconomic and social dynamics on the other.

There are many modifications to the RM. We even modified the RM for a specific problem unrelated to ecology, and the problem is an investment problem where the investor tries to make the most money possible from stocks. However, the investor can have three different approaches to strategies based on whether the investor prefers the money in cash or stock or does not mind, as long as the yield is at the maximum. To formulate such a problem, we must first declare sets and variables.

Description of the elements in Tab 4, 5, 6 and in formulas below are completely separated from previous notations of elements and formulas. Sets are described in Tab.4.

Tab. 4: Definition of sets

Symbol	Definition
$T=\{1,2,\dots,k\}$	Stages (years of the investment horizon), where k is total number of years.
$I=\{1,2,\dots,Z\}$	Admissible actions at each stage; Z is the number of distinct withdrawal policies that could be considered.

Source: Own Elaboration

A strategy over the whole horizon is an ordered k -tuple (i_1, i_2, \dots, i_k) with $i_t \in I$ for every t . The cartesian product I^k is simply the set of all such k -step strategies. Decision variables and auxiliary quantities are described in Tab.5.

Tab. 5: Definition of variables and auxiliary quantities

Symbol	Meaning (all evaluated at stage t unless stated otherwise)
N_t	Monetary value of the stock at the start of stage t .
π_{t,i_t}	Cash withdrawn in stage t when action i_t is chosen.
<i>CASH</i>	Total cash withdrawn over the entire horizon (objective).
<i>STOCK</i>	Final value of the remaining stock (secondary objective, if needed).

Source: Own Elaboration

Parameters are described in Tab.6.

Tab. 6: Definition of parameters

Symbol	Meaning
\mathcal{C}	Growth (or depreciation) factor that multiplies the stock from one stage to the next.
a_{t,i_t}	Percentage of stock withdrawn if action i_t is taken at stage t .
A	Regulatory or self-imposed upper bound on the withdrawal percentage in any single stage.

Source: Own Elaboration

There are two objective functions, primary and secondary. Primary goal – maximise total cash withdrawals:

$$CASH = \max_{(i_1, \dots, i_k) \in I^k} \sum_{t=1}^k \pi_{t,i_t} \quad (13)$$

Optional secondary goal – maximise residual stock at the horizon

$$STOCK = N_{k+1} \quad (14)$$

where N_{k+1} is obtained from the recursion below.

Constraints :

1. Withdrawal-percentage limit

$$a_{t,i_t} \leq A \quad t \in T, i_t \in I \quad (15)$$

2. Cash actually withdrawn in stage t

$$\pi_{t,i_t} = a_{t,i_t} (N_t \mathcal{C}) \quad t \in T, i_t \in I \quad (16)$$

3. Evolution of the stock

$$N_{t+1} = (N_t \mathcal{C}) - \pi_{t,i_t} \quad t = 1, \dots, k-1 \quad t \in T, i_t \in I \quad (17)$$

We used this model on the illustrative example of problem. The data is made up, however it reflects possible data from reality. We have calculated four different scenarios of this problem. The common values of variables were:

$$N_1 = 100$$

$$k=3$$

$$Z=2$$

For all illustrative scenarios, there are two possible actions $Z: (a_{t,1_t}; a_{t,2_t}) = (0; A)$.

The changing parameters in each of these problems were \mathcal{C} and A . In the Tab.7 are marked these four various combinations:

Tab. 7: Four different illustrative scenarios.

1. Combination: $\mathcal{C}=1,25; A=50\%$	2. Combination: $\mathcal{C}=1,2; A=50\%$
3. Combination: $\mathcal{C}=1,5; A=33\%$	4. Combination: $\mathcal{C}=1,35; A=80\%$

Source: Own Elaboration

We could use Excel and the add-on Solver to find the optimal solution for this model. This model is nonlinear, and for that matter we cannot use the simplex LP algorithm in the Solver, so we used the Generalized Reduced Gradient algorithm instead. The calculations of a problem with different input parameters, noted in Tab. 7, can be found in Tab. 8.

Decision-makers could have three approaches to this problem. First, he does not trust the investment project and only cares about cash, which he has at the end of the investment period, which has the most liquid form, approach *CASH*. The second approach is that the decisionmaker fully trusts the investment project and believes that the offered stock will have total value even after time k , and he wants to have as much money bound to those stocks as possible, that is approach *STOCK*. The third approach is that the decisionmaker believes in the investment project but not fully and wants to have as much money as possible at the end of time k ; that approach is called *CASH+STOCK*.

Tab. 8: Results of Solver calculation of various combinations of problems parameters

	$C=1,25;A=50\%$	$C=1,2;A=50\%$	$C=1,5;A=33\%$	$C=1,35;A=80\%$
<i>CASH</i>	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,50,50) CASH=126.96	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (50,50,50) CASH=117.6	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (33,33,33) CASH=149.25	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,80) CASH=196.83
<i>STOCK</i>	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0) STOCK=195.31	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0) STOCK=172.8	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0) STOCK=337.5	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0) STOCK=245.04
<i>CASH+STOCK</i>	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0 or 50) CASH+ STOCK=195.31	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0 or 50) CASH+ STOCK=172.8	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0 or 33) CASH+ STOCK=337.5	$(a_{1,i_1^*}, a_{2,i_2^*}, a_{3,i_3^*})$ = (0,0,0 or 80) CASH+ STOCK=245.04

Source: Own Elaboration

4 Conclusion

This article aimed to show how dynamic programming could be used to find the optimal use of processes in a simple problem. We believe that this feat was accomplished. We showed sample problems and the means of solving them. We even created a custom model that could be further extended to solve various economic or ecology-based problems. We used this method on a test problem, using manual calculation and even calculation based on the non-linear programming algorithm called Generalized Reduced Gradient.

The following research will focus on further developing our custom model. It will be developed to calculate various management optimisation problems using replication model inspired mathematical model, as we believe it could bring different views on optimising problems in dynamic and deterministic settings, leading to more accurate, faster results. It can also be used to merge or synchronise different management optimisation problems, such as environmental and financial management problems, leading to a more sustainable and profitable future.

The paper was elaborated in the scope of the grant assignment ESG I-24-111-00 Exploring the use of artificial intelligence to support the creation of scientific research papers and the data mining within them

References

1. Abiy, T., Pang, H., & Williams, C. (2017). Dijkstra's Shortest Path Algorithm: Brilliant math & science wiki. Brilliant. <https://brilliant.org/wiki/dijkstras-short-path-finder>
2. Baeldung, W. (2023, March 24). Epsilon-Greedy Q-Learning. Baeldung on Computer Science. <https://www.baeldung.com/cs/epsilon-greedy-q-learning>
3. Bellman, R. (1954). *The theory of dynamic programming*. Bulletin of the American Mathematical Society, 60(6), 503-515. <https://doi.org/10.1090/S0002-9904-1954-09848-8>
4. Carlson, D. A., Haurie, A., & Leizarowitz, A. (1991). *Infinite-horizon optimal control: Deterministic & stochastic systems*. Springer. <https://doi.org/10.1007/978-3-642-76755-5>
5. Chebil, K., & Khemakhem, M. (2015). A dynamic programming algorithm for the Knapsack problem with setup. *Computers & Operations Research*, 64, 40–50. <https://doi.org/10.1016/j.cor.2015.05.005>
6. Davidsen, C., Pereira-Cardenal, S. J., Liu, S., Mo, X., Rosbjerg, D., & Bauer-Gottwein, P. (2015). *Using stochastic dynamic programming to support water resources management in the Ziya River Basin, China*. *Journal of Water Resources Planning and Management*, 141(7), 04014086. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000482](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000482)
7. Jaffar, J., Santosa, A. E., & Voicu, R. (2008). *Efficient memoisation for dynamic programming with ad-hoc constraints*. In *Proceedings of AAI-08* (pp. 297-302). <https://doi.org/10.1609/aaai.v22i2.2190>
8. Laščiak, A., Sojka, J., Šimkovic, J., Mañas, M., Hozlár, E., Unčovský, L., Chobot, M., Hušek, R., & Ulašín, V. (1983). *Optimálne programovanie*. Alfa.
9. Lu, Q., & Wang, Z. (2013). *A stochastic dynamic-programming approach for the equipment-replacement optimisation problem under utilisation uncertainty*. *Transportation Research Procedia*, 7, 400-409. <https://doi.org/10.1016/j.trpro.2013.06.013>
10. Muccione, V., Lontzek, T., Huggel, C., Ott, P., & Salzmann, N. (2023). An application of dynamic programming to local adaptation decision-making. *Natural Hazards*, 119(1), 523–544. <https://doi.org/10.1007/s11069-023-06135-2>
11. Munch, S. B., & Brias, A. (2024). *Empirical dynamic programming for model-free ecosystem-based management*. *Methods in Ecology and Evolution*, 15(4), 769–778. <https://doi.org/10.1111/2041-210X.14302>
12. Sieniutycz, S., & Jeżowski†, J. (2013). Dynamic Optimization Problems. *Energy Optimization in Process Systems and Fuel Cells*, 45–84. <https://doi.org/10.1016/b978-0-08-098221-2.00002-3>
13. So, M. K., & Kek, S. L. (2020). *Workforce-size problem in manufacturing with a dynamic-programming approach*. *AIP Conference Proceedings*, 2266, 090005. <https://doi.org/10.1063/5.0018444>
14. Sutton, R. S., & Barto, A. G. (2018). *Bellman equations and dynamic programming - university of Texas at Austin*. Introduction to Reinforcement Learning. https://login.cs.utexas.edu/sites/default/files/legacy_files/research/documents/6%20Bellman%20Eqs%20and%20DP.pdf
15. Taha, H. A. (2017). *Operations research: An introduction*. Pearson.
16. Wimmer, S., Hu, S., & Nipkow, T. (2018). *Verified memoisation and dynamic programming*. In *Interactive Theorem Proving 2018* (LNCS 10895, pp. 579-596). https://doi.org/10.1007/978-3-319-94821-8_34

17. Yu, F., & Kuang, H. (2015). *Port multi-period investment-optimisation model based on supply-demand matching using dynamic programming*. *Journal of Shipping & Trade*, 2(4), 75-89. <https://doi.org/10.1515/JSSI-2015-0077>

Optimálne zaistenie poisťovateľa z pohľadu teórie rizika

Optimal insurance coverage for an insurer from the perspective of risk theory

Ján Gogola¹

Abstrakt

V aktuárstve v teórii krachu sa uplatňujú matematické modely na opis zraniteľnosti poisťovateľa voči krachu. Teoretický základ teórie krachu opisuje prebytok poisťovateľa v akomkoľvek budúcom čase ako náhodnú premennú, ktorej hodnota závisí od prijatého poistného a vyplatených poistných plnení. Poisťovňa chce udržať pravdepodobnosť krachu na čo najmenšej úrovni, alebo aspoň pod vopred stanovenú hranicu. Lundbergova nerovnosť poskytuje hornú hranicu pravdepodobnosti krachu v nekonečnom čase a je jedným z najznámejších výsledkov v teórii krachu. Jednou z možností pre poisťovateľa, ktorý chce znížiť pravdepodobnosť krachu, je uzavrieť zaistenie. Budeme uvažovať o dvoch typoch zaistenia: proporcionálnom zaistení a zaistení škodovej nadmierky. Zaistenie (z pohľadu poisťovateľa) považujeme za optimálne, ak minimalizuje pravdepodobnosť krachu. Cieľom tohto článku je ilustrovať, ako zmeny v parametri rizikovej prirážky poistného (používaného poisťovateľom i zaisťovateľom) ovplyvňujú pravdepodobnosť krachu pri oboch druhoch zaistenia. Nájdeme aj optimálny typ zaistenia za určitých podmienok.

Kľúčové slová

Zaistenie, Teória krachu, proporcionálne zaistenie, zaistenie škodovej nadmierky (XL zaistenie), koeficient korekcie, vlastný vrub, zložený Poissonov proces.

Abstract

In actuarial science ruin theory uses mathematical models to describe an insurer's vulnerability to ruin. Theoretical foundation of ruin theory describes an insurance company who experiences two opposing cash flows: incoming cash premiums and outgoing claims. The insurer's surplus at any future time is a random variable since its value depends on premiums and claims. The insurer want to keep the probability of ruin as small as possible, or at least below a predetermined bound. Lundberg's inequality provides an upper bound for the probability of ruin in infinite time and is one of the most famous results in ruin theory. One of the options for an insurer who wishes to reduce the probability of ruin is to effect reinsurance. We shall consider two kinds of reinsurance arrangement: proportional and excess of loss reinsurance. We could consider a reinsurance arrangement (from an insurer point of view) to be optimal if it minimizes the probability of ruin. The goal of this paper is to illustrate how changes in the premium loading factor (used by insurer and reinsurer) affect the probability of ruin in both kind of reinsurance. We find also the optimal type of reinsurance under certain conditions.

Key words

Reinsurance, Ruin theory, Proportional and Excess of Loss Reinsurance, Adjustment coefficient, Retention level, Compound Poisson process

JEL classification C18, G22

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra matematiky a aktuárstva, Dolnozemska cesta 1, 852 35 Bratislava, e-mail: jan.gogola@euba.sk.

1 Úvod

Teória krachu sa zaoberá modelmi kolektívneho rizika v dlhšom časovom období a slúži k posúdeniu náhodného výkyvu hodnoty prebytku poisťovne v časovom intervale niekoľkých rokov. Pod krachom poisťovne rozumieme skutočnosť, keď v určitom čase prebytok poisťovne U po prvýkrát klesne do záporných hodnôt. (Horáková, Mucha, 2006)

Zaistenie je opätovné poistenie časti rizika. Priamy poisťovateľ (alebo prvopoisťovateľ) postupuje časť svojho rizika zaistovní. Tú časť rizika, ktorú si ponecháva, sa nazýva vlastný vrub. Poisťovateľ za presun časti rizika platí zaistné. Zaistením sa poisťovateľ zbavuje časti rizika, ktoré je vyššie než môže poistiť alebo si ponechať na vlastný vrub. Zaistením poisťovateľ dosahuje väčšiu homogenitu poistného kmeňa.

Z hľadiska spôsobu delenia si poistného plnenia medzi poisťovateľom a zaistovníou delíme zaistenie na proporcionálne a neproporcionálne.

Cieľom nášho príspevku je preskúmať aký vplyv majú vybrané formy zaistenia a ich rozsah na pravdepodobnosť krachu.

2 Formulácia problému

Nech náhodná premenná X predstavuje hodnotu poistného plnenia individuálneho rizika. Predpokladajme, že existujú jej momenty $m_1(X) = E(X)$ a $m_2(X) = E(X^2)$, ktoré sú konečné.

Pri proporcionálnom zaistení sa poistné plnenie delí medzi poisťovateľa a zaistovníu vo vopred zjednanom pomere α , zvaného podiel na vlastný vrub priameho poisťovateľa. Zaistovník platí fixný podiel $1 - \alpha$ z každého poistného plnenia, bez ohľadu na ich výšku. Potom tá časť, ktorú hradí poisťovateľ, má hodnotu

$$Y = \alpha \cdot X. \quad (1)$$

Zaistovníka hradí zostávajúcu časť $1 - \alpha$ z každého poistného plnenia t.j. jej plnenie je

$$Z = (1 - \alpha) \cdot X. \quad (2)$$

Neproporcionálne zaistenie škodovej nadmierky (ang. *excess of loss XL*) je založené na dohode medzi poisťovateľom a zaistovníou, podľa ktorej zaistovníka vyplatí poisťovateľovi plnenie presahujúce vopred zjednanú sumu. Táto suma sa nazýva maximálna úroveň M na vlastný vrub poisťovateľa alebo priorita. Pravdepodobnosť, že výška plnenia je vyššia, než priorita je vo všeobecnosti malá.

Priebeh niektorých číselných charakteristík v závislosti od priority môžeme nájsť v (Fecenko, 2006).

Pri neproporcionálnom zaistení poisťovateľ hradí poistné plnenie v celej výške, pokiaľ hodnota plnenia X nepresiahne vlastný vrub M . Pri poistnom plnení väčšom než M , poisťovateľ hradí sumu M a zostatok zaplatí zaistovníka.

Potom pre plnenie poisťovateľa platí:

$$Y = \min\{X; M\}. \quad (3)$$

a pre plnenie zaistovne:

$$Z = \max\{0; X - M\}. \quad (4)$$

Vo všeobecnosti platí $X = Y + Z$.

3 Použité metódy

Prebytok poisťovateľa $U(t)$ v čase t možno vyjadriť v tvare

$$U(t) = U_0 + c \cdot t - S(t), \quad (5)$$

kde U_0 je počiatočná hodnota rezervného fondu alebo počiatočný kapitál poisťovateľa,

c je konštantná miera intenzity príjmu poistného. Predpokladáme spojitý príjem poistného v čase,

$S(t)$ je výška celkového poistného plnenia v čase t .

Prebytok poisťovateľa je náhodná premenná, ktorej hodnota závisí od výšky plnenia. V našom článku predpokladáme, že proces celkových poistných plnení za jednotku času je zložený Poissonov proces s parametrom λ .

Pravdepodobnosť krachu $\Psi(U)$ v nekonečnom časovom horizonte definujeme ako $\Psi(U) = P[U(t) < 0]$ pre nejaké $0 < t < +\infty$.

Jedným z najdôležitejších výsledkov teórie krachu je Lundbergova nerovnosť (Dickson, Waters, 1993), (Horáková, Mucha, 2006), (Pacáková, 2004), umožňujúca zhora ohraničiť pravdepodobnosť krachu poisťovne v nekonečnom časovom horizonte.

Podľa tejto nerovnosti pre každú hodnotu U počiatočného rezervného fondu poisťovne platí

$$\Psi(U) \leq e^{-RU}. \quad (6)$$

Pričom R je parameter, známy ako koeficient úpravy, resp. koeficient korekcie (ang. *adjustment coefficient* (Dickson, Waters, 1993)). Jeho hodnota závisí od rozdelenia individuálnych poistných plnení a na intenzite prijímania poistného. R môžeme považovať za mieru rizika.

V poisťovníctve sa pravá strana Lundbergovej nerovnosti používa ako aproximácia pri stanovení pravdepodobnosti krach poisťovne, ktorá sa znižuje s rastúcou hodnotou koeficientu úpravy R .

Preto v našom príspevku budeme maximalizovať koeficient úpravy R , pretože tým minimalizujeme Lundbergovu hornú hranicu pre pravdepodobnosť krachu.

Koeficient úpravy pre zložený Poissonov proces definujeme ako jediný kladný koreň rovnice

$$\lambda + c \cdot R = \lambda \cdot M_X(R), \quad (7)$$

kde $M_X(R)$ je momentová vytvárajúca funkcia v bode R .

Rovnica (7) je implicitná rovnica pre vyjadrenie koeficientu R . Pre niektoré typy rozdelení $F(x)$ individuálnych poistných plnení je možné nájsť explicitné vyjadrenie, pre niektoré (bohužiaľ väčšinu) iba numerické (približné) riešenie.

V prípade, keď individuálne poistné plnenie má exponenciálne rozdelenie s parametrom δ , koeficient úpravy R má jednoduché vyjadrenie $R = \frac{\delta \cdot \theta}{1 + \theta}$,

z ktorého vyplýva, že hodnota R závisí iba na rizikovej prirážke θ poisťovateľa a na parametre δ exponenciálneho rozdelenia.

Pokiaľ individuálne poistné plnenie má exponenciálne rozdelenie $Exp(\delta)$, je možné odvodiť aj jednoduchý a užitočný vzťah pre pravdepodobnosť krach $\Psi(U)$ v tvare

$$\Psi(U) = \frac{1}{1 + \theta} \cdot e^{-\frac{\delta \cdot \theta}{1 + \theta} U}. \quad (8)$$

Odvodenie tohoto vzťahu je možné nájsť v (Gerber, 1979).

4 Rozbor problematiky

Zaistenie je jednou z možností, ako môže poisťovňa minimalizovať pravdepodobnosť krachu, a to maximalizáciou koeficientu úpravy R .

Predpokladajme, že zníženie variability poistných plnení zvýši istotu poisťovateľa, a tým sa zníži aj pravdepodobnosť krachu. Z tohoto pohľadu sa bude zaistovacia zmluva, resp. rozsah zaistenia považovať za optimálny, pokiaľ povedie k minimalizácii pravdepodobnosti krachu $\Psi(U)$. Pretože je všeobecne ťažké nájsť jej presné riešenie, pokiaľ poznáme vzťah medzi $\Psi(U)$ a R , budeme maximalizovať hodnotu R .

Uvažujme dva druhy zaistenia:

- proporcionálne zaistenie (*proportional reinsurance*),
- zaistenie škodovej nadmierky (*excess of loss reinsurance*)

4.1 Maximalizácia koeficientu úpravy R pri proporcionálnom zaistení

Predpokladajme, že poisťovňa uzavrela proporcionálne zaistenie poistných plnení s podielom α na vlastný vrub.

Ďalej predpokladajme príjem poistného c_p za jednotku času (napr. rok) pred zaistením v tvare

$$c_p = (1 + \theta) \cdot \lambda \cdot m_1, \quad (9)$$

čo predstavuje priemerné celkové poistné plnenie za jednotku času pre zložený Poissonov proces s parametrom λ a s rizikovou prirážkou θ .

Predpokladáme, že zaistné sa počíta podľa vzťahu

$$c_z = (1 + \xi) \cdot (1 - \alpha) \cdot \lambda \cdot m_1, \quad (10)$$

kde ξ je riziková prirážka zaistovateľa.

Pretože zaistovňa platí podiel $1 - \alpha$ z každého poistného plnenia, výraz $E(Z) = (1 - \alpha) \cdot \lambda \cdot m_1$, predstavuje očakávané plnenie zaistovne za jednotku času. Z toho pre čistý príjem poistného priameho poisťovateľa po zaistení za jednotku času dostávame vzťah

$$c^* = [(1 + \theta) - (1 + \xi) \cdot (1 - \alpha)] \cdot \lambda \cdot m_1. \quad (11)$$

Ďalej budeme predpokladať, že $\xi \geq \theta$. V opačnom prípade by bolo možné pre poisťovateľa presunúť celé riziko na zaistovňu a mať istý zisk. Napríklad keby bolo $\theta = 0,2$ a $\xi = 0,1$, tak by poisťovateľ prijal poistné vo výške $1,2 \cdot \lambda \cdot m_1$. Potom by zaistil celé riziko za zaistné $1,1 \cdot \lambda \cdot m_1$ a ponechal si istý zisk $0,1 \cdot \lambda \cdot m_1$.

Aby bol čistý príjem poistného pre priameho poisťovateľa po zaistení kladný, musí platiť nerovnosť

$$(1 + \theta) > (1 + \xi) \cdot (1 - \alpha) \quad \text{resp.} \quad \alpha > \frac{\xi - \theta}{1 + \xi}. \quad (12)$$

Existuje však značný tlak na poisťovateľa, lebo jeho čistý príjem za jednotku času po zaistení $\alpha \cdot \lambda \cdot m_1$ musí byť vyšší než celkové očakávané plnenie za jednotku času. Preto požadujeme, aby platila nerovnosť $(1 + \theta) - (1 + \xi) \cdot (1 - \alpha) > \alpha$, alebo po úprave

$$\alpha > 1 - \frac{\theta}{\xi} \geq 0. \quad (13)$$

Posledná nerovnosť stanovuje minimálnu hodnotu rizikovej prirážky poisťovateľa, pretože $1 - \frac{\theta}{\xi} \geq \frac{\xi - \theta}{1 + \xi}$ pre $\theta \leq \xi$. Pokiaľ sú rizikové prirážky rovnaké, tak $\alpha > 0$ a v tom prípade môže vlastný vrub α nadobudnúť akúkoľvek hodnotu z intervalu $(0, 1)$. Pokiaľ je však $\theta < \xi$, tak poisťovateľ si musí ponechať časť rizika.

Určme koeficient úpravy R ako funkciu podielu na vlastný vrub α , pokiaľ poistné plnenie má exponenciálne rozdelenie s distribučnou funkciou $F(x) = 1 - e^{-0,1 \cdot x}$, $x > 0$, a poisťovňa aj zaistovňa používajú rovnakú rizikovú prirážku t.j. $\theta = \xi$.

Rozdelenie individuálneho plnenia poisťovateľa po zaistení je opäť exponenciálne, s parametrom $\frac{0,1}{\alpha}$. Vyplýva to z toho, že pokiaľ poistné plnenie poisťovateľa je $Y = \alpha \cdot X$, potom

$$F(y) = P(Y \leq y) = P\left(X \leq \frac{y}{\alpha}\right) = 1 - e^{-\frac{0,1 \cdot y}{\alpha}}.$$

Preto rovnica určujúca R má tvar

$$\begin{aligned}\lambda + c \cdot R &= \lambda \cdot M_Y(R) \\ \lambda + (1 + \theta) \cdot \lambda \cdot 10 \cdot \alpha \cdot R &= \lambda \cdot \int_0^\infty e^{Ry} \cdot \frac{0,1}{\alpha} \cdot e^{-\frac{0,1 \cdot y}{\alpha}} dy \\ 1 + (1 + \theta) \cdot 10 \cdot \alpha \cdot R &= \frac{1}{1 - 10 \cdot \alpha \cdot R} \\ R &= \frac{\theta}{(1 + \theta) \cdot 10 \cdot \alpha}, \text{ pre } 0 < \alpha \leq 1.\end{aligned}\tag{14}$$

Vidíme, že R je klesajúcou funkciou vlastného vrubu α . To znamená, že pokiaľ α klesá, rastie hodnota R , a teda klesá i pravdepodobnosť krachu poisťovne $\Psi(U)$ v nekonečnom časovom horizonte. Obrátene, pokiaľ α rastie, klesá hodnota R , a teda rastie hodnota pravdepodobnosti krachu $\Psi(U)$ v nekonečnom časovom horizonte. To je prirodzené, lebo čím menší vlastný rub, tým menšie je riziko krachu pre poisťovateľa.

Praktická ukážka 1. Uvažujme prípad, keď riziková prirážka poisťovateľa a zaistovne sa líši. Nech je distribučná funkcia poistného plnenia opäť $F(x) = 1 - e^{-0,1 \cdot x}$, $x > 0$, a riziková prirážka poisťovateľa a zaistovne sú

- a) $\theta = 0,1$ a $\xi = 0,15$,
- b) $\theta = 0,1$ a $\xi = 0,30$.

a) Čistý príjem poisťovateľa po zaistení podľa vzťahu (11) je $c^* = (11,5 \cdot \alpha - 0,5) \cdot \lambda$.

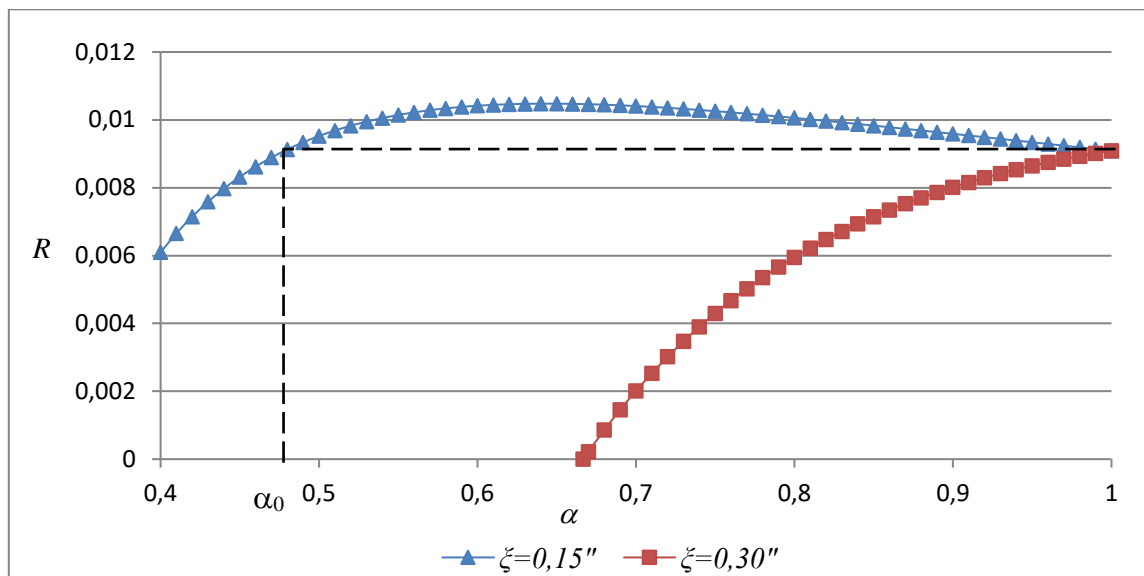
Nerovnosť (13) nám hovorí, že $\frac{1}{3} < \alpha \leq 1$, t.j. poisťovateľ si musí ponechať aspoň tretinu z každého plnenia.

Rovnica, ktorá definuje R , má tvar $\lambda + (11,5 \cdot \alpha - 0,5) \cdot \lambda \cdot R = \frac{\lambda}{1 - 10 \cdot \alpha \cdot R}$,

čo vedie k $R = \frac{1,5 \cdot \alpha - 0,5}{10 \cdot (11,5 \cdot \alpha^2 - 0,5 \cdot \alpha)}$ alebo $R = \frac{3 \cdot \alpha - 1}{230 \cdot \alpha^2 - 10 \cdot \alpha}$, pre $\frac{1}{3} < \alpha \leq 1$.

Vidíme, že koeficient úpravy R opäť závisí na hodnote vlastného vrubu α .

Obr. 1. ukazuje koeficient úpravy R ako funkciu podielu α na vlastný vrub priameho poisťovateľa pre hodnoty α z intervalu 0,4 až 1.

Obr. 1: Koeficient R ako funkcia podielu α na vlastný vrub

Zdroj: vlastné spracovanie

Z grafu vidíme, že existuje určité rozpätie hodnôt α , $\alpha_0 < \alpha < 1$, také, že pokiaľ je podiel α na vlastný vrub z tohoto intervalu, potom koeficient R prekročí hodnotu, ktorú dosiahneme pri $\alpha = 1$, teda pre prípad poistenia bez zaistenia.

V našom prípade je $\alpha_0 = 0,4783$.

Nájdime ešte takú hodnotu α , ktorá maximalizuje hodnotu R .

Deriváciou R podľa α dostaneme $\frac{dR}{d\alpha} = \frac{-690 \cdot \alpha^2 + 460 \cdot \alpha - 10}{(230 \cdot \alpha^2 - 10 \cdot \alpha)^2}$.

Maximálna hodnota R potom dostaneme, ak $\frac{dR}{d\alpha} = 0$.

Korene rovnice sú: $\alpha_1 = 0,0225$ a $\alpha_2 = 0,6442$. Potom maximálnu úroveň má koeficient úpravy R pre hodnotu, ktorá je z intervalu $(\frac{1}{3}; 1)$, t.j. $\alpha = 0,6442$.

Teda, pokiaľ je naším cieľom minimalizovať pravdepodobnosť krachu, tak optimálny podiel na vlastný vrub poisťovateľa má hodnotu 0,6442.

Poznamenajme ešte, že optimálne riešenie v jednom zmysle nemusí byť optimálnym v inom zmysle.

Napríklad, ak sa poisťovateľ rozhodne nevyužiť zaistenie (ponechá si všetko riziko), tak jeho očakávaný zisk za jednotku času je $\theta \cdot \lambda \cdot m_1$. Konkrétne v našom prípade $0,1 \cdot \lambda \cdot 10 = \lambda$.

Pokiaľ sa poisťovateľ rozhodne uskutočniť zaistenie s podielom $\alpha = 0,6442$ na vlastný vrub, tak jeho očakávaný zisk za jednotku času je „iba“ $0,4663 \cdot \lambda$, t.j. rozdiel príjmu

$$c^* = [(1 + \theta) - (1 + \xi) \cdot (1 - \alpha)] \cdot \lambda \cdot m_1$$

$$c^* = [(1 + 0,1) - (1 + 0,15) \cdot (1 - 0,6442)] \cdot \lambda \cdot 10 = 6,9079 \cdot \lambda$$

a očakávaného plnenia

$$\alpha \cdot \lambda \cdot m_1 = 6,442 \cdot \lambda.$$

b) Čistý príjem poisťovateľa po zaistení podľa vzťahu (11) je $c^* = (13 \cdot \alpha - 2) \cdot \lambda$.

Nerovnosť (13) nám hovorí, že $\frac{2}{3} < \alpha \leq 1$, t.j. poisťovateľ si musí ponechať aspoň dve tretiny z každého plnenia.

Rovnica, ktorá definuje R , má tvar $\lambda + (13 \cdot \alpha - 2) \cdot \lambda \cdot R = \frac{\lambda}{1 - 10 \cdot \alpha \cdot R}$,

čo vedie k $R = \frac{3 \cdot \alpha - 2}{10 \cdot (13 \cdot \alpha^2 - 2 \cdot \alpha)}$, pre $\frac{2}{3} < \alpha \leq 1$.

Z Obr. 1. vidíme, že R je rastúcou funkciou α , poisťovateľ by si musel ponechať celé riziko, aby maximalizoval koeficient úpravy R . V tomto prípade náklady na zaistenie prevážia nad znížením vo variabilite poistného plnenia.

4.2 Maximalizácia koeficientu úpravy R pri zaistení škodovej nadmierky

Pri zaistení škodovej nadmierky (XL - excess of loss) s maximálnou úrovňou M na vlastný vrub poisťovateľa sa zaistné poisťovne za jednotku času rovná

$$c_z = (1 + \xi) \cdot \lambda \cdot E(Z), \quad (15)$$

kde ξ je riziková prirážka zaistovateľa a $Z = \max\{0, X - M\}$.

Pre poistné plnenie Y poisťovateľa platí $Y = \min\{X, M\}$.

Čistý príjem poistného priameho poisťovateľa za jednotku času po zaistení je

$$c^* = (1 + \theta) \cdot \lambda \cdot m_1 - (1 + \xi) \cdot \lambda \cdot E(Z). \quad (16)$$

Z týchto predpokladov získame rovnicu definujúcu R v tvare

$$\begin{aligned} \lambda + c^* \cdot R &= \lambda \cdot M_Y(R) \\ \lambda + c^* \cdot R &= \lambda \cdot \left[\int_0^M e^{R \cdot x} \cdot f(x) dx + e^{R \cdot M} \cdot (1 - F(M)) \right]. \end{aligned} \quad (17)$$

Praktická ukážka 2. Nech individuálne poistné plnenie X má rovnomerné rozdelenie na intervale $(0; 20)$, teda $f(x) = 0,05$ pre $0 < x < 20$. Určíme koeficient úpravy R , keď poisťovňa aj zaistovňa používajú rovnakú rizikovú prirážku, t.j. $\theta = \xi = 0,1$.

Zo zadania vyplýva:

$$E(Z) = \int_M^\infty (x - M) \cdot f(x) dx = \int_M^{20} (x - M) \cdot 0,05 dx = 10 - M + 0,025 \cdot M^2,$$

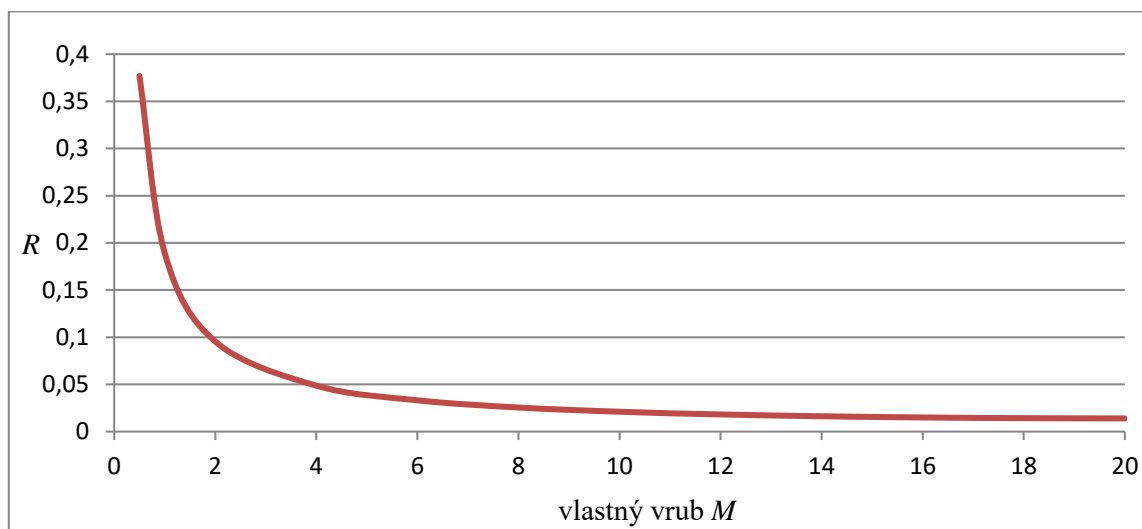
$$\text{a} \quad M_Y(R) = \int_0^M e^{R \cdot x} \cdot 0,05 dx + e^{R \cdot M} (1 - 0,05 \cdot M)$$

$$M_Y(R) = \frac{0,05}{R} \cdot (e^{R \cdot M} - 1) + e^{R \cdot M} \cdot (1 - 0,05 \cdot M).$$

Potom dostaneme rovnicu, ktorá definuje R v tvare

$$\begin{aligned} \lambda + (1,1 \cdot \lambda \cdot M - 1,1 \cdot 0,025 \cdot \lambda \cdot M^2) \cdot R &= \lambda \cdot \left[\frac{0,05}{R} \cdot (e^{R \cdot M} - 1) + e^{R \cdot M} (1 - 0,05 \cdot M) \right] \\ 1 + (1,1 \cdot M - 1,1 \cdot 0,025 \cdot M^2) \cdot R &= \frac{0,05}{R} (e^{R \cdot M} - 1) + e^{R \cdot M} (1 - 0,05 \cdot M). \end{aligned} \quad (18)$$

Rovnicu (18) môžeme spočítať numericky pomocou nástroja „Solver“ programu MS Excel (Brož, 2006). Obr. 2. nám ukazuje koeficient úpravy R ako funkciu M .

Obr. 2: R ako funkcia M , pre $\theta = \xi = 0,1$.

Zdroj: vlastné spracovanie

Vidíme, že každá úroveň M na vlastný vrub je možná, keď sa rizikové prirážky rovnajú. R je klesajúcou funkciou M , a $\lim_{M \rightarrow 0^+} R(M) = +\infty$.

Praktická ukážka 3. Nech platia rovnaké predpoklady ako v predchádzajúcom príklade, iba $\theta < \xi$. Určme koeficient úpravy R , ak

- $\theta = 0,1$ a $\xi = 0,15$,
- $\theta = 0,1$ a $\xi = 0,30$.

a) Pokiaľ je $\theta = 0,1$ a $\xi = 0,15$, tak $c^* = 1,1 \cdot \lambda \cdot 10 - 1,15 \cdot \lambda \cdot (10 - M + 0,025 \cdot M^2)$ resp. po úprave

$$c^* = -0,5 \cdot \lambda + 1,15 \cdot \lambda \cdot M - M - 1,15 \cdot 0,025 \cdot \lambda \cdot M^2.$$

Táto hodnota musí prevýšiť očakávané plnenie poisťovateľa očistené o zaistenie.

Očakávané plnenie poisťovateľa potom je:

$$\lambda \cdot E(X) - \lambda \cdot E(Z) = \lambda \cdot 10 - \lambda \cdot (10 - M + 0,025 \cdot M^2) = \lambda \cdot (M - 0,025 \cdot M^2).$$

Dostávame nerovnicu $-0,5 + 1,15 \cdot M - 1,15 \cdot 0,025 \cdot M^2 > M - 0,025 \cdot M^2$

alebo po úprave

$$3 \cdot M^2 - 120 \cdot M + 400 < 0. \quad (19)$$

Riešením nerovnice (19) je interval (3,67; 36,33).

To znamená, že minimálna úroveň M na vlastný vrub je 3,67.

Teda rovnako ako pri proporcionálnom zaistení s rôznymi hodnotami θ a ξ , existuje istá minimálna úroveň, ktorú si poisťovateľ musí ponechať.

- Pokiaľ je $\theta = 0,1$ a $\xi = 0,30$,

dostávame nerovnicu $-2 + 1,3 \cdot M - 1,3 \cdot 0,025 \cdot M^2 > M - 0,025 \cdot M^2$

alebo po úprave

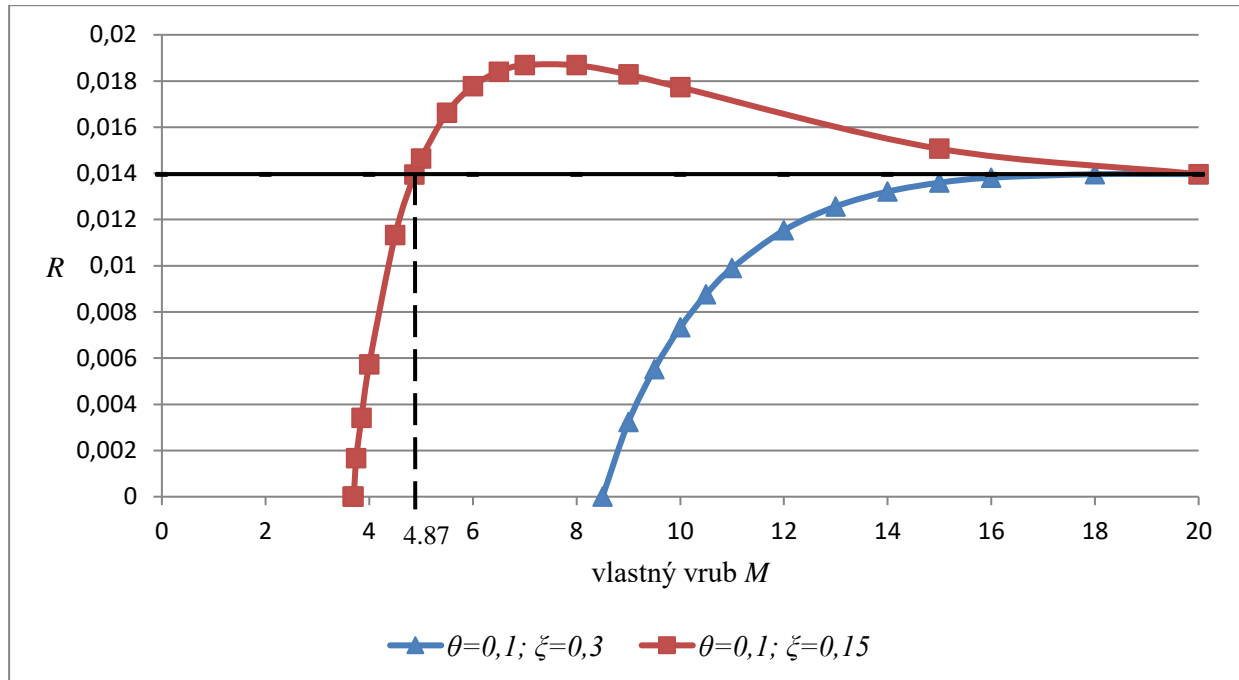
$$3 \cdot M^2 - 120 \cdot M + 800 < 0. \quad (20)$$

Riešením nerovnice (20) je interval (8,45; 31,55).

Teda minimálna úroveň M na vlastný vrub je 8,45.

Na Obr. 3. vidíme koeficient úpravy R ako funkciu úrovne M na vlastný vrub poisťovateľa.

Obr. 3: Koeficient R ako funkcia M .



Zdroj: vlastné spracovanie

V prípade bez zaistenia, teda pre $M = 20$, má koeficient úpravy hodnotu 0,1397 (bez ohľadu na rizikovú prirážku zaistovne).

Na Obr. 3 môžeme vidieť, že pre $\xi = 0,15$ platí $R(M) > R(20)$, ak $4,87 < M < 20$ a $R(M) < R(20)$, ak $3,67 < M < 4,87$.

Pre $\xi = 0,3$ platí $R(M) < R(20)$ pre všetky prípustné hodnoty M .

Preto, ak $\xi = 0,15$ a $\theta = 0,1$, priamy poisťovateľ môže zvýšiť hodnotu koeficientu úpravy R efektívnym zaistením tak, že úroveň M na vlastný vrub zvolí vyšší než 4,87.

V prípade, že $\xi = 0,30$ a $\theta = 0,1$, poisťovateľ by si musel ponechať celé riziko, aby maximalizoval koeficient úpravy R .

Zamyslime sa ešte nad tým, čo ak budeme predpokladať rovnaké podmienky pre oba typy zaistenia, aké sú pravdepodobnosti krachu poisťovateľa?

Aby bolo porovnanie možné, predpokladajme, že očakávané poistné plnenie poisťovateľa pri oboch typoch zaistenia sa rovnajú, t.j.

$$E[\min \{X; M\}] = E[\alpha \cdot X]. \tag{21}$$

Uvažujme poistné plnenie s distribučnou funkciou $F(x) = 1 - e^{-0,1 \cdot x}$, $x > 0$ a rizikovú prirážku poisťovateľa $\theta = 0,1$ a rizikovú prirážku zaistovne $\xi = 0,15$ (rovnako ako v Príklade 1.). Ďalej predpokladajme, že poisťovateľ uzavrel proporcionálne zaistenie s podielom $\alpha = 0,6$ na vlastný vrub.

Potom $E(Y) = E(\alpha X) = \alpha \cdot E(X) = 0,6 \cdot 10 = 6$,

a $c^* = [(1 + \theta) - (1 + \xi) \cdot (1 - \alpha)] \cdot \lambda \cdot m_1 = (1,1 - 1,15 \cdot 0,4) \cdot \lambda \cdot 10 = 6,4 \cdot \lambda$.

Rovnica pre výpočet R má tvar: $\lambda + c^* \cdot R = \lambda \cdot M_Y(R)$,

$$\lambda + 6,4 \cdot \lambda \cdot R = \lambda \cdot \frac{1}{1 - 6 \cdot R}$$

$$R = 0,01042.$$

Uvažujme teraz zaistenie škodovej nadmierky s maximálnou úrovňou M na vlastný vrub poisťovateľa. Hodnota vlastného vrubu M musí byť taká, aby očakávaná hodnota poistného plnenia poisťovateľa po zaistení bola 6 (rovnaká hodnota ako pri proporcionálnom zaistení).

Potom $E(Y) = \int_0^M x \cdot f(x) dx + M \cdot (1 - F(M)) = 6$,

$$10 \cdot (1 - e^{-0,1 \cdot M}) = 6$$

$$e^{-0,1 \cdot M} = 0,4$$

$$M = 9,163.$$

Ak teraz predpokladáme, že $\theta = 0,1$ a $\xi = 0,15$, tak pre rovnicu na výpočet R dostávame:

$$\lambda + c^* \cdot R = \lambda \cdot \left(\int_0^M e^{R \cdot x} \cdot 0,1 \cdot e^{-0,1 \cdot x} dx + e^{R \cdot M} \cdot e^{-0,1 \cdot M} \right)$$

$$\lambda + 6,4 \cdot \lambda \cdot R = \lambda \cdot \left(0,1 \cdot \frac{1 - e^{-(0,1-R) \cdot M}}{0,1 - R} + e^{-(0,1-R) \cdot M} \right)$$

Za M dosadíme hodnotu 9,163 a rovnicu riešime numericky.

Výsledkom je hodnota $R = 0,01635$.

Koeficient úpravy pri zaistení škodovej nadmierky je väčší než pri proporcionálnom zaistení. Teda za inak nemenných podmienok je zaistenie škodovej nadmierky (XL zaistenie) výhodnejšie pre poisťovateľa, pretože dáva menšiu pravdepodobnosť krachu.

5 Záver

V našom príspevku sa zaoberáme vplyvom rizikovej prirážky poisťovateľa a zaist'ovne na pravdepodobnosť krachu poisťovateľa. Z výsledkov nám vyplýva, že pri určitých hodnotách rizikových prirážok dokážeme maximalizovať koeficient úpravy a tým minimalizovať pravdepodobnosť krachu, pri niektorých, ak je cena zaistenia príliš vysoká, je lepšie si ponechať celé riziko.

Pokiaľ porovnáme proporcionálne zaistenie a zaistenie škodovej nadmierky z pohľadu poisťovateľa je výhodnejšie zaistenie škodovej nadmierky. Do praxe sa tento výsledok môže premietnuť tak, že zaist'ovňa stanoví vyššiu rizikovú prirážku pre zaistenie škodovej nadmierky než pre proporcionálne zaistenie (za inak nemenných podmienok).

Pri zaistení sa poisťovateľ musí rozhodnúť, či chce maximalizovať zisk, pričom súčasne zvyšuje riziko krachu, alebo minimalizovať riziko.

Poisťovňa musí nájsť kompromis medzi mierou zisku a pravdepodobnosti krachu.

Literatúra

1. Brož, M. (2006). *Microsoft Excel – Vzorce, funkce, výpočty*, Computer press Praha,, ISBN 9788025110881.
2. Dickson, D. C. M., Waters, H. R. (1993). *Ruin theory*, University of Melbourne and Heriot-Watt University, Edinburgh.
3. Fecenko, J. (2006). *Neživotné poistenie*, Vydavateľstvo EKONÓM, Bratislava, ISBN 80-225-2191-4.
4. Gerber, H. U. (1979). *An Introduction to Mathematical Risk Theory*, Philadelphia, PA, S.S. Huebner Foundation.
5. Horáková, G., Mucha, V. (2006). *Teória rizika v poistení, I. časť*, Bratislava, Vydavateľstvo EKONÓM.
6. Pacáková, V. (2004). *Aplikovaná poisťná štatistika*, IURA EDITION, Bratislava, ISBN 80-8078-004-8.

Analýza kvapôčkových zavlažovacích systémov na princípe fuzzy inferenčného mechanizmu

Analysis of Drip Irrigation Systems Based on the Principle of a Fuzzy Inference Mechanism

Slávka Jalčová¹, Jozef Koyš¹, Miroslav Hudec¹

Abstrakt

V dnešnej dobe evidujeme narastajúcu potrebu efektívneho využívania prírodných zdrojov, vrátane vody. Táto práca skúma štúdie, ktoré využívajú inovatívne riešenia pre optimalizáciu zavlažovania v poľnohospodárstve založené na fuzzy odvodzovaní a IoT pre kvapôčkovú závlahu. Výsledky skúmania ukazujú, že fuzzy zavlažovacie systémy môžu výrazne zlepšiť poľnohospodársku produktivitu a šetrenie vody. Fuzzy inferenčný mechanizmus je vysvetliteľný, transparentný a adaptovateľný pre používateľa. V práci sme následne navrhli systém, ktorý je prispôsobiteľný špecifickým potrebám konkrétnych pestovateľov. Tento systém predstavuje cenovo dostupný návrh zavlažovacieho systému využívajúceho mikrokontrolerovú dosku Arduino Nano, rôzne typy senzorov a fuzzy pravidlá. Na základe vstupných hodnôt, výsledku odvodzovania a defuzzifikácie sa vypočíta impulz pre intenzitu zavlažovania.

Kľúčové slová

Fuzzy logika, zavlažovanie, optimalizácia, báza fuzzy pravidiel

Abstract

Nowadays, we observe an increasing need for the efficient use of natural resources, including water. This work examines studies that use innovative solutions for optimizing irrigation in agriculture based on fuzzy inference and IoT for drip irrigation. The results of the research show that fuzzy irrigation systems can significantly improve agricultural productivity and water conservation. The fuzzy inference mechanism is explainable, transparent, and adaptable for the user. In the work, we subsequently designed a system that is adaptable to the specific needs of individual growers. This system represents a cost-effective design of an irrigation system using the Arduino Nano microcontroller board, various types of sensors, and fuzzy rules. On the basis of the input values, the result of inference, and defuzzification, an impulse for irrigation intensity is calculated.

Key words

Fuzzy logic, irrigation, optimization, fuzzy rule base

JEL classification

Q550, C630, Q130

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35 Bratislava, sjalcoval@student.euba.sk, jkoys1@student.euba.sk, miroslav.hudec@euba.sk.

1 Úvod

V súčasnosti sa bežne stretávame s nedostatkom prírodných zdrojov. Je potrebné, aby sme sa naučili využívať zdroje, ktoré máme k dispozícii a zbytočne nimi neplytvať. Demografický rast, hospodárska činnosť a zmena klímy zvyšujú sezónny aj pretrvávajúci nedostatok vody na celom svete. Veľká časť odberu všetkej vody sa používa v poľnohospodárstve a to najmä na zavlažovanie (Salayová, 2023).

Voda je podstatnou zložkou biosféry a popri pôde má prvoradý význam pre zabezpečenie výživy ľudstva. Pre rastliny je dôležité nielen jej množstvo za rok, ale aj jej výskyt vo vegetačnom období. Nedostatok vody ohrozuje globálnu stabilitu a potravinovú bezpečnosť (consilium.europa.eu, n. d.).

Poľnohospodárstvo značným spôsobom zabezpečuje ekonomickú prosperitu a sociálnu stabilitu krajiny. Striedanie hustých dažďov a období sucha môže výrazne prispievať k rozvoju alebo úpadku poľnohospodárskej produkcie štátu (Mohammed et al., 2021). Účelná a správna distribúcia zavlažovacej vody vedie k výraznému zlepšeniu v poľnohospodárskej produktivite a spotrebe vody z hľadiska jej šetrenia (Grow Market, 2022; Salayová, 2023).

Aby sme prehodnotili postavenie vody v jednotlivých ekonomikách, je potrebná účasť všetkých zainteresovaných strán od miestnej až po globálnu úroveň, na dosiahnutie misií, ktoré riešia najdôležitejšie výzvy globálnej vodnej krízy (Knoteková, 2024). Medzi **kritické misie na riešenie** mnohostranných **výziev vodnej krízy** patrí:

- Revolučne zmeniť potravinové systémy zlepšením zavlažovania a znížením závislosti od škodlivých hnojív.
- Zachovať a obnoviť prírodné biotopy na ochranu kľúčových zdrojov vody.
- Vytvoriť cirkulárnu vodnú ekonomiku na maximalizáciu obnovy zdrojov a zníženie odpadu.
- Zabezpečiť prístup k bezpečnej vode na prevenciu detskej úmrtnosti spôsobenej chorobami súvisiacimi s vodou do roku 2030 (ECA EUROPA, 2021; Opatrenia EÚ, 2017).

V našej práci sa zameriame na prvý bod, zlepšenie zavlažovania, čím sa poskytne podpora rastlinám vo vývoji a tým sa môže redukovať pridávanie umelých hnojív.

K Slovensku, krajine s obmedzenými nerastnými zdrojmi, patrí najväčšie bohatstvo v podobe vody a pôdy. Nedostatok, ale aj nadbytok vody negatívne vplyva na pórovitosť a štruktúru úrodnej ornej zeme (Bloomling, n.d.). Optimálne množstvo vody v pôde je definované poľnou vodnou kapacitou PVK, ktorou sa rozumie schopnosť pôdy udržať v sebe určité množstvo vody (ECA EUROPA, 2021).

Pri zavlažovaní sa zohľadňujú rôzne merateľné faktory, ako sú intenzita slnečného svitu, vlhkosť a ďalšie. Zavlažovanie by malo byť vykonávané v správnom čase (ráno alebo večer), aby sa minimalizovalo odparovanie vody. Rovnomerné zavlažovanie je kľúčové, aby sa predišlo suchým alebo premočeným miestam. Pravidlá zavlažovania zahŕňajú aj správnu hĺbku zavlažovania, aby sa podporoval silný koreňový systém. Merania nie sú vždy presné a preto sú vhodnejšie flexibilne pravidlá. Ak sú merania intervalové, potom pravidlá zavlažovania by klasickým prístupom museli byť reprezentované pomocou pomerne komplikovanej reálnej funkcie. Ak sú merania diskretizáciou výrazne redukované na presné hodnoty, potom je regulácia zavlažovania skokovitá. Fuzzy pravidlové systémy poskytujú flexibilitu a adaptovateľnosť na podmienky pri menšom počte pravidiel. Tieto systémy využívajú fuzzy logiku, ktorá umožňuje pracovať s neurčitou a nepresnou v údajoch, čím zvyšuje efektivitu a presnosť zavlažovania (Karakuzu, 2000).

V našej práci sme preto najprv analyzovali štúdie zaoberajúce sa IoT zariadeniami pre kvapôčkovú závlahu, ktoré využívajú fuzzy inferenčný mechanizmus. Na základe tejto analýzy sme navrhli model a preskúmali jeho adaptovateľnosť a nákladovosť.

2 Úvodné informácie o zavlažovaní a fuzzy logike

V nasledujúcej časti sa zameriame na konkrétne metódy optimálneho zavlažovania, ktoré prispievajú k šetreniu vodných zdrojov a vysvetlíme základné princípy fuzzy logiky.

2.1 Optimálne zavlažovanie ako spôsob šetrenia vodných zdrojov

Zavlažovanie je nahradenie alebo doplnenie dažďových zrážok vodou z iného zdroja za účelom pestovania plodín alebo rastlín. Základným činiteľom pri stanovení správneho výberu závlahy je znalosť jej účelu. Zdroj (Wexbo.com, n.d.) uvádza, že závlahy delíme podľa toho, aký cieľ chceme závlahou dosiahnuť (doplnenie vody, prihnojenie, ochrana proti mrazu, atď.) do nasledujúcich skupín:

- doplnková závlaha,
- hnojivá závlaha,
- špeciálna závlaha.

Zavlažovanie je kľúčovým aspektom poľnohospodárskej výroby. Moderné technológie umožnili prechod od tradičných metód, ako sú záplavy a zavlažovanie brázd, k úspornejším režimom, ako je kvapková, postrekovacia a priesaková závlaha. Tieto metódy teraz možno riadiť aj prostredníctvom mobilných zariadení so systémom Android/iOS, čo znižuje potrebu manuálnych zásahov.

Inteligentné zavlažovacie systémy v poľnohospodárstve využívajú senzory, automatické riadenie, výpočtovú techniku a bezdrôtové siete na zber a spracovanie údajov o stave prostredia (Wexbo.com, n.d.; Bloomling, n.d.; Grow Market, 2022; Bwambale et al., 2022; Mendes et al., 2019). Tieto systémy vykonávajú aj riadenie zavlažovania, správu údajov a generovanie automatických výstrah. Práve súbor týchto funkcií umožňuje efektívnejšie rozhodovanie a zvyšuje mieru automatizácie, čím sa podporuje prechod k technologicky náročnejšiemu a presnejšiemu poľnohospodárstvu (Wexbo.com, n.d.).

Zaujímalo nás, na čo sa práce zaoberajúce sa témou inteligentného kvapôčkového zavlažovania zameriavali, aké princípy meraní zvolili, aké zistenia práce priniesli a aké je ponaučenie.

2.2 Základné princípy fuzzy logiky a odvodzovania

Fuzzy logika pracuje s nepresnými a nejednoznačnými informáciami. Môže byť nápomocná najmä ak nemáme presne namerané hodnoty, alebo expert nevie stanoviť presné hranice. Pri fuzzy logike operujeme s lingvistickými výrazmi.

V *modus ponens* odvodzovacom systéme na základe premís, vieme odôvodniť záver: „Ak sú P a Q výroky a pritom je P pravdou, tak je pravdou aj Q.“ (Enderton, 2001) Na tomto princípe funguje *generalizovaný modus ponens*, kde je rozdiel v intenzitách príslušnosti do pozorovania a flexibilita pravidiel (Gorzałczany, 2002). Odvodzovanie je komplexnejšie, ale lepšie zachytí realitu pri menšom počte pravidiel.

Všeobecne, fuzzy logický kontrolný systém je vytvorený zo štyroch blokov (Obrázok 1). Prvý je fuzzifikáčny, ktorý konvertuje vstupné premenné na fuzzy premenné a priraduje im stupne príslušnosti nameraných hodnôt. Potom, druhý blok je inferenčný mechanizmus, ktorý vyvodzuje fuzzy výsledok z fuzzifikovaného vstupu a báze if-then pravidiel (Gorzałczany, 2002). Štvrtý blok je defuzifikácia, ktorý konvertuje výsledok na presnú hodnotu potrebnú na reakciu na dané vstupy.

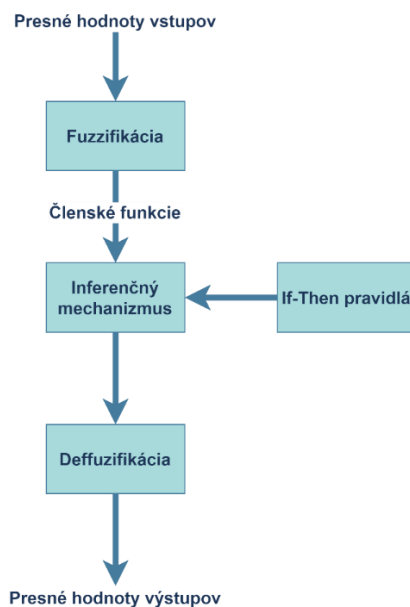
Tento princíp vieme aplikovať aj na princíp zavlažovania. Ak nemáme presné pozorovania, len vyjadrenia ako „prší“, „pôda je vlhká“ a vtedy môžeme použiť fuzzy

inferenčný mechanizmus: „Ak trochu prší, pôda bude jemne vlhká.“ Z toho vieme vyvodit' vety: „Prší,“ a „Pôda je vlhká, pretože prší“.

V nami analyzovaných prácach (Singh et al., 2022; Neugebauer et al., 2023; Hoque et al., 2023) vidíme rôzne prístupy k tvorbe fuzzy bázy pravidiel, pretože každá štúdia obsahuje rôzne vstupné a výstupné premenné. Vo všeobecnosti však vieme povedať, že sú vstupné premenné, ktoré sú v nepriamej úmere s výstupnou premennou a potom také, ktoré sú v priamej úmere s výstupnou premennou. Naplnenie fuzzy bázy pravidlami zabezpečí ďalšie pokračovanie v modelovaní rôznych stavov pôdy. Tento systém je univerzálny, pretože každý majiteľ zavlažovacieho systému si môže podmienky pre vlhkosť a zavlažovanie pôdy nastaviť podľa konkrétnej pestovanej plodiny, rastliny. Regulácia je založená na defuzzifikácii, ktorú je vhodné realizovať ťažiskovou metódou pre vyššiu presnosť a odolnosť (Klir & Yuan, 1995).

Na ďalšiu funkcionálnu sú potrebné funkcie príslušnosti vstupných a výstupných premenných. Fuzzy množiny podporujú neostre hranice a umožňujú intenzity príslušností. Vo všeobecnosti sa používajú trojuholníkové, lichobežníkové a singleton fuzzy množiny, pretože umožňujú zachytiť lingvistický výraz *Nízka*, *Približne...*, *Stredná*, *Vysoká* a podobne. Ako príklad uvádzame lichobežníkovú fuzzy množinu, pretože trojuholníková je v podstate to isté ako lichobežníková, iba má koncentrované jadro do jedného prvku s intenzitou príslušnosti rovnou jednej.

Obrázok 1: Diagram kontrolného systému využívajúceho fuzzy logiku

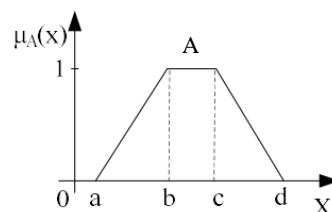


Zdroj: Vlastné spracovanie podľa (Singh et al., 2022).

Lichobežníková fuzzy množina (Obrázok 2) sa využíva na modelovanie neurčitosti. Na výstupe je možné použiť aj nelineárnu funkciu, pričom fuzzy množiny by mali približne spĺňať požiadavku Ruspini partície, t. j. aby súčet hodnôt príslušnosti pre každý bod bol približne rovný 1. Lichobežníková fuzzy množina je definovaná dolným (a) a horným (d) ohraňením nosiča a dolnou (b) a hornou (c) hranicou jadra takto:

$$\mu_A(x) = \begin{cases} 0, & x \leq a \vee x \geq b \\ \frac{x-a}{b-a}, & x \in (a, b) \\ 1, & x \in \langle b, c \rangle \\ \frac{d-x}{d-c}, & x \in \langle c, d \rangle \end{cases} \quad (1)$$

Obrázok 2: Lichobežníková fuzzy množina



Zdroj: Spracovanie upravené podľa (Klir & Yuan, 1995)

3 Analýza existujúcich riešení pre zavlažovanie

Výskumníci z Indie (Singh et al., 2022) sa v navrhnutom riešení zamerali na tvorbu kontrolného systému s fuzzy logikou. V práci použili autori fuzzy inferenčný systém vyvinutý v prostredí Matlab, ktorý využíva Mamdani prístup fuzzy inferencie (Gorzałczany, 2002; Hudec, 2016). Vo výsledku je AND operátor realizovaný vypočítaním **minima**, pričom OR operátor vypočítaním **maxima**. Štyri vstupné premenné sú *Vlhkosť Pôdy (Soil Moisture)*, *Slnčné Žiarenie (Solar Irradiance)*, *Teplota Vzduchu (Air Temperature)* a *Vlhkosť Vzduchu (Air Humidity)*. Výstupnou premennou, *Napätie Čerpadla (Pump Voltage)*, je kontrolovaná báza fuzzy pravidiel definovaná v Matlab-e. Autori práce zvolili premennú *Napätie Čerpadla*, pretože množstvo vody potrebnej v pôde sa môže regulovať elektrickým napätím čerpadla. Otázka je ako regulovať napätie tak, aby zavlažovanie bolo efektívne.

Pri Mamdaniho inferenčnom systéme pre jednoduchý prípad dvoch vstupov sú premennými *Vlhkosť Pôdy (Soil Moisture)* a *Teplota Vzduchu (Air Temperature)*, výstupom je intenzita zavlažovania. Takýto fuzzy systém následne na základe pravidiel vyhodnocuje aktuálny stav prostredia a určuje, aká intenzita zavlažovania je v danom momente vhodná. Typické fuzzy pravidlo môže mať tvar – „ak je vlhkosť pôdy nízka a teplota vzduchu vysoká, potom intenzita zavlažovania je vysoká“. Pravdivosť takéhoto pravidla sa vyhodnotí ako minimum príslušností aktuálnych vstupov k daným jazykovým hodnotám. Výsledkom je úroveň, ktorou sa uplatní výstupná hodnota v rámci fuzzy množiny „vysoká intenzita zavlažovania“.

Zavlažovací systém (Singh et al., 2022) nechali v prostredí pracovať na fuzzy báze pravidiel v rôznych časoch počas dňa, od východu slnka až po západ slnka, v rôznych dňoch, aby získali vstupné hodnoty z rôznych senzorov na sledovanie zmien výstupného napätia a prietoku vody z čerpadla. Cieľom tohto merania bolo overiť, či navrhnutý fuzzy systém reaguje na meniace sa environmentálne podmienky v súlade s očakávaným správaním, teda či výstupné napätie čerpadla (ako výsledok odvodzovania) odráža logiku nastavených pravidiel. Overenie funkčnosti systému prebehlo najmä na úrovni reakcie výstupu na vstupy. Skutočný prínos pre poľnohospodársku prax však môže vyžadovať samostatné hodnotenie od odborníkov z terénu alebo koncových používateľov.

Podľa nameraných hodnôt sa dá interpretovať, že vstupné napätie čerpadla vody sa mení v závislosti od environmentálnych hodnôt meraných rôznymi senzormi. Výsledky overili správnosť ich predpokladov, napríklad: „Ak je vlhkosť pôdy vysoká a zároveň slnečné žiarenie je tlmené, zatiaľ čo teplota vzduchu je studená a vlhkosť zas vysoká, vtedy je nutnosť zavedenia vody do pôdy minimálna“. Permutáciou všetkých vstupných premenných v tomto poradí by sa potreba vody mala zvyšovať, keďže potreba vody pre tento konkrétny prvý príklad je ten najnižší (Singh et al., 2022).

Príkladom formulácie pravidla touto logikou by mohlo byť: „Ak vlhkosť pôdy je vysoká a zároveň teplota vzduchu je nízka, tak potom je napätie čerpadla veľmi nízke.“

Výsledky uvedenej práce ukázali, že Indii by navrhovaný fuzzy zavlažovací kvapôčkový systém umožnil efektívne korigovať mieru napätia čerpadla a následne ovplyvňovať množstvo vody využitej na zavlažovanie. Tento model je vysvetliteľný pre farmára. Keďže India je krajina s veľkým množstvom obyvateľov a pestujú sa v nej rôzne plodiny (ryža, bavlna, cukrová trstina, idnigo, atď.) (Bloomling, n.d.).

Autori práce (Hoque et al., 2023) sa nezamerali na konkrétnu krajinu, či podnebné pásmo. Ich vstupné premenné boli *Faktor prahovej hodnoty vlhkosti pôdy*, *Faktor prahovej hodnoty slnečného žiarenia*, *Faktor prahovej hodnoty vlhkosti*, *Faktor prahovej hodnoty teploty* a výstupnou premennou bol *Faktor prahovej hodnoty dĺžky zavlažovania*. V práci bol využitý rovnaký spôsob fuzzy odvodzovania a realizovaný elektrický obvod zavlažovacieho systému. Výsledky potvrdzujú, že ovplyvnením času zavlažovania na základe vstupov nameraných senzormi a fuzzy odvodzovania sa optimalizoval proces zavlažovania.

Práca (Neugebauer et al., 2023) vznikla spoluprácou vedcov z Poľského Olsztyna a Tureckého Izmiru. Táto práca mala premenné *Čas dňa*, *Vlhkosť pôdy*, *Teplota* a výstupnú premennú *Intenzita zavlažovania*. Boli použité takisto trojuholníkové a lichobežníkové tvary fuzzy množín, len pri výstupnej premennej sa objavil tvar singleton (bol použitý pri vyjadrení minimálnej a maximálnej intenzity zavlažovania, teda 0 a 100).

Práca (Mohammed et al., 2021) je zameraná na oblasť Maroka. Zavlažovací systém je podobný ostatným analyzovaným prácam. Vstupné premenné sú *Vlhkosť pôdy*, *Teplota* a *Slnečné žiarenie*. Merania pomocou navrhovaného zavlažovacieho systému boli vykonané v lete a na jeseň. Práca je charakterom a postupom podobná ostatným analyzovaným prácam s rozdielom, že zber údajov z testovacieho systému prebieha v 60 minútových intervaloch.

Preštudovali sme aj prácu (Sujono & Nainggolan, 2024), aby sme zistili, či môžeme navrhnúť elektrický obvod kvapôčkového zavlažovacieho systému, ktorý by využíval fuzzy logiku a zároveň bol kompatibilný so zariadeniami Internetu vecí. Z práce môžeme konštatovať nasledujúci záver: Z celkového testu vyplýva, že automatický systém kvapkovej závlahy môže byť založený na úrovni vlhkosti pôdy a teploty rastlín. Systém môže byť založený na fuzzy Mamdaniho metóde, kde touto metódou možno monitorovať parametre snímačov a následne regulovať ventily.

Všetky nami skúmané práce využili Mamdani prístup fuzzy inferencie a všetky dokázali, že je možné nastaviť intenzitu zavlažovania konkrétnych plodín/rastlín na základe vstupných údajov získaných z externých zariadení a vytvorených fuzzy pravidiel.

4 Náš návrh cenovo dostupného riešenia

Na základe skúmania prác v predošlej časti sme zhotovovali návrh technického zariadenia. Na nastavenie zariadenia je potrebné určiť logiku a pravidlá zavlažovania plodiny. Zvolili sme kapustu obyčajnú, ktorá je univerzálna plodina pre pestovanie na celom území Slovenska.

Zvolili sme vstupné premenné *Pôda: Vlhkosť, Ovzdušie: Vlhkosť, Ovzdušie: Teplota* a *Časť dňa*. Premennú *Časť dňa* uvažujeme pre zvyky a tradície slovenských pestovateľov, ktorí sa riadia pravidlom, že cez obed sa nikdy nezavlažuje. Závlaha na pravé poludnie a v nasledujúcich hodinách spáli kvety, listy a najmä korene rastliny.

Výstupnou premennou je *Napätie čerpadla*. Najprv sme vyslovili základné pravidlo, kedy sa má zavlažovať najvyššou intenzitou:

„Ak je časť dňa ráno, alebo večer a noc a zároveň je vlhkosť pôdy nízka, a zároveň je teplota vzduchu nízka, pričom je vlhkosť vzduchu nízka, nech je napätie čerpadla veľmi vysoké.“

Z tohto najnižšieho príkladu, kedy sú všetky vstupné premenné nízke a výstupná premenná má najvyššiu hodnotu, vieme opačným spôsobom definovať, kedy sa kapusta zavlažovať nemá, a to: „Ak je časť dňa obed a zároveň je vlhkosť pôdy vysoká, a zároveň je teplota vzduchu vysoká, pričom je vlhkosť vzduchu vysoká, nech je napätie čerpadla veľmi nízke.“

Všetky stavy medzi týmto najnižším a najvyšším extrémom môžu byť dosiahnuté permutáciou hodnôt vstupných údajov.

Na modelovanie sme použili lichobežníkové fuzzy množiny a údaje o hodnotách premenných fuzzy pravidiel sú v *Tabuľka 1: Parametre modelu*. V grafickej časti tabuľky uvádzame fuzzy rozklady pravidiel, pričom parametre premenných sú označené písmenami. Tieto rozklady boli vytvorené tak, aby splnili požiadavky na fuzzy partície (Ruspini, 1969, pp. 22-32) a indikátory kvality jazykovej premennej (Alonso et al, 2021).

V praktickom prípade kapusty obyčajnej, by to mohlo napríklad byť: Vstupná premenná *Pôda: Vlhkosť*, je jednoznačne nízka v hodnotách od 0 do $a_{vp} = 25\%$, jednoznačne priemerná v hodnotách $b_{vp} = 37,5\%$ až $c_{vp} = 62,5\%$ a jednoznačne vysoká v hodnotách $d_{vp} = 75\%$ a viac, v tomto prípade 100%. Medzi hodnotami $a_{vp} = 25\%$ a $b_{vp} = 37,5\%$ je plynulý prechod medzi nízkou a strednou vlhkosťou. Podobne, medzi hodnotami $c_{vp} = 62,5\%$ a $d_{vp} = 75\%$ je plynulý prechod medzi strednou a vysokou vlhkosťou. Postup by bol obdobný aj pri ostatných premenných. Parametre si môže vhodne nastaviť farmár pre každú pestovanú plodinu individuálne, podľa odporúčania, skúseností či vlastného uváženia.

Usúdili sme, že potrebujeme zostaviť spoľahlivý, ale jednoducho ovládateľný systém, ktorý by vykonával logiku podľa nami stanovených pravidiel. Zvolili sme zariadenie Arduino.

Mikrokontrolérová doska Arduino Nano je napájaná cez zbernicu s USB portom napätím 5 V, ktorej úlohou je aj odosielanie príkazov z Modelu do zariadenia Arduino Nano na základe externých vstupov. Zbernica okrem napájania slúži pre komunikáciu s vyhodnocovacou aplikáciou (napr. Simulink). Na Arduino Nano sú taktiež pripojené potrebné senzory, ktoré merajú okolité podmienky. Potrebnými senzormi pre meranie okolitých podmienok sú senzor teploty a vlhkosti ovzdušia (DHT11), senzor vlhkosti pôdy a senzor slnečného žiarenia. Na základe údajov zachytených pomocou senzorov dokáže mikrokontrolér zopnúť magnetické relé, ktoré následne spustí pumpu s jednosmerným prúdom (ďalej len DC) potrebnú pre zavlažovanie. DC pumpa sa skladá z DC motorčeka napájaného 3 až 7 V napätím a vodnej pumpy. Napájanie DC motorčeka je realizované zo striedavej siete cez adaptér na zníženie a usmernenie napäťovej úrovne z 220 V striedavého prúdu (ďalej len AC) na 12 V DC. Model je zobrazený na Obrázku 3.

Následne je hodnota napätia znížená cez step-down menič na hodnotu napätia, ktorá je požadovaná pre chod DC pumpy. LED pripojený na zariadenie slúži ako ad hoc kontrola pre rýchle zobrazenie potrebných údajov, ktoré môžu byť pre pestovateľa užitočné (napr. vonkajšia teplota).

Tabuľka 1: Parametre modelu

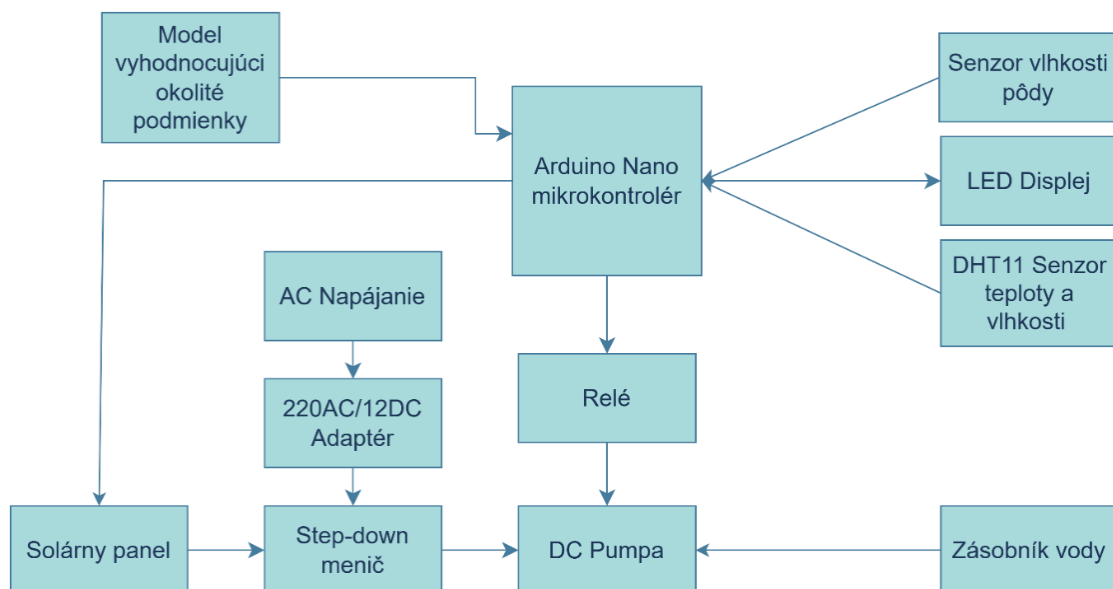
Premenná	Merná jednotka	Jazyková premenná	Graf
Pôda: Vlhkosť	%	Nízka	
		Priemerná	
		Vysoká	
Ovzdušie: Vlhkosť	%	Nízka	
		Priemerná	
		Vysoká	
Ovzdušie: Teplota	°C	Nízka	
		Stredná	
		Vysoká	
Časť dňa	Hodiny	Ráno	
		Obed	
		Večer	
Napätie čerpadla	V	Veľmi nízke	
		Nízke	
		Stredné	
		Vysoké	
		Veľmi vysoké	

Zdroj: Vlastné spracovanie

Na vytvorenie fuzzy regulátora a návrh pravidiel bol použitý Matlab a jeho nadstavba Simulink, pričom počas vývoja bol mikrokontrolér Arduino Nano pripojený k počítaču. V tejto fáze slúžil počítač nielen na napájanie dosky, ale aj ako rozhranie pre odosielanie príkazov a zber údajov.

V aktuálnej konfigurácii je Arduino závislé od počítača počas zavlažovania, čo je z hľadiska terénneho nasadenia menej praktické. Preto do budúcnosti plánujeme systém vylepšiť tak, aby fuzzy inferenčný mechanizmus bežal priamo na zariadení bez potreby trvalého pripojenia k počítaču. Keďže klasické Arduino Nano má veľmi obmedzené výpočtové možnosti, spustenie kompletného fuzzy regulátora priamo na tejto doske nie je realistické. Jednou z vhodných alternatív by mohlo byť nasadenie platformy Raspberry Pi, ktorá umožňuje bežať fuzzy logiku v jazyku Python priamo na zariadení, spolu s riadením vstupno-výstupných signálov. Táto možnosť by v budúcnosti mohla výrazne znížiť závislosť od externého počítača a zjednodušiť použiteľnosť systému v teréne.

Obrázok 3: Schéma modelu



Zdroj: Vlastné spracovanie

Približný návrh ceny zariadenia pre vybrané komponenty je v *Tabuľke 2*. V návrhu ceny nie je zahrnutá cena za prácu, kabeláž, cín a náradie potrebné pri konštruovaní zariadenia. Komponenty môžu byť od rôznych výrobcov s odlišnými parametrami.

Týmto modelom a približnými cenami jednotlivých komponentov, by sme chceli poukázať na to, že zostavenie zariadenia nie je až tak finančne náročné, aké je v konečnom dôsledku užitočné ako pre spoločnosť, tak pre malého farmára, či samostatnú domácnosť.

Tabuľka 2: Komponenty pre nami navrhovaný zavlažovací systém.

Názov komponentu	Cena (€ s DPH)
Arduino Nano	13,53
DC pumpa RS-360SH 3-12V	4,80
Senzor teploty a vlhkosti DHT11	7,51
Senzor vlhkosti pôdy	6,18
AC Adaptér	32,31
Step-down menič LM2596	1,60
Relé	4,09
Solárny panel	37,80
LED displej	3,80
Spolu	111,62

Zdroje elektronických obchodov: Mouser a Techfun.sk.

5 Diskusia

Síce sa Slovensko nachádza v miernom podnebnom pásme, ale v zavlažovaní na princípe fuzzy logiky vidíme budúcnosť i na našom území. V roku 2023 tvorilo poľnohospodárstvo 2 % podiel na Slovenskom HDP, 11,06 % na HDP Maroka, 16 % na HDP Indie a 2,75 % na HDP Poľska, preto sme skúmali, ako sa v týchto štátoch skúma problém zavlažovania. (Our World in Data, n.d.)

Výskumníci z týchto krajín v prácach (Singh et al., 2022; Neugebauer et al., 2023; Hoque et al., 2023) zdôrazňovali výrazný vplyv poľnohospodárstva na budúci vývoj ich ekonomiky a preto sa i my domnievame, že so zvyšujúcimi sa teplotami v posledných rokoch, sa ani Slovensko nevyhne rozhodnutiam ohľadne zavádzania inteligentného zavlažovania do praxe.

V našej práci sme sa snažili prísť s návrhom modelu fuzzy zavlažovacieho kvapôčkového systému. Uznávame, že návrh by sa dal zdokonaľiť. Jedno zo zlepšení by predstavovalo využitie solárneho panelu nielen na zaznamenávanie intenzity slnečného žiarenia, ale aj na napájanie zavlažovacieho systému. Pridanie Wi-Fi modulu do zariadenia pre umožnenie bezdrôtovej komunikácie by sme takisto považovali za užitočnú funkcionálnosť.

Uvažujeme, že kvapôčkový zavlažovací systém sa môže využiť nielen na distribúciu vody, ale aj na distribúciu hnojiva a pesticídov, za predpokladu, že nami navrhovaný model by majiteľ upravil podľa svojich konkrétnych potrieb a potrieb vegetácie. Samozrejme, bolo by potrebné ošetriť aj nebezpečenstvo, že v prípade výpadku prúdu by sa náš model automaticky sám nespustil a preto pestovateľ, ktorý by sa na náš model chcel spoliehať počas jeho neprítomnosti, by mal zvážiť modifikáciu obvodu záložným zdrojom.

6 Záver

Slovensko nepatrí medzi krajiny, ktorých poľnohospodárstvo tvorí väčšinu HDP, ale efektívne využitie vody aj vďaka fuzzy zavlažovacím systémom by mohlo napomôcť k zvýšeniu potravinovej sebestačnosti.

V práci sme na základe analýz existujúcich riešení navrhli model, ktorý pracuje na princípe Mamdani modelu fuzzy odvodzovania. Táto flexibilita umožňuje nastavenie pravidiel výrazmi prirodzeného jazyka a prispôsobenie parametrov fuzzy množín cieľovej plodiny.

Zariadenie na reguláciu závlahy predstavuje atraktívne riešenie pre poľnohospodárov, ktorí pestujú rastliny komerčne. Vďaka svojej cenovej dostupnosti však môže nájsť uplatnenie aj medzi drobnými pestovateľmi.

Literatúra

1. (Opatrenia EÚ, 2017) Opatrenia EÚ v oblasti energetiky a zmeny klímy. Európsky dvor audítorov – Opatrenia EÚ v oblasti energetiky a zmeny klímy. (n.d.). <https://op.europa.eu/webpub/eca/lr-energy-and-climate/sk/>
2. Alonso, J. M., Castiello, C., Magdalena, L., & Mencar, C. (2021) *Explainable Fuzzy Systems: Paving the way from Interpretable Fuzzy Systems to Explainable AI Systems*. Springer, Cham.
3. Bloomling. (n.d.). *Šetrite Vodu Pomocou Kvapkovej závlahy*. Internetový obchod Bloomling. <https://www.bloomling.sk/info/sprievodca-zahradkarstvom/uspورا-vody-pomocou-kvapkovej-zavlahy>
4. Bwambale, E., Abagale, F. K., & Anornu, G. K. (2022). Smart irrigation monitoring and control strategies for improving water use efficiency in precision agriculture: A review. *Agricultural Water Management*, 260, 107324.
5. ECA EUROPA (2021). *Special report 20/2021: Sustainable water use in EU agriculture*. https://www.eca.europa.eu/Lists/ECADocuments/SR21_20/SR_CAP-and-water_EN.pdf
6. Enderton, H.B. (2001): *A Mathematical Introduction to Logic*. Academic Press, San Diego
7. From farm to fork - consilium.europa.eu. (n.d.). <https://www.consilium.europa.eu/en/policies/from-farm-to-fork/>
8. Gorzałczany, M. (2002): *Computational Intelligence Systems and Applications*. Physica-Verlag, Heidelberg
9. Grow Market. (2022, November 22). *Kvapkové Zavlažovanie znamená pohodlné a úsporné Zavlažovanie*. www.growmarket.sk. <https://www.growmarket.sk/a/kapkova-zavlahaznamena-vyhodne-a-hospodarnezalevani?srsId=AfmBOoq9t4nfW6K241CAWOABqUN6sj5huONxwLUUiJI6jDXjfkJugEIG>
10. Hoque, M. J., Islam, M. S., & Khaliluzzaman, M. (2023). A Fuzzy Logic-and Internet of Things-Based Smart Irrigation System. *Engineering Proceedings*, 58(1), 93.
11. Hudec, M. (2016) *Fuzziness in Information Systems – How to Deal with Crisp and Fuzzy Data in Selection, Classification, and Summarization*. Springer, Cham.
12. Karakuzu, C. (2000) An experimental comparison of fuzzy, neuro and classical control techniques. *21st IEEE Convention of the Electrical and Electronic Engineers in Israel*, Tel-Aviv, pp. 160-166.
13. Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic, theory and applications*. Prentice Hall, New Jersey.
14. Knoteková, D. (2024, October 21). *Expertí Varujú: Globálna Vodná Kríza Ohrozuje 8% Stratu HDP do Roku 2050*. SITA.sk. <https://sita.sk/klima/experti-varuju-globalna-vodna-kriza-ohrozuje-8-stratu-hdp-do-roku-205/>
15. Mendes, W. R., Araújo, F. M. U., Dutta, R., & Heeren, D. M. (2019). Fuzzy control system for variable rate irrigation using remote sensing. *Expert systems with applications*, 124, 13-24.
16. Mohammed, B., Bakkay, H., Migan-Dubois, A., Adel, M., & Rabhi, A. (2021). An intelligent irrigation system based on fuzzy logic control: A case study for Moroccan oriental climate region. In *2nd international conference on Embedded Systems and Artificial Intelligence (ESAI'21)*.
17. Mouser. (n.d.). https://www.mouser.com/ProductDetail/PowerFilm/MPT15-150?q_s=BjLw7L4Cy78xXZvPjeDZ/w%3D%3D
18. Neugebauer, M., Akdeniz, C., Demir, V., & Yurdem, H. (2023). Fuzzy logic control for watering system. *Scientific Reports*, 13(1), 18485.

19. Ruspini, E. H. (1969). A new approach to clustering. *Information and control*, 15(1), 22-32.
20. Salayová, V. (2023, March 23). *Zem je na Pokraji Globálnej Vodnej krízy. takmer 3 miliardy ľudí Nemá prístup K čistej vode*. iMeteo.sk. <https://www.imeteo.sk/spravy/zem-je-na-pokraji-globalnej-vodnej-krizy-takmer-3-miliardy-ludi-nema-pristup-k-cistej-vode>
21. *Share of GDP from Agriculture*. Our World in Data. (n.d.). <https://ourworldindata.org/grapher/agriculture-share-gdp?tab=chart&country=SVK~MAR~IND>
22. Singh, A. K., Tariq, T., Ahmer, M. F., Sharma, G., Bokoro, P. N., & Shongwe, T. (2022). Intelligent control of irrigation systems using fuzzy logic controller. *Energies*, 15(19), 7199.
23. Sujono, H. A., & Nainggolan, R. W. (2024). Drip irrigation control system based on Mamdani fuzzy logic and internet of things (IoT). *Prz. Elektrotech*, 1, 65-69.
24. Techfun.sk. (n.d.). Techfun.sk – Naj Arduino shop. https://techfun.sk/produkt/led-display-16x2-modre-podsvietenie-spajkovany-i2c-modul/?currency=EUR&gad_source=1&gclid=CjwKCAiAzba9BhBhEiwA7glbam2ZNjIVcI_2LU8IK4-BOi2KyEgoo0cS36-PBWtS2rCGHWtIKK7N7RoCSKEQAvD_BwE
25. Wexbo.com. (n.d.). *Čo Je Kvapková Závlaha?*. MAX GARDEN - záhradné centrum pre Vašu záhradu. https://maxgarden.sk/a/co-je-kvapkova-zavlaha-31/?srsltid=AfmBOooJY_X3Ovt_43CN9tsAbIsZsswL61LtUQZKFjjABX_b9Xh5buhp

Detailné porovnanie vybraných aplikácií na podporu tímovej komunikácie využiteľných v pracovnom prostredí

A detailed comparison of selected applications to support team communication usable in work environment

Pavol Jurík¹, Natália Babjaková²

Abstrakt

V dnešnom globalizovanom svete sa firmy snažia o väčšie zjednodušenie komunikácie za účelom zvýšenia efektivity a konkurencieschopnosti. Vzhľadom na to sa aplikácie na podporu tímovej komunikácie stali výkonnými nástrojmi na transformáciu spolupráce na pracovisku. Komunikácia je jedným z najdôležitejších faktorov v každom podniku a schopnosť zlepšovať tieto faktory je významná pre rozvoj podniku. Aplikácie na podporu tímovej komunikácie umožňujú komunikáciu v reálnom čase prostredníctvom okamžitých správ, čo umožňuje pracovníkom rýchlo zdieľať informácie. Táto schopnosť je užitočná najmä pri prijímaní operatívnych rozhodnutí a rýchlom riešení problémov. Aplikácie na podporu tímovej komunikácie umožňujú zamestnancom efektívnejšie spolupracovať, tímy pracovníkov môžu zdieľať nápady a riešiť problémy spoločnosti „jedným kliknutím“. Používanie Aplikácie na podporu tímovej komunikácie môže viesť k lepšej výkonnosti a produktivite zamestnancov, pretože ide o efektívnejší spôsob komunikácie v porovnaní so zasielaním a prijímaním emailov. V tomto článku sa zameriavame na porovnanie aplikácií Microsoft Teams, Slack a Google Chat.

Kľúčové slová

Aplikácie na podporu tímovej komunikácie, firemná komunikácia, Microsoft Teams, Slack, Google Chat

Abstract

In today's globalized world, companies strive for greater simplification of communication in order to increase efficiency and competitiveness. With this in mind, team communication applications have become powerful tools for transforming workplace collaboration. Communication is one of the most important factors in any business, and the ability to improve these factors is significant for business development. Team communication applications enable real-time communication through instant messaging, allowing workers to quickly share information. This ability is especially useful when making operational decisions and quickly solving problems. Team communication applications allow employees to collaborate more effectively, teams of workers can share ideas and solve company problems "with one click". Using an team communication application can lead to better performance and productivity of employees, because it is a more effective way of communication compared to sending and receiving emails. In this article, we focus on comparing Microsoft Teams, Slack and Google Chat.

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35 Bratislava, pavol.jurik@euba.sk.

² absolventka študijného programu Hospodárska informatika na Fakulte hospodárskej informatiky, Ekonomickej univerzity v Bratislave, Dolnozemska cesta 1, 852 35 Bratislava, nbabjakova2@student.euba.sk.

Key words

Team communication applications, corporate communication, Microsoft Teams, Slack, Google Chat

JEL classification

D83, L86

1 Úvod

Aplikácia na podporu tímovej komunikácie je softvérový program, ktorý umožňuje tímom používateľov komunikovať v reálnom čase prostredníctvom internetu. Takéto aplikácie je možné používať na rôznych zariadeniach vrátane smartfónov, tabletov a počítačov. Ponúkajú množstvo funkcií ako napríklad posielanie textových a hlasových správ, hlasové a video hovory, zdieľanie súborov a skupinové čety. Ďalšími funkciami môže byť posielanie emotikonov, nálepiek a iných interaktívnych prvkov medzi používateľmi. K dispozícii je viacero typov takýchto aplikácií vrátane aplikácií určených na osobné použitie, ako je napríklad WhatsApp a Messenger alebo aplikácií určených na firemné použitie, ako sú Slack a Microsoft Teams. Aplikácie na podporu tímovej komunikácie sa môžu používať za účelom spolupráce na pracovných projektoch, komunikácie s kolegami alebo s cieľom udržiavania kontaktu s priateľmi a rodinou, respektíve pri udržiavaní spoločenských kontaktov.

Aplikácie na podporu tímovej komunikácie sú zabezpečené v záujme ochrany súkromia, chránia informácie používateľov a udržiavajú ich konverzácie v súkromí. Mnohé aplikácie na podporu tímovej komunikácie sú zabezpečené šifrovaním medzi koncovými zariadeniami (E2EE – end-to-end šifrovanie). Ide o bezpečný komunikačný proces, ktorý zabraňuje tretím stranám v prístupe k údajom prenášaným z jedného koncového bodu do druhého (Kim, 2024). Šifrovanie údajov je proces používania algoritmu, ktorý transformuje štandardné textové znaky do nečitateľného formátu. Tento proces používa šifrovacie kľúče na zakódovanie údajov tak, aby boli údaje dostupné len autorizovaným používateľom. Šifrovanie medzi koncovými zariadeniami je proces šifrovania, ktorý zabezpečuje komunikáciu z jedného koncového bodu do druhého (IBM, 2024).

2 Stručná charakteristika vybraných aplikácií na podporu tímovej komunikácie

V tejto kapitole sa zameriame na stručnú charakteristiku nami vybraných aplikácií na podporu tímovej komunikácie, ktoré budeme ďalej porovnávať.

2.1 Microsoft Teams

Spoločnosť Microsoft má veľa skúseností s nástrojmi na spoluprácu a komunikáciu a Microsoft Teams je vývojom jej predchádzajúcich riešení. Teams zdedil všetky skúsenosti a know-how získané z iných podnikových riešení a za posledných 20 rokov je prirodzeným nástupcom Microsoft Lync a Skype for Business. Microsoft Teams prepája ostatné služby a aplikácie spoločnosti Microsoft používané v podnikovom prostredí, vďaka čomu znižuje súvisiace náklady (Ferreira, 2020).

Microsoft Teams je k dispozícii ako webová aplikácia a je možné k nej pristupovať priamo z prehliadača na adrese <https://teams.microsoft.com>. Webová verzia umožňuje využívať hlavné funkcie aplikácie Microsoft Teams, ale v závislosti od používaného prehliadača je možné sa stretnúť s určitými obmedzeniami. Plnohodnotné skúsenosti zo služby Microsoft Teams možno dosiahnuť len pomocou desktopových klientov dostupných pre systémy Windows, macOS a Linux. Microsoft Teams v systéme Windows poskytuje na

stiahnutie inštaláčne súbory MSI v 32-bitovej, 64-bitovej a ARM64 architektúre. Architektúra x86 (32-bitová vs. 64-bitová) aplikácie Teams je nezávislá od architektúry systému Windows a balíka Office, ktorý je nainštalovaný. Používatelia počítačov Mac môžu nainštalovať aplikáciu Teams pomocou inštaláčneho súboru PKG pre počítače so systémom macOS. Mobilné aplikácie Teams sú k dispozícii pre systémy Android a iOS, sú určené pre používateľov, ktorým umožňujú vzájomné zvukové hovory a konverzácie pomocou číetov. Mobilné aplikácie sú dostupné v mobilných obchodoch Google Play a Apple App Store.

Microsoft Teams ponúka viacero rôznych verzií Microsoft Teams bezplatnú verziu, Microsoft Teams Essentials, Microsoft 365 Business Basic a Microsoft 365 Business Standard. V tab. 1 uvádzame vybrané rozdiely medzi ponúkanými verziami z hľadiska ich funkcionality na podporu komunikácie.

Tab. 1: Verzie Microsoft Teams

Charakteristika	Microsoft Teams - bezplatná verzia	Microsoft Teams Essentials	Microsoft 365 Business Basic	Microsoft 365 Business Standard
<i>Cena</i>	zadarmo	4,70€ pre 1 používateľa na 1 mesiac	5,60€ pre 1 používateľa na 1 mesiac	11,70€ pre 1 používateľa na 1 mesiac
<i>Maximálny počet používateľov</i>	100	300	300	300
<i>Neobmedzený počet správ a vyhľadávanie v tímoch</i>	Áno	Áno	Áno	Áno
<i>Prístup pre hostí</i>	Nie	Áno	Áno	Áno
<i>Prílohy súborov v čete</i>	2GB/používateľ	2GB/používateľ	1TB/používateľ	1TB/používateľ
<i>Ukladanie súborov</i>	5GB	10GB	1TB	1TB
<i>Spolupráca v reálnom čase v mobil. aplikácii Microsoft 365 (Word, Excel, Power Point,...)</i>	Áno	Áno	Áno	Áno
<i>Viac ako 250 integrovaných aplikácií a služieb pre Teams</i>	Nie	Áno	Áno	Áno
<i>Čet medzi pracovnými a osobnými kontami</i>	Áno	Áno	Áno	Áno

Zdroj: spracované podľa: (Microsoft.com, 2024)

2.2 Slack

Slack je nástroj na spoluprácu, ktorý je možné používať ako aj pomocou webového prehliadača, tak aj ako aplikáciu v počítači alebo mobilnom zariadení. Bez potreby inštalácie aplikácie fungujú aj iné služby, ako napríklad Zoom, Teams alebo Discord. Slack používa takzvané pracovné priestory nazývané aj kanály na udržiavanie organizácie projektov a na uľahčenie komunikácie medzi členmi tímu (Bernstein, 2020). Takisto podporuje aj asynchrónnu prácu, keď je práca organizovaná v kanáloch, prístup k potrebným informáciám je umožnený bez ohľadu na polohu používateľa, časové pásmo alebo funkciu. Čo sa týka Slacku všetci v organizácii majú prístup k rovnakým zdieľaným informáciám, v ktorých sa dá vyhľadávať. Pri spolupráci tímov v kanáloch je možné informácie zdieľať so všetkými naraz, čo pomáha tímom zostať v súlade a rýchlejšie prijímať rozhodnutia (Slack.com, 2024c). Existujú viaceré verzie Slacku, pričom základné rozdiely medzi nimi uvádzame v tab. 2.

Tab. 2: Verzie Slacku

Charakteristika	Slack - bezplatná verzia	Slack Pro	Slack Business+	Slack Enterprise Grid
<i>Cena</i>	zadarmo	8,25€ pre 1 používateľa na 1 mesiac	14,10€ pre 1 používateľa na 1 mesiac	Cena je známa až na základe dopytu
<i>História správ</i>	90-dňový prístup	Neobmedzená história	Neobmedzená história	Neobmedzená história
<i>Možnosť integrácie s inými aplikáciami</i>	Max. 10 aplikácií	Neobmedzený počet aplikácií	Neobmedzený počet aplikácií	Neobmedzený počet aplikácií
<i>Skupinové verzus individuálne konverzácie</i>	Audio a video konverzácie len 1:1	Skupinové audio a video konverzácie	Skupinové audio a video konverzácie	Skupinové audio a video konverzácie
<i>Protokoly auditu</i>	Nie	Nie	Nie	Áno
<i>Počet „workspace“ (virtuálny pracovný priestor)</i>	1	1	1	Neobmedzený
<i>Zdieľanie dokumentov</i>	Nie	Áno	Áno	Áno, vrátane vlastných šablón
<i>Export údajov z kanálov a konverzácií</i>	Nie	Nie	Áno	Áno
<i>Zabudovaný zoznam zamestnancov</i>	Nie	Nie	Nie	Áno
<i>Prevenencia voči strate dát, offline záloha</i>	Nie	Nie	Nie	Áno

Zdroj: spracované podľa: (Slack.com, 2024a)

2.3 Google Chat

Google Chat, predtým známy ako Hangouts Chat, je komunikačný softvér vyvinutý spoločnosťou Google ako súčasť Google Workspace. Je určený na uľahčenie priameho posielania správ a tímovej spolupráce. Vďaka integrácii s ostatnými nástrojmi Google Workspace, ako sú Gmail, Disk a Dokumenty, ponúka Google Chat bezproblémové prostredie pre používateľov Google. Jeho rozhranie je jednoduché a zameriava sa na zefektívnenie komunikácie a spolupráce v rámci tímov a organizácií. Pomocou Google Chatu je možné posielat správy komukoľvek, kto má adresu Gmail alebo Google účet, takisto poskytuje aj možnosť nastavenia skupinových četov s viacerými účastníkmi alebo posielanie správ jeden na jedného. Google Chat je dostupný pre webové prehliadače, mobilné zariadenia a ako samostatná aplikácia. Existujú viaceré verzie Google Workspace, ktoré sa však od seba príliš nelíšia, čo sa týka podpory komunikácie a četu (tab. 3).

Tab. 3: Verzie Google Workspace

Charakteristika	Google Workspace Business Starter	Google Workspace Business Standard	Google Workspace Business Plus	Google Workspace Enterprise
<i>Cena</i>	5,75€ pre 1 používateľa na 1 mesiac	11,50€ pre 1 používateľa na 1 mesiac	17,25€ pre 1 používateľa na 1 mesiac	Cena je známa až na základe dopytu
<i>Videokonferencie a hlasové konferen.</i>	100 účastníkov	150 účastníkov	500 účastníkov	1000 účastníkov
<i>Záznamy stretnutí uložené na Google Disk</i>	Nie	Áno	Áno	Áno
<i>Google Disk (cloudové úložisko)</i>	30GB na 1 používateľa	2TB na 1 používateľa	5TB na 1 používateľa	5TB na 1 používateľa s možnosťou požiadať o viac
<i>Tímové zasielanie správ v čete</i>	Áno	Áno	Áno	Áno
<i>Zapnutie alebo vypnutie histórie</i>	Áno	Áno	Áno	Áno
<i>Individuálny externý čet</i>	Áno	Áno	Áno	Áno
<i>Pokročilé četovacie priestory</i>	Nie	Áno	Áno	Áno

Zdroj: spracované podľa: (Google.com, 2024b)

Google Chat poskytuje funkciu nazývanú „priestory“ alebo anglicky „spaces“. Táto funkcia umožňuje trvalý tímový čet a spoluprácu s možnosťou zdieľania súborov, pridelovania úloh a prezerania histórie správ. Priestory sú ideálne pre skupiny akéhokoľvek druhu, ktoré chcú pracovať na projektoch alebo komunikovať na základe spoločných záujmov (Google.com, 2024a).

Spomínané priestory umožňujú:

- zameranie konverzácie na spoločné oblasti záujmu, projekty a organizačné

- oznámenia;
- zdieľanie súborov a pridelovanie jednotlivých úloh členom ;
- pridanie aplikácií na vytváranie pracovných postupov;
- vytváranie vlákna pre veci, ktoré vyžadujú podrobnú diskusiu.

V čete je možné komunikovať so skupinou používateľov prostredníctvom priestorov alebo pomocou priamych správ. Tieto dve formy komunikácie fungujú rôznymi spôsobmi a slúžia na rôzne účely. Priestory sú centrálnym miestom, kde môžu používatelia zdieľať súbory, pridelovať úlohy a byť v spojení. Priame správy slúžia na čítanie priamo so skupinou.

3 Metodika výskumu

Za účelom objektívnejšieho vyhodnotenia vybraných aplikácií na podporu tímovej komunikácie sme vykonali dotazníkový prieskum, v ktorom sme sa používateľov týchto aplikácií pýtali na ich spokojnosť, prípadne zistené nedostatky. Na interpretáciu získaných dát bola použitá deskriptívna štatistika, ktorá poslúžila na vizuálnu prezentáciu dát pomocou grafov a tabuliek. Všetky otázky boli vyhodnocované v Exceli. Dotazník pozostával zo štyroch základných otázok, kde bol zisťovaný vek, pohlavie respondentov a oblasť, v ktorej respondenti pracujú. Štvrtá otázka bola zameraná na to, akú aplikáciu respondenti primárne používajú na komunikáciu v pracovnom prostredí, kde si respondenti mohli vybrať z troch porovnávaných aplikácií, a to Microsoft Teams, Slack alebo Google Chat, alebo mohli uviesť inú aplikáciu, ktorú používajú. V prípade, ak si respondenti vybrali Microsoft Teams, Slack alebo Google Chat, tak sa im zobrazila sekcia, ktorá sa zaoberala otázkami na nimi zvolenú aplikáciu a pozostávala z 10 otázok. Ak respondenti zvolili ako odpoveď v 4 otázke inú aplikáciu ako Microsoft Teams, Slack alebo Google Chat, tak sa im automaticky dotazník ukončil.

Prieskumu sa aktívne zúčastnilo celkovo 303 respondentov zo Slovenskej republiky. Prvá otázka zisťovala vek respondentov. Najviac respondentov bolo vo veku 21 až 30 rokov (246 respondentov), nasledovala skupina respondentov vo veku 41 a viac rokov (24 respondentov), ďalej skupina respondentov vo veku 31 až 40 rokov (21 respondentov) a najmenšie zastúpenie mala skupina respondentov do 20 rokov (12 respondentov). Pri konkrétnom veku bol najviac zastúpený vek 22 rokov (51 respondentov) a následne 21 rokov (38 respondentov). Pomer mužov a žien zapojených do výskumu bol približne rovnaký. Výskumu sa zúčastnilo 154 mužov a 149 žien.

V tretej otázke bolo predmetom záujmu to, v akej oblasti respondenti pracujú. Najväčší podiel respondentov pracuje v oblasti informačných technológií, kde bolo zaznamenaných 89 respondentov. Nasledovalo školstvo a vzdelávanie s 53 respondentmi, administratíva (33), zdravotníctvo (29), štátna správa (25) a financie a bankovníctvo (22). Ďalej cestovný ruch a marketing a reklama zhodne s 15 respondentami. Naopak, najmenší počet respondentov bol zaznamenaný v energetike (1), architektúre a dizajne (1), námorníctve (1), strojárstve (3), obchode a službách (4), automobilovom priemysle (6), stavebníctve (6), výskume (9) a práve a právnych službách (10). Každá z týchto oblastí mala iba niekoľko respondentov.

Po vyhodnotení dotazníka sme následne vykonali SWOT analýzu vybraných aplikácií na podporu tímovej komunikácie, aby sme poukázali na ich silné a slabšie stránky, príležitosti a prípadné hrozby.

4 Výsledky

Z prieskumu vyplýva, že najviac používanou aplikáciou na podporu tímovej komunikácie využívanou na komunikáciu v pracovnom prostredí je Microsoft Teams, ktorú preferovalo 116 respondentov. Nasleduje Slack s 66 respondentmi a Google Chat s 65 respondentmi. Okrem týchto troch hlavných aplikácií na podporu tímovej komunikácie uviedli respondenti aj iné

aplikácie: Messenger (15), WhatsApp (14), Outlook (10), Skype (7), Gmail (5 respondentov). A Viber, Cisco Jabber, Discord, Signal a Thunderbird zhodne s 1 respondentom. Týmto iným aplikáciám sme sa však ďalej už nevenovali.

Dotazníkový prieskum bol vykonaný za účelom toho, aby sme po jeho vyhodnotení mohli zostaviť SWOT analýzu, ktorá identifikuje silné stránky, slabé stránky, príležitosti a hrozby tej ktorej aplikácie. Na základe sumarizácie odpovedí z dotazníka bola teda pre každú aplikáciu zvlášť vykonaná SWOT analýza. Dotazník bol zameraný na jednotlivé funkcie vybraných aplikácií, kde používatelia hodnotili spokojnosť s vybranou funkciou na stupnici od 1 do 5, pričom 1 – Veľmi spokojný/á, 2 – Skôr spokojný/á, 3 – Neutrálne, 4 – Skôr nespokojný/á a 5 – Veľmi nespokojný/á. V dotazníku sa zisťovalo aj to, aké nedostatky alebo obmedzenia majú dané aplikácie. V prípade ak je súčet odpovedí: neutrálne, skôr nespokojný/á a veľmi nespokojný/á väčší ako jedna tretina z celkového počtu odpovedí, tak daná funkcia bola považovaná za slabú stránku aplikácie. Silné stránky aplikácií sú zoradené v tabuľkách SWOT analýzy od najlepšej po najhoršiu, podľa súčtu odpovedí: veľmi spokojný/á a skôr spokojný/á. Slabé stránky sú taktiež zoradené zostupne podľa súčtu odpovedí, takže ich zoznam sa začína slabou stránkou, ktorú môžeme považovať za najviac intenzívnu, pretože je najviac početná v odpovediach.

4.1 SWOT analýza aplikácie Microsoft Teams z hľadiska komunikačných funkcií

Ako môžeme vidieť v tab. 4, najviac spokojní sú používatelia aplikácie Microsoft Teams s funkciou posielania správ. Viac ako jedna tretina používateľov aplikácie uviedla ako jej slabé stránky kalendár a plánovač a integráciu Microsoft Teamsu s Microsoft Office. Ako príležitosť vidíme zlepšovanie niektorých funkcií aplikácie a takisto aj úpravu vzhľadu aplikácie na modernejší. Za hrozbu môžeme považovať stratu používateľov z dôvodu nespokojnosti s niektorými funkciami, únik citlivých informácií a konkurenciu zo strany iných podobných aplikácií.

Tab. 4: SWOT analýza Microsoft Teamsu

Silné stránky	Slabé stránky
<ul style="list-style-type: none"> • Funkcia posielania správ • Pracovanie v aplikácii • Funkcia hlasových hovorov a videohovorov • Používateľské rozhranie • Funkcia zdieľania súborov • Práca v kanáloch 	<ul style="list-style-type: none"> • Kalendár a plánovač • Integrácia s Microsoft Office • Zamrzanie aplikácie počas videohovorov • Slabé notifikácie • Nemoderný vzhľad • Vypadávanie zvuku pri audiohovoroch • Neustále upozornenia na aktualizácie • Prepínanie medzi kontami počas hovoru nie je možné • Problémy s otváraním súborov • Neintuitívne vytváranie kanálov
Príležitosti	Hrozby
<ul style="list-style-type: none"> • Zlepšovanie funkcií: kalendár a plánovač, integrácia s Microsoft Office, vyladenie notifikácií, zvýšenie intuitívnosti vytvárania kanálov • Modernizácia vzhľadu aplikácie 	<ul style="list-style-type: none"> • Strata používateľov z dôvodu nespokojnosti s niektorými funkciami aplikácie • Konkurencia zo strany iných podobných aplikácií • Únik citlivých informácií

Zdroj: (vlastná tvorba)

4.2 SWOT analýza aplikácie Slack z hľadiska komunikačných funkcií

Ako môžeme vidieť v tab. 5 najviac používateľov aplikácie Slack považuje za jej silné stránky prácu v kanáloch a následne funkciu posielania správ. Za slabé stránky považujú používatelia neprehľadnosť aplikácie, 90-dňovú históriu správ v bezplatnej verzii Slacku a všetky ostatné aspekty spomenuté v tab. 5. Ako príležitosť na zlepšenie aplikácie vidíme pridanie integrovaného kalendára, upravenie vzhľadu aplikácie a zjednodušenie prihlasovania do aplikácie. Rovnako ako pri aplikácii Microsoft Teams aj pri Slacku môžeme považovať za hrozbu stratu používateľov z dôvodu nespokojnosti s niektorými funkciami a konkurenciu zo strany iných podobných aplikácií.

Tab. 5: SWOT analýza Slacku

Silné stránky	Slabé stránky
<ul style="list-style-type: none"> • Práca v kanáloch • Funkcia posielania správ • Používateľské rozhranie • Pracovanie v aplikácii • Funkcia hlasových hovorov a videohovorov • Funkcia zdieľania súborov • Integrácia s Microsoft Office 	<ul style="list-style-type: none"> • Neprehľadnosť aplikácie • Len 90-dňová história správ v bezplatnej verzii • Chýba integrovaný kalendár • Zložité prihlasovanie • Automatické odhlasovanie po určitom čase • Problémy so zvukovým prenosom v hovoroch
Príležitosti	Hrozby
<ul style="list-style-type: none"> • Tvorba integrovaného kalendára • Úprava vzhľadu aplikácie • Zjednodušenie prihlasovania do aplikácie • Predĺženie, resp. rozšírenie histórie v bezplatnej verzii • Vyriešenie problému s automatickým a príliš skorým odhlasovaním používateľov 	<ul style="list-style-type: none"> • Strata používateľov z dôvodu nespokojnosti s niektorými funkciami aplikácie • Konkurencia zo strany iných podobných aplikácií • Únik citlivých informácií

Zdroj: (vlastná tvorba)

4.3 SWOT analýza aplikácie Google Chat z hľadiska komunikačných funkcií

Ako môžeme vidieť v tab. 6, najviac spokojní sú používatelia aplikácie Google Chat s používateľským rozhraním aplikácie. Ako slabé stránky uviedli používatelia pomalosť aplikácie, časté aktualizácie a aktualizácie bez schválenia používateľa, slabú kvalitu hovorov a neprehľadnosť aplikácie. Ako príležitosti, ktoré by pomohli aplikácii vidíme úpravu vzhľadu aplikácie a pridanie upozornení pred aktualizáciami aplikácie. Hrozby pri aplikácii Google Chat sú rovnaké ako pri aplikácii Microsoft Teams a Slack (tab. 6).

Tab. 6: SWOT analýza Google Chatu

Silné stránky	Slabé stránky
<ul style="list-style-type: none"> • Používateľské rozhranie • Funkcia zdieľania súborov • Kalendár a plánovač • Práca v kanáloch • Pracovanie v aplikácii • Funkcia posielania správ • Integrácia s Microsoft Office 	<ul style="list-style-type: none"> • Pomalá aplikácia • Časté aktualizácie, aktualizácie bez schválenia používateľa • Slabá kvalita hovorov • Neprehľadná aplikácia
Príležitosti	Hrozby
<ul style="list-style-type: none"> • Zrýchlenie aplikácie • Zlepšenie kvality hovorov • Úprava vzhľadu aplikácie • Pridanie upozornení pred aktualizáciami aplikácie 	<ul style="list-style-type: none"> • Strata používateľov z dôvodu nespokojnosti s niektorými funkciami aplikácie • Konkurencia zo strany iných podobných aplikácií • Únik citlivých informácií

Zdroj: (vlastná tvorba)

4.4 Porovnanie výkonových nárokov jednotlivých aplikácií na pracovnú stanicu

Pre porovnanie jednotlivých výkonov na stanicu daných aplikácií sme vytvorili porovnávaciu tabuľku (tab. 7), ktorá poskytuje prehľad o tom, aké hardvérové a softvérové požiadavky kladú tieto aplikácie na pracovné stanice používateľov.

Tab. 7: Porovnanie výkonových nárokov jednotlivých aplikácií na pracovnú stanicu

	Microsoft Teams	Slack	Google Chat
Procesor (CPU)	1,1 GHz a rýchlejší, dvojjadrový	Neuvádza sa	Intel alebo AMD, dvojjadrový
Pamäť RAM	4 GB	Neuvádza sa	4 GB
Sieť	Stabilné internetové pripojenie	Stabilné internetové pripojenie	Stabilné internetové pripojenie
Operačný systém	Windows 11, 10, Windows 10 na ARM, Windows 8.1, Windows Server 2019, 2016 2012 R2	Windows 10 verzia 21H2 a vyššia, Windows Server 2016	Windows 7, 8, 10 (64-bit)

Zdroj: (vlastná tvorba)

4.5 Porovnanie jednotlivých aplikácií z hľadiska bezpečnosti

V aplikácii Microsoft Teams je zabudovaných viacero bezpečnostných opatrení na ochranu údajov:

- **End-to-end šifrovanie (E2EE)** – zabezpečuje šifrovanie video hovorov, hlasových hovorov a zdieľania obrazovky, čím zabraňuje prístupu tretích strán k údajom;
- **Azure Active Directory (Azure aktívny adresár)** – umožňuje jednotné prihlasovanie, multifaktorovú autentifikáciu a podmienený prístup;
- **Transport Layer Security (Prenosová vrstva)** – TLS šifruje údaje vrátane správ, súborov a stretnutí v pokoji aj pri prenose a zabraňuje odpočúvaniu;

- **Normy súladu** – Microsoft Teams je v súlade s normami: HIPAA, ISO 27001, ISO 27018, SSAE16 SOC 1 a SOC 2, GDPR;
- **Monitorovanie komunikácie** – MS Teams je vybavený funkciou monitorovania, ktorá správcom umožňuje sledovať prebiehajúce konverzácie. Monitorovanie sa musí vykonávať manuálne a obe strany s ním musia súhlasiť. Správcovia môžu zaregistrovať kľúčové slová, ktoré spustia upozornenie vždy, keď sa o nich hovorí. Vďaka tomu stačí aby používatelia používali kľúčové slová, aby upozornili správcov na bezpečnostné problémy ;
- **Správy o činnosti** – poskytujú prehľad o aktivitách používateľov v aplikácii vrátane čísel a stretnutí;
- **Čety pod dohľadom** – Microsoft Teams je vybavený bezpečnostným opatrením, ktoré obmedzuje súkromné správy medzi používateľmi, pokiaľ nie sú do četu pridaní určití používatelia. Táto funkcia môže byť užitočná pri pomoci vzdelávacím inštitúciám pri obmedzovaní súkromných interakcií medzi študentmi, takže čety môžu prebiehať len pod dohľadom učiteľa;
- **Microsoft Defender (Microsoft obranca)** – táto funkcia je zameraná na zabránenie šírenia škodlivých aktivít spojených so zdieľaním súborov. Súbory, ktoré sú bezpečné, sú označené ako „bezpečné odkazy“, takže používatelia sa môžu rozhodnúť, ktoré súbory stiahnu a ktorým sa vyhnú (Houssier, 2023).

Zabudované bezpečnostné opatrenia aplikácie Slack:

- **Certifikácie** - SOC 2, SOC 3, ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, ISO/IEC 27701, GovSlack SOC 3, tieto certifikácie potvrdzujú, že Slack spĺňa bezpečnostné štandardy;
- **GDPR** – Slack je v súlade s GDPR - všeobecné nariadenie na ochranu osobných údajov;
- **Normy HIPAA a FINRA** – Slack je možné nakonfigurovať na dodržiavanie predpisov HIPAA (Zákon o prenosnosti a zodpovednosti za zdravotné poistenie) a FINRA (Regulačný orgán pre finančný priemysel);
- **FedRAMP** (Federálny program riadenia rizík a autorizácie) – Slack má oprávnenie, aby spĺňal potreby organizácií vo verejnom sektore v oblasti dodržiavania predpisov;
- **Enterprise Key Management (Správa podnikových kľúčov)** – umožňuje administrátorom čiastočnú kontrolu nad šifrovaním údajov;
- **Integrácia s externými zabezpečovacími nástrojmi** – aplikácia Slack ponúka aj integráciu s externými zabezpečovacími nástrojmi (Slack.com, 2024b).

Zabudované bezpečnostné opatrenia aplikácie Google Chat:

- **Šifrovanie údajov** – Google Chat zabezpečuje šifrovanie údajov pri prenose aj v pokoji, tieto dáta sú chránené viacúrovňovým zabezpečením vrátane popredných šifrovacích technológií – HTTPS (Zabezpečený hypertextový prenosový protokol) a Transport Layer Security (Prenosová vrstva);
- **Bezpečnostné upozornenia** – slúžia na ochranu osobných údajov a hlásenie podozrivých prihlásení, škodlivého webu, súboru alebo aplikácie;
- **Blokovanie škodlivých reklám** – Google Chat zabezpečuje blokovanie škodlivých reklám, ktoré môžu obsahovať malvér, prekrývajú obsah alebo propagujú falošné výrobky;
- **Cloudová ochrana dát;**
- **Norma ISO/IEC 27001** – jedna z najuznávanejších a medzinárodne prijímaných nezávislých noriem zabezpečenia (Google.com, 2024c).

5 Záver

V článku sme zanalyzovali silné a slabé stránky vybraných aplikácií určených na podporu tímovej komunikácie. Konkrétne sme sa zamerali na Microsoft Teams, Slack a Google Chat. Naším cieľom nebolo určiť, ktorá z týchto aplikácií je najlepšia, pretože každá z nich má svoje výhody, nevýhody a určité špecifiká. SWOT analýzu pre jednotlivé aplikácie sme zostavili na základe výsledkov prieskumu, ktorý sme realizovali na vzorke 303 respondentov zo Slovenskej republiky. Respondenti uvádzali v dotazníku svoje osobné skúsenosti s danými aplikáciami prevádzkovanými na svojich vlastných počítačoch. Ide teda o sumarizáciu ich subjektívnych názorov. Zároveň je potrebné podotknúť, že tieto aplikácie sa neustále vyvíjajú, a preto sú výsledky tohto prieskumu skôr orientačné.

Záverom môžeme vyjadriť presvedčenie, že použitie vhodnej aplikácie na podporu tímovej komunikácie môže nepochybne zvýšiť efektivitu komunikácie medzi zamestnancami v pracovnom prostredí, čo sa môže pozitívne prejavovať pri vykonávaní procesov v danej organizácii. Komunikácia je dôležitá na všetkých úrovniach manažmentu organizácie – či už je to operatívna, taktická alebo strategická úroveň a implementácia ľubovoľného nástroja na jej zefektívnenie môže byť prínosom. Pritom je však z hľadiska bezpečnosti dobré siahnúť po overených a profesionálnych riešeniach, aby sa predišlo úniku citlivých informácií. Uvedené tri aplikácie spĺňajú obvyklé bezpečnostné štandardy a sú po tejto stránke vhodné.

Literatúra

1. Bernstein, J. (2020). *Slack Made Easy: Communication and Collaboration Simplified*. Independently published.
2. Ferreira, J. (2020). *Hands-On Microsoft Teams: A practical guide to enhancing enterprise collaboration with Microsoft Teams and Office 365*. Packt.
3. Google.com. (2024a). *Messaging and team collaboration*. <https://workspace.google.com/products/chat/>.
4. Google.com. (2024b). *Vyzkousejte Google Workspace na 14 dní. A otestujte si u toho i službu Gemini*. <https://workspace.google.com/pricing.html>.
5. Google.com. (2024c). *Work safer with Google Workspace*. workspace.google.com . <https://workspace.google.com/security/>.
6. Houssier, E. (2023, February 28). *MICROSOFT TEAMS SECURITY: IS YOUR DATA SECURE?*. Powell Security. <https://powell-software.com/resources/blog/microsoft-teams-security/>.
7. IBM.com. (2024, August 18). *What is end-to-end encryption (E2EE)?* <https://www.ibm.com/topics/end-to-end-encryption>.
8. Key, K. (2024, August 1). *The Best Private Messaging Apps for 2024*. PCMag.com. <https://www.pcmag.com/picks/best-secure-messaging-apps>.
9. Microsoft.com. (2024). *Find the right Microsoft Teams for your needs*. <https://www.microsoft.com/en/microsoft-teams/compare-microsoft-teams-business-options?market=af#x31ffe6258c7141f5bbb29f20b87ba3c3>.
10. Slack.com. (2024a). *Make Teamwork more productive*. <https://slack.com/pricing>.
11. Slack.com. (2024b). *SECURITY AT SLACK: Enterprise-grade data protection*. <https://slack.com/trust/security>.
12. Slack.com. (2024c). *What is Slack?* <https://slack.com/help/articles/115004071768-What-is-Slack->.

Regionálne disparity v spotrebných výdavkoch domácností na Slovensku

Regional Disparities in Household Consumption Expenditure in Slovakia

Lívia Krajčíková¹

Abstrakt

Porozumenie regionálnym rozdielom vo výdavkoch domácností je nevyhnutné pre efektívne formovanie sociálnej a regionálnej politiky. Cieľom tohto článku je analyzovať regionálne rozdiely vo výdavkoch slovenských domácností na základe dát zo zisťovania Rodinné účty. Analýza sa zameriava na 12 kategórií výdavkov podľa klasifikácie COICOP a porovnáva ich medzi ôsmimi kraji Slovenska s využitím viacrozmerných štatistických metód – viacrozmerná analýza rozptylu a permutačná viacrozmerná analýza rozptylu. Výsledky ukazujú, že hoci Bratislavský kraj vykazuje najvyššie celkové výdavky, štruktúra výdavkov v relatívnom vyjadrení je medzi kraji pomerne podobná, s dominanciou základných potrieb (bývanie a potraviny). Výsledky boli doplnené Bonferroniho post-hoc testom, ktorý identifikoval štatisticky významné rozdiely medzi jednotlivými kraji. Bratislavský kraj vykazuje signifikantne vyššie výdavky oproti všetkým ostatným krajom v kategóriách bývanie; reštaurácie a hotely; pošty a telekomunikácie; odevy a obuv.

Kľúčové slová

rodinné účty, domácnosti, výdavky, spotreba, viacrozmerná analýza rozptylu, permutačná viacrozmerná analýza rozptylu

Abstract

Understanding regional differences in household expenditure is essential for the effective formulation of social and regional policy. The aim of this paper is to analyse regional differences in the expenditure of Slovak households based on data from the Household Budget Survey. The analysis focuses on 12 categories of expenditures according to the COICOP classification and compares them across the eight regions of Slovakia using multivariate statistical methods – Multivariate Analysis of Variance and Permutational Multivariate Analysis of Variance. The results show that although the Bratislava region reports the highest total expenditures, the relative structure of expenditures is similar across regions, dominated by basic needs (housing and food). The results were supplemented by the Bonferroni post-hoc test, which identified statistically significant differences between individual regions. The Bratislava region shows significantly higher expenditures compared to all other regions in the categories of housing; restaurants and hotels; postal services and telecommunications; and clothing and footwear.

Key words

household budget survey, households, expenditure, consumption, Multivariate Analysis of Variance, Permutational Multivariate Analysis of Variance

JEL classification

R29, C19

¹ Bratislava University of Economics and Business, Faculty of Economic Informatics, Department of Statistics, Dolnozemska cesta 1, 852 35 Bratislava, livia.krajcikova@euba.sk.

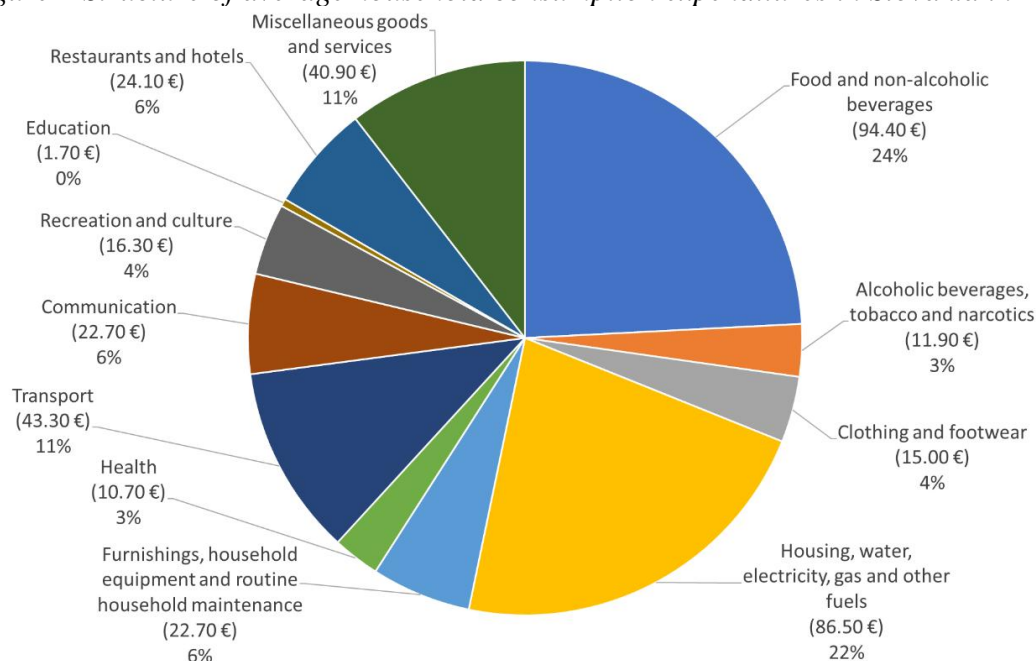
1 Introduction

Analysis of regional disparities in household consumption expenditure is an important part of examining socio-economic disparities in Europe. Piekut and Knapkova (2025) focus on differences in consumption expenditure between households in Western and Eastern Europe from 2000 to 2021. Using cluster analysis and regression analysis, they identify two distinct consumption patterns: while households in Eastern Europe allocate a higher share of their expenditures to basic needs (e.g., food and non-alcoholic beverages), households in Western Europe invest more in recreation, culture, and housing. The results suggest partial convergence in the consumption of certain categories, such as food and miscellaneous goods and services, indicating an improvement in the standard of living in Eastern Europe. However, significant differences persist in areas such as education, telecommunications, and clothing.

These findings are also relevant in the context of regional differences in consumption expenditure in Slovakia, where economic and historical factors similarly influence household spending patterns. This paper analyses these differences using data from the Household Budget Survey and applies the one-way Multivariate Analysis of Variance (MANOVA) method to identify statistically significant differences in consumption expenditure across Slovak regions in 2022. The aim is to identify structural differences between regions and assess their statistical significance, thus contributing to the broader discussion on socio-economic inequalities at the regional level.

According to Eurostat (2025), the Household Budget Survey (HBS) is a national statistical survey conducted in EU countries and coordinated by Eurostat, which collects data on household consumption expenditure on goods and services. These data are supplemented by information on household size and composition, income, and the characteristics of individuals living in private households. Household Budget Survey began to be implemented in most EU Member States in the early 1960s, and since 1988, Eurostat has collected and published data from this survey every five years. The survey focuses on consumption expenditure, i.e., what people spend on goods and services to satisfy their needs. Data from the survey also serve as an important input—especially at the national level—for determining weights in the consumer basket used to calculate the Consumer Price Index (CPI).

Figure 1 Structure of average household consumption expenditures in Slovakia in 2022



Source: Database of the Statistical Office of the Slovak Republic

To classify consumption expenditure, the international Classification of Individual Consumption According to Purpose (COICOP) is used, as recommended by Eurostat for Household Budget Statistics. According to this classification, consumption expenditure is divided into 12 main divisions, as outlined in the publication *Classification of Individual Consumption According to Purpose 2018* (United Nations, 2023): Food and non-alcoholic beverages; Alcoholic beverages, tobacco and narcotics; Clothing and footwear; Housing, water, electricity, gas and other fuels; Furnishings, household equipment and routine household maintenance; Health; Transport; Communication; Recreation and culture; Education; Restaurants and hotels; Miscellaneous goods and services.

To begin with, we present the overall structure of the average consumption expenditure of Slovak households in 2022 (Figure 1). The data are based on the database of the Statistical Office of the Slovak Republic (SOSR) and include 12 categories of expenditure according to the COICOP classification. Figure 1 clearly shows that almost half of total household expenditures are allocated to basic necessities such as food and non-alcoholic beverages and housing, including energy. In 2022, the highest monthly per capita expenditure in Slovakia was on food and non-alcoholic beverages, amounting to 94.40€, which represented approximately 24% of total household consumption expenditures. The second highest category was housing, water, electricity, gas, and other fuels, with an average monthly cost of 86.50€, which represented approximately 22% of total household consumption expenditures (Figure 1). A significant portion of the household budget is also allocated to transport; restaurants and hotels; and miscellaneous goods and services. However, the share of each category can vary significantly depending on the socio-economic status of the household and the region in which its members live.

2 Literature review

Household expenditure reflects the economic behaviour and living standards of households, making them the subject of numerous academic studies. The following section provides an overview of relevant research focused on the regional analysis of household consumption expenditures.

Piekut and Piekut (2022) analysed the expenditure patterns of European households between 2004 and 2020 and identified groups of countries with similar consumption structures using cluster analysis. They found that household expenditure increased in most countries, with key categories being food, housing, and services. The COVID-19 pandemic caused a temporary decrease in consumption across all areas except food and non-alcoholic beverages. The results showed that similarities in countries' consumption patterns change over time, although some countries maintained stable positions within the created clusters. This study provides a valuable methodological framework for examining regional differences in household expenditures and their evolution over time.

Lazíková (2017) analysed the development of household income and expenditure in Slovakia between 2000 and 2015, with an emphasis on regional differences. She pointed out that despite the increase in net income and expenditure of households in recent years, significant disparities between regions persist—particularly between the Bratislava and Prešov regions. She identified key factors influencing this development, such as Slovakia's accession to the EU, the adoption of the euro, and the economic crisis, emphasizing that these events did not contribute to reducing regional disparities.

Hupková et al. (2018) focused on economic and social disparities in Slovak regions during 2005–2015, examining household expenditure and the factors influencing them. The results of regression and correlation analyses confirmed a strong dependence of expenditures on nominal gross monthly wages and the at-risk-of-poverty rate in all Slovak regions. The

strongest dependency was observed in the Trnava and Košice regions. In Nitra and Košice, a negative correlation was identified between the risk of poverty and household expenditure.

In the context of analysing regional differences in household financial situations, the study by Kozak et al. (2022) is also relevant. It focused on evaluating the financial situation of Polish households and its regional differentiation in 2018. The authors used both one-dimensional comparisons of indicators and multidimensional assessment using the TOPSIS method. The results showed significant regional differences. The main factors behind the differences included economic and demographic characteristics such as employment rates, the education level of the household head, and the ability to repay debts.

Regional differences in the expenditures and incomes of Polish households over a longer period (2000–2019) were analysed in the study by Kasprzyk and Leszczyńska (2021). Based on data from the Household Budget Survey, the research identified increasing household income and expenditure over time. The study also confirmed the existence of regional disparities in household spending and income. Despite a decreasing trend in regional inequality over time, eastern regions remained economically the weakest, with their financial situation mainly influenced by socio-economic and demographic factors.

3 Methodology

To analyse regional differences in household expenditure, we used the MANOVA and Permutational Multivariate Analysis of Variance (PERMANOVA) methods. The assumptions for applying MANOVA—multivariate normality and homogeneity of covariance matrices across groups—were not met; therefore, we decided to supplement the analysis with PERMANOVA. Unlike MANOVA, PERMANOVA does not require strict fulfilment of the assumptions, and thus represents a suitable alternative for verifying regional differences in expenditure structure. For a more detailed comparison between individual regions, the Bonferroni post-hoc test was subsequently applied.

Multivariate Analysis of Variance is a statistical method that extends the univariate analysis of variance (ANOVA) to the case when there is more than one dependent variable. While ANOVA tests for differences in the means of a single dependent variable across multiple groups, MANOVA allows us to examine differences in the mean vectors of multiple dependent variables between groups. Typical applications of MANOVA include, for example, investigating differences in sets of psychological test scores, where each person completes multiple tests and we want to determine whether the result vectors differ among various groups (Hebák, Hustopecký, 1987).

MANOVA focuses on testing whether the mean vectors of the dependent variables are the same across different groups. It is used to test the null hypothesis

$$H_0: \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \dots = \boldsymbol{\mu}_p$$

which states that the mean vectors of the dependent variables are equal across all groups. The vector $\boldsymbol{\mu}_i$ denotes the mean vector of the variables for group i . The alternative hypothesis states that at least one of the mean vectors is different:

$$H_1: \boldsymbol{\mu}_i \neq \boldsymbol{\mu}_j \text{ for at least one } i \neq j.$$

If the null hypothesis is rejected, it means that there is a statistically significant difference between the groups in some (or all) of the dependent variables.

PERMANOVA is a non-parametric statistical permutation test used to compare groups across multiple dependent variables. This test is especially useful when analysing data that are not normally distributed or when the assumption of homoscedasticity (equal variances) is not met. PERMANOVA uses a permutation-based approach, which involves randomly permuting the data to obtain empirical distributions of test statistics. In PERMANOVA, various distance

metrics can be selected (e.g., Euclidean distance, Manhattan distance), increasing the flexibility of the analysis (Anderson, 2017).

3.1 Fundamental Concepts of MANOVA

The main idea of analysis of variance is the decomposition of the total variability (T) into a between-group component (B) and a within-group component (W). Individual multivariate observations are denoted as \mathbf{x}_{hi} , where the index $h = 1, 2, \dots, k$ indicates the group, and the index $i = 1, 2, \dots, n_h$ identifies the object. The sample size for group $h = 1, 2, \dots, k$ is n_h and the total number of observations is $n = \sum_{h=1}^k n_h$. To decompose the variability, we first compute the following statistics within each group: the vector of group means $\bar{\mathbf{x}}_h$ (1) and the group covariance matrices \mathbf{S}_h (2) (Hebák, Hustopecký, 1987):

$$\bar{\mathbf{x}}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} \mathbf{x}_{hi} \quad (1)$$

$$\mathbf{S}_h = \frac{1}{n_h - 1} \sum_{i=1}^{n_h} (\mathbf{x}_{hi} - \bar{\mathbf{x}}_h)(\mathbf{x}_{hi} - \bar{\mathbf{x}}_h)^T \quad (2)$$

Additionally, we compute the overall mean vector $\bar{\mathbf{x}}$ (3) and the pooled estimate of the covariance matrix $\bar{\mathbf{S}}$ (4).

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{h=1}^k \bar{\mathbf{x}}_h n_h \quad (3)$$

$$\bar{\mathbf{S}} = \frac{\sum_{h=1}^k \mathbf{S}_h (n_h - 1)}{n - k} \quad (4)$$

The calculations of MANOVA test statistics are based on matrices of between-group and within-group variability.

- Between-group variability matrix (B):

$$\mathbf{B} = \sum_{h=1}^k n_h (\bar{\mathbf{x}}_h - \bar{\mathbf{x}})(\bar{\mathbf{x}}_h - \bar{\mathbf{x}})^T \quad (5)$$

- Within-group variability matrix (W):

$$\mathbf{W} = \sum_{h=1}^k (n_h - 1) \mathbf{S}_h \quad (6)$$

- Total variability matrix (T):

$$\mathbf{T} = \mathbf{B} + \mathbf{W} \quad (7)$$

3.2 Assumptions of the MANOVA Model

For MANOVA to yield reliable results, the following assumptions must be met:

- randomness of the sample,
- independence between the defined groups (absence of multicollinearity),
- existence of multivariate normal distribution within groups,
- equality of covariance matrices between groups,
- homogeneity of variances between groups (homoscedasticity).

To test the equality of covariance matrices, Box's M test is used (Hebák, Hustopecký, 1987). The null hypothesis of equal covariance matrices, $H_0: \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \dots = \boldsymbol{\Sigma}_k$, is rejected if the value of the test statistic

$$K = \frac{1}{C_p} (n - k) \ln |\bar{\mathbf{S}}| - \sum_{h=1}^k (n_h - 1) \ln |\mathbf{S}_h| \quad (8)$$

exceeds the critical value of the chi-square distribution $\chi_{1-\alpha}^2 [(k-1)p(p+1)/2]$. In this formula, C_p is a constant that improves the approximation:

$$C_p = 1 + \frac{2p^2 + 3p - 1}{6(k-1)(p+1)} \left(\sum_{h=1}^k \frac{1}{n_h - 1} - \frac{1}{n - k} \right), \quad (9)$$

where p is the number of dependent variables (i. e., the number of expenditure categories analysed simultaneously).

To verify the assumption of multivariate normality, several tests can be applied, such as Mardia's test (based on multivariate skewness and kurtosis), Royston's test (a multivariate version of the Shapiro-Wilk test), or the Henze-Zirkler test, which is based on a non-negative functional distance that measures the deviation between two distribution functions. Details of these computations can be found in Korkmaz et al. (2014). The null hypothesis tested is that the empirical and theoretical (normal) probability distributions do not significantly differ, versus the alternative hypothesis that they do differ.

To diagnose multicollinearity, several approaches may be used, including sample pairwise correlation coefficients, the determinant of the correlation matrix, the smallest eigenvalue of the correlation matrix, the condition index, the Variance Inflation Factor (VIF), the M statistic, or the Farrar-Glauber test for variable independence, which is discussed in more detail by Šoltés (2019).

To assess homoscedasticity, several tests can be used. These test the null hypothesis that the random component has constant variance (homoscedastic), against the alternative hypothesis that the variance is not constant (heteroscedastic). Bartlett's test for homoscedasticity is a universal test suitable for both balanced and unbalanced samples (a balanced sample has equal numbers of observations in each group, while an unbalanced sample has varying numbers). Cochran's test and Hartley's test are suitable for balanced samples. In our case, we applied Levene's test, which can be used even when the assumption of normality is violated in one or more subgroups. Details of these tests can be found in Pacáková et al. (2012).

3.3 Test Statistics for MANOVA

In MANOVA, several test statistics are used to decide whether to reject or accept the null hypothesis. The values of these statistics are calculated based on the between-group variability matrix \mathbf{B} and the within-group variability matrix \mathbf{W} or based on the eigenvalues λ_g of the matrix $\mathbf{B}\mathbf{W}^{-1}$, where $g = 1, 2, \dots, s$, where s is the number of nonzero eigenvalues of the matrix $\mathbf{B}\mathbf{W}^{-1}$, and it holds that $s = \min(p, k - 1)$, with p being the number of dependent variables and k being the number of groups (Carey, 1998).

Wilk's Lambda – ratio of within-group variability to total variability

$$\lambda = \frac{|\mathbf{W}|}{|\mathbf{T}|} = \frac{|\mathbf{W}|}{|\mathbf{W} + \mathbf{B}|} = \prod_{g=1}^s \frac{1}{1 + \lambda_g} \quad (10)$$

Hotelling-Lawley's Trace

$$T^2 = st(\mathbf{B}\mathbf{W}^{-1}) \equiv \sum_{g=1}^s \lambda_g \quad (11)$$

Roy's Largest Root

$$V = \max \lambda_g \quad (12)$$

Pillai's Trace – considered the most powerful and robust of the mentioned statistics

$$Pillai's\ trace = st[\mathbf{B}(\mathbf{B} + \mathbf{W})^{-1}] = \sum_{g=1}^s \frac{\lambda_g}{1 + \lambda_g} \quad (13)$$

Each of these test statistics can be approximated by the F -distribution, which allows us to draw conclusions about the hypotheses.

If the calculated F -statistic is greater than the critical value of F (based on the chosen significance level and degrees of freedom), we reject the null hypothesis and conclude that statistically significant differences exist between the studied groups.

3.4 Multiple Comparison Methods

If the null hypothesis is rejected in favour of the alternative hypothesis about the inequality of group means, we obtain only a vague statement indicating that there exists at least one pair (or more) of means that differ. In such cases, the analysis is extended with multiple comparison methods, which help identify specific pairs of group means that differ significantly (Pacáková et al., 2012). The null hypothesis for each of these methods takes the form:

$$H_0: \mu_i = \mu_{i'} (i, i' = 1, 2, \dots, k; i \neq i')$$

Various multiple comparison methods can be applied, such as Scheffé's method, Tukey's method, Bonferroni method, Duncan's method, or Kramer's method. In our case, we applied the Bonferroni method, which is suitable for unbalanced samples. Formulas for calculating the critical values of selected multiple comparison methods are presented in Pacáková et al. (2012).

The analysis is based on anonymized microdata from the Household Budget Survey for 2022, provided by the Statistical Office of the Slovak Republic for scientific purposes. All analyses (MANOVA, PERMANOVA, Bonferroni post-hoc tests) were conducted using the Python programming language in the Jupyter Lab environment.

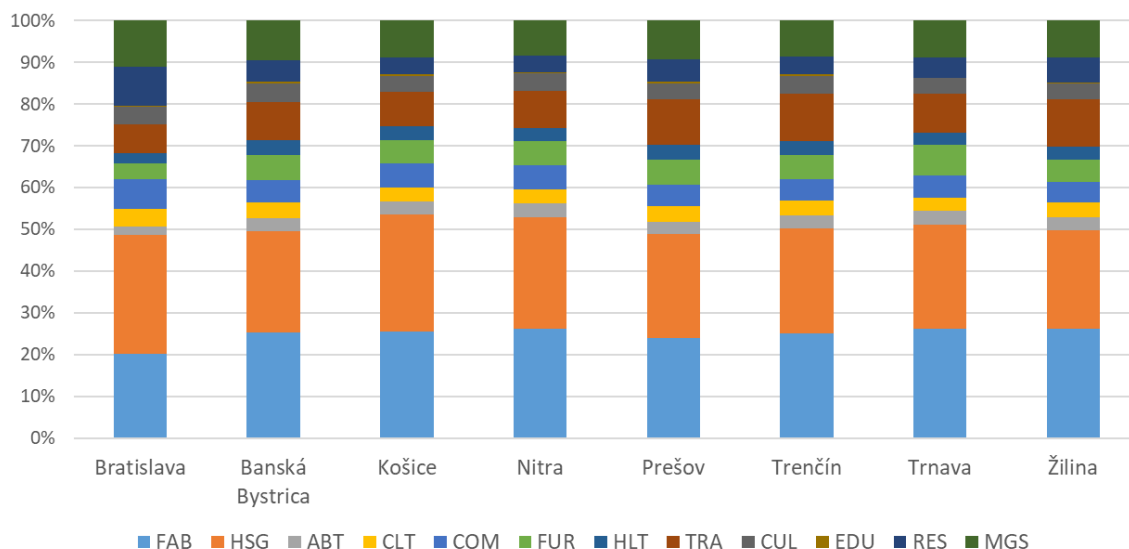
4 Results and Discussion

We began by looking at the average consumption expenditures of Slovak households and the average net income of households across different Slovak regions.

Piekut and Knapkova (2025) reveal two distinct consumer patterns among European households. According to their study, households in Eastern European countries tend to prioritize basic needs, such as food and non-alcoholic beverages, while households in Western Europe allocate more spending to other needs and services, including recreation, culture, and housing. Slovakia, which is usually classified as an Eastern European country based on household consumer behaviour, shows a trend towards the Western European consumption model, according to the findings of Piekut and Knapkova (2025). Figure 2 shows that across Slovak regions, the share of spending on basic needs (housing and food) remains high—around 50%. Although households do spend on recreation, culture, and other goods and services, the share of these categories is not significant enough to confirm a clear shift toward the Western consumption pattern. However, this shift may not be most visible at the regional level, but rather in the structure of household expenditures by household type. It is likely that differences in consumer behaviour are more pronounced between households with varying income levels, sizes, or compositions, rather than between regions. A more detailed analysis by household type could therefore better reveal any tendencies of convergence toward the Western European model.

Spending category abbreviations used in the graphs and tables are based on the classification presented in the introduction: FAB – Food and non-alcoholic beverages, ABT – Alcoholic beverages, tobacco and narcotics, CLT – Clothing and footwear, HSG – Housing, water, electricity, gas, and other fuels, FUR – Furnishings, household equipment, and routine household maintenance, HLT – Health, TRA – Transport, COM – Communication, CUL – Recreation and culture, EDU – Education, RES – Restaurants and hotels, MGS – Miscellaneous goods and services.

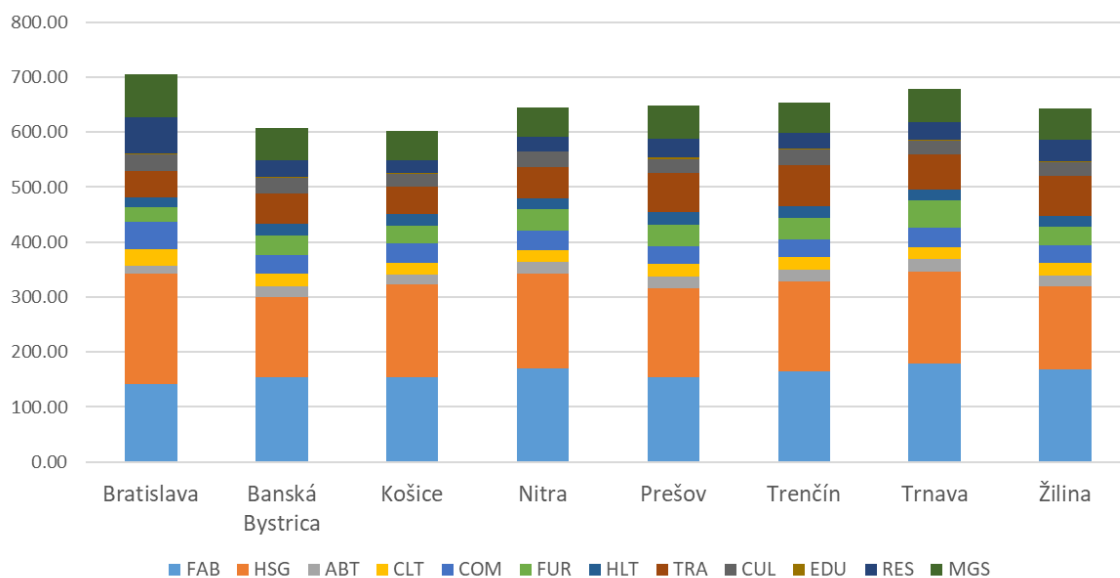
Figure 2 Share of individual expenditure categories in the total household expenditures in the given region



Source: Own processing based on data from the Household Budget Survey (SOSR).

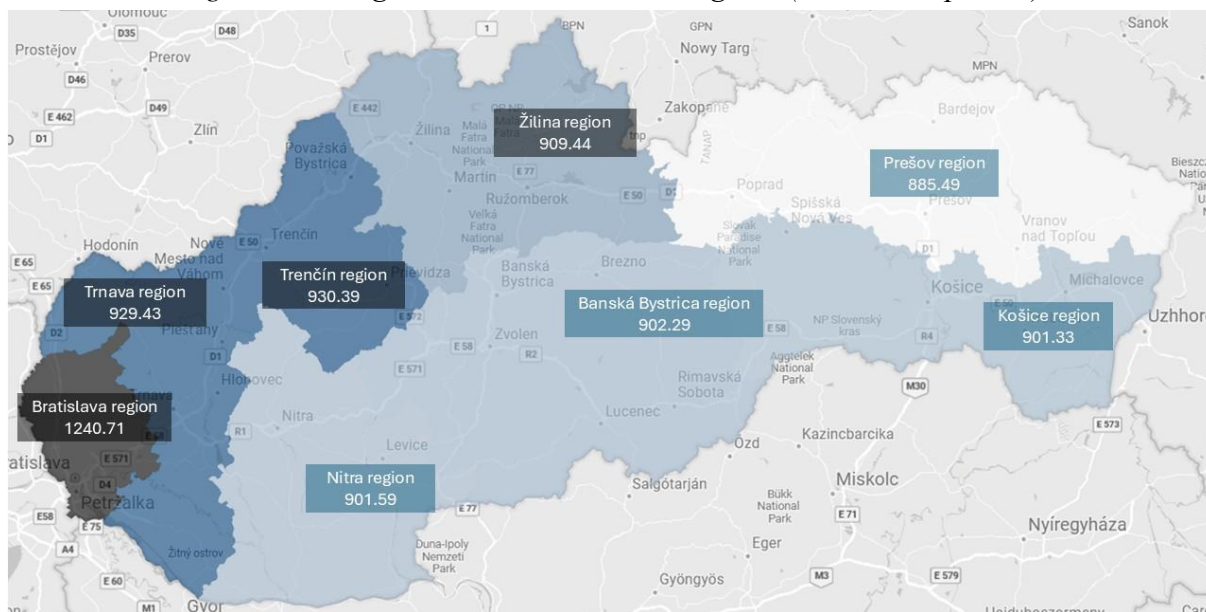
According to Figure 3, households in the Bratislava region have the highest total expenditures, while those in the Košice region show the lowest. The difference between these two regions amounts to approximately €100 per person per month, highlighting significant regional disparities in consumption expenditures. Despite these differences in absolute spending levels, the structure of consumption across regions is very similar.

Figure 3 Average monthly household expenditures in Slovak regions according to 12 COICOP categories (in €/month/person)



Source: Own processing based on data from the Household Budget Survey (SOSR).

In our analysis, the average net household incomes were aggregated at the regional level. The results show substantial regional differences in average net income (Figure 4), with the Bratislava region reaching the highest average income (1240.71€/person/month), while the lowest values were recorded in the Prešov region (885.49€/person/month).

Figure 4 Average net income in Slovak regions (in €/month/person)

Source: Own processing based on data from the Household Budget Survey (SOSR) in PowerBI.

In Figure 4, these income differences are illustrated on a map using a colour scale from white (lowest income) to black (highest income). The contrast between the Bratislava region and the other regions is clearly visible, which may be related to the concentration of economic opportunities and higher wage levels in this area. The remaining regions range between approximately 885€ and 930€/person/month, showing relatively minor differences between them.

Similarly to the findings of Lazíková (2017), who revealed significant income and expenditure differences among Slovak regions for the period 2005–2015, our 2022 analysis also shows that the Bratislava region records the highest average net income per person. Lazíková noted that the Bratislava region had the highest income levels and the lowest share of spending on basic needs, in line with Engel's law. However, in our analysis, we observe that the relative structure of household expenditures in the Bratislava region is similar to other regions—around 50% of spending goes to basic needs (housing and food). Nevertheless, in absolute terms, the Bratislava region spends significantly more on less essential categories, such as miscellaneous goods and services or restaurants and hotels. This difference is therefore more visible in absolute expenditure amounts rather than in their relative structure.

Before proceeding with the multivariate analysis, we tested the assumptions necessary for the valid application of MANOVA. Multivariate normality was assessed using the Henze-Zirkler's test, where the p-value was <0.001, leading to the rejection of the null hypothesis that the data originate from a multivariate normal distribution. Homogeneity of covariance matrices was tested using Box's M test, which also resulted in the p-value <0.001, thus rejecting the null hypothesis of equality of covariance matrices across groups. Homoskedasticity was evaluated by Levene's test for each of the 12 consumption categories separately; for all variables, the p-value was <0.001, indicating heteroskedasticity.

Since none of the assumptions for MANOVA were satisfied, caution is necessary when interpreting its results. Nevertheless, MANOVA was performed, and the results showed a statistically significant difference in household expenditures across Slovak regions for all four test statistics (Wilk's lambda, Pillai's trace, Hotelling-Lawley trace, Roy's largest root) at the 0.05 significance level (Figure 5).

Figure 5 Results of the MANOVA test

Multivariate linear model					
Intercept	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.3249	12.0000	4972.0000	860.9382	0.0000
Pillai's trace	0.6751	12.0000	4972.0000	860.9382	0.0000
Hotelling-Lawley trace	2.0779	12.0000	4972.0000	860.9382	0.0000
Roy's greatest root	2.0779	12.0000	4972.0000	860.9382	0.0000
Kraj	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.7943	84.0000	30459.4727	13.8981	0.0000
Pillai's trace	0.2153	84.0000	34846.0000	13.1642	0.0000
Hotelling-Lawley trace	0.2472	84.0000	21352.2280	14.6285	0.0000
Roy's greatest root	0.1918	12.0000	4978.0000	79.5839	0.0000

Source: Own processing in Jupyter Lab (Python).

Due to the violation of key assumptions, permutational multivariate analysis of variance (PERMANOVA) was subsequently applied as a robust alternative, and it confirmed the existence of statistically significant regional differences in the expenditure structure of Slovak households.

Figure 6 Results of the PERMANOVA test

method name	PERMANOVA
test statistic name	pseudo-F
sample size	4991
number of groups	8
test statistic	11.069869
p-value	0.001
number of permutations	999
Name: PERMANOVA results, dtype: object	

Source: Own processing in Jupyter Lab (Python).

The results of the PERMANOVA test (Figure 6) confirm statistically significant differences in household expenditures across Slovak regions (pseudo-F = 11.07, p-value = 0.001). The analysis included 4991 households divided into eight groups based on their region. 999 permutations were used, ensuring robustness of the results.

After confirming statistically significant regional differences, we conducted additional pairwise comparisons between regions for each expenditure category individually. Table 1 presents the results of the Bonferroni post-hoc test for all pairs of Slovak regions. The test examines whether there are statistically significant differences in average expenditures between regions within each COICOP category. For each regional pair, the p-value and the difference in average expenditures are shown. Statistically significant values at the $\alpha = 0.05$ significance level are highlighted in blue.

The results show that average spending on food and non-alcoholic beverages in the Bratislava region is significantly lower (at the 0.05 level) compared to all other Slovak regions. This trend may be linked to differences in income levels, availability of stores, and dietary habits. Residents of the Bratislava region generally have higher incomes, allowing them to eat out more often, which in turn is reflected in higher average spending on restaurants and hotels

compared to other regions. The largest difference in average spending on restaurants and hotels is between the Bratislava and Košice regions, amounting to 41.48€.

Table 1 Results of the Bonferroni post-hoc test for regional differences in household expenditures

Region 1	Region 2		FAB	ABT	CLT	HSG	FUR	HLT	TRA	COM	CUL	EDU	RES	MGS
BA	TT	diff	-36.56	-8.24	8.61	33.08	-22.98	-1.93	-14.87	13.43	4.65	0.01	33.44	17.53
		p-value	0.000	0.000	0.000	0.000	0.000	0.213	0.106	0.000	0.077	0.978	0.000	0.000
BA	TN	diff	-22.42	-5.88	6.07	36.99	-10.53	-5.13	-25.59	16.28	2.25	-0.98	37.07	22.14
		p-value	0.000	0.000	0.001	0.000	0.000	0.002	0.006	0.000	0.352	0.107	0.000	0.000
BA	NR	diff	-27.86	-6.82	8.08	28.70	-11.14	-1.77	-9.56	13.19	2.45	-0.26	40.19	23.83
		p-value	0.000	0.000	0.000	0.000	0.000	0.216	0.150	0.000	0.310	0.428	0.000	0.000
BA	ZA	diff	-25.82	-4.49	5.83	48.91	-6.07	-3.13	-23.82	17.58	5.09	-0.20	26.66	21.80
		p-value	0.000	0.001	0.001	0.000	0.004	0.032	0.002	0.000	0.000	0.030	0.553	0.000
BA	BB	diff	-12.08	-4.61	5.90	54.19	-8.19	-4.66	-7.13	17.26	3.19	-1.56	34.77	19.83
		p-value	0.002	0.001	0.001	0.000	0.001	0.011	0.321	0.000	0.207	0.014	0.000	0.000
BA	PO	diff	-12.98	-4.98	5.20	39.23	-11.91	-5.39	-22.44	17.02	5.13	-2.19	31.83	17.38
		p-value	0.001	0.000	0.004	0.000	0.000	0.001	0.008	0.000	0.033	0.002	0.000	0.000
BA	KE	diff	-12.31	-3.40	8.17	31.92	-5.70	-3.81	-0.98	15.19	7.26	-0.86	41.48	24.59
		p-value	0.001	0.014	0.000	0.000	0.023	0.010	0.902	0.000	0.002	0.094	0.000	0.000
TT	TN	diff	14.14	2.36	-2.54	3.90	12.45	-3.20	-10.72	2.85	-2.40	-1.00	3.63	4.60
		p-value	0.001	0.197	0.093	0.332	0.035	0.041	0.374	0.011	0.284	0.151	0.218	0.144
TT	NR	diff	8.70	1.42	-0.54	-4.39	11.84	0.16	5.31	-0.23	-2.20	-0.28	6.74	6.30
		p-value	0.045	0.434	0.754	0.342	0.041	0.899	0.563	0.848	0.340	0.442	0.014	0.031
TT	ZA	diff	10.75	3.75	-2.79	15.83	16.91	-1.20	-8.95	4.16	0.44	-0.21	-6.78	4.27
		p-value	0.015	0.024	0.053	0.000	0.003	0.362	0.392	0.000	0.839	0.560	0.037	0.151
TT	BB	diff	24.48	3.63	-2.71	21.10	14.79	-2.73	7.74	3.83	-1.46	-1.57	1.33	2.30
		p-value	0.000	0.029	0.048	0.000	0.011	0.133	0.433	0.001	0.542	0.030	0.657	0.495
TT	PO	diff	23.58	3.27	-3.41	6.15	11.07	-3.46	-7.57	3.60	0.47	-2.20	-1.61	-0.15
		p-value	0.000	0.056	0.019	0.126	0.055	0.028	0.496	0.001	0.833	0.006	0.571	0.962
TT	KE	diff	24.26	4.84	-0.44	-1.16	17.28	-1.88	13.89	1.76	2.61	-0.88	8.04	7.06
		p-value	0.000	0.003	0.760	0.750	0.002	0.168	0.186	0.107	0.230	0.131	0.002	0.034
TN	NR	diff	-5.44	-0.93	2.00	-8.29	-0.61	3.36	16.03	-3.08	0.20	0.72	3.12	1.69
		p-value	0.177	0.603	0.240	0.074	0.857	0.020	0.089	0.009	0.922	0.236	0.222	0.551
TN	ZA	diff	-3.39	1.39	-0.25	11.93	4.46	2.00	1.77	1.31	2.84	0.78	-10.41	-0.34
		p-value	0.406	0.397	0.866	0.002	0.113	0.174	0.867	0.200	0.136	0.214	0.001	0.907
TN	BB	diff	10.34	1.27	-0.18	17.20	2.34	0.47	18.46	0.98	0.94	-0.58	-2.30	-2.30
		p-value	0.011	0.443	0.901	0.000	0.444	0.805	0.068	0.357	0.663	0.509	0.409	0.481
TN	PO	diff	9.44	0.91	-0.87	2.25	-1.38	-0.27	3.15	0.75	2.88	-1.20	-5.24	-4.76
		p-value	0.015	0.592	0.554	0.584	0.655	0.875	0.779	0.477	0.150	0.198	0.048	0.129
TN	KE	diff	10.12	2.49	2.10	-5.06	4.84	1.32	24.61	-1.08	5.01	0.12	4.41	2.45
		p-value	0.008	0.130	0.150	0.178	0.129	0.381	0.021	0.307	0.010	0.874	0.069	0.446
NR	ZA	diff	2.05	2.33	-2.25	20.22	5.07	-1.36	-14.26	4.39	2.64	0.06	-13.52	-2.03
		p-value	0.613	0.157	0.170	0.000	0.088	0.265	0.064	0.000	0.185	0.833	0.000	0.448
NR	BB	diff	15.78	2.20	-2.18	25.49	2.95	-2.89	2.43	4.06	0.74	-1.30	-5.42	-4.00
		p-value	0.000	0.183	0.175	0.000	0.355	0.084	0.727	0.000	0.739	0.042	0.037	0.189
NR	PO	diff	14.88	1.84	-2.88	10.53	-0.77	-3.63	-12.88	3.83	2.67	-1.92	-8.36	-6.45
		p-value	0.000	0.276	0.082	0.023	0.810	0.013	0.127	0.001	0.197	0.007	0.001	0.027
NR	KE	diff	15.56	3.42	0.10	3.23	5.44	-2.04	8.58	2.00	4.81	-0.60	1.29	0.76
		p-value	0.000	0.037	0.953	0.452	0.096	0.106	0.275	0.080	0.017	0.231	0.565	0.800
ZA	BB	diff	13.73	-0.12	0.07	5.27	-2.12	-1.53	16.69	-0.32	-1.90	-1.36	8.11	-1.97
		p-value	0.001	0.933	0.957	0.155	0.413	0.372	0.045	0.746	0.357	0.039	0.009	0.526
ZA	PO	diff	12.84	-0.48	-0.63	-9.68	-5.84	-2.27	1.38	-0.56	0.04	-1.98	5.17	-4.42
		p-value	0.001	0.751	0.656	0.013	0.026	0.127	0.887	0.568	0.985	0.007	0.081	0.136
ZA	KE	diff	13.51	1.09	2.35	-16.99	0.37	-0.68	22.84	-2.39	2.17	-0.66	14.82	2.79
		p-value	0.000	0.459	0.092	0.000	0.892	0.595	0.012	0.016	0.240	0.202	0.000	0.363
BB	PO	diff	-0.90	-0.36	-0.70	-14.95	-3.72	-0.74	-15.31	-0.24	1.94	-0.63	-2.94	-2.46
		p-value	0.818	0.814	0.604	0.000	0.195	0.699	0.092	0.818	0.368	0.513	0.275	0.462
BB	KE	diff	-0.22	1.22	2.27	-22.26	2.49	0.85	6.15	-2.07	4.07	0.70	6.71	4.76
		p-value	0.954	0.411	0.089	0.000	0.405	0.625	0.467	0.047	0.051	0.370	0.007	0.163
PO	KE	diff	0.68	1.58	2.97	-7.31	6.21	1.59	21.46	-1.83	2.13	1.32	9.65	7.21
		p-value	0.853	0.302	0.034	0.051	0.039	0.296	0.027	0.074	0.270	0.116	0.000	0.028

Source: Own processing in Jupyter Lab (Python).

Average spending on housing, water, electricity, gas, and other fuels is significantly higher in the Bratislava region compared to all other Slovak regions, which is an expected result given the high property and rental prices in the capital. Specifically, households in Bratislava spend on average 54.19€ more than those in the Banská Bystrica region, 48.91€ more than in the Žilina region, and 39.23€ more than in the Prešov region.

The Bratislava region also shows statistically significantly higher average expenditures than other regions in the following categories: clothing and footwear; communication; miscellaneous goods and services (e.g., hairdressing, jewellery, insurance, consultancy, financial services, funeral services, etc.) This may be attributed to the higher standard of living in Bratislava, a wider selection of shopping opportunities, or a specific urban lifestyle. On the other hand, average spending on alcoholic beverages, tobacco, and narcotics, as well as furniture and household equipment, is significantly lower in the Bratislava region compared to other regions.

At the 0.05 significance level, significantly higher average expenditures on housing, water, electricity, gas, and other fuels were also observed in the Trenčín, Trnava, Nitra, Prešov, and Košice regions, compared to the Žilina and Banská Bystrica regions.

Average spending on food and non-alcoholic beverages was significantly higher in the Trnava region compared to all other regions at the 0.05 level. This spending was also significantly higher in the Trenčín, Žilina, and Nitra regions compared to Banská Bystrica, Prešov, and Košice.

Beyond the Bratislava region, average expenditures on restaurants and hotels were also significantly higher (at the 0.05 level) in the Prešov, Banská Bystrica, Žilina, and Trnava regions compared to the Nitra and Košice regions. On the contrary, significantly lower expenditures on restaurants and hotels were recorded in Trnava, Trenčín, and Banská Bystrica compared to the Žilina region.

In the Trnava region, significantly higher expenditures (at the 0.05 level) were observed on furniture, household equipment, and routine household maintenance compared to Trenčín, Bratislava, Nitra, Žilina, Banská Bystrica, and Košice. Likewise, higher average expenditures on communication were recorded in Trnava compared to Trenčín, Žilina, Banská Bystrica, and Prešov.

Significant differences at the 0.05 level also emerged in the transport category. Significantly higher average transport expenditures were found in the Trenčín, Žilina, and Prešov regions compared to Bratislava and Košice. This may be linked to high commuting rates to the capital city for work.

Overall, these results confirm the existence of regional differences in household consumption behaviour, influenced by a combination of factors such as income level, cost of living, availability of services, and cultural habits of the population.

Piekut and Knapkova (2025) used cluster analysis to analyse data from Household Budget Survey to logically group EU countries with similar consumer behaviour. Kozak et al. (2022) evaluated the consumer behaviour of Polish households at the regional level using the TOPSIS method. This method allows for the ranking of regions based on socio-economic indicators and ordering them by expenditure levels. We decided to test whether statistically significant differences in household expenditure structure exist among Slovak regions, and therefore applied the MANOVA method.

5 Conclusion

The results of the analysis confirm the existence of statistically significant regional differences in the structure of consumer expenditures of Slovak households. Despite a similar expenditure structure across regions in relative terms, multivariate analysis using PERMANOVA and post-hoc tests identified significant differences between specific regions.

The Bratislava Region stands out from the rest – not only in the total level of expenditures but also in the nature of consumption. Significantly higher spending on housing, restaurants and hotels, clothing and footwear, as well as postal and telecommunications services, point to a higher standard of living, a different lifestyle, and likely a higher level of technological advancement and service availability.

However, apart from the Bratislava Region, notable differences were also observed between other regions in certain expenditure categories. For example, in the Trnava Region, expenditures on food and non-alcoholic beverages were significantly higher than in all other regions, which may indicate a different household structure or price level in the region. Similarly, higher housing expenditures, besides Bratislava, were also observed in the Trenčín, Trnava, Nitra, Prešov, and Košice Regions compared to the Žilina and Banská Bystrica regions.

Increased expenditure on restaurants and hotels in the Prešov, Žilina, Banská Bystrica, and Trnava Regions compared to the Nitra and Košice Regions may indicate the development of tourism or a greater focus on experience-based consumption. The Trnava Region also reported significantly higher expenditures on furniture and household equipment, as well as on postal and telecommunications services.

Significant differences also appeared in the area of transport – higher expenditures were recorded in the Trenčín, Žilina, and Prešov Regions compared to Bratislava and Košice. This could be related to higher commuting rates or limited availability of public transport in these regions.

These findings highlight not only the specific characteristics of each region but also broader disparities between western and eastern Slovakia, where western regions – especially Bratislava and Trnava – show higher levels of spending and consumption in many areas. These differences reflect the economic performance of regions, availability of services, household income levels, and their consumer preferences.

These results can be useful for shaping regional policies aimed at reducing inequalities and supporting the balanced development of Slovak regions.

This paper is an output of the science project VEGA no. 1/0285/24: The Impact of Inflation on Poverty and Social Exclusion in Slovakia and the EU; and Early Stage Grant number A-25-103/3020-11: Macroeconomic forecasting using advanced machine learning.

References

1. Anderson, M. J. (2017). Permutational multivariate analysis of variance (PERMANOVA). *Wiley StatsRef: Statistics Reference Online*, 1–15. Available at: <https://doi.org/10.1002/9781118445112.stat07841>
2. Carey, G. (1998). *Multivariate analysis of variance (MANOVA): I theory*. Academic Press Boston.
3. Department of Economic and Social Affairs of the United Nations. (2023). *Classification of Individual Consumption According to Purpose 2018*. United Nations.
4. Eurostat. (2025). *Household budget survey - statistics on consumption expenditure*. Available at : https://ec.europa.eu/eurostat/statisticsexplained/index.php?title=Household_budget_survey_-_statistics_on_consumption_expenditure
5. Hebák, P., & Hustopecký, J. (1987). *Vícerozměrné statistické metody s aplikacemi: Celost. vysokošk. Příručka Pro Stud. Ekon. fakult. SNTL ; Alfa*.
6. Hupková, D., Mihálová, V., Dobošová, Ľ., & Laurová, M. (2018). Examining the dependence of expenditures on wage and poverty rates in regions of the Slovak Republic. *International Scientific Days 2018. Towards Productive, Sustainable and Resilient Global*

- Agriculture and Food Systems: Proceedings*, 959–966. Available at: <https://doi.org/10.15414/isd2018.s4.06>
7. Kasprzyk, B., & Leszczyńska, M. (2021). Changes in the incomes and expenditures of households in Poland and their regional relations. *Zeszyty Naukowe SGGW, Polityki Europejskie, Finanse i Marketing*, (25 (74)), 40–53. Available at: <https://doi.org/10.22630/pefim.2021.25.74.4>
 8. Korkmaz, S., Goksuluk, D., & Zararsiz, G. (2014). MVN: An R package for assessing multivariate normality. *The R Journal*, 6(2), 151. Available at: <https://doi.org/10.32614/rj-2014-031>
 9. Kozak, J. M., Mrówczyńska-Kamińska, A., & Wołoszyn, A. (2022). Multidimensional assessment of the financial position of Polish households and its regional diversity. *Heliyon*, 8(5). Available at: <https://doi.org/10.1016/j.heliyon.2022.e09483>
 10. Lazíková, Z. (2017). Development of incomes and expenditures in Slovak households. *Acta Regionalia et Environmentalica*, 14(2), 37–44. Available at: <https://doi.org/10.1515/aree-2017-0006>
 11. Pacáková, V. et al. (2012). *Štatistická indukcia pre ekonómov*. Vydavateľstvo EKONÓM.
 12. Piekut, M., & Knapkova, M. (2025). Patterns and convergence in household spending: Insights from western and Eastern Europe. *Amfiteatru Economic*, 27(68), 180–195. Available at: <https://doi.org/10.24818/ea/2025/68/180>
 13. Piekut, M., & Piekut, K. (2022). Changes in patterns of consumer spending in European households. *Sustainability*, 14(19), 12794. Available at: <https://doi.org/10.3390/su141912794>
 14. Šoltés, E. (2019). *Regresná a korelačná analýza s aplikáciami v softvéri SAS*. Letra Edu.

Určenie rizikovej skupiny investora fuzzy logikou pri využití európskej regulácie UCITS

Determining investor's risk group using fuzzy logic under the UCITS

Richard Martinus¹

Abstrakt

Existuje mnoho prístupov, ktoré skúmajú členenie rizika investora. Najčastejšie používaným prístupom správcovských spoločností a brokerov je investičný dotazník, ktorý určuje rizikovú skupinu, do ktorej investor patrí. Častým problémom tohto prístupu je, že potenciálny investor je plne zaradený do jednej kategórie, aj keď môže byť blízko hranice s druhou kategóriou. Tento nedostatok je možné zlepšiť pomocou využitia európskej smernice o podnikoch kolektívneho investovania do prevoditeľných cenných papierov (z angl. UCITS), ktorá stanovuje 7 rizikových skupín podľa syntetického ukazovateľa rizika a strát SRRI. Využitím fuzzy logiky, určujeme s akou príslušnosťou investor spĺňa každú jednu otázku. Je tak možné presnejšie definovať profil investora cez SRRI vyššou granularitou rizikových skupín. Na analýzu používame programovací jazyk Python v prostredí Jupyter Notebook. Tento článok má za cieľ priniesť presnejšiu klasifikáciu investorov do rizikových skupín, nakoľko tvorba investičného dotazníka fuzzy logikou s využitím európskej smernice ešte doposiaľ nebola implementovaná.

Kľúčové slová

UCITS, Fuzzy, Manažment rizika, SRRI

Abstract

There are many approaches that examine the classification of investor risk. The most used approach offered by asset management companies and brokers is an investment questionnaire, which determines the risk group to which the investor belongs. A common problem with this approach is that a potential investor is fully classified into one category, even though he may be close to the border with the second category. This deficiency can be improved by using the European Collective Investment in Transferable Securities directive (UCITS), which sets out 7 risk groups according to the synthetic risk and loss indicator SRRI. Using fuzzy logic, we determine with what affiliation the investor meets each question. This makes it possible to more accurately define the investor profile through SRRI with a higher granularity of risk groups. The Python programming language in the Jupyter Notebook environment is used for the analysis. This article aims to bring a more accurate classification of investors into risk groups, since the creation of an investment questionnaire with fuzzy logic using the European directive has not yet been implemented.

Key words

UCITS, Fuzzy, Risk management, SRRI

JEL classification

C44, G11, G18

¹ Bratislava University of Economics and Business, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemska cesta 1, 852 35 Bratislava, richard.misek@euba.sk.

1 Introduction

Allocating an investor's assets and assigning them to an adequate risk group is a difficult task in modern investing. Investors often demand the highest returns, the risk of which they are unable to manage. It is the failure to manage risk (market fluctuations) that leads to the sale of assets in a period of unfavorable development of financial markets and subsequent high losses. In order to set up a suitable investment product for a potential investor, it is necessary to classify investor according to the risk profile and assign investor to a risk group, as defined in the Securities Act 566/2001 Coll (Národná rada Slovenskej republiky, 2001). Investment questionnaires in banks, asset management companies and other asset providers in Slovakia are carried out based on so-called exclusionary conditions (hereinafter K.O. conditions), which integrate the investor into one of the three risk groups. If the questions in the questionnaire are answered in favor of high-risk aversion, the investor is rightly limited in his choice and is subject to a restriction on the allocation of his assets to more risky funds. The disadvantage of questionnaires using this approach is that they assign investors to only 3 risk groups, which is often not enough. As mentioned, investors are sorted into group of dynamic investors, balanced investors and conservative investors. The first group to represent investors with the greatest risk aversion is the group of conservative investors. These investors are allowed to buy bonds and bond funds due to their low risk. Bonds provide a fixed income in the form of interest rates, which can help protect investors from inflation. In the event of bankruptcy, bondholders are paid before shareholders (Johnson, 2018).

For investors who are willing to take on a greater degree of risk, there is a set for balanced investors, which represents a mixed approach, consisting of bond, real estate, and equity parts. The last set stands for dynamic investors, these investors have a high level of risk acceptance and can thus achieve the highest returns. This set allows investors to create a portfolio from all available assets and the allocation is usually fully focused on stocks. Companies and financial advisors use terms such as conservative, balanced and dynamic investors to describe investment strategies and portfolios that correspond to clients' different levels of risk tolerance and investment objectives. These terms are not defined in the law but are commonly accepted in the investment community and their meaning is derived from related EU legislation and rules, such as MiFID - Markets in Financial Instruments Directive (European Parliament and Council, 2004). They often appear in investment literature and educational materials (Omololu, E, 2023).

The current deficiency of investment questionnaires results is assigning different investors to the same set, while they may be investors who have significantly different levels of risk acceptance or risk aversion. It is possible to have an investor that is on the border of two groups and therefore his portfolio composition should overlap between these groups. However, since the currently used questionnaires strictly categorize investors only into 3 groups, the investor that falls in between will end up with a sub-optimal recommendation of portfolio composition since the range of risk aversion in specific group is too big. An option is to create more risk classes as defines UCITS, which stands for Undertakings for collective investments in transferable securities (European Parliament and Council, 2014). To solve this problem, this article applies fuzzy logic and fuzzy sets to assign the exact allocation of assets, specifically adjusted to the investor's responses. The next problem is uncertainty in answers, when respondents are not sure which answer to choose, so they can pick random answers from provided answers which influence the results. This problem will be solved with more specific answer options represented with fill-in answer options, instead of multiple predefined options.

2 Methodology

This chapter examines the current state of questionnaires and analyzes the assignment of investors to risk groups. At the same time, we will describe why the current setup is insufficient. The second subchapter describes the fuzzy set approach that will be used to refine the allocation to be specific and fitting to each investor. Final subchapter takes into consideration UCITS regulation for European Union funds and used formulas for computing each risk class.

2.1 Usage of the Investment Questionnaires

The first chapter discusses the reason for using the investment questionnaire. This questionnaire is a necessity for starting the investment and its creation must comply with the Securities Act 566/2001 Coll. From the results of the questionnaires, it is possible to purchase only those underlying assets whose risk corresponds to the risk profile of the investor. By buying riskier assets beyond their level of risk acceptance, the investor exposes themselves to the danger of not managing emotions and fluctuations in the markets, due to which the portfolio may be sold at a loss (Financial Conduct Authority, 2021).

The current solution for these questionnaires is to use classical sets, where restrictive K.O conditions determine the result of the questionnaire and each of the investors can belong to only one set. In general, if an investor is unable to take risks when investing, he belongs to a conservative investor. If an investor is willing to take the highest risk, he is classified as a dynamic investor. If the investment questionnaire consisted of only one question in the wording: indicate your risk aversion when investing, it would be possible to distinguish between the two investors and assign them to appropriate groups. Let us denote by the letter X a universal set of elements (in our case all potential investors) and the set C (in our case a subgroup of investors) as a subset of the set X . Based on set theory, one can say whether each element x of the universal set X belongs to the set C . The belonging of the element x to the set C , which is a subset of the universe X , (which is a basic mathematic concept developed in (Cantor, 1874) is expressed by a characteristic function φ_K as follows:

$$\begin{aligned}\varphi_K &: X \rightarrow \{0, 1\}, \\ \varphi_K(x) &= 1, \text{ if } x \in C, \\ \varphi_K(x) &= 0, \text{ if } x \notin C\end{aligned}\tag{1}$$

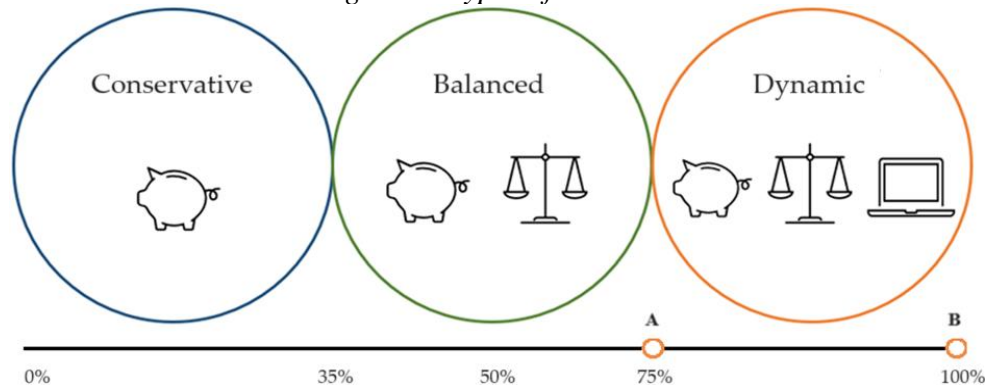
In this case, set C could represent a group of conservative investors with maximum risk aversion, and element x is the representative of the investor who filled out the investment questionnaire. Based on the answers, investors are assigned a percentage of risk, based on which it is possible to include the investor in a risk group. In general, percentages in a portfolio can be used to determine an investor's risk profile. A conservative investor might have a higher share of conservative investments, such as bonds or money markets, while a dynamic investor might have a higher share of stocks or other riskier assets. These ratios can be adapted to the individual goals and needs of the investor (Nukala, V.B., Prasada Rao, S.S, 2021). When identifying, for instance, conservative investors, the characteristic function could be determined as follows:

$$\begin{aligned}\varphi_C(x) &= 1, \text{ if } x \leq 35\%, \\ \varphi_C(x) &= 0, \text{ if } x > 36\%\end{aligned}\tag{2}$$

Investment questionnaires set out three sets to which investors can be assigned. The boundaries of each set are shown in Figure 1, where:

- Conservative investor achieves a result from 0% to no more than 35%
- A balanced investor achieves a result of at least 36% to no more than 74%
- A dynamic investor achieves a result of at least 75% to 100%

Figure 1: Types of investors



Source: Own processing (2025)

Based on the questionnaire, individual investors are assigned to classes and therefore can buy securities, the composition of which varies depending on their risk group. Investors in the higher risk group have the opportunity to buy products from their own risk group and as well the products available to the lower risk groups. If the product were only available to the riskiest group, conservative investors would not have access to it. At first glance, these three groups may seem sufficient. However, there is a difference that would be best shown by an example of two investors. Let both investors be in the set of dynamic investors, but investor *A* has a significantly lower risk acceptance (75%) - on the border with a balanced investor. Investor *B* is fully dynamic and has a maximum risk acceptance rate (100%). According to the investment questionnaire, both would be offered the same products in identical proportions as they both belong to the dynamic investor group. This article tends to disagree with this approach, as each of the investors is specific and their portfolios should be more personalized, based on their risk profile. The inaccuracy of this approach can be easily read from Figure 1. For this reason, this article will explore the application of fuzzy logic in order to improve the categorization of investors based on the results of their investment questionnaire.

2.2 Fuzzy logic

Fuzzy logic was introduced as an extension of classical sets (Zadeh, 1965). The difference from the classical sets lies in determining belonging to the set on the interval between 0 and 1. In the example above, the allocation of investors in a given set using classical sets could not be clearly distinguished, resulting in different investors being recommended the same portfolio composition. Using intensities (affiliations) of belonging to a set, it is possible to determine the exact values of affiliations of belonging and thus better optimize the asset allocation for a given investor. If fuzzy logic was used, the investor could have specific portfolio type with higher granularity and so it would be much more precise than using only three risk groups. We can distinguish between conservative and ultra conservative investors and their risk aversion.

A high variety of fuzzy sets exist. In this work, we applied: R-fuzzy, L-fuzzy, triangular, trapezoidal, and singleton. Each of the sets has a specific and suitable applicability for calculating affiliations.

R fuzzy is defined with two parameters. Parameter a represents, where support of fuzzy set starts and parameter b represents where core of fuzzy set starts (3).

$$\mu_R(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b \end{cases} \quad (3)$$

L fuzzy is defined with two parameters. Parameter a represents, where core of fuzzy set ends and parameter b represents where support of fuzzy set ends (4).

$$\mu_L(x) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases} \quad (4)$$

Singleton (S) fuzzy is defined with one parameter. Parameter m represents an element, from all elements that assumes the value of one – the core of fuzzy set is defined in only one point (5).

$$\mu_A(x) = \begin{cases} 0, & x \neq m \\ 1, & x = m \end{cases} \quad (5)$$

Trapezoidal (Trap) fuzzy is defined with four parameters. Parameter a represents start of support, d represents end of support, b represents start of the core, c represents end of the core.

$$\mu_A(x) = \begin{cases} 0, & x \leq a \vee x \geq d \\ \frac{x-a}{b-a}, & x \in (a, b) \\ 1, & x \in (b, c) \\ \frac{d-x}{d-c}, & x \in (c, d) \end{cases} \quad (6)$$

More about fuzzy sets theory is in Fuzzy set theory and its applications can be found in multiple articles such as (Zimmermann, 2001).

2.3 UCITS

UCITS represent a European Union regulatory framework governing collective investment funds. It was introduced to ensure a level of investor protection and to ensure the free distribution of these funds within the EU. UCITS funds are popular with both retail and professional investors, who must comply with strict rules on diversification, transparency and risk management.

UCITS regulations require a key investor information document (KIID), which contains important information about the fund, including the investment strategy and risk profile. It is in this document that the Synthetic Risk and Reward Indicator is found. This indicator is mostly

used with funds that are offered to retail investors. The SRRI was introduced, to simplify the information for investors in matter of potential riskiness and return of the specific fund.

SRRI as an indicator was introduced as part of EU legislation on the key documents that must be shared with investors, and they contain important information about the fund. This indicator is a mandatory part of the KIID for all Undertakings for Collective Investment in Transferable Securities (UCITS) funds. As shown, the intention of SRRI is to provide simple, visually accessible information, which makes it easier for investors to compare different investment products. Another benefit of SRRI is its transparency of the fund score, which is widely used with brokerage companies which intention is to provide funds for their clients, which matches clients' risk profile. The higher the score of SRRI is, the higher risk and return can investors expect. SRRI is expressed through a single number that ranges from 1 to 7. This level reflects different levels of risk, with a lower number meaning lower risk and lower return, while a higher number means higher returns. The indicator is calculated based on the historical volatility of the fund, which is a measure of the fluctuations in the value of investments in the past. Volatility is used as the main indicator of risk, which is measured with standard deviation for last 5 years. It reflects how significantly an investment can change over time.

Score from 1 to 2 (marginally 3) belongs to the funds with lower risk. Typically, they invest in low-risk assets like government bonds, term deposits, short-term bonds, or other cash equivalents. Such funds are often referred to as conservative or suitable for conservative investors.

The score from 3 to 5 belongs to the funds with higher risk. These funds are likely to have structure based on combination of bonds, stocks and other financial instruments. Bond do not have to be just issued by government. Higher risk bonds with higher yields are corporate bonds, which might be contained in these compositions. These funds are suitable for balanced investors, which cannot manage high risk funds.

Score from (marginally 5) 6 to 7 are two of the highest possible scores from SRRI range. High risk funds mostly allocate their financial means into stocks or other riskier assets. These funds have the tendency for the long period of time to have higher returns than funds with conservative assets like government bonds or funds that generate income from real estate's rents.

The volatility on which the SRRI is calculated is measured using the historical standard deviation of the fund's returns. Standard deviation measures how much a fund's returns deviate from the average return over a period. A period of 5 years of historical data is usually used to calculate the SRRI, if available. General methodology is shown in Methodology for the calculation of the synthetic risk and reward indicator (Committee of European Securities Regulators, 2009). According to the Committee the volatility of the fund shall be computed, and then rescaled to a yearly basis, using the following standard method:

$$volatility = \sigma_f = \sqrt{\frac{m}{T-1} \sum_{t=1}^T (r_{f,t} - r'_f)^2} \quad (7)$$

Where the returns of the fund ($r_{f,t}$) are measured over T non overlapping periods of duration of $1/m$. This means $m=52$ and $T = 260$ for weekly returns, and $m=12$ and $T=60$ for monthly returns; and where r'_f is arithmetic mean of the returns of the fund over the T periods:

(8)

$$r'_f = \frac{1}{T} \sum_{t=1}^T r_{f,t}$$

Committee states that the synthetic risk and reward indicator will correspond to an integer number designed to rank the fund over a scale from 1 to 7, according to its increasing level of volatility. Each score from 1 to 7 represents the final value behind the volatility intervals:

Figure 2: Types of investors

Risk Class	Volatility Intervals	
	equal or above	less than
1	0%	0.5%
2	0.5%	2%
3	2%	5%
4	5%	10%
5	10%	15%
6	15%	25%
7	25%	

Source: own processing (2025)

3 Results

We can divide questions in an investment questionnaire into three groups. First group A represents questions focused on investor's experience and knowledge with securities. The second group represents the financial situation of the investor. The last third group focuses on the risk and time horizons of investors.

Here are the questions asked in the questionnaire, where all answers have been filled by option "fill in blank" and not with using predefined options:

Q1A-Right: How many years of experience do you have in investing?

- Where a stands for zero experience, b stands for 3-year experience (3).

Q2A-Trapezoidal: What is the percentual annual return of S&P500 index in the last 10 years?

- Where a stands for 7% return, b stands for 8% return, c stands for 12% return, d stands for 13% return (6).

Q3A-Singleton: What is my portfolio value when I invested 1 000 units, the first year dropped by 10% and second year rose 10%?

- Where m stands for 990 units (5).

Q1B-Right: What is your average monthly net income?

- Where a stands for 615 representing minimum net wage for Slovak citizen, b stands 1130 representing average net wage of Slovak citizen (3).

Q2B-Right: What amount can you save monthly?

- Where a stands for 0 units, b stands for 10% of net wage from Q1B (3).

Q3B-Right: What is your total amount of savings reserve?

- Where a stands for 0 units, b stands for value of 6 times of net wage from Q1B (3).

Q4B-Left: How much do your loans cost monthly?

- Where a stands for value of 36% of net wage, b stands for value of 43% of net wage from Q1B (4).

Q1C-Right: How many years do you plan to invest?

- Where a stands for 3 years, b stands for 5 years (3).

Q2C-Left: If the value of your portfolio dropped 20% in a year, how many percent of total portfolio would you sell.

- Where a stands for 0%, b stands for 5% of portfolio value (4).

Q3C-Left: What is your age?

- Where a stands for 54 years, b stands for 64 years of age (3).

As the input we provide randomly generated data that simulates real answers. This input data from the investment questionnaire is shown in figure 3 below, where we represent 481 records:

Figure 3: Investment questionnaire records

	A	B	C	D	E	F	G	H	I	J
1	Q1A-R-exp	Q2A-TRAP-sp500	Q3A-S-drop	Q1B-R-income	Q2B-R-save	Q3B-R-savings	Q4B-L-loan	Q1C-R-horizon	Q2C-L-drop	Q3C-L-age
2	6	7	957	2243	236	10467	40	16	0	49
3	16	10	1058	2022	509	1348	45	12	13	37
4	16	14	1088	2076	247	14532	31	3	9	68
5	9	14	990	1503	520	10020	22	26	0	24
6	12	12	961	2235	62	15645	32	15	10	25
7	18	6	990	1302	279	13888	21	2	0	50
8	6	4	999	1160	343	18947	0	18	0	22
9	15	7	1075	844	59	7877	34	20	14	18
10	6	10	1092	1250	108	12500	0	17	11	18
11	3	4	1029	1814	217	22373	0	5	0	37
12	8	8	990	1171	276	14822	20	10	0	44

Source: own processing (2025)

After obtaining answers we use the fuzzy logic which sets the affiliation to each fuzzy set. Used formulas in Python are represented at figure 4, which is equal to formulas defined in subchapter 2.2. In the whole process we used the programming language Python.

Figure 4: Fuzzy sets

```

def L_fuzzy(a,x,z):
    if (x<=a):
        return 1
    elif (a<x and x<z):
        return (z-x)/(z-a)
    elif (x>=z):
        return 0

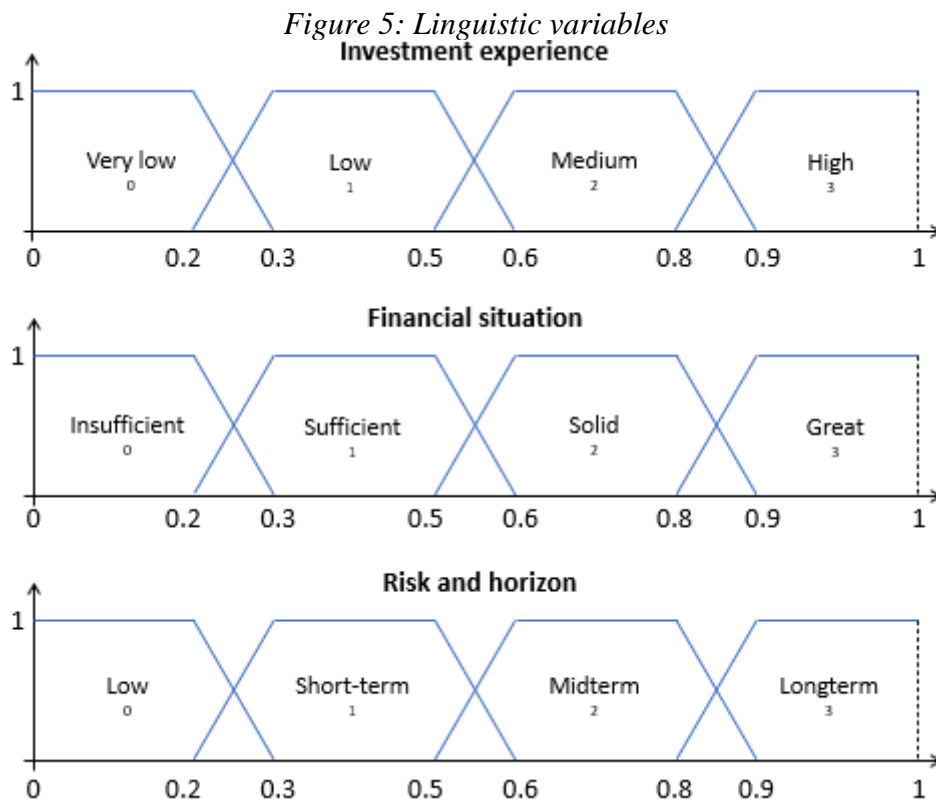
def R_fuzzy(a,x,z):
    if(x<=a):
        return 0
    elif(a<x and x<z):
        return (x-a)/(z-a)
    elif(x>=z):
        return 1

def S_fuzzy(x,m):
    if(x!=m):
        return 0
    elif(x==m):
        return 1

def Trap_fuzzy(a,l,x,r,z):
    if(x<=a or x>=z):
        return 0;
    elif(x>a and x<l):
        return R_fuzzy(a,x,l)
    elif(x>=l and x<=r):
        return 1
    elif(x>r and x<z):
        return L_fuzzy(r,x,z)
    
```

Source: own processing (2025)

After the computation is done, we proceed to averaging the responses of individual groups to new three groups QA, QB, QC. These values we substitute with the use of linguistic variables as shown at Figure 4, which are used in our built matrix which determines the final SRI score.



The matrix is filled with 64 possible options and 7 outputs. The matrix used does not have a specific general notation. Since this is a new approach, we have proposed the notation in the following figure 6.

Figure 6: Defined SRRI matrix

INPUT	SRRI	INPUT	SRRI	INPUT	SRRI	INPUT	SRRI
0,0,0	1	1,0,0	1	2,0,0	2	3,0,0	3
0,0,1	1	1,0,1	2	2,0,1	2	3,0,1	3
0,0,2	2	1,0,2	2	2,0,2	3	3,0,2	4
0,0,3	3	1,0,3	3	2,0,3	4	3,0,3	4
0,1,0	1	1,1,0	2	2,1,0	2	3,1,0	3
0,1,1	2	1,1,1	3	2,1,1	3	3,1,1	4
0,1,2	2	1,1,2	3	2,1,2	4	3,1,2	5
0,1,3	3	1,1,3	4	2,1,3	5	3,1,3	6
0,2,0	2	1,2,0	2	2,2,0	3	3,2,0	4
0,2,1	2	1,2,1	3	2,2,1	4	3,2,1	5
0,2,2	3	1,2,2	4	2,2,2	5	3,2,2	6
0,2,3	4	1,2,3	5	2,2,3	6	3,2,3	7
0,3,0	3	1,3,0	3	2,3,0	4	3,3,0	4
0,3,1	3	1,3,1	4	2,3,1	5	3,3,1	6
0,3,2	4	1,3,2	5	2,3,2	6	3,3,2	7
0,3,3	4	1,3,3	6	2,3,3	7	3,3,3	7

Source: own processing (2025)

The last step was to show the SRRI score from 1 to 7 range and compare it with three risk groups used until now. This is shown at figure 7.

Figure 7: Output of investment questionnaire

	Q1A-R	Q2A-TRAP	Q3A-S	Q1B-R	Q2B-R	Q3B-R	Q4B-R	Q1C-R	Q2C-L	Q3C-L	QA	QB	QC	JP-A	JP-B	JP-C	SRRI	SRRI_profile
0	1	0	0	1	0.03	1	0	1	0	1	0.33	0.51	0.67	Low	Sufficient	Midterm	3	Balanced
1	1	1	0	1	0.05	1	0	1	0	1	0.67	0.51	0.67	Medium	Sufficient	Midterm	4	Balanced
2	1	0	0	1	0.07	1	0	1	0	1	0.33	0.52	0.67	Low	Sufficient	Midterm	3	Balanced
393	1	0	0	1	0.1	1	0	1	0	1	0.33	0.52	0.67	Low	Sufficient	Midterm	3	Balanced
394	1	1	1	1	0.04	1	0	1	1	1	1.0	0.51	1.0	High	Sufficient	Longterm	6	Dynamic
395	1	1	0	0.33	0.13	1	0	1	0	1	0.67	0.36	0.67	Medium	Sufficient	Midterm	4	Balanced
396	0.67	1	1	1	0.06	1	0	1	0	1	0.89	0.52	0.67	High	Sufficient	Midterm	5	Balanced
397	1	1	1	1	0.04	1	0	1	0	1	1.0	0.51	0.67	High	Sufficient	Midterm	5	Balanced
398	0.33	0	0	0	0.07	0	0	1	0	1	0.11	0.02	0.67	Very low	Insufficient	Midterm	2	Conservative
399	1	0	1	1	0.02	1	0	1	0	1	0.67	0.5	0.67	Medium	Sufficient	Midterm	4	Balanced
400	1	0	0	0	0.22	0	0	1	0	1	0.33	0.06	0.67	Low	Insufficient	Midterm	2	Conservative
401	1	0	1	0.71	0.06	1	0	1	1	1	0.67	0.44	1.0	Medium	Sufficient	Longterm	5	Balanced

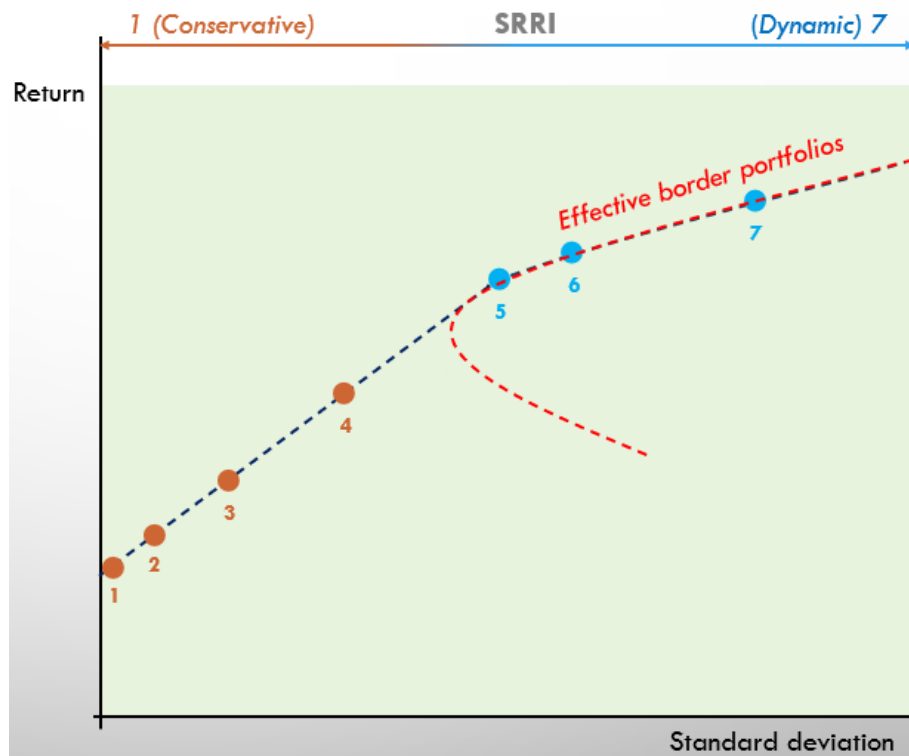
Source: own processing (2025)

With this approach, we can precisely define how risky a fund is ideal for an investor and more precisely determine the risk profile using SRRI.

4 Further research

There are many ways to build a portfolio based on set value of risk. If we were to try to adhere to the European regulation for collective investment UCITS, we should continue in our research to use the standard deviation as a representative of risk while building portfolios. In this case, it is possible to use the theory of modern portfolio, where it is possible to create an effective border of portfolios with standard deviation on horizontal axis and estimated return on vertical axis. We can subsequently divide these portfolios, which would be made up of equity securities, into SRRI risk groups according to the interval of the lower and upper volatility limits. This allows us to build groups of effective portfolios, which can then be optimized and the best portfolio for each group can be selected. Portfolios with lower SRRI classes must be combined with bonds or other low-risk assets. Here, it is possible to start, for example, from the possibility of using the capital market line and allocating the bond component, which represents a risk-free asset, in aliquots according to the degree of risk, while the capital market line will pass through the risk-free asset and the optimal portfolio, made up of purely equity assets. This approach shown at figure below is one of many that we are considering in the future as a complement to this work.

Figure 8: Effective border portfolios



Source: own processing (2025)

5 Conclusion

Risk management, portfolio selection and weight allocation is a challenging discipline and there is no clear answer to how to build an optimal portfolio that will perform best from all available options in future. A certain level of optimality can be reached by building a portfolio that is more individualized for the investor. The original format of the investment questionnaire assigned investors to one from three categories only. Here existed edge cases and not specifically defined sets of who and when is going to be assigned in specific risk set. An alternative proposed in this article is the applicability of fuzzy sets, where affiliations to individual sets based on answers from questionnaires can be assigned more accurately under the regulation UCITS. All restrictive conditions from answers in questionnaire are considered, using fuzzy sets and linguistic variables. We build the SRRI matrix with 64 specific options, that leads to 7 risk groups. The offered approach is more flexible, and the greatest advantage is the dynamism of questions, which are self-adaptable on each investor. The extension of this contribution could in the future involve applying a new approach that replaced the use of SRRI in 2014 with the current SRI approach found in Packaged Retail Investment and Insurance Products. The SRI approach is based on calculating a score from 1 to 7, taking into account market risk measure (MRM) and credit risk measure (CRM). In this approach, the calculation of volatility is applied using Value at Risk. The resulting portfolios would therefore not only depend on returns and standard deviation but also on the expanded model applied in the SRI methodology. We believe our new approach for evaluation of investors' risk is more accurate and transparent.

Funding: This article was supported by the Grant Agency of Slovak Republic: VEGA grant no. 1/0120/23 "Environmental models as a tool for ecological and economic decisions making".

Literature

1. Cantor, G. (1874). On a Property of the Collection of All Real Algebraic Numbers.
2. Committee of European Securities Regulators. (2009). *Methodology for the calculation of the synthetic risk and reward indicator* (CESR/09-1026). European Securities and Markets Authority.
https://www.esma.europa.eu/sites/default/files/library/2015/11/09_1026_final_kid_srri_methodology_for_publication.pdf
3. European Parliament and Council. (2004). *Directive 2004/39/EC of the European Parliament and of the Council of 21 April 2004 on markets in financial instruments*. Official Journal of the European Union, L 145, 1–44. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32004L0039>
4. European Parliament and Council. (2014). *Directive 2014/91/EU of the European Parliament and of the Council of 23 July 2014 amending Directive 2009/65/EC on the coordination of laws, regulations and administrative provisions relating to undertakings for collective investment in transferable securities (UCITS)*. Official Journal of the European Union, L 257, 186–213. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32014L0091>
5. Financial Conduct Authority. (2021). *5 questions to ask before you invest*. Retrieved November 27, 2024 from <https://www.fca.org.uk/investsmart/5-questions-ask-you-invest/>
6. Johnson, R. (2018). *Why Bonds Are Safer Than Stock*. Retrieved November 23, 2024, from <https://pocketsense.com/bonds-safer-stock-5558.html>
7. Národná rada Slovenskej republiky. (2001). *Zákon č. 566/2001 Z. z. o cenných papieroch a investičných službách*. Zbierka zákonov Slovenskej republiky.
8. Nukala, V.B., Prasada Rao, S.S. (2021) *Role of debt-to-equity ratio in project investment valuation, assessing risk and return in capital markets*. *Futur Bus J* 7, 13.
9. Omololu, E. (2023). *Different Types of Investment Portfolios*. *Savvy New Canadians*. Retrieved January 14, 2024, from <https://www.savvynewcanadians.com/different-types-investment-portfolios/>
10. Zadeh, L. (1965). *Fuzzy sets. Information and Control*.
11. Zimmermann, H. J. (2001). *Fuzzy set theory—and its applications* (4th ed.). Springer.
<https://doi.org/10.1007/978-94-010-0646-0>

Metódy strojového učenia aplikované pre investičné stratégie

Machine learning methods applied for investment strategies

Richard Martinus¹

Abstrakt

Strojové učenie sa čoraz častejšie stáva kľúčovým a preferovaným nástrojom pri tvorení a optimalizácii investičných portfólií. Jeho využitie spočíva v presnejších predikciách, adaptívnom rozhodovaní a vlastnosti naučiť sa podľa vzorov ako reagovať na trhové udalosti. V oblasti financií je strojové učenie uplatňované pomocou regresných modelov, neurónových sietí alebo aj učením formou odmeňovania, kde cieľom je predpovedať cenové pohyby aktív. Využitím neurónových sietí je možná identifikácia komplexných vzorov v historických dátach, zatiaľ čo učenie na základe odmeňovania optimalizuje stratégie obchodovania podľa spätnej väzby. V článku poukazujeme aj na výhody a výzvy spojené s používaním strojového učenia v investovaní ako je interpretovateľnosť modelov alebo kvalita dát.

Kľúčové slová

Strojové učenie, portfólio, investičné stratégie

Abstract

Machine learning is increasingly becoming a key and preferred tool in the creation and optimization of investment portfolios. Its use lies in more accurate predictions, adaptive decision-making and the ability to learn from patterns how to respond to market events. In the field of finance, machine learning is applied using regression models, neural networks or even reinforcement learning, where the goal is to predict asset price movements. Using neural networks, it is possible to identify complex patterns in historical data, while reinforcement learning optimizes trading strategies based on feedback. We also discuss the advantages and challenges associated with the use of machine learning in investing, such as model interpretability or data quality.

Key words

Machine learning, portfolio, investing strategies

JEL classification

C45, C53, G11, G17

¹ Bratislava University of Economics and Business, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemska cesta 1, 852 35 Bratislava, richard.misek@euba.sk.

1 Introduction

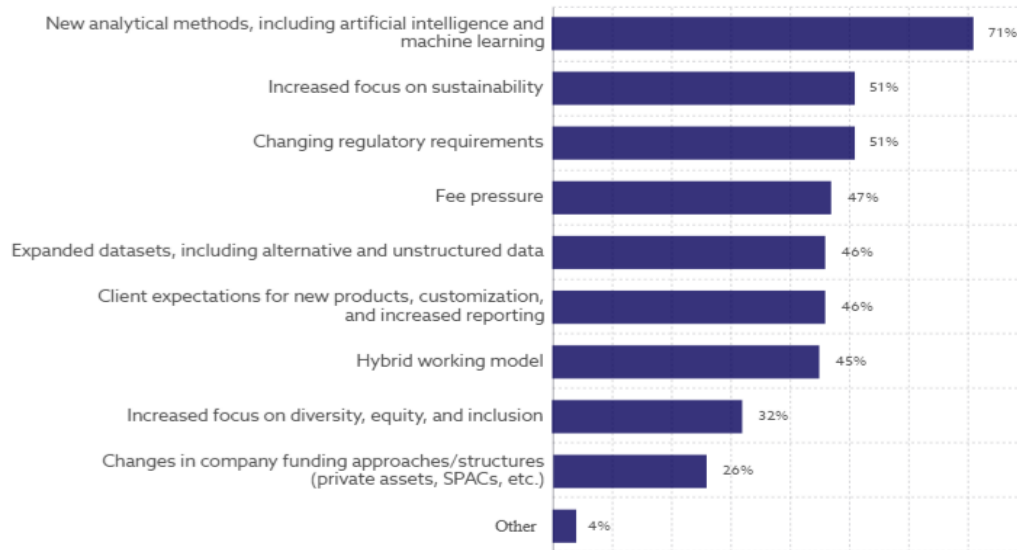
Learning is the foundation of every human being from birth, starting with learning to eat, taking the first steps, and continuing through school. Learning can be used not only in humans, but also in technology. In the world of information technology, machine learning (ML) can be defined as the science of computers' ability to learn without being explicitly programmed. Machine learning is a subset of artificial intelligence (AI) and is an impressive way to approach solving complex problems. This revolutionary approach to analyzing, processing, and interpreting data allows computer systems to learn and subsequently improve in the area they are used for. Improvement in machine learning is the ability to improve the results of an algorithm with minimal human intervention. Unlike traditional programming, where a programmer manually defines rules, machine learning allows a system to adapt to new situations based on existing data. There are many algorithms and different machine learning techniques, the most common of which are supervised learning, unsupervised learning, and reward-based learning. In supervised learning, a model is trained on labeled data, where the outputs are labeled. The goal here is to teach the model to predict outputs based on new inputs. Unsupervised learning, as the name suggests, is based on learning from unlabeled input data. The goal is to discover patterns and groups or structures in the data that are related to each other. Reinforcement learning is based on learning from interaction with the environment. The goal is to maximize reward or minimize loss, and this type can be imagined by awarding points for good decisions and deducting points for bad decisions. Various techniques can also be used in predicting financial markets, which is generally a difficult, if not impossible, discipline. It is never possible to accurately predict events such as inflation, natural disasters, pandemics or other macroeconomic events that can change the market trend. Analyses can only approximate the future market direction and the analyses created should only form guidelines for predicting asset prices. We chose the topic of machine learning because it is an increasingly attractive approach to solving complex problems not only in finance and allows for continuous improvement of the algorithm. This article is focused on a deeper understanding of machine learning, its methods and techniques in the field of finance, where we will illustrate the application of machine learning in predicting the prices of financial assets. The goal is to examine how machine learning can determine the future value of an asset or at least approach the value that may occur in the future.

2 Literature review

Machine learning and artificial intelligence are rapidly becoming mainstream approaches to solving various investment strategies. Its history dates back to the 1950s, where the most famous article that is considered the foundation of machine learning is “Computing Machinery and Intelligence” (Turing, 1950). Turing was concerned with the question of whether machines can think and introduced the Turing test, which evaluates a machine’s intelligence according to its ability to communicate like a human. Here he outlined the idea that machines could learn from data and experience, just like humans. The first program that learned from experience was created for the game of checkers, where learning was used in the form of rewards and heuristic search. The program improved here, so that it played against itself (Samuel, 1959). Subsequently, new approaches and contributions were developed and discovered that addressed the topic of machine learning. When it comes to investing, there are several approaches that can be used, such as active trading, where the investor buys and sells assets over a short period of time (days or weeks), but it can also be a long-term investment strategy. In this chapter, we will present several articles studied abroad, in which we will describe the solution procedure that the authors chose to solve this problem. One of the first articles focused on investing using machine learning is a article from the 1980s that applied neural networks to price prediction

and was constructed when predicting IBM stocks. The author focused on the solution using MLP multilayer perceptrons, which he used to predict stock prices (White, 1988). Among the first articles is the valuation and hedging of derivative securities using machine learning methods. The article focused on non-parametric methods such as neural networks. The authors' approach enabled a more accurate approach to modeling nonlinear patterns in markets, while a new tool was created for optimizing hedging strategies (Lo & Poggio, 1994). The article "Application of machine learning in algorithmic investment strategies on global stock markets" focuses on the use of machine learning in algorithmic investment strategies. Several models are analyzed here, such as neural networks, naive Bayesian models or regression trees for generating trading signals on the Warsaw WIG20 index, the German DAX index and the American S&P500 index and selected Central and Eastern European indices in the period from January 2002 to the end of March 2023. Individual strategies and the benchmark buy-and-hold strategy were compared with each other, where risk and return were compared. The results showed that machine learning models outperformed passive strategies, with the best performing models being the polynomial support vector machine for the WIG20 and S&P500 indices. The best performing model for the DAX index was the linear support vector machine (Grudniewicz & Ślepaczuk, 2023). Another study presents financial market indicators that can be used for investment strategies when investing in the global stock market. The authors propose to create a design of unmanaged and managed global stock market volatility using pairwise correlation and vector autoregressive models. The effectiveness of these indicators was analyzed using logistic regression, SVM, and the random forest algorithm. Strategies for global stock market prediction and regional allocation for emerging markets were proposed. The results showed that these indicators are useful in times of crisis. The study is one of the first attempts to apply financial indicators in investment strategies (Lee & Cho & Kwon & Sohn, 2018). Focusing on the development of a sustainable quantitative investment model is also possible using machine learning. The model that the authors investigated includes stock selection and algorithmic trading, using principal component analysis and EVA (Economic Value Added) indicators for stock selection. Testing the model on the US stock market showed that the LSTM networks can predict stock prices more accurately from the used techniques. The strategy generated higher returns than the market in all market conditions. (Li & Wang & Ahmad & Huang & Khan, 2023). We will also practically apply the LSTM technique in this article. Studies are often focused on long-term investment strategies within Value Investing. This approach is based on estimating the intrinsic value of an individual company and investing in undervalued stocks, which represents a safety margin. The goal of this study was to automate stock selection by combining the principles of Value Investing and historical financial data. The classification models used classify stocks based on investment activity. The model's success rate reached more than 80%, identifying stocks with a potential for 15% annualized returns over a three-year horizon from the recommended purchase date. (Priel & Rokach, 2024). The overall benefit and increasing use of machine learning is also illustrated in a survey by the CFA Institute conducted in 2022, where 71% of respondents stated that new analytical methods created by artificial intelligence and machine learning will significantly affect job positions in the next 5 to 10 years. An overall overview can be seen in Figure 1.

Figure 1: CFA survey



Source: CFA Institute

3 Methodology and research methods

Machine learning is based on the idea that a computer should be able to learn independently without human assistance and then be able to apply what it has learned in practice. This chapter will introduce three categories of learning methods (supervised learning, unsupervised learning, and reinforcement learning), and will also introduce the algorithms that are often used in these types, the principle of testing and training data, which form the basic building blocks of machine learning, and measures of forecast accuracy.

3.1 Data selection

Testing and training data are the basis for machine learning, in achieving the correctness of algorithms and can be divided into two sets. **Training data set** represents the first series of data that is used in the system. This data is used to improve or train the model and usually represents between 70% and 80% of the total data set. It is therefore used to train machine learning algorithms so that they can make predictions, which can be for example classifications based on relationships and patterns found in this data set. When improving the model, we recognize inputs that represent characteristics or attributes that the model analyzes. An example could be predicting house prices, where the inputs can be location, plot size or number of rooms. The inputs are followed by outputs that represent target values or categories that we want the model to predict. If we were to stick to the example above, then for house prices the output would be the price of the house. The quality and size of the training data set strongly affects the performance of a machine learning model. In general, the training process consists of data collection and preparation, data partitioning, model training, model evaluation, and model optimization. Without training data, models would not be able to learn and generate predictions of results such as classification. **Testing data set** is the second data set, which usually represents 20% to 30% of the data from the total set. This data is just as important as the training data, although its use in the time sequence is after the training data. It is used to evaluate the performance of the model after it has been trained. Evaluation consists of verifying whether the model can correctly predict or classify new data and thus assess its ability to generalize. Test data also consists of inputs and outputs. Inputs represent characteristics or attributes that the model analyzes. These inputs can be, for example, measurements, parameters, or other relevant information. Outputs are target values or categories that the model predicts. They are used to compare with model predictions to evaluate the accuracy of the model. When testing data, the

usual first step is to divide the data into a set, evaluate the model, followed by evaluation methods and generalization with validation. We distinguish several evaluation methods, for example, accuracy, precision, recall, F1-score, measures of forecast accuracy such as mean square error (MSE), root mean square error (RMSE) or mean absolute error (MAE). We will discuss quality assessment at the end of this chapter. Testing data allows us to assess whether a trained model is overfitted or underfitted and is crucial for objective model evaluation (Chatzilygeroudis & Hatzilygeroudis & Perikos, 2021). Important issues in model evaluation, optimization, and selection are the overfitting and underfitting. Each of these issues represents an extreme in the model's ability to learn from training data and generalize correctly to new, unknown data. **Undertraining** occurs when a model is unable to capture patterns and relationships in the training data well. The model is too simple, and its prediction capabilities are weak. This leads to low accuracy on both training and test data. The first cause may be that the model is too simple, whereas if the model is not complex enough, it cannot capture patterns in the data. Other causes include a lack of features (input variables), too much regularization, which penalizes large weights and leads to a model that is too simple, or a lack of training data. An undertrained model is unable to properly capture the essence of the data and has low learning from the training data. Undertraining can be avoided by using a more complex model that better captures patterns in the data, adding relevant features, reducing the degree of regularization, or increasing the amount and quality of training data. **Overtraining** occurs when a model learns training data too well, even with the noise of this data and anomalies, which leads to high accuracy on training data, but poor generalization to new unknown data. Common causes of overtraining are an overly complex model with many parameters that adapt too well to the training data. To avoid overtraining, measures can be taken such as using regularization techniques that penalize large weights such as L1 and L2, using cross-validation to evaluate the model, reducing the complexity of the model, e.g. by reducing the number of parameters or layers in the neural netarticle, increasing the amount and quality of training data (Demyanov, 2015).

3.2 Types of learning in ML

Supervised learning is one of the main approaches in machine learning, where models learn from data that contains input and output pairs. The model is trained on training data that contains inputs and their corresponding outputs. The goal of the model is to learn the mapping between inputs and outputs so that it can make predictions about unknown and future data (target or explanatory variable). A prerequisite for its use is the existence of a sufficiently large training dataset that contains correctly labeled input–output pairs. It is therefore strongly dependent on a sufficient amount of quality data. In supervised learning, we distinguish two types of supervised learning: classification and regression. The difference in **unsupervised learning** is that the data is not labeled, and the algorithms try to discover, model, and describe patterns in the data in order to learn new information. The most used algorithm is clustering. In the case of **reinforcement learning**, no labeled or unlabeled training examples are used to train the model. Reinforcement learning occurs by creating a system (agent), which we deploy into the environment and let it learn through interaction with the environment. The only thing we need to determine about it are the rules for how it can behave in a given environment and the so-called reward function. With its help, the agent can evaluate whether the decision it has just made was beneficial for it or not. Then, using the trial-and-error method, like a human, it tries out individual options and learns how to ideally behave in individual situations. A model example for reinforcement learning is the game of chess - Deep blue defeated Kasparov, 1997. Detailed information can be found in the article Reinforcement Learning: an Overview (Glennec, 2000).

3.3 Forecast accuracy rates

The accuracy measures are based on a measure of the forecast error – the difference between the actual and estimated values $e_i = y_i - \hat{y}_i$ (where e_i is called the residual). The goal is to minimize forecast errors, that is, the better model is the one with the lower value of the calculated forecast accuracy measure. We do this with the following two formulas:

The **mean square error** (MSE) is used to measure the average size of the errors in the forecasts, where more emphasis is placed on larger errors. Lower MSE values predict better model quality. MSE is sensitive to extreme values, since larger errors have a quadratic effect on the result.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

The **root mean square error** (RMSE) is a metric derived from the MSE. Lower RMSE values (as with MSE) indicate a better model. RMSE places greater weight on large errors. RMSE is used to determine how far predictions are, on average, from the true values.

$$RMSE = \sqrt{MSE} \quad (2)$$

4 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a type of recurrent neural netarticle (RNN) specifically designed to solve problems in sequential data processing, where the outputs of previous steps are used as inputs for subsequent steps. This approach allows RNNs to retain information about previous states and process data with time dependencies. RNNs are useful for tasks such as time series, natural language processing, and speech recognition. LSTM is used in similar areas such as natural language processing, speech recognition, and time series. In natural language processing, LSTMs are used for tasks such as text translation, text generation, and sentiment analysis. In speech recognition, LSTMs are used to convert speech to text. Time series prediction is the prediction of future values based on historical data, and for example, music generation is based on learning musical sequences and creating new compositions. With LSTM, the goal is to provide a neural netarticle with a short-term memory spanning thousands of gradient steps, which represents a long short-term memory. This netarticle can be used in tasks that use memories of events that happened many (thousands and millions) steps earlier. LSTM solves problems with long-term dependencies that recurrent neural networks cannot handle due to the problem of vanishing gradients. LSTM allows for efficient identification and retention of relevant information over long periods of time, which is crucial for successful processing of sequential data. With LSTM, we recognize basic properties such as cells and gates (Sherstinsky, 2020).

A **cell** remembers values at any given time interval, and three gates regulate the flow of information into and out of the cell. An LSTM is made up of cells, which are the basic building blocks of the netarticle. A cell has an internal state that can be transmitted over many time steps, allowing the LSTM to “remember” information over long sequences.

Three main gates are used in an LSTM to control information:

The **input gate** decides which pieces of new information are stored in the current state, using the same system as the forget gates, in other words, it determines which new information is added to the cell.

The **forget gate** decides which information from the previous state to discard, by assigning a value between 0 and 1 to the previous state. A value of 1 means to keep the information, and a value of 0 means to discard the information. It therefore decides which information from the previous state is to be forgotten.

The **output gate** controls which information in the current state is to be output by assigning a value from 0 to 1 to the information with respect to the previous and current state. It thus determines which part of the cell state will be used as output.

The most common type of learning for LSTM is supervised learning, where the model is trained on labeled data. The application here is for time series prediction, where predicting future values is based on historical data, for example, predicting stock prices. Natural language processing is represented by tasks such as machine translation, text generation, sentiment analysis, or speech-to-text recognition. Sequence classification can be, for example, sequence classification (spam filter) or audio signal categorization.

5 Machine learning application with LSTM approach

Predicting the exact development of securities prices is an impossible discipline. The factors that affect a company's price on the market are mainly economic indicators, and only then can price predictions be considered. In this example, the price is determined without considering macroeconomic factors - without adapting fundamental analysis. The result is a price determination for the next day, based on the data provided. This data is drawn from Yahoo Finance. We articleed with a ten-year period from May 15, 2014, to May 15, 2024, and the analyzed company was the French company LVMH Moët Hennessy - Louis Vuitton S.A. The first step was to import libraries and create an empty data frame, into which we downloaded the price for the given period using the Yahoo Finance API (application programming interface) and found out what shape or size our data is. The data matrix represents 2561 rows and 6 columns and is shown in Figure 2.

Figure 2: Filling data into a data frame

```

1 # Import necessary libraries for data manipulation and visualization
2 import math
3 import pandas_datareader as web # For accessing financial data
4 import numpy as np # For numerical computations
5 import pandas as pd # For data manipulation
6 from sklearn.preprocessing import MinMaxScaler # For normalizing data
7 from keras.models import Sequential # For building the neural network
8 from keras.layers import Dense, LSTM # For adding layers to the neural network
9 import matplotlib.pyplot as plt # For plotting graphs
10 plt.style.use('fivethirtyeight') # For setting the plot style
11 import yfinance as yf # For accessing financial data from Yahoo Finance

1 # Create an empty DataFrame to store the data
2 df = pd.DataFrame()
3
4 # Fetch historical stock price data for the ticker 'ASME.DE' from Yahoo Finance
5 df = yf.download('MC.PA', start='2014-05-15', end='2024-05-15')
6
7 # Display the fetched data
8 df

[*****100%*****] 1 of 1 completed

      Date      Open      High      Low      Close  Adj Close  Volume
2014-05-15 00:00:00+02:00  142.250000  144.649994  141.899994  143.550003  120.781311  1157770
2014-05-16 00:00:00+02:00  143.850006  144.149994  142.550003  143.750000  120.949539  812583
2014-05-19 00:00:00+02:00  143.750000  143.949997  141.600006  142.350006  119.771829  797627
2014-05-20 00:00:00+02:00  142.699997  143.000000  141.250000  142.750000  120.108192  528324
2014-05-21 00:00:00+02:00  142.600006  143.949997  141.699997  143.949997  121.117836  807472
...
...
...
...
...
2024-05-08 00:00:00+02:00  790.000000  795.700012  786.000000  787.900024  787.900024  204110
2024-05-09 00:00:00+02:00  785.000000  789.000000  778.099976  789.000000  789.000000  150180
2024-05-10 00:00:00+02:00  795.099976  796.400024  787.000000  788.400024  788.400024  303924
2024-05-13 00:00:00+02:00  792.599976  793.099976  782.099976  782.599976  782.599976  167046
2024-05-14 00:00:00+02:00  783.599976  791.900024  780.099976  791.900024  791.900024  196008

2561 rows x 6 columns

1 # Get the number of rows and columns in the dataset
2 rows, columns = df.shape
3
4 # Print the number of rows and columns
5 print(f"The dataset contains {rows} rows and {columns} columns.")

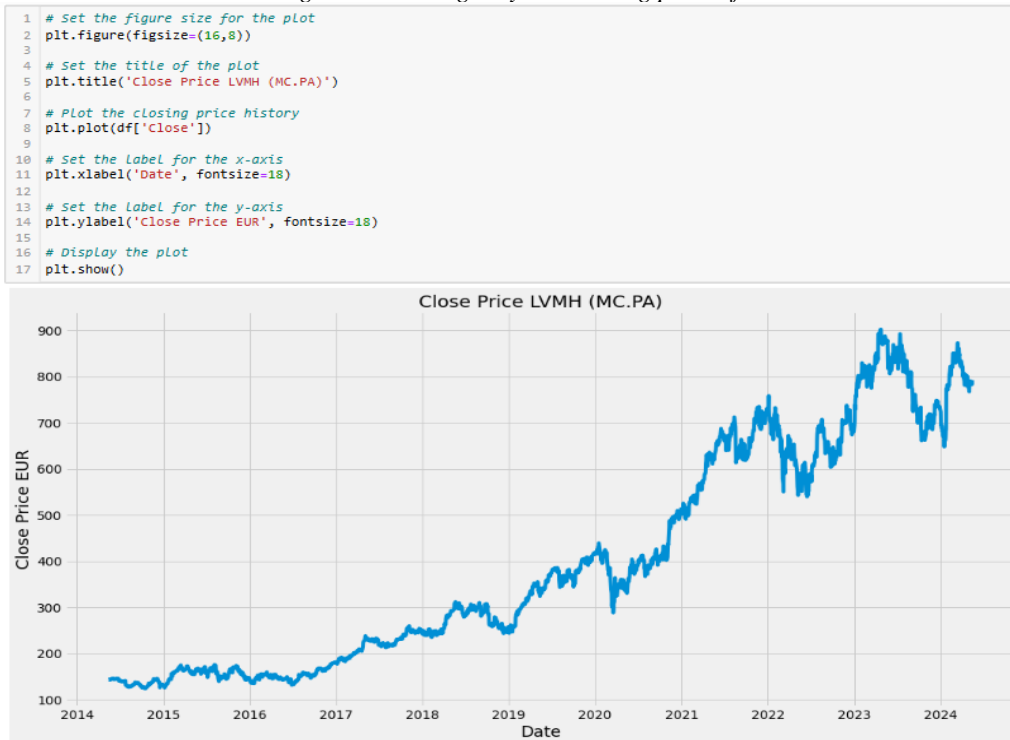
The dataset contains 2561 rows and 6 columns.

```

Source: own processing

We then filtered out the closing price and displayed it on the chart using the `plt.show()` function. We labeled both axes and named the chart, which we show in Figure 3.

Figure 3: Plotting 10 years closing price of LVMH



Source: own processing

Then, we create a new dataframe in Figure 4, which will contain only the closing price, and convert the dataframe to an array. After the conversion, we set the training data range to 80% and print out their number (2049). Using the MinMaxScaler function, we scale the data so that all values lie within the specified range, and then we train the scaling model on the dataset, which is where the data transformation occurs.

Figure 4: Creating training data sets

```

1 # Create a new DataFrame with only the 'Close' column
2 data = df.filter(['Close'])
3
4 # Convert the DataFrame to a numpy array
5 dataset = data.values
6
7 # Get the number of rows to train the model on (80% of the dataset)
8 training_data_len = math.ceil(len(dataset) * 0.8)
9
10 # Print the number of training rows
11 training_data_len

```

2049

```

1 # Initialize the MinMaxScaler with the feature range (0, 1)
2 scaler = MinMaxScaler(feature_range=(0, 1))
3
4 # Fit the scaler to the dataset and transform the data
5 scaled_data = scaler.fit_transform(dataset)
6
7 # Display the scaled data
8 scaled_data

```

```

array([[0.0254417 ],
       [0.02569868],
       [0.02389978],
       ...,
       [0.85403151],
       [0.84657883],
       [0.85852878]])

```

Source: own processing

After the MinMaxScaler function, we create a training dataset, assigning the previous 60 days to the `x_train` dataset and assigning the current day to the `y_train` variable.

Figure 5: Creating a training dataset and converting it to numpy arrays

```

1 # Create the training dataset
2 # Create the scaled training data set
3 train_data = scaled_data[0:training_data_len, :]
4
5 # Initialize the training data sets
6 x_train = []
7 y_train = []
8
9 # Split the data into x_train and y_train datasets
10 for i in range(60, len(train_data)):
11     # Append the past 60 days of data to x_train
12     x_train.append(train_data[i-60:i, 0])
13     # Append the current day (i-th day) to y_train
14     y_train.append(train_data[i, 0])
15     # Print the first few sets for verification
16     if i <= 61:
17         print("x_train:")
18         print(x_train)
19         print("y_train:")
20         print(y_train)
21         print()
22
23 # Convert x_train and y_train to numpy arrays
24 x_train, y_train = np.array(x_train), np.array(y_train)

```

Source: own processing

At the figure 6 we show output of the created arrays.

Figure 6: Creating a training dataset and converting it to numpy arrays

```

x_train:
[array([[0.0254417 , 0.02569868, 0.02389978, 0.02441375, 0.02595567,
0.0254417 , 0.02492771, 0.02634115, 0.02685512, 0.0262769 ,
0.02666238, 0.02852553, 0.02691937, 0.0273691 , 0.02762608,
0.02826855, 0.02505622, 0.0253132 , 0.02679088, 0.02717635,
0.02698362, 0.02653388, 0.02646965, 0.02781882, 0.02711212,
0.0282043 , 0.0282043 , 0.02794732, 0.02839705, 0.02563443,
0.02248635, 0.02132991, 0.02190813, 0.02216511, 0.02081593,
0.02325732, 0.02197238, 0.02043045, 0.01888853, 0.02023771,
0.02004498, 0.02004498, 0.02132991, 0.01920976, 0.02088018,
0.01901703, 0.01882428, 0.01888853, 0.01991648, 0.0206232 ,
0.02248635, 0.01015097, 0.01002249, 0.01021522, 0.00835207,
0.00623194, 0.0057822 , 0.00475425, 0.0077096 , 0.00732412]]])

y_train:
[0.004111785349492053]

x_train:
[array([[0.0254417 , 0.02569868, 0.02389978, 0.02441375, 0.02595567,
0.0254417 , 0.02492771, 0.02634115, 0.02685512, 0.0262769 ,
0.02666238, 0.02852553, 0.02691937, 0.0273691 , 0.02762608,
0.02826855, 0.02505622, 0.0253132 , 0.02679088, 0.02717635,
0.02698362, 0.02653388, 0.02646965, 0.02781882, 0.02711212,
0.0282043 , 0.0282043 , 0.02794732, 0.02839705, 0.02563443,
0.02248635, 0.02132991, 0.02190813, 0.02216511, 0.02081593,
0.02325732, 0.02197238, 0.02043045, 0.01888853, 0.02023771,
0.02004498, 0.02004498, 0.02132991, 0.01920976, 0.02088018,
0.01901703, 0.01882428, 0.01888853, 0.01991648, 0.0206232 ,
0.02248635, 0.01015097, 0.01002249, 0.01021522, 0.00835207,
0.00623194, 0.0057822 , 0.00475425, 0.0077096 , 0.00732412]), array([[0.02569868, 0.02389978, 0.02441375, 0.02595567, 0.0254417 ,
0.02492771, 0.02634115, 0.02685512, 0.0262769 , 0.02666238,
0.02852553, 0.02691937, 0.0273691 , 0.02762608, 0.02826855,
0.02505622, 0.0253132 , 0.02679088, 0.02717635, 0.02698362,
0.02653388, 0.02646965, 0.02781882, 0.02711212, 0.0282043 ,
0.0282043 , 0.02794732, 0.02839705, 0.02563443, 0.02248635,
0.02132991, 0.02190813, 0.02216511, 0.02081593, 0.02325732,
0.02197238, 0.02043045, 0.01888853, 0.02023771, 0.02004498,
0.02004498, 0.02132991, 0.01920976, 0.02088018, 0.01901703,
0.01882428, 0.01888853, 0.01991648, 0.0206232 , 0.02248635,
0.01015097, 0.01002249, 0.01021522, 0.00835207, 0.00623194,
0.0057822 , 0.00475425, 0.0077096 , 0.00732412, 0.00411179]])]

y_train:
[0.004111785349492053, 0.0037263109873162947]

```

Source: own processing

We then convert the training data to numpy arrays and display the shapes of the training sets. We need to create a three-dimensional shape in order to use the LSTM model, the creation of which is shown in Figure 7.

Figure 7: Use of LSTM

```

1 # Convert the x_train and y_train lists to numpy arrays
2 x_train, y_train = np.array(x_train), np.array(y_train)
3
4 # Display the shapes of the resulting numpy arrays
5 print(f"x_train shape: {x_train.shape}")
6 print(f"y_train shape: {y_train.shape}")
7
8 # Reshape the x_train data to be 3-dimensional for the LSTM model
9 x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
10
11 # Display the shape of the reshaped x_train data
12 x_train.shape

x_train shape: (1989, 60)
y_train shape: (1989,)

(1989, 60, 1)

1 Initialize a Sequential model
2 del = Sequential()
3
4 Add the first LSTM Layer with 50 units, return_sequences=True since we are adding more LSTM layers after this one, and input
5 del.add(LSTM(50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
6
7 Add the second LSTM Layer with 50 units and return_sequences=False since this is the last LSTM layer
8 del.add(LSTM(50, return_sequences=False))
9
10 Add a Dense Layer with 25 units
11 del.add(Dense(25))
12
13 Add a Dense Layer with 1 unit (output layer)
14 del.add(Dense(1))

1 #Compile the model
2 model.compile(optimizer='adam', loss='mean_squared_error')

1 #Train the model
2 model.fit(x_train, y_train, batch_size=1, epochs=1)

1989/1989 [=====] - 37s 18ms/step - loss: 2.3962e-04
<keras.callbacks.History at 0x1f8d82a64f0>

1 # Select the scaled data from training_data_len - 60 to the end to create the testing dataset
2 test_data = scaled_data[training_data_len - 60:, :]
3
4 # Create empty lists to store the features (x_test) and target values (y_test) for testing
5 x_test = []
6 y_test = dataset[training_data_len:, :] # Target values for testing data (unscaled)
7
8 # Populate x_test with sequences of 60 days for testing
9 for i in range(60, len(test_data)):
10     # Append sequences of past 60 days for each testing data point to x_test
11     x_test.append(test_data[i - 60:i, 0])
12

1 #Convert the data into a numpy array
2 x_test = np.array(x_test)

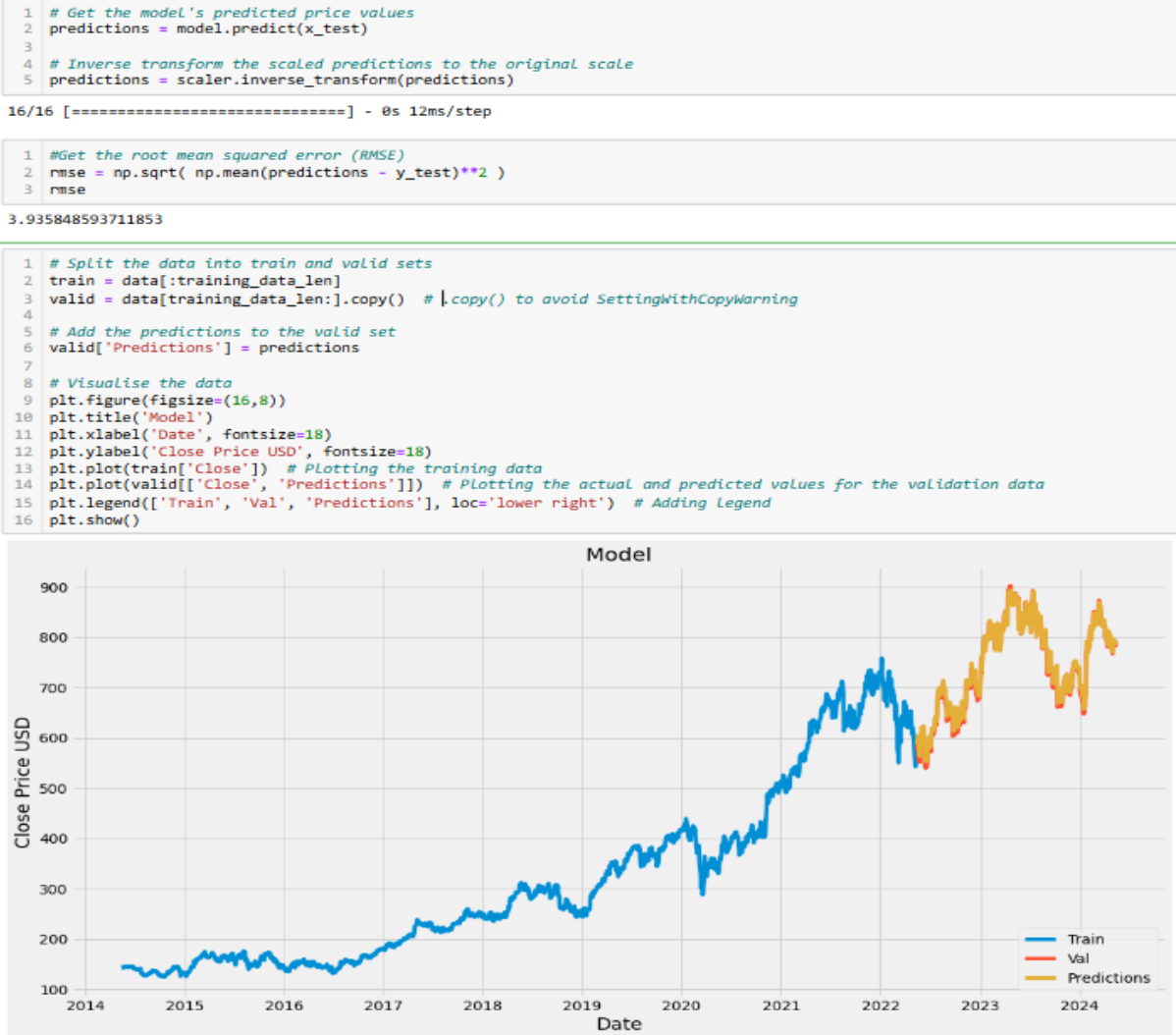
1 #Reshape the data to 3D shape
2 x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

```

Source: own processing

We run the model prediction and calculate the RMSE (2), which currently gives a value of 3.9358. In Figure 7, we see the data split into training and validation and plot the output along with the real data. We see that the prediction was successful, and the price was set with a high probability at the real value.

Figure 8: Visualization of model outputs



Source: own processing

The results shown in Figure 9 show the price prediction versus the actual closing prices of the asset.

Figure 9: Example of predicted values versus actual closing prices

```

1 #Show the valid and predicted prices
2 valid

```

	Close	Predictions
Date		
2022-05-16 00:00:00+02:00	575.000000	595.800293
2022-05-17 00:00:00+02:00	590.099976	588.232117
2022-05-18 00:00:00+02:00	578.500000	603.457947
2022-05-19 00:00:00+02:00	568.500000	590.343201
2022-05-20 00:00:00+02:00	556.599976	579.988098
...
2024-05-08 00:00:00+02:00	787.900024	794.584839
2024-05-09 00:00:00+02:00	789.000000	791.004456
2024-05-10 00:00:00+02:00	788.400024	792.454651
2024-05-13 00:00:00+02:00	782.599976	791.254211
2024-05-14 00:00:00+02:00	791.900024	784.652222

512 rows × 2 columns

Source: own processing

The final step is to run the algorithm to predict the next value that the model has not yet been trained on internally. The results differ but remain close to real value with a partial deviation. The procedure is shown in Figure 10.

Figure 10: Running the prediction algorithm

```

1 # Fill in company details into a table
2 lvmh_quote = pd.DataFrame()
3
4 # Download historical stock price data for the 'MC.PA' ticker (LVMH) from Yahoo Finance
5 lvmh_quote = yf.download('MC.PA', start='2014-05-15', end='2024-05-15')
6
7 # Filter the DataFrame to include only the 'Close' column
8 lvmh_quote = lvmh_quote.filter(['Close'])
9
10 # Print the data for verification
11 print(lvmh_quote)
12
13 # Get the last 60 days of closing prices and scale the data
14 last_60_days = lvmh_quote[-60:].values
15 last_60_days_scaled = scaler.transform(last_60_days)
16
17 # Create an empty list and append the past 60 days
18 X_test = []
19 X_test.append(last_60_days_scaled)
20
21 # Convert X_test to a numpy array and reshape the data
22 X_test = np.array(X_test)
23 X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
24
25 # Get the predicted scaled price
26 pred_price = model.predict(X_test)
27
28 # Undo the scaling to get the predicted price in the original scale
29 pred_price = scaler.inverse_transform(pred_price)
30
31 # Print the predicted price for next day
32 print(pred_price)

```

[*****100%*****] 1 of 1 completed

Date	Close
2014-05-15 00:00:00+02:00	143.550003
2014-05-16 00:00:00+02:00	143.750000
2014-05-19 00:00:00+02:00	142.350006
2014-05-20 00:00:00+02:00	142.750000
2014-05-21 00:00:00+02:00	143.949997
...	...
2024-05-08 00:00:00+02:00	787.900024
2024-05-09 00:00:00+02:00	789.000000
2024-05-10 00:00:00+02:00	788.400024
2024-05-13 00:00:00+02:00	782.599976
2024-05-14 00:00:00+02:00	791.900024

[2561 rows x 1 columns]
1/1 [=====] - 0s 31ms/step
[[796.30865]]

```

1 # Fill in company details into a table
2 lvmh_quote2 = pd.DataFrame()
3
4 # Download data for 'ASML' ticker for a specific date
5 lvmh_quote2 = yf.download('MC.PA', start='2024-05-15', end='2024-05-16')
6
7 # Check if data was retrieved successfully
8 if not lvmh_quote2.empty:
9     # Filter the DataFrame to include only the 'Close' column
10    lvmh_quote2 = lvmh_quote2.filter(['Close'])
11
12    # Display the 'Close' price for the specified date
13    print(lvmh_quote2['Close'])
14 else:
15    print("No data found for the specified date range. The symbol may be delisted or unavailable.")
16

```

[*****100%*****] 1 of 1 completed

Date	Close
2024-05-15 00:00:00+02:00	789.0

Name: Close, dtype: float64

Source: own processing

This completes the code that predicts the future value of an asset based on data from the past ten years. The algorithm can be further improved using regularization techniques and validation, which may be considered in the future when optimizing this example.

Conclusion

At the end of the article, we would like to evaluate the results, the main findings and the progress of the article. At the beginning of the article, we set ourselves the goal of examining how it is possible to determine the future value of an asset using machine learning or at least get closer to the value that may occur in the future. In order to be able to work on these findings, we gradually introduced the types of learning we recognize in the methodology. We introduced the basic algorithms for these types of learning and determined the occasions when they are appropriate to be used. Subsequently, we introduced LSTM, which is key in the calculations performed by our algorithms. After defining and explaining these terms, we proceeded to examine the prediction of stock price development using machine learning. Machine learning uses testing, training data and LSTM. The work was carried out with the company LVMH Moët Hennessy. In this article, we focused on the European market, since the company is French. In the model, we used a ten-year history, where the indicative data were the closing daily values of LVMH. Based on these values, we created a model using an algorithm that tries to predict the future value of the asset and plots the predicted values against the actual values from the past. The created model consists of machine learning and is focused on predicting stock price. Asset prices can never be fully predicted and thus serve only as an aid in predicting price development. The model does not include other elements of either technical or fundamental analysis. The principle on which the model articles is LSTM. The results show that the predicted price is close to real prices and the resulting code presents that the predicted price is also close to real values. An improvement of this model in the future could be the application of multiple epochs, the application of regularization techniques or a change in the algorithm, which could then be compared with the results of other models. Improvement could also lie in the application of quarterly or annual reports of the company. If these reports are positive, they can greatly influence the price upwards, and in the case of negative results on economic growth and the company's intended expansion, the price can drop sharply by many percent. These aspects are not included in the model. Another possible improvement is the inclusion of a reaction to resistance and support zones, which are mostly reflection zones if they are reached (the price reflects from them in the reverse direction). The last significant improvement could also be the reaction to lowering or increasing interest rates together with inflation reports, which are superior to the technical analysis of the company. When incorporating and creating such a complicated model, it would be possible to get closer to the real price of assets, but it is still true that the model cannot be created in such a way that it accurately creates future asset prices. In conclusion, we can say that machine learning is becoming an integral part of our everyday life and its importance is constantly growing. Uses and applications can be found from healthcare to finance. Machine learning is revolutionizing the way we approach problems and make decisions. This method is still under development and the longer it is studied, the more accurate results we will be able to achieve using various algorithms. From the perspective of the future of machine learning, it is clear that development will not stop. Progress in this area can be seen especially in recent years from companies such as Nvidia, Alphabet, Meta or Tesla. These companies are gradually becoming major players in the world of machine learning and are thus able to use their resources and improve processes better. Machine learning has the potential to improve the quality of our daily lives, but it can also solve global problems. The work is fully applicable to other assets as well, not just one selected stock. The same principle is used for other assets or funds that are composed of multiple assets. In cases where multiple assets are analyzed in parallel, the program can be modified to add various securities. If we later attempt to model a portfolio from these stocks, it is necessary to combine them into a portfolio before the calculation, the price of which will then be predicted.

We believe that with further development and innovation, machine learning will continue to advance and will play a key role in shaping a better and more sustainable world.

Funding: This article was supported by the Grant Agency of Slovak Republic: VEGA grant no. 1/0120/23 “Environmental models as a tool for ecological and economic decisions making”.

Literature

1. Chatzilygeroudis, K., Hatzilygeroudis, I. & Perikos, I. (2021). Machine Learning Basics.
2. Demyanov, S. (2015). Regularization Methods for Neural networks and Related Models.
3. *Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice (1st ed.)*.
4. Glorennec, P. Y. (2000). Reinforcement Learning: an Overview. Retrieved 22.6.2024 from: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b373b0c6e3b4fef4ac0534965708fc382343f8dc>.
5. Grudniewicz, J. & Ślepaczuk, R. (2023). Application of machine learning in algorithmic investment strategies on global stock markets. *Research in International Business and Finance*.
6. Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks (NBER Articleing Paper No. w4718). National Bureau of Economic Research. <https://ssrn.com/abstract=236673>
7. Lee, T. & Cho, J. & Kwon, D. & Sohn, S. (2018). Global stock market investment strategies based on financial netarticle indicators using machine learning techniques. *Expert Systems with Applications*.
8. Li, J & Wang X. & Ahmad, S. & Huang, X. & Khan, Y. A. (2023), Optimization of investment strategies through machine learning, *Heliyon*.
9. Lo, A. W., & MacKinlay, A. C. (1994). A Nonparametric Approach to Pricing and Hedging Derivative Securities. *The Journal of Finance*, 49(5), 1815-1839. <https://doi.org/10.1111/j.1540-6261.1994.tb00081.x>
10. Priel, R. & Rokach, L. (2024). Machine learning-based stock picking using value investing and quality features. *Neural Computing and Applications*.
11. Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229. <https://doi.org/10.1147/rd.33.0210>
12. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Netarticle (RNN) and Long Short-Term Memory (LSTM) netarticle. *Physica D: Nonlinear Phenomena*.
13. Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433-460. <https://doi.org/10.1093/mind/LIX.236.433>
14. White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. *1988 IEEE International Conference on Neural networks (Vol. 2, pp. 451-458)*. IEEE. <https://doi.org/10.1109/ICNN.1988.23959>

Pokročilé modelovanie rizika v povinnom zmluvnom poistení: Využitie štatistických metód a strojového učenia

Advanced risk modeling in motor third party liability insurance: Using statistical methods and machine learning

Peter Schmidt¹, Silvia Zelinová², Eva Rakovská³

Abstrakt

Cieľom článku je analyzovať možnosti kombinácie tradičných štatistických metód a modelov strojového učenia (ML) pri spracovaní údajov v oblasti povinného zmluvného poistenia (PZP). Východiskom je použitie generalizovaných lineárnych modelov a kontrastnej analýzy na štatistickú segmentáciu poistencov podľa rizikových faktorov, ako sú vek vodiča, značka a výkon vozidla či lokalita. V ďalšej fáze boli implementované ML algoritmy, najmä Random Forest, na imputáciu chýbajúcich údajov a predikciu závažnosti poistných udalostí (claim severity). Výsledky ukazujú, že strojové učenie dosahuje vyššiu predikčnú presnosť než tradičné modely, pričom zároveň identifikujú kľúčové faktory rizika. Kombinovaný analytický prístup spája výhody oboch metodík – vysvetliteľnosť a regulačnú transparentnosť štatistiky so škálovateľnosťou a výpočtovým výkonom ML. Záverom štúdia odporúča hybridný modelovací rámec ako optimálny nástroj pre segmentáciu poistencov, dynamickú cenotvorbu a efektívne riadenie rizika v PZP.

Kľúčové slová

PZP, poistné riziko, ML, predikcia poistného plnenia, Random Forest

Abstract

This article aims to analyze the potential of combining traditional statistical methods with machine learning (ML) models in the context of compulsory motor third party liability insurance (MTPL). The study begins with the application of generalized linear models and contrast analysis to statistically segment policyholders based on risk factors such as driver age, vehicle brand and engine power, or location. In the next phase, ML algorithms - particularly Random Forest - are used for imputing missing data and predicting claim severity. The results show that machine learning achieves higher predictive accuracy ($R^2 = 0.76$) than traditional models ($R^2 = 0.71$), identifying key risk drivers. The combined analytical approach leverages the strengths of both methodologies—interpretability and regulatory transparency of statistics, alongside the scalability and computational power of ML. The study concludes by

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35, Bratislava, Slovensko, peter.schmidt@euba.sk (*korešpondujúci autor).

² Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky Katedra matematiky a aktuárstva, Dolnozemska cesta 1, 852 35, Bratislava, Slovensko silvia.zelinova@euba.sk.

³ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35, Bratislava, Slovensko, eva.rakovska@euba.sk.

recommending a hybrid modeling framework as an optimal tool for policyholder segmentation, dynamic pricing, and effective risk management in MTPL.

Key words

motor third party liability insurance, insurance risk, machine learning, claim prediction, Random Forest

JEL classification

C51, C55, G22, C38

1 Úvod

Motorové vozidlo je dôležitým prvkom mobility v moderných spoločnostiach, no jeho prevádzka so sebou prináša riziko dopravných nehôd a následných finančných škôd. Povinné zmluvné poistenie (PZP) je preto kľúčovým pilierom poistného systému v mnohých krajinách vrátane Slovenska. Tento produkt zabezpečuje krytie škôd spôsobených prevádzkou motorového vozidla tretím stranám, čím chráni nielen vodičov, ale aj ostatných účastníkov cestnej premávky. Vzhľadom na jeho povinný charakter je efektívne a presné spracovanie veľkého množstva poistných údajov nevyhnutnosťou pre poisťovne pri riadení rizík, segmentácii poistencov a optimalizácii poistných produktov.

Tradične sa v oblasti PZP využívajú aktuárske a štatistické metódy na výpočet poistného, pričom zovšeobecnené lineárne modely (GLM) sú jedným z hlavných nástrojov na stanovenie rizikových profilov poistencov (Šoltés et al., 2019). V posledných rokoch však v poisťovníctve získavajú čoraz väčšiu popularitu modely strojového učenia (ML), ktoré umožňujú efektívnejšie spracovanie a analýzu rozsiahlych datasetov, predikciu poistných udalostí, či dokonca imputáciu chýbajúcich údajov (Richman, 2018). Spojenie tradičných štatistických metód s ML modelmi preto predstavuje inovatívny prístup k segmentácii poistencov, cenotvorbe a zlepšeniu kvality spracovania poistných dát (Hanafy et al., 2021).

Hlavným cieľom tohto článku je analyzovať možnosti kombinácie štatistických a ML metód pri spracovaní dát v oblasti PZP. V prvej časti sa zameriame na využitie kontrastnej analýzy pri segmentácii poistencov a odhade závažnosti poistných udalostí. Následne predstavíme postupy implementácie ML modelov na spracovanie poistných údajov, predovšetkým metódu Random Forest na imputáciu chýbajúcich hodnôt a predikciu poistného rizika (Rawat et al., 2021). Výsledkom bude porovnanie efektívnosti týchto prístupov a diskusia o možnostiach ich kombinácie v poistnej praxi.

2 Teoretický rámec

Spracovanie dát v povinnom zmluvnom poistení (PZP) je komplexný proces, ktorý si vyžaduje kombináciu aktuárskych a štatistických metód spolu s modernými prístupmi založenými na strojovom učení (ML). Cieľom tohto rámca je vysvetliť základné teoretické východiská oboch prístupov a ukázať ich potenciálne prepojenie pri analýze rizika, segmentácii poistencov a určovaní výšky poistného.

2.1 Aktuárske a štatistické metódy v segmentácii poistencov

V tradičných prístupoch k výpočtu poistného sa využívajú modely z oblasti aktuárskych vied a štatistiky. Zovšeobecnené lineárne modely (GLM) patria medzi najčastejšie používané

metódy v neživotnom poistení, pretože umožňujú efektívne modelovanie pravdepodobnosti poistných udalostí a ich závažnosti (Reiff et al., 2022). Pri výpočte poistného sa zohľadňujú viaceré premenné, ako napríklad:

- Technické parametre vozidla (výkon, hmotnosť, značka, rok výroby),
- Charakteristiky vodiča (vek, lokalita, história škodovosti),
- Účel využitia vozidla (osobné, firemné, taxislužba).

Kontrastná analýza v segmentácii poistencov

Jednou z menej využívaných, no veľmi efektívnych štatistických metód v poisťovníctve je kontrastná analýza. Tá umožňuje podrobnejšie skúmať rozdiely medzi skupinami poistencov a identifikovať relevantné faktory, ktoré najviac ovplyvňujú výšku škôd. Na rozdiel od tradičných priemerných hodnôt, ktoré môžu byť skreslené v dôsledku nerovnomerného rozdelenia údajov, kontrastná analýza využíva marginalizované priemery (Least Squares Means – LS-means) na elimináciu vplyvu ostatných faktorov.

V oblasti PZP umožňuje kontrastná analýza:

- Presnejšie rozdelenie poistencov do homogénnych skupín, kde v rámci segmentu neexistujú významné rozdiely v škodovosti;
- Odhalenie interakčných efektov medzi premennými, napríklad vplyv veku vodiča na závažnosť škôd v závislosti od značky vozidla;
- Lepšie modelovanie cenotvorby poistného, keďže poisťovne môžu stanoviť spravodlivejšie sadzby na základe štatisticky významných rozdielov medzi skupinami vodičov.

V tradičných aktuárskych modeloch sa však často zanedbáva komplexnosť vzťahov medzi jednotlivými faktormi, čo môže viesť k suboptimálnym výsledkom. Strojové učenie zohráva kľúčovú úlohu práve v tejto fáze analytického procesu, keďže umožňuje identifikáciu latentných vzorov v dátach a zároveň poskytuje nástroje na efektívne spracovanie rozsiahlych dátových súborov (Baudry & Robert, 2019).

2.2 Strojové učenie a jeho využitie v analýze PZP

Modely strojového učenia (ML) umožňujú poisťovniam nielen presnejšie predikovať riziká, ale aj efektívnejšie spracovávať veľké objemy dát. V oblasti PZP sa ML využíva predovšetkým na:

- Imputáciu chýbajúcich hodnôt – pomocou metód ako Random Forest Imputation je možné doplniť chýbajúce údaje v databázach bez skreslenia výsledkov;
- Predikciu pravdepodobnosti poistných udalostí – ML algoritmy dokážu identifikovať vodičov s vyšším rizikom nehody na základe historických údajov;
- Predikcia poistného na základe parametrov a škodovosti klienta;
- Odhalenie anomálií a podvodov – pokročilé algoritmy ako neuronové siete alebo autoenkóдеры dokážu detegovať podozrivé poistné udalosti.

Random Forest a jeho využitie pri spracovaní PZP dát

Jeden z najpoužívanejších algoritmov v poisťovníctve je Random Forest, ktorý patrí medzi metódy súborového učenia (ensemble learning). Tento algoritmus využíva viacero rozhodovacích stromov na vytvorenie robustného prediktívneho modelu. Jeho hlavné výhody zahŕňajú:

- Odolnosť voči pretrénovaniu – vďaka bootstrap agregácii (bagging) je model menej náchylný na chyby spôsobené šumom v dátach;

- Flexibilitu pri spracovaní rôznych typov údajov – dokáže pracovať s číselnými aj kategorizovanými premennými;
- Možnosť identifikácie kľúčových faktorov – pomocou metódy feature importance je možné určiť, ktoré premenné najviac ovplyvňujú škodovosť.

V poisťovníctve sa Random Forest využíva na imputáciu chýbajúcich hodnôt, kde predikuje chýbajúce údaje na základe podobnosti s existujúcimi poisťovacími zmluvami. Tento prístup umožňuje poisťovniam zachovať úplnosť databáz a minimalizovať chyby spôsobené nekompletnými údajmi.

2.3 Synergia tradičných aktuárskych metód a strojového učenia

Aj keď ML modely ponúkajú mnoho výhod, nie sú dokonalým riešením (Grize et al., 2020). Štatistické metódy, ako GLM a kontrastná analýza, poskytujú interpretovateľnosť výsledkov a overiteľnosť modelov, čo je dôležité v regulačnom prostredí poisťovníctva. Na druhej strane, ML modely dokážu efektívne spracovať nelineárne vzťahy a pracovať s rozsiahlymi datasetmi. Preto kombinácia oboch prístupov:

- Štatistické modely (GLM + kontrastná analýza) → Zabezpečujú teoretickú správnosť, vysvetliteľnosť a konzistentnosť výsledkov;
- ML modely (Random Forest, neurónové siete) → Pomáhajú odhaliť skryté vzory a optimalizovať procesy spracovania dát;

predstavuje optimálne riešenie.

Táto kombinácia umožňuje poisťovniam efektívne predikovať riziká, minimalizovať chyby v údajoch a zabezpečiť spravodlivejšie oceňovanie poisťového.

3 Metodika

Táto časť načrtáva metodický prístup použitý na analýzu údajov v kontexte povinného zmluvného poistenia zodpovednosti za škodu spôsobenú prevádzkou motorového vozidla. Dôraz sa kladie na súbor údajov, techniky predbežného spracovania údajov, stratégie štatistického modelovania a implementáciu algoritmov strojového učenia. Cieľom je poskytnúť transparentný a replikovateľný rámec na analýzu rozsiahlych poisťovacích údajov pomocou tradičných aj moderných analytických nástrojov.

3.1 Opis datasetu

Empirická analýza je založená na vlastnom súbore údajov poskytnutom nemenovanou poisťovňou, ktorý pokrýva poisťné zmluvy a záznamy o škodách počas deviatich rokov (2016–2024). Súbor údajov obsahuje viac ako 540 000 poisťných zmlúv a približne 33 500 poisťných udalostí súvisiacich s PZP, čo ponúka bohatý základ pre štatistické modelovanie a aplikácie strojového učenia. Všetky údaje v dátovom súbore boli anonymizované.

Z úplnej databázy bol extrahovaný výber relevantných premenných a kategorizovaný do troch hlavných skupín:

- *Technické parametre vozidla:*
 - Výkon motora (kW),
 - Hmotnosť vozidla (kg),
 - Rok výroby,
 - Značka vozidla,
- *Charakteristiky poisťníka:*
 - Vek vodiča,
 - Lokalita (okres trvalého bydliska),

- História škodovosti (bonus/malus),
- *Poistné a poistné udalosti:*
 - Výška ročného poistného (€),
 - Počet poistných udalostí,
 - Výška škôd (claim severity – CS).

Pred analýzou prešiel súbor údajov komplexným procesom čistenia. Všetky záznamy s neúplnými alebo neobnoviteľnými informáciami boli vylúčené. Okrem toho sa vykonala séria kontrol konzistencie, aby sa zabezpečila spoľahlivosť údajov. To zahŕňalo overenie číselných formátov, odstránenie duplicitných záznamov a identifikáciu odľahlých hodnôt alebo anomálií, ktoré by mohli naznačovať chyby pri zadávaní údajov alebo štrukturálne nezrovnalosti. Takáto príprava údajov je nevyhnutná pre štatistickú integritu aj robustnosť modelov strojového učenia. Zabezpečuje, že následné analýzy nie sú ovplyvnené chybnými záznamami a že modely sú trénované na kvalitných reprezentatívnych údajoch.

3.2 Predspracovanie dát

Efektívne predspracovanie údajov je kritickým krokom pri vytváraní robustných štatistických modelov a modelov strojového učenia (James et al., 2023). Zabezpečuje kvalitu údajov, rieši nezrovnalosti a pripravuje súbor údajov na spoľahlivú a nezaujatú analýzu. Pracovný postup predbežného spracovania pre túto štúdiu zahŕňal niekoľko kľúčových fáz (Provost, 2023).

Čistenie a kontrola kvality dát

Počiatočná fáza zahŕňala komplexnú kontrolu súboru údajov s cieľom identifikovať a opraviť problémy s kvalitou údajov. Riešilo sa niekoľko bežných problémov:

Identifikácia chýbajúcich hodnôt – Najčastejšie medzery boli pozorované pri premenných ako výkon motora, hmotnosť vozidla, bydlisko a rok narodenia.

Odstránenie extrémnych alebo nepravdepodobných hodnôt – Boli vylúčené záznamy s nereálnymi údajmi, ako sú uvedené vozidlá s výkonom motora 0 kW alebo vek vodiča pod 18 alebo nad 100 rokov.

Korekcia dátových formátov – bola vykonaná štandardizácia pre číselné polia (napr. oddelovače desatinných miest) a kódovanie znakov, aby sa zabezpečila konzistentnosť v rámci celého súboru údajov.

Imputácia chýbajúcich hodnôt

Na účely riešenia chýbajúcich hodnôt sme testovali viaceré stratégie imputácie (Li et al., 2023), pričom sme kládli dôraz na výber takej metódy, ktorá minimalizuje riziko narušenia pôvodnej štruktúry a logiky segmentácie dátového súboru:

- Priemerná imputácia bola vylúčená z dôvodu jej tendencie skresľovať rozptyl a potenciálne skreslenie modelov založených na segmentoch;
- Multiple Imputation by Chained Equations (MICE) vykazovala mierny výkon pre niektoré premenné a bola zvažovaná v predbežných skúškach;
- Náhodná imputácia lesa bola nakoniec vybraná ako najefektívnejšia metóda, ktorá vyvažuje presnosť so zachovaním premenných vzťahov.

Algoritmus Random Forest bol aplikovaný špeciálne na doplnenie chýbajúcich hodnôt pre technické vlastnosti vozidla, menovite výkon motora, hmotnosť a rok výroby, učením sa vzorov z podobných úplných záznamov.

3.3 Štatistické metódy a modely strojového učenia

Táto fáza analýzy bola zameraná na získanie poznatkov z vyčisteného súboru údajov prostredníctvom štatistického modelovania a techník strojového učenia. Použil sa dvojcestný prístup: štatistická segmentácia založená na kontraste, po ktorej nasledovalo prediktívne modelovanie.

Segmentácia poistencov pomocou kontrastnej analýzy

Na segmentovanie poistencov do štatisticky významných skupín sme použili zovšeobecnené lineárne modely (GLM) v spojení s kontrastnou analýzou. Tento proces sa opieral o výpočet priemerov najmenších štvorcov (LS-means) na neutralizáciu vplyvu mäťúcich premenných a na izoláciu účinkov kľúčových segmentačných faktorov.

3.4 Postup segmentácie

Postup segmentácie poistencov prebiehal v niekoľkých analytických krokoch, ktorých cieľom bolo vytvoriť štatisticky homogénne skupiny na základe relevantných rizikových faktorov.

- Identifikácia faktorov segmentácie – vrátane veku vodiča, značky vozidla, geografickej polohy a výkonu motora;
- Aplikácia GLM – na modelovanie závažnosti reklamácie (CS) ako funkcie týchto prediktorov;
- Kontrastná analýza – na testovanie štatisticky významných rozdielov medzi identifikovanými segmentmi;
- Spresnenie segmentov – zlúčenie alebo rozdelenie skupín na základe štatistickej významnosti, aby sa zabezpečila homogenita v rámci segmentov a heterogenita medzi nimi.

Konečná segmentácia priniesla znížený počet štatisticky odlišných zhlukov poistencov a profilov vozidiel, z ktorých každý sa vyznačuje významnými rozdielmi v priemernej závažnosti poistných udalostí.

Modelovanie claim severity pomocou Random Forest

Po segmentácii sme využili Random Forest (Breiman, 2001) model na predikciu claim severity. Tento model bol trénovaný na predikciu výšky poistných škôd na základe:

- veku vodiča,
- výkonu a hmotnosti vozidla,
- značky vozidla,
- histórie škodovosti.

Parametre modelu:

- počet stromov: 500,
- hĺbka stromov: Neobmedzená,
- bootstrap vzorkovanie: Áno,
- kritérium rozdelenia: MSE (Mean Squared Error).

Model bol trénovaný na 80 % údajov daného súboru a testovaný na zostávajúcich 20 %, čím sa zabezpečilo, že metriky výkonu odzrkadľujú zovšeobecnenie na neviditeľné údaje.

Výhody použitia Random Forest:

- Dokáže pracovať s veľkými datasetmi a heterogénnymi dátami,
- Minimalizuje multikolaritu medzi premennými,

- Má vysokú predikčnú presnosť a je odolný voči pretrénovaniu.

Validácia modelov

Na vyhodnotenie výkonnosti štatistických modelov a modelov strojového učenia sa použili viaceré overovacie metriky:

- Pre štatistický model (GLM + kontrastná analýza):
 - P-hodnota faktorov ($\leq 0,05$),
 - R^2 a AIC kritérium.
- Pre Random Forest model:
 - Mean Absolute Error (MAE),
 - Root Mean Squared Error (RMSE),
 - Feature Importance Analysis.

Výsledky oboch modelovacích ciest boli porovnané s cieľom identifikovať najefektívnejšiu metodiku pre segmentáciu poistencov a predikciu závažnosti poistných udalostí v kontexte PZP.

4 Výsledky a diskusia

Táto časť prezentuje výsledky segmentácie poistencov, predikciu závažnosti poistnej udalosti a porovnávacie vyhodnotenie štatistických modelov a modelov strojového učenia. Diskusia poukazuje na silné stránky a obmedzenia každého prístupu a identifikuje najvplyvnejšie faktory ovplyvňujúce závažnosť poistných udalostí v kontexte poistenia zodpovednosti za škodu spôsobenú prevádzkou motorového vozidla (PZP).

4.1 Výsledky segmentácie poistencov pomocou kontrastnej analýzy

Segmentácia poistencov bola vykonaná pomocou všeobecných lineárnych modelov GLM v kombinácii s kontrastnou analýzou. Tento prístup umožnil vytvorenie štatisticky homogénnych podskupín na základe vybraných rizikových faktorov.

Hlavné zistenia sú nasledovné:

- **Vek vodiča:**
 - Najvyššia priemerná závažnosť poistnej udalosti bola pozorovaná u vodičov vo veku 25 rokov alebo mladších. Rozdiel v závažnosti reklamácie v porovnaní s vodičmi vo veku 25–55 rokov bol štatisticky významný ($p < 0,01$).
 - Vodiči nad 55 rokov vykazovali mierne nižšiu priemernú závažnosť nárokov ako vodiči v strednom veku, ale tento rozdiel nebol štatisticky významný ($p = 0,12$).
 - Na základe týchto výsledkov boli vodiči zoskupení do troch segmentov:
 - Skupina A (≤ 25 rokov) – Vysoké riziko; zvýšené poistné
 - Skupina B (26–55 rokov) – Stredné riziko
 - Skupina C (56+ rokov) – Nízke riziko
- **Značka vozidla:**
 - Vodiči prémiových/luxusných vozidiel značiek ako BMW, Mercedes a Audi vykazovali vyššiu závažnosť nárokov v porovnaní s vodičmi štandardných značiek ako Škoda, Ford alebo Opel.
 - Rozdiel medzi prémiovými a neprémiovými značkami bol štatisticky významný ($p < 0,05$).

- **Lokalita:**
 - Vodiči z veľkých miest (Bratislava, Košice) mali o 25 % vyššiu závažnosť poistných udalostí oproti vodičom z menších miest a vidieka.
 - Rozdiel bol štatisticky významný ($p < 0,01$).
- **Výkon a hmotnosť vozidla:**
 - Vyšší výkon motora bol silne pozitívne korelovaný so závažnosťou poistných udalostí ($r = 0,68$).
 - Hmotnosť mala mierny negatívny vplyv ($r = 0,21$), čo naznačuje, že ťažšie vozidlá spôsobujú závažnejšie škody.

Celkovo sa segmentačný prístup založený na kontrastnej analýze ukázal ako účinnejší pri rozlišovaní rizikových skupín ako tradičné metódy, ktoré sa často spoliehajú len na základné kategorizácie veku alebo výkonu vozidla.

4.2 Výsledky predikcie závažnosti poistných udalostí pomocou Random Forest modelu

Na predikciu výšky škôd bol aplikovaný Random Forest model, ktorý bol trénovaný na 80 % datasetu a testovaný na zvyšných 20 %.

Výsledky modelu:

- MAE (Mean Absolute Error): 312,54 €
- RMSE (Root Mean Squared Error): 528,87 €
- R^2 (koeficient determinácie): 0,76

Tab. 1: Dôležitosť faktorov v modeli Random Forest

Faktor	Relatívna dôležitosť (%)
Výkon motora (kW)	29,3 %
Vek vodiča	23,7 %
Značka vozidla	18,2 %
Lokalita vodiča	14,5 %
Hmotnosť vozidla	7,9 %
História škodovosti	6,4 %

Zdroj: vlastné spracovanie

Model potvrdil, že výkon motora a vek vodiča sú najvýznamnejšie faktory ovplyvňujúce výšku škôd. Zároveň ukázal mierne lepšiu predikčnú presnosť v porovnaní s GLM modelom ($R^2 = 0,76$ vs. $R^2 = 0,71$ pri GLM).

4.3 Porovnanie štatistických a ML prístupov

Hlavné zistenia z porovnania štatistických prístupov a strojového učenia sú zobrazené v tabuľke 2.

Tab. 2: Porovnanie GLM a ML

Kritérium	GLM + kontrastná analýza	Random Forest
Segmentácia poistencov	Áno, manuálne definované skupiny	Automatická segmentácia
Interpretovateľnosť	Vysoká (jasné vzťahy medzi premennými)	Nízka (black-box model)

Presnosť predikcie (R^2)	0,71	0,76
Práca s veľkými dátami	Mierne obmedzená	Výborná
Schopnosť zachytiť nelineárne vzťahy	Obmedzená	Vysoká

Zdroj: vlastné spracovani

Môžeme ich zhrnúť do nasledujúcich bodov:

- GLM v kombinácii s analýzou kontrastu poskytuje vysokú mieru interpretovateľnosti, čo je kľúčové najmä v regulovaných prostrediach, kde sa vyžaduje transparentnosť a možnosť spätnej validácie modelu;
- Model Random Forest poskytuje vyššiu presnosť predikcie, zachytáva zložité nelineárne vzory a je vhodnejší pre prostredia s veľkým objemom údajov;
- Hybridný prístup – integrácia štatistického modelovania so strojovým učením – prináša najefektívnejšie riešenie, ktoré vyvažuje transparentnosť s presnosťou.

Štatistické modely zabezpečujú teoretickú konzistentnosť a súlad s predpismi, zatiaľ čo modely ML prispievajú k zlepšenej predikcii rizík a prevádzkovej efektívnosti

4.4 Diskusia a implikácie pre prax

Výsledky tejto štúdie naznačujú, že poisťovne môžu výrazne zlepšiť segmentáciu poistencov aj predikciu rizík kombináciou štatistických metód a metód strojového učenia. Z tohto výskumu vyplýva niekoľko praktických dôsledkov:

- Optimalizácia cien: Modely vylepšené ML umožňujú poisťovniam nastaviť presnejšie a individualizované sadzby poistného na základe jemne vyladených parametrov rizika.
- Dynamické modelovanie rizík: Využitie strojového učenia umožňuje priebežnú aktualizáciu rizikových profilov na základe aktuálnych údajov, čím sa zvyšuje flexibilita a operatívnosť procesov poistného upisovania.
- Regulačné výzvy: Používanie zložitých modelov ML vyvoláva problémy s vysvetliteľnosťou modelov, ktoré môžu predstavovať problémy v rámci regulačných rámcov, ako je Solventnosť II ((Trombetti, 2017). Kľúčovým hľadiskom zostáva zabezpečenie zrozumiteľnosti a opodstatnenosti výstupov modelu.

Zistenia podporujú zmiešaný analytický prístup, v ktorom sa modely ML primárne používajú na spracovanie údajov, predikčnú analytiku a detekciu vzorov, zatiaľ čo štatistické techniky sú zachované na validáciu, interpretáciu a regulačné vykazovanie. Táto synergia nielen zvyšuje výkonnosť modelu, ale pomáha aj udržiavať dôveru regulačných orgánov aj poistencov.

5 Záver

Táto štúdia sa zamerala na skúmanie integrácie štatistických metód a modelov strojového učenia (ML) na analýzu údajov v oblasti poistenia zodpovednosti za škodu spôsobenú prevádzkou motorového vozidla (PZP). Cieľom bolo posúdiť, ako môže synergia týchto prístupov zlepšiť segmentáciu poistencov, zlepšiť predikciu rizika a podporiť efektívnejšie stratégie oceňovania poistného. Kľúčové zistenia štúdie sú zhrnuté nižšie.

Po prvé, použitie kontrastnej analýzy v kombinácii so všeobecnými lineárnymi modelmi (GLM) umožnilo presnejšiu segmentáciu poistencov do homogénnych rizikových skupín. Začlenením premenných, ako je vek vodiča, značka vozidla a geografická poloha, sa poskytol

hlbší pohľad na rizikové profily a podporil sa vývoj spravodlivejších modelov oceňovania založených na dôkazoch.

Po druhé, algoritmus Random Forest preukázal vyššiu presnosť predikcie ($R^2 = 0,76$) pri modelovaní závažnosti nároku, čím prekonal tradičné štatistické prístupy ($R^2 = 0,71$). Model efektívne zachytil nelineárne vzťahy a identifikoval kľúčové rizikové faktory, najmä výkon motora a vek vodiča, ktoré významne ovplyvňujú výsledky poistných udalostí.

Porovnanie prístupov ukázalo, že kombinácia štatistických modelov a modelov strojového učenia prináša optimálne výsledky. Zatiaľ čo štatistické modely poskytujú interpretovateľnosť, teoretickú konzistentnosť a transparentnosť, ktoré sú nevyhnutné pre súlad s predpismi, techniky strojového učenia prinášajú vynikajúcu predikčnú silu, škálovateľnosť a schopnosť spracovávať veľké a zložité súbory údajov. Tento hybridný rámec podporuje informovanejšie rozhodnutia pri upisovaní poistného a vyššiu efektívnosť pri spracovaní údajov. Z praktického hľadiska zistenia naznačujú, že poisťovne môžu výrazne zlepšiť riadenie rizík a cenové procesy prijatím kombinovaného prístupu. Integrácia štatistických a ML metodológií umožňuje presnejšiu segmentáciu, dynamické modelovanie rizík a škálovateľnú analýzu údajov, čo v konečnom dôsledku vedie k presnejším a spravodlivejším prémiovým štruktúram. Prijatie modelov ML však prináša určité regulačné a etické výzvy. Najmä povaha „čiernej skrinky“ mnohých algoritmov ML môže brániť transparentnosti a sťažiť regulačné schvaľovanie, najmä v rámci, ako je Solventnosť II alebo IFRS 17. Preto je nevyhnutné zabezpečiť preukázateľnosť a spravodlivosť vo výstupoch modelu.

Pri pohľade do budúcnosti možno identifikovať niekoľko smerov budúceho výskumu. Rozšírenie o ďalšie techniky ML: Ďalšie štúdie by mali preskúmať použitie alternatívnych algoritmov, ako sú neurónové siete alebo stroje na zvýšenie gradientu, na vyhodnotenie ich výkonnosti pri predikcii rizika PZP. Analýza údajov v reálnom čase: Implementácia modelov ML schopných spracovávať streamingové údaje by mohla umožniť poisťovniam dynamicky reagovať na zmeny v individuálnych rizikových profiloch. Etické a regulačné hľadiská: Na zosúladenie aplikácií ML s vyvíjajúcimi sa regulačnými požiadavkami a spoločenskými očakávaniami je nevyhnutné neustále skúmanie ochrany údajov, algoritmickej spravodlivosti a preukázateľnosti modelov.

Na záver, kombinácia štatistickej analýzy a strojového učenia predstavuje účinný prístup k modelovaniu rizík a optimalizácii produktov v poisťovníctve. Keďže priemysel čelí rastúcemu tlaku na presnosť, efektívnosť a dodržiavanie predpisov, táto integrovaná stratégia ponúka cestu ku konkurenčnej výhode a lepšiemu rozhodovaniu. Výsledky tejto štúdie potvrdzujú, že spojenie kontrastnej analýzy s modelmi ML umožňuje robustnú segmentáciu poistencov a presnú predikciu závažnosti poistných udalostí v oblasti PZP. Budúci výskum by mal uprednostňovať pokročilé prediktívne modelovanie, integráciu telematických údajov a vývoj transparentných rámcov strojového učenia kompatibilných s reguláciou.

Príspevok bol spracovaný v rámci riešenia grantovej úlohy VEGA 1/0096/23 Vybrané metódy riadenia rizík pri implementácii čiastkových interných modelov na stanovenie kapitálovej požiadavky na solventnosť.

Literatúra

1. Baudry, M., & Robert, M. (2019). A machine learning approach for individual claims reserving in insurance. *ASMBI*, 35(5), 1127–1155.
2. Breiman, L. (2001). *Machine Learning*, 45(1), 5–32. doi:10.1023/a:1010933404324

3. Chollet, F. (2019). *Deep Learning v jazyku Python: Knihovny Keras, TensorFlow*. TensorFlow. Grada Publishing.
4. Grize, Y.-L., Fischer, W., & Lützelshwab, C. (2020). Machine learning applications in nonlife insurance. *Applied Stochastic Models in Business and Industry*, 36(4), 523–537. doi:10.1002/asmb.2543
5. James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning: With applications in Python*. Springer.
6. Li, A., Perry, R., Huynh, C., Tomita, T. M., Mehta, R., Arroyo, J., Vogelstein, J. (2023). Manifold oblique random forests: Towards closing the gap on convolutional deep networks. *SIAM Journal on Mathematics of Data Science*, 5(1), 77–96. doi:10.1137/21m1449117
7. Rawat, S., Rawat, A., Kumar, D., & Sabitha, A. S. (2021). Application of machine learning and data visualization techniques for decision support in the insurance sector. *International Journal of Information Management Data Insights*, 1(2).
8. Reiff, M., Šoltés, E., Komara, S., Šoltésová, T., & Zelinová, S. (2022). Segmentation and estimation of claim severity in motor third-party liability insurance through contrast analysis. *Equilibrium*, 17(3), 803–842. doi:10.24136/eq.2022.028
9. Šoltés, E., Zelinová, S., & Bilíková, M. (2019). General linear model: An effective tool for analysis of claim severity in motor third party liability insurance. *Statistics in Transition New Series*, 20(4), 13–31. doi:10.21307/stattrans-2019-032
10. Trombetti, G. (2017). *Solvency II*. Eurilink.

Porovnanie exekučnej efektívnosti programovacích jazykov Python a Java pri spracovaní dát

Comparison of the code execution efficiency of programming languages Python and Java used in data processing

Pavol Sojka¹

Abstrakt

Článok sa zameriava na efektívnosť vykonávania operácií v programovacích jazykoch Python a Java z hľadiska časovej náročnosti. Oba jazyky sa používajú v bežnom programovaní aj v dátovej vede. Dátová veda sa v súčasnosti rozrástla do veľkého rozsahu a tento článok by mal poskytnúť lepší prehľad o tom, ktorý z vybraných jazykov má vyšší výkon pri bežných činnostiach v základnom spracovaní dátových setov. Pripravili sme experiment a v praxi bežne používaný scenár v oboch jazykoch, ktorý zahŕňa prípravu dát, ich spracovanie a výpočet a následnú interpretáciu vrátených výsledkov.

Kľúčové slová

Python, Java, množiny údajov, spracovanie údajov

Abstract

The article focuses on the efficiency of execution of the programming languages computations in Python and Java in terms of time consumption. Both languages are used in common programming and also in data science. Data science has grown to a large scale nowadays and this article should provide a better overview of which of the selected languages has better performance in terms of basic data handling. We have prepared an experiment and a scenario commonly used in practice in both languages, which includes data preparation, data processing and calculation, and subsequent interpretation of the returned results.

Key words

Python, Java, datasets, data processing

JEL classification

C8

1 Úvod

V súčasnosti je veľký dopyt po dátových analytikoch a programátoroch, pretože dátová veda a priemysel v posledných rokoch výrazne narástli. S týmto trendom sa objavujú otázky, ako napríklad, ktorý programovací jazyk by sme mali použiť na splnenie požiadaviek na

¹ Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra aplikovanej informatiky, Dolnozemska cesta 1, 852 35 Bratislava, Slovakia, pavol.sojka@euba.sk.

spracovanie dát. Mnohé odpovede možno nájsť na internete alebo v literatúre. Môžeme zväziť tieto odporúčania na základe rozšírenosti jazyka, krivky učenia, dostupnosti zdrojov poznatkov a podobne. V našom článku sme si vybrali jedno kritérium na porovnanie a toto kritérium je založené striktné na meraní efektívnosti na základe hardvérových parametrov s prednastaveným softvérovým scenárom.

2 Príprava prostredia

Ako primárny operačný systém, na ktorom budú vykonávané experimenty, sme si vybrali Linux Ubuntu 22.04. Naše prostredie bolo umiestnené na platforme Google Cloud na virtuálnom stroji s hardvérovými parametrami začínajúcimi na ôsmich CPU (centrálne procesorová jednotka, Intel Xeon @ 2,20 GHz, family 6, model 79) a šestnástich gigabajtoch RAM (operačná pamäť). Počet CPU sa postupne menil a merali sme dopad na efektívnosť vykonávania kódu z hľadiska času. Môže sa zdať, že výsledky sú ľahko predvídateľné, ale našim cieľom nebolo len dokázať, že zníženie počtu CPU má negatívny vplyv na výkon, ale tiež presne merať hodnoty vo zvolenom prostredí a skúmať, či tento vplyv je rovnaký alebo podobný u oboch programovacích jazykov.

Po úspešnom nasadení operačného systému sme nainštalovali jazyk Python (verzia Python 3.10.12) a jazyk Java (verzia OpenJDK 21.0.6) z repozitárov operačného systému. Spolu s Pythonom sme nainštalovali aj knižnicu Pandas. Táto knižnica je navrhnutá na efektívnu prácu s dátovými množinami uloženými vo formátoch CSV, XLSX a ďalších. Na úspešnú inštaláciu kompilátora Java je potrebné nainštalovať Java framework a jeho vývojové nástroje.

Naším cieľom bolo otestovať rýchlosť nášho naprogramovaného kódu v týchto krokoch:

- Otvoriť súbor CSV
- Načítať stĺpce dátovej sady do pamäte
- Vypočítať súčet celého stĺpca „Tržby“ a „Zisk“
- Vytlačiť názov súboru a výsledok na obrazovku
- Vytlačiť výsledný čas vykonávania a ukončiť aplikáciu

Rozhodli sme sa generovať syntetické dátové sady vo formáte CSV (hodnoty oddelené čiarkami). Na tento účel sme použili aplikáciu naprogramovanú v jazyku Java pôvodne pre iné experimenty vykonané v minulosti. Vygenerovali sme 500 testovacích súborov s údajmi o 20 riadkoch a 15 stĺpcoch. Naš program môže generovať dátové sady so stovkami tisíc riadkov, ale pre naše experimenty by malo byť 500 súborov x 20 riadkov dostatočných. Vygenerované súbory boli skopírované do adresárov Python a Java.

3 Výsledky

Každé meranie sme vykonali trikrát za sebou v oboch jazykoch a zaznamenané časy sme uložili do tabuľky. Po troch cykloch sme vypli server a postupne menili počet používaných CPU z ôsmich na dve. Pozorovali sme malé zmeny v časoch vykonávania kódu, ale vyskytli sa len menšie rozdiely, až pri dvoch CPU bola zmena času významná. Ako vidno z tabuliek nižšie, môžeme povedať, že program v Jave bol efektívnejší ako v jazyku Python, pretože Java je kompilovaná do bajtkódu spusteného na Java Virtual Machine, ktorý je vždy rýchlejší ako interpretovaný jazyk – Python. Oba programy bežali v konzolovom okne Linux bash. Prvý meraný cyklus v oboch jazykoch bol vždy pomalší ako ďalšie dva. Tento efekt možno vysvetliť prednačítaním systémových knižníc do pamäti po prvom behu a preto sme merali časy až po

prvom spustení, aby sme tento efekt eliminovali. Časy v tabuľkách sú už matematicky zaokrúhlené kvôli prehľadnosti, pôvodne boli počítané rozdiely pri vyššej presnosti.

Tab. 1: Časy vykonávania (8 CPU)

	CPU: 8, Pamäť: 16 GB		
	Python (s)	Java (s)	Rozdiel (python/Java)
1. meranie	0,65	0,15	4,28
2. meranie	0,63	0,17	3,66
3. meranie	0,62	0,16	3,91
priemer	0,63	0,16	3,94

Zdroj: vlastné spracovanie

Tab. 2: Časy vykonávania (6 CPU)

	CPU: 6, Pamäť: 16 GB		
	Python (s)	Java (s)	Rozdiel
1. meranie	0,63	0,16	3,97
2. meranie	0,68	0,15	4,50
3. meranie	0,63	0,16	3,91
priemer	0,65	0,16	4,12

Zdroj: vlastné spracovanie

Tab. 3: Časy vykonávania (4 CPU)

	CPU: 4, Pamäť: 16 GB		
	Python (s)	Java (s)	Rozdiel
1. meranie	0,68	0,18	3,72
2. meranie	0,68	0,21	3,28
3. meranie	0,67	0,17	3,88
priemer	0,68	0,19	3,61

Zdroj: vlastné spracovanie

Tab. 4: Časy vykonávania (2 CPU)

	CPU: 2, Pamäť: 16 GB		
	Python (s)	Java (s)	Rozdiel
1. meranie	0,70	0,33	2,09
2. meranie	0,71	0,28	2,56
3. meranie	0,72	0,30	2,43
priemer	0,71	0,30	2,34

Zdroj: vlastné spracovanie

Ako môžeme vidieť z údajov v tabuľkách, jazyk Java je vždy efektívnejší ako jazyk Python. Efektívnosť oboch jazykov mierne klesá vždy, keď sa zníži počet jadier (CPU). Tabuľky 1–3 ukazujú, že efektívnosť Pythonu je podobná a v jazyku Java rýchlo klesá v poslednom scenári iba s dvoma CPU. Ako bolo spomenuté vyššie, prvé meranie je pomalšie ako ďalšie a tento efekt ovplyvňujú interné procesy riadené operačným systémom. Rozdiely v jazykoch Python a Java sú na prvý pohľad zrejmé. V porovnaní s Pythonom a knižnicou Pandas bude Java vždy efektívnejšia, pretože Java je staticky typovaný programovací jazyk a kompilátor pozná každú premennú a výraz pri behu (Adhikari, 2024). Na druhej strane Python je dynamicky typovaný programovací jazyk a preto sa typ premennej odvodzuje na základe jej použitia (Kohli, 2021). Ďalší rozdiel je, že Java je kompilovaná do bajtkódu, ktorý sa následne vykonáva na Java Virtual Machine (Adhikari, 2024). Python s jeho najbežnejším použitím je interpretovaný vždy zo zdrojového kódu (interpreter následne už volá skompilované knižnice), čo je jasne viditeľné z časov vykonávania (nižší čas spracovania = vyššia efektívnosť aplikácie pri ostatných nezmenených podmienkach). Na optimalizáciu procesov nášho konceptu by sme mali zahrnúť špecifické algoritmy vhodné pre daný typ úlohy, hlbší vhl'ad do konkrétneho programacieho jazyka, výber vhodnejších programovacích jazykov pre strojové učenie alebo výber výkonnejších knižníc jazyka Python použitých pri strojovom učení (Esposito & Esposito, 2020). Náš článok má za cieľ základné predstavenie často používaných jazykov strojového učenia, ktoré zahŕňajú jazyky ako Java, Python a R (Kumar, 2024). Často používaným nástrojom v praxi je tiež rámec Spark (Mehrotra & Grade, 2019), ktorý využíva efektívnosť Java a knižnice Spark používaných na manipuláciu s obrovským množstvom dát, ako aj platformy Hadoop (Nordeen, 2020) používané na udržiavanie takýchto veľkých dátových sád. Nasledujú časti kódu, ktoré sme použili v našom testovacom prostredí.

Obr. 1: Ukážka kódu v jazyku Python

```
for f in csv_files:
    df = pd.read_csv
    (f, sep=';')
    print('Location:', f)
    print('File Name:', f.split("\\")[ -1])
    print(f)
```

```
print('Total profit:', df['profit'].sum())
print('Total sales:', df['sale'].sum())
print("-----")
```

Zdroj: vlastné spracovanie

Obr. 2: Ukážka kódu v jazyku Java

```
public static void compute(String filename) {
    try (BufferedReader bufferedSource = new BufferedReader(new FileReader(filename))) {
        int start = 1;
        double hodnota = 0.0;
        double hodnota2 = 0.0;
        double sum = 0;
        double sum2 = 0;
        String line;

        while ((line = bufferedSource.readLine()) != null) {
            String[] cols = line.split(";");
            for (int i = 0; i < cols.length; i++) {
                cols[i] = cols[i].trim();
            }

            if (start != 1) {
                hodnota = Double.parseDouble(cols[4]);
                sum += hodnota;
                hodnota2 = Double.parseDouble(cols[5]);
                sum2 += hodnota2;
            }
            start = 0;
        }
        System.out.println(filename);
    }
}
```

```
System.out.println("Celkove trzby:" + sum);
System.out.println("Celkovy zisk:" + sum2);
System.out.println("-----");
```

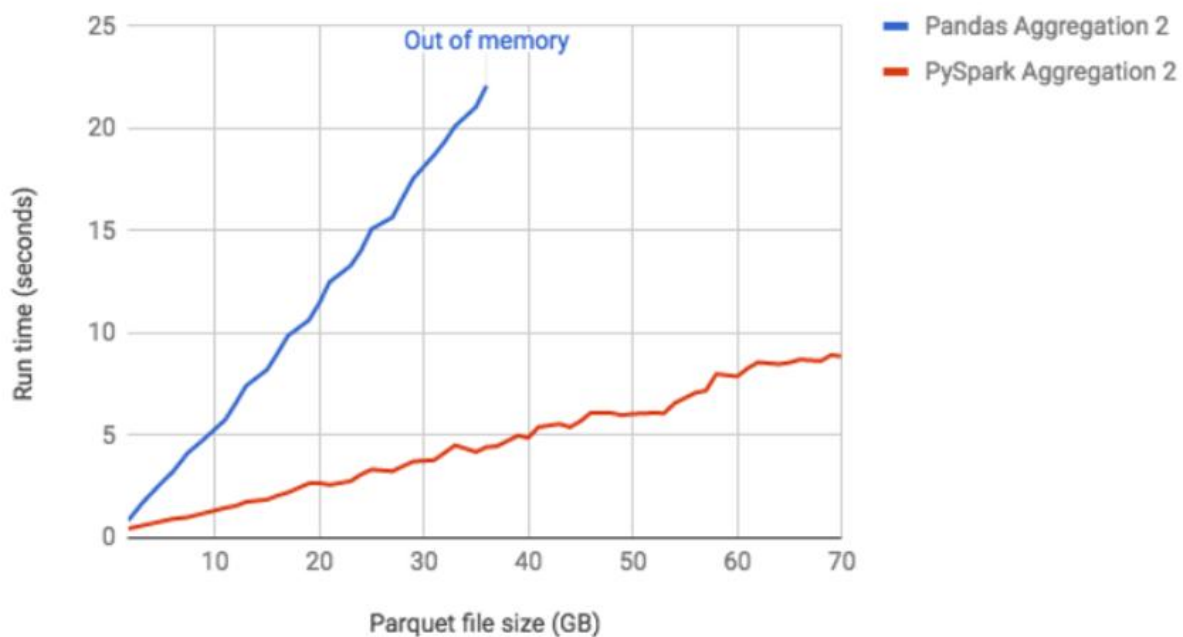
Zdroj: vlastné spracovanie

4 Diskusia

Náš prístup by mohol byť optimalizovaný pre lepšiu výkonnosť pomocou mnohých nástrojov a knižníc, ako sú Hadoop, Spark a špecializované rámce používané oboma jazykmi. Môžu sa objaviť ďalšie otázky, aký jazyk použiť, keďže mnohé iné jazyky (napr. Mojo, Julia a iné) zaznamenali v posledných rokoch veľký rozvoj. Python sa môže používať aj s rámcom Apache Spark. V tomto rámci je Python optimalizovaný pre špecializované knižnice, ktoré sú oveľa efektívnejšie ako samotný Python (Aven, 2018). Ako ukazuje nasledujúci graf, čas behu niekoľkých dopytov v PySpark (jazyk Python bežiaci spolu s Apache Spark) v porovnaní s rovnakými dotazmi v knižnici Pandas. Je zrejmé, že (Py)Spark skutočne umožňuje pracovať s väčšími údajmi efektívnejším spôsobom (Element 61, 2020). V grafe je znázornená (rovnako ako v našom článku) celková efektívnosť funkcie „sum“ v knižnici Python Pandas v porovnaní s knižnicou PySpark.

Obr. 3: Agregatívny dopyt Pandas vs PySpark

Pandas VS PySpark: aggregation query 2



Zdroj: Element (Element 61, 2020)

5 Záver

V našom článku sme sa zamerali na porovnanie dvoch jazykov aplikovaných na jednoduchú výpočtovú úlohu, aby sme ukázali efektívnosť použitých jazykov. Na základe našich výsledkov sme preukázali, ktorý z vybraných jazykov je z hľadiska výkonnosti lepší.

Existujú aj ďalšie aspekty, ktoré by sa mohli zohľadniť, a to je napríklad rozšírenosť konkrétneho jazyka a krivka učenia. Vzhľadom na tieto možnosti je Python víťazom z dôvodu jeho rozšíreného používania a známosti. Jazyk Java je pre nováčikov trochu komplikovaný na rýchle pochopenie. Na druhej strane Python je celosvetovo uznávaný jazyk s veľkou používateľskou základňou a rozsiahlou znalostnou bázou na riešenie problémov. Naučiť sa tento jazyk nie je pre nováčikov také náročné ako jazyk Java. Z našich úryvkov kódu možno pozorovať, že jazyk Python je ľahko čitateľný a jednoduchý. Tieto výhody a nevýhody možno podporiť alebo vyvrátiť kombináciou nástrojov, ako je Apache Spark, ktorý implementuje podporu pre Javu aj Python. Špeciálne pripravené knižnice by mohli zvýšiť efektívnosť jazyka Python, ale stále nebude taký výkonný ako čisto kompilované jazyky. Na záver pridávame obrázok najpoužívanejších jazykov v dátovej vede do roku 2024. Index popularity programovacieho jazyka bol vytvorený na základe analýzy frekvencie vyhľadávania príručiek a tutoriálov pre jednotlivé jazyky vo vyhľadávacej službe Google (PYPL, 2024).

Obr. 4: Programovacie jazyky používané v dátovej vede

Worldwide, Jul 2024 :				
Rank	Change	Language	Share	1-year trend
1		Python	29.35 %	+1.5 %
2		Java	15.6 %	-0.2 %
3		JavaScript	8.49 %	-0.8 %
4		C#	6.9 %	+0.1 %
5		C/C++	6.37 %	-0.1 %
6	↑	R	4.73 %	+0.3 %
7	↓	PHP	4.49 %	-0.5 %
8		TypeScript	2.96 %	-0.1 %
9		Swift	2.78 %	+0.2 %
10	↑	Rust	2.55 %	+0.4 %

Zdroj: PYPL (PYPL, 2024)

Literatúra

1. Adhikari, S. (2024). Exploring Scientific Computing with Java: A Practical Guide for Logic and Application Building (English Edition). India: Bpb Publications.
2. Aven, J. (2018). Data Analytics with Spark Using Python. Pearson Education.
3. Element 61. (2020). How to use Python and Pandas while embracing the power of Spark. Element 61. <https://www.element61.be/en/resource/how-use-python-and-pandas-while-embracing-power-spark>
4. Esposito, D., Esposito, F. (2020). Introducing Machine Learning. Pearson Education.
5. Kohli, M. (2021). Basic Core Python Programming. BPB Publications.
6. Kumar, A. (2024). Ultimate Java for Data Analytics and Machine Learning: Unlock Java's Ecosystem for Data Analysis and Machine Learning Using WEKA, JavaML, JFreeChart, and Deeplearning4j (English Edition). India: Orange Education Pvt Ltd.
7. Mehrotra, S., Grade, A. (2019). Apache Spark Quick Start Guide. Packt Publishing.

8. Nordeen, A. (2020). Learn Hadoop in 24 Hours. Guru99.
9. PYPL (2024). PYPL PopularitY of Programming Language. PYPL.
<https://pypl.github.io/PYPL.html>

EXTERNÍ RECENZENTI

Tomáš Bacigál

Peter Červenka

Petr Fiala

Patrik Mihalech

Erika Mináriková

Ján Pittner

Patrícia Teplanová

Marián Tokarský

Jana Tušimová

Adam Vranka

POKYNY PRE AUTOROV

Rozsah:

- vedecké state a diskusie 10 až 15 strán. Základnou požiadavkou je originalita príspevku a komplexnosť jeho spracovania. Prijímame príspevky v slovenskom, českom a anglickom jazyku (uprednostňujú sa príspevky v anglickom jazyku);
- informácie maximálne 2 strany;
- recenzie maximálne 2 strany.

Forma:

Použite textový editor MS WORD, verzia 2 000 a vyššia. Šablóna pre písanie článkov je na webovej stránke:

<https://fhi.euba.sk/veda-a-vyskum/vedecke-casopisy/ekonomika-a-informatika/o-casopise>

a v elektronickom systéme na stránke:

<http://ei.fhi.sk/index.php/EAI>

Príspevky predkladajú autori elektronicky vo formáte .doc/.docx do systému na stránke <http://ei.fhi.sk/index.php/EAI>. Príspevky sú recenzované. Redakčná rada zabezpečí interné a externé posúdenie textu príspevku. Autor príspevku je povinný zapracovať pripomienky z posudkov najneskôr do 2 týždňov od doručenia e-mailov so žiadosťou o vykonanie oponentských posudkov v elektronickom systéme časopisu a zaslať príspevok so zapracovanými pripomienkami vo formáte .doc/.docx prostredníctvom elektronického systému časopisu *Ekonomika a informatika*. Konečné rozhodnutie o publikovaní príspevku urobí redakčná rada časopisu. Autor pred zverejnením príslušného čísla časopisu *Ekonomika a informatika* odsúhlasí formátovanie elektronickej verzie článku. Fakulta hospodárskej informatiky si vyhradzuje právo zverejniť príspevky schválené redakčnou radou v elektronickej forme časopisu *Ekonomika a informatika*.

Autorské honoráre sa neplatia. Predložením príspevku do elektronického systému vedeckého časopisu *Ekonomika a informatika* dáva autor príspevku vydavateľovi právo, aby bezplatne publikoval text príspevku v časopise *Ekonomika a informatika* v elektronickej forme vo formáte .pdf.

EKONOMIKA A INFORMATIKA

Vedecký časopis Fakulty hospodárskej informatiky Ekonomickej univerzity v Bratislave a občianskeho združenia Slovenská spoločnosť pre hospodársku informatiku.

Poslaním vedeckého časopisu je publikovať teoretické a aplikačné poznatky získané v ekonomickom výskume a hospodárskej praxi z oblastí hospodárskej informatiky, účtovníctva a audítorstva, ekonometrie a operačného výskumu, aplikovanej štatistiky a aktuárstva, s akcentom na aktuálne otázky harmonizácie, integrácie a kompatibility s európskou a svetovou metodológiou a praxou.

Uverejňuje vedecké state a diskusie, recenzie a informácie o dizertačných a habilitačných prácach, inauguračných prednáškach a vedeckých podujatiach v slovenskom, českom alebo anglickom jazyku, ktoré sú výsledkom vedeckovýskumnej činnosti autorov, vedeckých aktivít doktorandov, medzinárodnej výskumnej a pedagogickej spolupráce a ich aplikácie v ekonomickej praxi.

ECONOMICS AND INFORMATICS

A scientific journal of the Faculty of Economic Informatics of University of Economics in Bratislava and the Slovak Economic Informatics Association.

Mission of the scientific journal is to publish theoretical and application knowledge acquired in economic research and practice in the areas of economic informatics, accounting and auditing, applied statistics, actuarial science, econometrics and operations research, with emphasis on the current issues of harmonization, integration and compatibility with the European and global methodology and practice.

The journal publishes scientific articles and paper discussions, reviews and information on doctoral and habilitation theses, inauguration lectures and scientific events in Slovak, Czech or English language, which are results of scientific and research activity of authors, scientific activities of doctoral students, international research and educational cooperation and their application in the economic practice.

EKONOMIKA A INFORMATIKA

Vydáva: Fakulta hospodárskej informatiky Ekonomickej univerzity v Bratislave a Slovenská spoločnosť pre hospodársku informatiku

Vychádza: 2x ročne