EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/I/2021/421000077974

KLASIFIKÁCIA ÚDAJOV AGREGAČNÝMI FUNKCIAMI

Diplomová práca

2021 **Bc. Erika Mináriková**

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

KLASIFIKÁCIA ÚDAJOV AGREGAČNÝMI FUNKCIAMI

Diplomová práca

Študijný program: Informačný manažment
Študijný odbor: Ekonómia a manažment
Školiace pracovisko: Katedra aplikovanej informatiky
Vedúci záverečnej práce: doc. Dr. Ing. Miroslav Hudec

Bratislava 2021 **Bc. Erika Mináriková**

Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Erika Mináriková
Študijný program: informačný manažment (Jednoodborové štúdium, inžiniersky II. st., denná forma)
Študijný odbor: ekonómia a manažment
Typ záverečnej práce: Inžinierska záverečná práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Klasifikácia údajov agregačnými funkciami

Anotácia: Klasifikácia je často realizovaná pravidlami typu IF-THEN nad relevantnými atribútmi záznamov. V takejto klasifikácii treba vytvoriť a manažovať IF-THEN pravidlá, čo nie je vždy jednoduché. Alternatívou môžu byť agregačné funkcie (uninormny, ordinálne súčty a pod.), ktoré každému záznamu pridelia jeho skóre.

Cieľom práce je vytvoriť klasifikačný priestor pomocou ordinálnych súčtov, preskúmať vhodnosť jednotlivých funkcií, na experimentoch preskúmať výhody a slabé stránky týchto funkcií a odvodiť záver.

Vedúci: doc. Dr. Ing. Miroslav Hudec
Katedra: KAI FHI - Katedra aplikovanej informatiky FHI
Vedúci katedry: Ing. Mgr. Peter Schmidt, PhD.

Dátum zadania: 27.10.2019

Dátum schválenia: 29.10.2019

Ing. Mgr. Peter Schmidt, PhD.
vedúci katedry

Poďakovanie

Ďakujem vedúcemu diplomovej práce, doc. Dr. Ing. Miroslavovi Hudecovi, za pomoc, odborný dohľad, rady a pripomienky, ktoré mi poskytol pri písaní a vypracovaní mojej diplomovej práce.

ABSTRAKT

MINÁRIKOVÁ, Erika: *Klasifikácia údajov agregačnými funkciami.* - Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra aplikovanej informatiky. – Vedúci záverečnej práce: doc. Dr. Ing. Miroslav Hudec. – Bratislava: FHI EU, 2021, 69 s.

Na naše rozhodnutia vplýva mnoho javov a preto nám často slúžia systémy ako poradný nástroj. Avšak, aby sme sa mohli spoľahnúť na rozhodnutie stroja a dôverovať mu, musíme chápať, ako prišiel k výsledkom a vedieť ich vysvetliť a jednoducho interpretovať. Klasifikácia je metóda, ktorá nám veľké množstvo dát rozdeľuje do príslušných kategórií. V našej práci sa sústredíme na triedenie dát do troch kategórií: *áno, nie, možno* s inklináciou ku *áno* alebo *nie*, vďaka čomu vieme zachytiť neistotu reality. Našim cieľom je vytvorenie vysvetliteľného modelu klasifikácie. Na dosiahnutie cieľa sme vytvorili klasifikáciu pomocou zmiešaných agregačných funkcií, ordinálnych súčtov tvorených konjunktívnymi a disjunktívnymi funkciami. Práca je rozdelená do štyroch kapitol, obsahuje 14 obrázkov, 11 tabuliek a 32 vzorcov. Prvá kapitola skúma rôzne existujúce klasifikačné metódy a poukazujeme na ich výhody ako aj nevýhody. Navrhli sme klasifikáciu pomocou agregačných funkcií a teoreticky sme si popísali ich základné vlastnosti. Druhá kapitola sa sústredí na ciele, metódy a metodiku práce. Tretia kapitola nám prinesie vytvorenie klasifikačného modelu pomocou ordinálnych súčtov a jeho aplikovanie na rôzne experimenty s dátami. V našej práci sme sa zaoberali modelom, ktorý spĺňa podmienky vysvetliteľnosti a interpretovateľnosti a je perspektívnym modelom pre využitie umelej inteligencie. Štvrtá kapitola opisuje dosiahnutú výsledky práce a prináša ďalšie návrhy na jej rozšírenie. Vytvorením transparentného a parametrizovateľného modelu sme navrhli možnosť využitia interaktívneho strojového učenia na zistenie tej najvhodnejšej kombinácie funkcií na klasifikáciu a tak umožnili vytvoriť vysvetliteľnú aplikáciu v oblasti umelej inteligencie.

Kľúčové slová:

klasifikácia, ordinálne súčty, agregačné funkcie, vysvetliteľnosť, interpretovateľnosť

ABSTRACT

MINÁRIKOVÁ, Erika: *Data classification by agregation functions*. – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Applied Informatics. – Thesis supervisor: doc. Dr. Ing. Miroslav Hudec. – Bratislava: FHI EU, 2019, 69 pages

When we are making decisions, we are affected by many influences, so we use technical systems as our advisor. However, we have to understand the decision-making system to figure out how the machine reaches the solution in order to trust it and depend on it. Classification is a method which categorizes a large amount of data. In our work, we focus on separating entities into three classes: *yes*, *no* and *maybe* with the inclination towards to *yes* or *no*. Thanks to this approach, we can catch and interpret vagueness in real world. Our goal is to make an explainable classification model. To reach this goal, we investigate classification by the aggregation functions of mixed behavior with the variability in ordinal sums of conjunctive and disjunctive functions. The thesis is divided into four chapters. It contains 14 picture 11 tables and 32 functions. The first chapter is devoted to various existing classification algorithms, where we also examine their pros and cons. As a solution, we suggest classification by aggregation functions and we theoretically describe their properties. Second chapter is focused on objectives, methods and methodology of the work. The third chapter proposes the classification model by ordinal sums and its applications on experimental data. In our work, we focused on explainable and interpretable classification model as well as its perspective for using UI. The fourth chapter describes the results and outlines further research activities. A transparent and parametrized model gives the opportunity for the interactive machine learning to find out the most suitable combination of functions and therefore to create an explainable application using artificial intelligence.

Key words:

classification, ordinal sums, aggregation functions, explainability, interpretability

OBSAH

ÚVOD

Klasifikácia je široko využívaná metóda v mnohých systémoch, triedi neprehľadné množstvo dát na menšie množiny. V pravidlových systémoch na klasifikovanie musia používatelia zadať AK-POTOM pravidlá, čo nie je jednoduchou otázkou. Klasifikovanie napríklad neurónovými sieťami vyžaduje množstvo dát označených aj s výstupmi, ktoré nemáme vždy k dispozícií. V našej práci sme navrhli klasifikáciu pomocou ordinálnych súčtov, kde nutne nepotrebujeme takéto dáta, ale používateľ môže slovne popísať problém klasifikácie, na základe čoho sa vytvorí agregačný model pre klasifikáciu. Cieľom našej práce je vytvorenie modelu klasifikácie, ktorého výsledky sú ľahko interpretovateľné a tak aj jednoducho vysvetliteľné koncovému používateľovi. Vysvetliteľnosť sa stáva dôležitou súčasťou systémov hlavne v odvetví medicíny, ale aj v ekonomike či v každodennom živote, pretože sú čoraz väčšou súčasťou a aby sme ich mohli prijať, mali by sme im dôverovať.

V prvej kapitole sa sústredíme na už existujúce klasifikačné metódy a skúmame ich výhody a nevýhody. Klasifikačné algoritmy strojového učenia sú dnes populárne na vytvorenie aplikácií umelej inteligencie. Avšak, také algoritmy, ako napríklad neurónové siete, síce dosahujú veľmi dobré performačné výsledky, ale ich dosiahnutie je pre nás neznáme. Naopak, pravidlové systémy sú jednoznačne identifikovateľné, avšak ich komplexnosťou sa stávajú neprehľadnými. Preto sme navrhli nový klasifikačný model s implementovaním agregačných funkcií. Aké majú vlastnosti, a ktoré funkcie medzi nich patria, si bližšie opíšeme v našej práci.

V druhej kapitole si definujeme náš hlavný cieľ a súčasne čiastkové ciele, ktoré skúmaním a nakoniec vytvorením modelu chceme dosiahnuť. Popíšeme si aj metódy, ktoré použijeme na dosiahnutie našich cieľov.

V tretej kapitole sa konkrétne pozrieme na klasifikáciu pomocou zmiešaných agregačných funkcií. Ukážeme si, ako môžeme vytvoriť klasifikačný model, ako môžeme meniť jeho správanie, čo potrebujeme na to, aby sme mohli model aplikovať. Vytvorili sme si experimenty, na ktorých sme pozorovali, ako model klasifikácie funguje a interpretovali sme dosiahnuté výsledky.

V štvrtej kapitole navrhneme rozšírenie nášho klasifikačného modelu použitím ďalších agregačných funkcií alebo zakomponovaním lingvistických súhrnov do interpretácie či využitím interaktívneho strojového učenia, na vytvorenie aplikácie umelej inteligencie.

1 Súčasný stav klasifikácie údajov

Klasifikácia je systematické delenie do kategórií. Využíva sa v mnohých odvetviach, ako napríklad v biológií, medicíne, geografii, tak aj v biznise, kde sa našla veľká výhoda v kategorizovaní jednotlivých dát, ako zgrupovanie produktov, zákazníkov, zamestnancov, atď. [33][44] Využívaním konkrétnych algoritmov môžeme pospájať jednotlivé prvky biznisu do spoločných skupín. Najnovšie sa téma klasifikácie otvára v spojení s umelou inteligenciou a strojovým učením.

Cieľom klasifikačných metód je kategorizovať vstupné údaje do viacerých preddefinovaných tried. Rozlíšiť medzi sebou dva rôzne javy pomocou atribútov, vlastností, opisom alebo iným definovaným kritériom do kategórií, ktoré už dopredu poznáme a poznáme aj ich charakteristiky [24].

Žijeme v dobe informačnej spoločnosti a ako už aj sám názov nám napovedá, je to práve rýchlo rastúca úloha informácie, ktorá je dominantnou charakteristikou novej doby. Je preto nadmieru dôležité vlastnostiam informácie rozumieť, a to znamená študovať informáciu v najrôznejších podobách. Dáta zo snímačov, počítadiel, masmédií a iných zdrojov, nám vytvárajú obrovské množstvo dát, ktoré už nie je pre človeka jednoducho pochopiteľné. Z pohľadu business intelligence je organizácia dát nevyhnutná pre podnik [44]. Ich klasifikácia a teda hierarchické rozdelenie je dôležitým krokom na ich pochopenie, čo nám poskytne porozumieť vzťahom na trhu či jeho segmentácii. Spracovanie dát človekom je náročná úloha, ktorá by nám zabrala veľké množstvo času a taktiež by sme sa nemuseli dostať k správnym výsledkom. Postupne, ako sa automatizácia rozširovala, zlepšovali sa aj možnosti spracovania dát a teraz máme mnoho algoritmov, nástrojov, ktoré nám v tom pomáhajú.

Klasifikácia je proces predikcie triedy daných vlastností. Triedy sa niekedy nazývajú ako ciele alebo kategórie [31]. Prediktívne modelovanie klasifikácie je úlohou aproximácie funkcie mapovania zo vstupných premenných na diskrétne výstupné premenné.

Rozlišujeme klasifikáciu do dvoch tried, nazývanú ako binárna, kedy sú triedy zvyčajne *áno - nie*, *patrí-nepatrí* a pod. alebo do viacerých tried. Dvojhodnotová logika môže byť však v niektorých prípadoch nedostačujúca [33] [24]. V prípadoch, keď sa nevieme rozhodnúť, sa nám žiada ďalšia trieda, *možno*. Avšak, aj v tomto prípade môžeme mať príklady, kedy objekt viac inklinuje k *nie* alebo *áno*. Vtedy sa nám žiada použitie viachodnotovej klasifikácie. Túto oblasť pokrýva fuzzy klasifikácia [25].

Ak chceme prvky klasifikovať, musíme určiť podmienky, na základe ktorých vieme vytvoriť klasifikačný model, čo môže byť niekedy ťažkou úlohou. Ak klasifikačný model nedostatočne popíšeme, niektoré objekty môžu zostať nezaradené alebo budú zaradené do zlej kategórie. Taktiež si musíme dávať pozor, ak definujeme veľa podmienok, aby sa navzájom nevylučovali alebo aby sme sa v ich spleti nestratili. Experti niekedy nemusia vedieť exaktne matematicky alebo logicky formulovať podmienky, pretože nemusia mať na to znalosti. Vďaka fuzzy logike vieme pomocou lingvistických súhrnou realitu lepšie zachytiť. Pravidlá potom môžu znieť napríklad, *ak si zákazník kúpi tovar za vysokú cenu, potom dostane veľkú zľavu*. Fuzzy logika založená na paradigme výpočtov so slovami nám ponúka možnosť vyjadriť užitočné vlastnosti pomocou veľkého množstva lingvistických výrazov, ktoré sú ľahšie pochopiteľné pre používateľov a taktiež sa aj jednoduchšie definujú [43]. Nemusíme určiť presné číslo, ale povieme len či je hodnota vysoká alebo nízka, či bol zákazník málo alebo veľmi spokojný.

Umelá inteligencia nachádza uplatnenie v modernej ekonomike v mnohých príkladoch v praxi. Klasifikačné nástroje sa používali už pred rozšírením umelej inteligencie, ale ich využitie nabralo druhý potenciál masovým zberom dát.

S exponenciálnym rastom technológií nepotrebujeme len lepšie nástroje na pochopenie zozbieraných údajov, ktoré momentálne máme, ale dáta nám budú pribúdať neustále, takže musíme vedieť spracovať aj tie nové. Tento problém nám rieši práve strojové učenie, ktoré nám prináša také algoritmy, ktoré sa vedia učiť na základe príkladov a následne nám vedia vyriešiť aj problém s novými, inými dátami.

Umelá inteligencia (UI), nemá presnú definíciu, ale voľne povedané, UI je stroj naprogramovaný tak, aby simuloval ľudské myslenie a konanie na dosiahnutie stanovaného cieľa [31]. Je to simulácia ľudskej inteligencie.

Strojové učenie (Machine Learning - ML) je aplikácia umelej inteligencie, ktorá poskytuje systémom schopnosť automaticky sa učiť a zlepšovať zo skúseností bez dodatočného programovania človekom [31]. UI nám prináša pomocou ML zlepšovanie systémov bez dodatočného zásahu vývojára, iba na základe naučených dát.

Prax klasifikácie s UI nadobúda čoraz väčší zmysel v podnikovom prostredí. Využíva sa napríklad na vylepšenie podnikových rozhodnutí, zvýšenie produktivity, ale aj pri iných odvetviach, ako zistenie choroby, schvaľovanie úveru, predpoveď počasia a mnoho ďalších.

Dôvodom, prečo sa stáva viac populárnou, je možnosť získania konkurenčnej výhody na globálnom trhu [26], získať zákazníkov a tak prosperovať v dynamicky sa

rozrastajúcej ekonomike. Pomocou správnej klasifikácie môžeme napríklad vybrať správnu marketingovú stratégiu pre zviditeľnenie alebo na základe zozbieraných dát presne zákazníkovi odporučiť produkt jeho predstáv. Klasifikáciu môžeme taktiež využiť pri odmeňovaní našich zákazníkov zľavami na mieru, podľa ich preferencií a ich hodnote pre náš biznis.

Ako sme si už uviedli množstvo príkladov, kde by sa klasifikácia dala využiť, rovnako sa už v súčasnosti aj v mnohých prípadoch používa a mnohokrát si možno ani neuvedomujeme, že ide o proces klasifikácie. Uvedieme si pár príkladov, kde už reálne nasadenie klasifikačných nástrojov pomocou UI môžeme vidieť:

- vyhľadávače, Facebook, Netflix,
- automatické vozidlá, drony,
- medicínske predpovede,
- hry,
- virtuálni asistenti,
- rozpoznávanie obrázkov,
- spam filter,
- predikcia sudcovského rozhodnutia,
- online reklama.

Medzi predikčné algoritmy okrem klasifikácie patrí aj regresia. Rozdiel je v tom, že pri klasifikácii, ako sme spomínali, máme diskrétne výstupné množiny a pri regresii je výstupom spojitá premenná.

Klasifikačné algoritmy strojového učenia patria do kategórie kontrolovaného učenia, kedy vstupné údaje poskytujú aj výstupné kategórie alebo ciele riešenia [32]. Z modelovacej perspektívy, klasifikácia pomocou strojového učenia vyžaduje veľké množstvo vstupných a výstupných dát, z ktorých sa následne učí. Tu sa musíme preto spoliehať iba na presnosť dát, na základe ktorých algoritmus klasifikácie pracuje. Avšak, to nie je pravidlo. Dáta môžu obsahovať množstvo chýb, či už obsahovať zlú hodnotu alebo dáta nebudú dostatočne diverzifikované. Ak sa systém naučí zo zlých dát, tak aj výsledná klasifikácia bude zlá. Bližšie sa nepresnosti venujeme v kapitole 1.3. Aj keď tréningový dátový set nebude obsahovať žiadne nezrovnalosti, informácie, ktoré z dát získame môžu byť zložité a pochopiteľné len expertmi UI. Na definovanie podmienok klasifikácie musíme poznať algoritmus, ale aj dáta a tieto informácie získame od doménových expertov. Tí však nevedia správne matematicky formulovať podmienky, hlavne napríklad pre lekárov alebo expertov na marketing môže byť táto úloha veľmi obťažná. Nepoznáme algoritmy strojového učenia

a preto je ťažko pochopiteľné, ako dospeli k záverom. Aby sa ľudia mohli na systém spoľahnúť, chcú vedieť, ako a prečo UI rozdelila dáta do tried, čo algoritmom strojového učenia chýba a preto nie sú dôverné.

UI a ML mení súčasné pozeranie sa na technológie a spracovanie dát. V súčasnosti je k dispozícii veľa klasifikačných algoritmov, ale nie je možné dospieť k záveru, ktorý z nich je lepší ako ostatné. Závisí to od aplikácie a povahy dostupného súboru údajov. Veľmi obľúbeným algoritmom využívaným na klasifikáciu v UI sú neurónové siete, rozhodovací strom, Naive Bayes. Iným populárnym spôsobom klasifikácie je klasický pravidlový systém a pravidlové systémy rozšírené o fuzzy logiku. V ďalšej časti si ich základné vlastnosti v krátkosti predstavíme a porovnáme ich výhody a nevýhody.

1.1 Pravidlové systémy

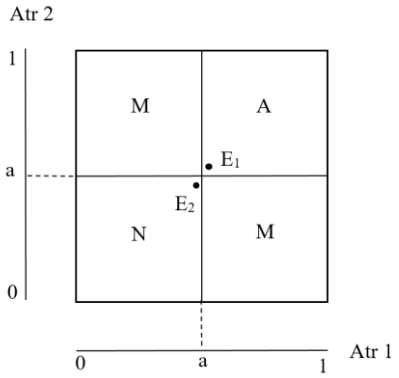
Najznámejšie klasifikovanie, ktorého výhody aj nevýhody boli popísané už v mnohých publikáciách je pomocou pravidlových systémov [24] [43]. Každá trieda klasifikácie je presne popísaná a objekt patrí do danej triedy alebo nepatrí. Problémom s ostrými hranicami tried sme sa venovali aj v bakalárskej práci [36], kde sme poukázali na jej nevýhody a predstavili sme riešenie opísaných problémov pomocou fuzzy pravidlových systémov.

Fuzzy pravidlové systémy predstavujú dôležitú oblasť vo využití fuzzy množín a fuzzy logiky. Rozširujú nám klasické pravidlové systémy s ostrými hranicami o flexibilitu, ponúkajú nám zachytenie nejednoznačných dát a zvyšujú nám pravdivosť výstupov, ktoré sa viac približujú k rozhodovaniu človeka. Vytvoreným fuzzy klasifikačným modelom môžeme matematicky zachytiť aj výrazy ako perspektívny a menej perspektívny zákazník alebo výkonný a menej výkonný zamestnanec. Opis situácie pomocou lingvistických výrazov je bližší k ľudskému vyjadreniu aj pochopeniu, ako zložité matematické funkcie.

Vyjadrením množín pomocou lingvistických výrazov nedefinujeme ich presné hranice, ale hranice sa budú navzájom prelínať. Preto prvok môže patriť do viacerých tried naraz s určitým stupňom príslušnosti, kedy platí suma jednotlivých príslušností sa rovná 1 [34]. Stupne príslušnosti nám predstavujú pravdivostné hodnoty z intervalu [0,1]. Spájaním dvoch alebo viacerých pravdivostných hodnôt, vypočítaním výslednej hodnoty príslušnosti, sa zaoberá viachodnotová logika [37]. Bola založená na jednoduchšie popísanie objektov reálneho sveta, kedy je ťažšie určiť či objekt jednoznačne patrí do danej skupiny alebo nie.

Keď porovnáme klasifikovanie pomocou klasického a fuzzy pravidlového systému na základe dvoch atribútov a ich parametrov, fuzzy klasifikácia nám prináša spravodlivejšie výsledky, teda výsledky zodpovedajúce bližšie realite. Fuzzy model sa odkláňa od klasických striktných hraníc a mení sa na hodnotu, príslušnosť do danej triedy. Rôzne matematické funkcie nám poskytujú širokú škálu možnosti výpočtu miery príslušnosti jednotlivých objektov do všetkých tried [25].

Grafické znázornenie pravidlového systému s ostrými hranicami do troch tried: *áno - A, nie – N, možno -M* môžeme vidieť na obrázku 1. Ilustračný príklad zobrazuje klasifikáciu pomocou dvoch atribútov, ale všeobecne každý pravidlový systém môže byť rozšírený na n (n Є N) atribútov.



Obrázok 1Klasický model klasifikácie s ostrými hranicami tried, zobrazujúci triedy: áno (A), nie (N), možno (M)
Zdroj : Vlastné spracovanie

Klasifikačný priestor opisujeme AK-POTOM pravidlami. Pravidlá opisujúce klasifikáciu s ostrými hranicami triedy:

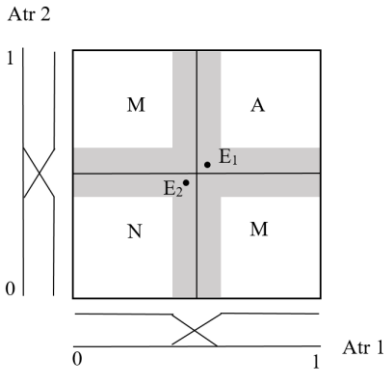
- AK Atr1 < a A Atr2 < a, POTOM N
- AK Atr1 ≤ a A Atr2 ≥ a, POTOM M
- AK Atr1 ≥ a A Atr2 ≤ a, POTOM M
- AK Atr1 > a A Atr2 > a, POTOM A

Na porovnanie obrázok 2 nám predstavuje klasifikáciu fuzzy pravidlovým systémom, kde vidíme, že prvok E_1 čiastočne patrí do všetkých štyroch tried, pričom $\sum_{i=1}^{4} \mu_i(E_1) = 1$. Na výpočet výslednej hodnoty, s akou príslušnosťou inklinuje prvok do triedy áno alebo nie, nám slúžia trojuholníkové normy (t-normy). Medzi najčastejšie používané t-normy patria minimová, súčinová a Lukasiewiczova [25] , vo fuzzy logike sa používajú na vytvorenie konjunkcie.

T-normy môžeme využiť aj na agregovanie atribútov opisujúci klasifikovaný objekt. Napríklad ak zákazník nakupuje často a zároveň vo vysokej hodnote tovaru, potom dostane vysokú zľavu. Teda t-normy využijeme aj v AK, aj v POTOM časti pravidla, kde môžeme implikáciu modelovať t-normou [19]. Funkčným vyjadrením výpočtu pravdivostných hodnôt rôznych akregačných funkcií, nielen t-normiem, sa budeme bližšie zaoberať v podkapitole 1.4.

Pravidlá opisujúce fuzzy klasifikačný priestor:

- AK Atr1 je nízky A Atr2 je nízky, POTOM N
- AK Atr1 je nízky A Atr2 je vysoký, POTOM M
- AK Atr1 je vysoký A Atr2 je nízky, POTOM M
- AK Atr1 je vysoký A Atr2 je vysoký, POTOM A



Obrázok 2 Fuzzy klasifikačný model troch tried: áno (A), nie (N), možno (M)
Zdroj : Vlastné spracovanie

Problém klasických pravidlových systémoch je, že nám poskytujú nespravodlivé rozdelenie do tried. Situáciu, kedy dva podobné vstupy dajú rozdielne výsledky v klasických pravidlových systémoch, sme vyriešili pomocou flexibilných tried, kedy podobné vstupy majú podobné výstupy. Nevýhodou, ktorú môžeme vidieť pri klasifikovaní pomocou fuzzy logiky je, že na výpočet príslušnosti do danej kategórie musíme experimentálne určiť viac parametrov. Teda tento druh klasifikácie je náročný na určenie veľkého množstva konštánt určujúcich vlastnosti tried [4]. Musíme označiť, ktorá hodnota je vysoká, ktorá nízka a to pri každej množine.

Aj keď počet definovaných tried môže byť teoreticky vďaka asociatívnosti t-noriem akýkoľvek, ideálny počet je od 3 do 9 tried [4]. Pri viacerých triedach sa môže stať

klasifikácia neprehľadná a zároveň náročná na dohľadanie vzťahov k jednotlivým vlastnostiam triedy, čiže vysvetliteľnosť výsledku by sa znižovala.

Fuzzy pravidlové systémy nám prinášajú uplatnenie v mnohých oblastiach a predstavuje nám silný nástroj na riešenie problémov. Ale pri zvyšovaní presnosti a teda pridávaní ďalších pravidiel, zvyšujeme komplexnosť systému, čo sa nám odráža zas na iných jeho vlastnostiach, ako je napríklad aj vysvetliteľnosť, ktorá je pri fuzzy systémoch kľúčovou.

Ako sme si už uviedli aj v predchádzajúcej kapitole, v súčasnosti sa do praxe viac a viac dostáva pojem umelá inteligencia a strojové učenie, ktoré nám prináša nové klasifikačné algoritmy. V ďalšej kapitole si uvedieme hlavné vlastnosti umelej inteligencie a strojového učenia.

1.2 Klasifikačné algoritmy strojového učenia

Umelá inteligencia je pomerne mladou vednou disciplínou, jej začiatok sa datuje rokom 1956. V tomto roku na konferencií v Dartmouth College, New Hampshire, po prvýkrát formulovali oblasť tejto vednej disciplíny a dostala pomenovanie umelá inteligencia.

Koncom 50. rokov F.Rosenblattem vyvinul model neurónu, schopný simulovať činnosť nervovej sústavy živých organizmov [22]. Týmto významným modelom položil základ výskumu v oblasti klasifikácie a rozpoznávania, teda v oblasti automatického zaraďovania objektov, javou a situácií do tried. Výskum jednoduchých neurónových sieti rozbehol taktiež vývoj prvých algoritmov strojového učenia.

Strojové učenie, ako aj samotná umelá inteligencia nemajú presne vymedzené definície. Zmyslom strojového učenia je program, ktorý si dokáže sám z príkladov odvodiť popisy alebo definície pojmu a na základe získania tohto poznatku následne vykonať klasifikáciu [31].

Klasifikácia je problém, ktorý vyžaduje algoritmy ML, ktoré sa učia ako priradiť dáta z domény k jednotlivých triedam. V praxi sa objavuje mnoho klasifikačných úloh a na každý pripadá špeciálny prístup riešenia. Preto sa aj vývoju klasifikačných algoritmov venuje náležitá pozornosť, aby výsledky boli čo najpresnejšie a kopírovali výsledky, ktoré by sme dostali od človeka. V ML rozlišujeme kontroľované a nekontroľované učenie. Pri kontoľovanom učení sú vstupné ako aj výstupné hodnoty opísané vlastnosťami. Na základe naučených súvislostí z dát vie klasifikovať nové dáta. Pri nekontoľovanom učení nevieme

výstupné hodnoty a algoritmus na základe vtupných dát nájde podobnosť a definuje triedy. Ďalej sme sa venovali iba kontrolovanému učeniu a spomíname iba dôležité aspekty súvisiace s našimi závermi, ďalšie podrobnnosti môžete vidieť v literatúrach, na ktoré sa odvolávame.

Najdôležitejšou podmienkou kontroľovaného učenia je dostupný set dát, čo nám môže prinášať nasledujúce problémy [30]:

1. Nereálne alebo málo pravdepodobné hodnoty
2. Chýbajúce hodnoty v dátovom sete (chýbajúce záznamy klasifikačnej triedy)
3. Nedostatočný počet záznamov
4. Dátový set obsahuje nerelevantné informácie
5. Problém nulovej chybovosti
6. Možná diskriminácia

1. Detekcia anomálií (outlier detection) nám slúži na nájdenie nereálnych alebo málo pravdepodobných hodnôt, ktoré by nám mohli skresliť výsledky klasifikačného modelu. Pred tým, ako dátový set použijeme na učenie, by sa mali také hodnoty prepísať alebo jednoducho odstrániť. Na obrázku 3 môžeme vidieť body E_1 a E_2 sú extrémne hodnoty, sú výnimkou medzi ostatnými oranžovými bodmi a bod E_3 nám predstavuje odľahlú hodnotu, pravdepodobne chybný záznam. Ak by sme si predstavili, že modré body predstavujú zákazníkov, ktorí si tovar kúpia, tak zákazníci E_1 a E_2 sú výnimky medzi skupinou zákazníkov s podobnými vlastnosťami, ktorí si daný tovar nekúpili. E_3 bude zákazník so zle zadanými hodnotami atribútov, pretože dané nadobudnuté hodnoty nie sú v definovanej škále.

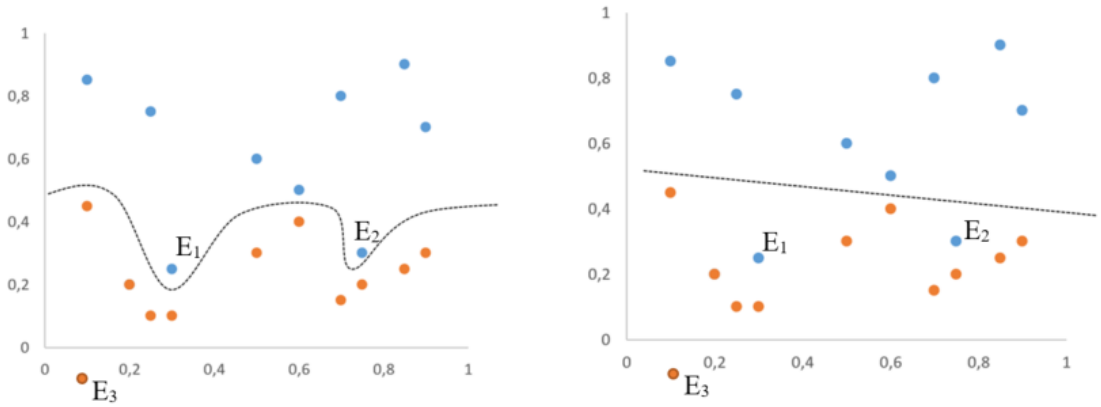
2. Množina trénovacích dát nie je dobrej kvality alebo presnejšie vstupné dáta nepokrývajú jednu významnú časť klasifikačného priestoru. Tú časť ktorú pokrývajú dáta, algoritmy ako napríklad neurónová sieť zvládne perfektne, ale keď prídu nové dáta, ktoré majú byť klasifikované do nepreskúmanej časti klasifikačného priestoru, neurónová sieť zlyhá [30].

3. Problémom sú aj úlohy, kde nie je veľký objem dát. Napríklad v odporúčacích systémoch sú to nie často kupované výrobky s väčším počtom atribútov (napr. domy a byty), oproti odporúčaní kníh a filmov (relatívne lacné a často kupované) [2] alebo choroby s nižšou frekvenciou výskytu [23].

4. Nadbytočné dáta v tréningovom sete dát zvyšuje komplexnosť, zabraňuje modelu rýchlejšie a efektívnejšie pracovať. Veľkým problémom je aj vzájomná závislosť dát, čo

ovplyvňuje presnosť. Vtedy potrebuje model ďalšie dáta, čo zvyšuje náročnosť aj problém získania veľkého objemu dát.

5. Pri trénovaní modelu strojového učenia môže nastať problém nulovej chybovosti alebo inak, sto percentnej presnosti. To znamená, že naučený model bude natoľko zložitý, že všetky tréningové dáta sa budú rovnať očakávanému výsledku. Problém je to práve preto, pretože stratíme generalizáciu, ktorá je podstatou pri strojovom učení. Výsledkom nových testovacích dát je veľká chybovosť, pretože naučený model je príliš komplexný. Situáciu nám znázorňuje obrázok 3.



Obrázok 3 Porovnanie 2 klasifikačných modelov: graf vľavo nám predstavuje nulovú chybovosť a graf vpravo predstavuje generalizovaný model

Zdroj : Vlastné spracovanie

6. Nekonzistentnosť dát nám spôsobuje problém, kedy naučený model klasifikácie sa naučí aj napríklad zlozvyky spoločnosti [28]. Napríklad, ak zozbierané dáta viac preferovali mužov pri prijímaní do povolania, aj naučený model bude taktiež preferovať mužov.

V nasledujúcej kapitole sa budeme podrobnejšie týmito problémami zaoberať a aj navrhnutými riešeniami.

Neurónové siete sú jeden z najznámejších algoritmov využívaným pri strojovom učení. Počítače sa učia na základe analýzy vstupných dát, ktoré sú už dopredu označené, teda hovoria nám aj o výstupe [31]. Neurónové siete na začiatku vyžadujú veľké množstvo dát, ktoré im slúžia ako základ na učenie. Po dostatočnom definovaní jednotlivých tried, vie počítač sám roztriediť nové vstupné údaje na základe už naučených skúseností.

Neurónové siete fungujú na základe princípu fungovania neurónov v ľudskom mozgu. Vstupné dáta z nezávislých premenných sa najprv kombinujú (najčastejšie pomocou

funkcie sčítania) a potom sa používajú ako vstupy do určitého typu aktivačnej funkcie, ktorá produkuje výstupné dáta [21]. Je to sieť zostavených z vypočítaných jednotiek (neurónov), ktoré sú navzájom poprepájané ohodnotenými váhami [6]. V procese učenia sa aktualizujú hodnoty váhových spojení. Výstupné dáta sa porovnávajú s očakávanými hodnotami a nastavenie siete sa optimalizuje dovtedy, kým sa pomocou procesu spätného učenia nedosiahne najmenšia možná chyba a optimálne prepojenie medzi vstupmi a výstupmi.

Základ neurónových sietí tvoria vrstvy a na základe počtu vrstiev rozlišujeme rôzne druhy [21]. Každý obsahuje minimálne tri vrstvy uzlov (neurónov): vstupná vrstva, skrytá vrstva/ vrstvy a výstupná vrstva.

V modeli môže byť niekoľko skrytých vrstiev v závislosti od zložitosti funkcie, ktorú bude model mapovať. Mať viac skrytých vrstiev umožní modelovať zložité vzťahy. Keď je však veľa skrytých vrstiev, trénovanie a nastavenie vĺn trvá veľa času.

Avšak najväčšou nevýhodou neurónových sietí je ťažké porozumenie spôsobu uloženia znalostí v ich štruktúre. V rozsiahlych maticiach váh sú nájdené zákonitosti uložené tak distribuovaným spôsobom, že do nich nevidíme [31]. Ďalšou nevýhodou je zlá interpretovateľnosť modelu v porovnaní s inými modelmi, ako sú rozhodovacie stromy alebo už uvedené pravidlové systémy, kvôli neznámemu symbolickému významu za naučenými váhami.

Algoritmus rozhodovacieho stromu je jeden z klasických algoritmov strojového učenia. Skladá sa z listov, ktoré nám predstavujú ohodnotenia a uzlov, ktoré obsahujú testy hodnôt atribútov [31]. Pri klasifikácií prechádzame stromom od hora na dol a porovnávame postupne testy v uzloch. Výsledky nám určujú, ktorou vetvou ďalej budeme pokračovať. Čiže rozhodovacie stromy nám postupne zisťujú vlastnosti objektu pýtaním sa otázok a zužovaním výsledných možností [5]. Pri veľkom množstve dát dostaneme veľa uzlov a môžeme sa stratiť v zložitosti riešenia.

Jednoduchý algoritmus na klasifikáciu s dobrými výsledkami je aj Naive Bayes, ktorý výsledok hľadá na základe spoločných vlastností definovaných tried [5]. Keďže atribúty sú závislé, predpoklad nemusí byť platný a taktiež sa dostávame do problému pri nulovej pravdepodobnosti.

Ďalšou možnosťou sú hybridné systémy, kde sú jednotlivé časti klasifikácie riešené podľa toho, ktoré systémy sú na daný problém vhodné. Napríklad neuro-fuzzy systémy. Tie nám však prinášajú taktiež ďalšie nevýhody. Napríklad veľké nároky na vstupné a výstupné dáta na určenie jednotlivých parametrov v klasifikačných podmienkach.

Algoritmy umelej inteligencie sú veľmi presné a flexibilné a môžu byť aplikované vo veľkom množstve aplikácií. Avšak, výsledný model je považovaný za takzvanú „čiernu skrinku“, teda modely, ktorých správanie nie je možné vysvetliť.

Vývojom vednej disciplíny umelá inteligencia sa ukázalo, že rozhodujúce pre vysokú efektivitu systémov umelej inteligencie sú znalosti. Významnú časť expertných znalostí tvoria znalosti nepresné, neurčité, vágne. Preto je teória spracovania neurčitosti v znalostiach a dátach veľmi dôležitou časťou teórie umelej inteligencie.

V nasledujúcej kapitole sa budeme venovať problematike neurčitosti znalostí a vysvetliteľnosti algoritmov.

1.3 Vysvetliteľnosť v strojovom učení

Ako sme si uviedli, všetky algoritmy strojového učenia využívajú veľké množstvo vstupných dát, aby sa naučili správnemu výsledku. Rovnako aj pri pravidlových a fuzzy pravidlových systémoch potrebujeme dáta opisujúce množiny, v tomto prípade aj experta, aby určil pravidlá.

Ak uvažujeme o probléme získania pravdivých a kompletných vstupných a výstupných údajov, na základe ktorých vyberáme naše riešenie, tak tento problém je spoločný pre všetky modelovacie procesy či sa jedná o fuzzy alebo nie [4]. Otázka výberu vstupných a výstupných dát a vlastností týchto údajov sú spoločné pre všetky systémy.

Veľká väčšina údajov a znalostí, ktoré využívame pri rozhodovaní či riadení, je do určitej miery neistá. Aj keď niektorý výrok považujeme za pravdivý, na základe prostredia sa môže jeho pravdivosť meniť [10]. Zovšeobecnená teória informácie si zobrala za úlohu študovať informáciu vo vzťahu k neurčitosti. Objavom teórie fuzzy množín a teórie fuzzy mier sa rozširuje rámec pre formalizovanie neurčitosti. Vlastnosť fuzzy logiky je prirodzeným jazykom opísať existujúce javy pomocou fuzzy pravidiel. Pre matematické formalizovanie je kľúčová, pretože nám ponúka logický záver vysvetliteľný pre človeka.

Dôvody, prečo je vysvetliteľnosť podstatná [4]:

- **Integrácia**: Vo vysvetliteľných modeloch získané vedomosti môžeme jednoducho overiť a pridať do množiny znalostí. Presnejšie môžeme overiť, či daný výstup je novým odvodeným faktom a môže byť pridaný do pôvodných podmienok.
- **Interakcia**: Využívanie prirodzeného jazyka ako nástroja na interpretovanie výsledkov prináša jednoduchšiu interakciu stroja a človeka. Interakciou sa myslí

tvorenie podmienok, pravidiel človekom pre stroj, ako aj interpretáciu výsledkov strojom pre človeka.

- **Overiteľnosť**: Získané vedomosti môžeme ľahko overiť podľa už známych znalostí. Vďaka tejto schopnosti vieme ľahko určiť prípadné nekonzistentnosti, ktoré môžu mať rôzne príčiny. Takáto detekcia je dôležitá pri zdokonaľovaní procesov.
- **Dôvernosť**: Najdôležitejším dôvodom použitia vysvetliteľných algoritmov je ľahké presvedčenie používateľa o ich funkčnosti a spoľahlivosti. Ak je systém schopný vysvetliť ako dospel k výsledku, používateľ môže byť spokojný, ako je výstup spracovaný.

Vysvetliteľnosť systémov má okamžitý dopad na aplikácie reálneho sveta. Ich využitie je vítané vo všetkých oblastiach, kde centrom je človek. Napríklad:

- Životné prostredie:

  Životné prostredie je veľmi širokou témou s veľkým množstvom premenných a nepresných charakteristík a v súčasnosti aj veľmi aktuálnou témou. Využitie počítačových výpočtov figuruje hlavne pri tvorbe modelu rozhodovania. Na základe vysvetliteľného modelu môžeme pochopiť vplyv okolitých javov na výsledné rozhodnutie systému. Napríklad na spravodlivé rozdelenie prostriedkov pre obnovu na základe prínosov a ďalších parametrov.
- Financie:

  V tomto sektore je vzťah počítačov a človeka veľmi blízky. Finanční experti používajú počítačové riešenie pri tvorení rozhodnutí na finančných trhoch, kde sa priamo výsledky dotýkajú človeka. Napríklad, ak by banka použila klasifikačný model založený na neurónovej sieti a výstupom by bolo, že klient nedostane úver, klient by sa zaujímal o dôvody, prečo nie. Ak nevieme vysvetliť klientovi prečo, môžeme o neho prísť.
- Priemysel:

  Využitie vysvetliteľných systémov je používané napríklad pri zistení príčiny chyby alebo v robotike ako lokalizačný systém, na základe analýzy pohybu.
- Medicína:

  V medicíne je nesmierne dôležité, aby sme sa na systém mohli spoľahnúť. Príkladom využitia je napríklad podporný rozhodovací systém na diagnostiku chorôb.

- Spoločnosť:

  V dnešnej dobe sme ovplyvňovaní novými technológiami nie len v pracovnom, ale aj sociálnom prostredí. Použitím vysvetliteľných systémov napríklad pri analýze správania na sociálnych siatiach, nám pomôže lepšie pochopiť ich dopad.

Na základe zozbieraných informácií považujeme vlastnosť vysvetliteľnosť v klasifikačných algoritmoch za veľmi dôležitú. Na klasifikáciu nám slúži mnoho nástrojov, od klasického pravidlového systému s ostrými hranicami tried, až po algoritmy umelej inteligencie.

Pravidlové systémy a ich rozšírenie o fuzzy logiku nám prinášajú vysvetliteľnosť riešenia, ale sa stávajú pri veľkom počte atribútov neprehľadné, komplexné a nezobraziteľné [24].

Algoritmy umelej inteligencie nám prinášajú síce dobré výsledky, ako z mnohých experimentov a prác [30] môžeme vidieť, ale neodpovedajú nám na otázku prečo a ako. Ako sme sa dostali k výsledku? Prečo objekt bol zaradený do danej triedy? Chýba nám vysvetliteľnosť a ľahká interpretovateľnosť a teda aj dôvera v systémy.

Vďaka mnohým výhodám sa algoritmy UI začali široko využívať najmä v priemyselnej oblasti, začali sa naďalej skúmať a zvyšovať nároky na ich výkon, čo viedlo k väčšej komplexnosti systémov. Aj keď nové algoritmy spĺňajú požiadavku na zvýšenie výkonnosti, systémy sa zmenili na tzv. „čierne skrinky“ a preto vyvolávajú neistotu vo výsledkoch. Nevieme, ako k jednotlivým záverom prišli a preto sa tento výstup stáva menej dôveryhodným, čo nám zabraňuje využiť technológiu v kritických oblastiach, ako je medicína, kde je presnosť nevyhnutná. Na základe tohto vývoja sa preto nadobúda otázka, ako zvýšiť dôveru v systémy a teda prichádza do záujmu vysvetliteľná UI. [28] nám ponúka prehľad literatúry a taxonómiu metód, ktoré sa danou problematikou zaujímajú.

Množstvo dát využité na učenie modelu nám zvyšuje predikčnú či klasifikačnú schopnosť a taktiež zvyšujú požadovanú presnosť. Avšak na druhej strane vysvetliť vnútorné fungovanie systému sa stáva o to obťažnejšie, otázku správnosti dát nám vôbec neriešia a preto výsledky sú ťažko interpretovateľné koncovému používateľovi.

Ako sme si naznačili, máme nepriamu úmeru medzi výkonom UI modelov, správnosťou dosiahnutých výsledkov a vysvetliteľosťou, resp. interpretovateľnosťou modelu. Rozdiel medzi týmito dvomi pojmami je, že interpretovateľnosť nám hovorí o tom, prečo sme daný výsledok dostali. Predstavuje vzťah medzi vstupom a výstupom, teda dôvod,

na základe ktorého zo vstupu bol konkrétny výstup. Vysvetliteľosť sa viaže k logike modelu strojového učenia, vnútornej štruktúre modelu, jeho implementácií. Predstavuje mechanizmus modelu, ako dosiahol výsledok, ako sa učil. Hovorí nám o pochopení algoritmu, ktorým sme dostali výsledky. K odstráneniu „čiernej skrinky“ potrebujeme naplniť oba pojmy, ako aj prečo. Ak sú dosiahnuté oba pojmy, vznikajú takzvané „biele skrinky“ (white-box/glass-box). Na jednej strane máme „čierne skrinky“ s vysokým výkonom a nízkou vysvetliteľnosťou a na druhej máme „biele skrinky“, ktoré sú vysvetliteľné a jednoducho interpretovateľné ale ich štruktúra, dizajn, je jednoduchý a komplexné systémy ťažko zachytí, resp. pri komplexných systémoch sa stáva štruktúra neprehľadná a preto sa nám aj vysvetliteľnosť stráca.

Systémy, ktoré sú ťažko interpretovateľné, nie sú tak dôveryhodné, hlavne v odvetviach, kde je nevyhnutná aj z morálnych dôvodov (medicína, samo jazdiace autá) alebo aby sme zostali spravodliví, ako napríklad v bankovníctve.

Pri otázke, ako dosiahnuť viac dôveryhodné systémy, sa dostávame k ďalšiemu pojmu a to je interaktívne Strojové učenie (UI). Tento pojem sa ešte veľmi neobjavuje [22]. Ako sme si uviedli, tak jedným pohľadom na vysvetliteľnosť je zmeniť „black-box“ na „white-box“ a to môžeme dosiahnuť tým, že človek by bol súčasťou algoritmu systému. Algoritmus by nebol 100% automatizovaný, len na základe vstupných dát, ale bol by ovplyvnený podmienkami experta. Človek by na začiatku nie len určil pravidlá a označil vstupy a výstupy, ale bol by aj súčasťou algoritmu a manažoval ho, aby napríklad možnosti, o ktorých nemáme dostatočné informácie na naučenie modelu, boli zahrnuté s rovnakou intenzitou ako ostatné [23].

V našej práci sa zaoberáme modelom, ktorý spĺňa podmienky vysvetliteľnosti a interpretovateľnosti a je perspektívnym modelom pre využitie UI. Preto mne sa zaoberali ďalším spôsobom klasifikácie údajov a to pomocou agregačných funkcií. Agregačné funkcie sú presne definované a matematicky dokázané a odvodené. To nám zabezpečuje, že presne vieme, ako sme sa k výsledku dostali. Agregačných funkcií existuje veľké množstvo a v ďalších častiach teórie sa budeme snažiť priblížiť základné a najznámejšie z nich. Ukážeme si mnohé príklady, pri ktorých použijeme rôzne funkcie, aby výsledok odpovedal realite. Tu sa dostávame k problému, ako správne vybrať, ktorá funkcia bude tá najvhodnejšia na konkrétny príklad klasifikácie. Túto odpoveď často pozná expert, ktorý vie, ako má výsledok vyzerať alebo nám na základe už definovaných príkladov môže pomôcť aj umelá inteligencia. Keďže v procese bude figurovať aj človek, výsledok nám nebude neznámy.

1.4 Agregačné funkcie

Ak máme hovoriť o agregačných funkciách, pozrime sa najskôr všeobecne, čo agregačné funkcie sú. Je to matematický nástroj, ktorého funkciou je zníženie počtu čísel do jedinečného reprezentujúceho, zmysluplného čísla [13]. Všeobecne hovoríme o zgrupovaní informácií do jednotného výsledku. O číslach hovoríme práve preto, pretože v počítačových modeloch nakoniec pracujeme s číslami.

Agregačné funkcie hrajú dôležitú rolu v mnohých technických otázkach, ktorým dnes čelíme. Sú široko využívané v matematike a aplikovanej matematike (štatistike, matematika rozhodovania), počítačových a technických vedách (napr. umelá inteligencia, operačný výskum, teória informácií, rozpoznávanie vzorov a analýza obrazu, dolovanie dát), ekonomike a financiách (napr. teória hier, teória hlasovania, rozhodovanie), spoločenských vedách (napr. reprezentačné meranie, matematická psychológia), ako aj mnohých ďalších oblastí aplikovanej fyziky a prírodných vedách.

Široké využitie agregačných funkcií nám hovorí, ako je táto téma overená, široko študovaná, matematicky podložená a vysvetliteľná a preto jej využitie popisujeme aj v tejto práci. Agregácia, zhlukovanie informácií, je dôležitou súčasťou systémov založených na znalostiach, od rozhodovacích procesov po strojové učenie. Môžeme teda povedať, že agregácia spája súčasne rôzne informácie, za účelom dosiahnutia záveru alebo rozhodnutia.

Našim zámerom je vybrať také agregačné funkcie, ktoré nám transformujú vstupy na výstupy tak, aby bol porovnateľný s realitou. Zmysel agregačných funkcií (taktiež volaných aj agregačné operátory) je spojiť, kombinovať vstupy. Vstupy napríklad vo fuzzy logike sú reprezentované ako stupne príslušnosti, alebo ich môžeme nazvať stupne preferencie, sila dôkazu alebo miera podpory hypotézy, atď. [8] Teda máme definované hodnoty, ako daný objekt spĺňa rôzne vlastnosti kategórie a našou otázkou je, aká je výsledná kombinácia, aký je náš výstup.

Agregačná funkcia je funkcia n > 1 argumentov, ktorá mapuje n-rozmerný (dimenziový) vektor na hodnoty z intervalu [0,1], f: $[0,1]^n \rightarrow [0,1]$ s vlastnosťami:

- limitujúce podmienky: f (1, 1 ... 1) = 1
  f (0, 0 ... 0) = 0
- monotónnosť: $x \leq y \Rightarrow f(x) \leq f(y)$ pre každé x, y $\in [0,1]^n$. (1)

Agregačné funkcie delíme do nasledovných tried [14]:

- konjunktívne (výsledok je medzi 0 a minimálnou hodnotou vstupného vektora, vrátane hraničných hodnôt),
- disjunktívne (výsledok je medzi maximálnou hodnotou vstupného vektora a 1, vrátane hraničných hodnôt),
- strednohodnotové (výsledok je medzi minimálnou a maximálnou hodnotou vstupného vektora, vrátane hraničných hodnôt),
- ich kombinácia – zmiešané.

Podmienky (1) nám predstavujú vlastnosti, ktoré má každá agregačná funkcia. Inak povedané, funkcia, ktorá spĺňa minimálne (1) sadu podmienok, môžeme považovať za agregačnú funkciu. Z tohto vyplýva, že napr. implikácia a ekvivalencia nie sú agregačné funkcie. Agregačné funkcie môžeme opísať aj ďalšími dodatočnými vlastnosťami, ktoré však nemusia platiť pre všetky funkcie, ale sú doplnkové pre konkrétne typy funkcií. Sú nimi kontinuita, asociatívnosť, symetria, existencia absorbujúceho elementu, existencia neutrálneho elementu, idempotencia, kompenzácia, vyvažovanie, posilnenie, invariancia (nemenosť), sémantická interpretovateľnosť, možnosť pridania váh [8] [13] [15].

Možno sa vám naskytla otázka či sú podmienky agregačných funkcií moc ohraničujúce. Teda v reálnom svete sa väčšinou nedostávame do situácie, kedy máme hodnoty z intervalu iba od 0 po 1, ale vyskytujú sa nám rôzne hodnoty. Otázkou zostáva či použitie rozdielnych rozsahov prinesie niečo nové do matematickej agregácie. Odpoveďou je nie. Agregačné funkcie sú izomorfné, to znamená, všetky agregačné funkcie na intervale $[a, b]$ môžu byť zamenené pomocou jednoduchej lineárnej transformácie na agregačnú funkciu na intervale $[0, 1]$ [8]. Preto výber rozsahu je otázkou vysvetliteľnosti a nie typu agragačnej funkcie.

Transformácia z jedného intervalu na druhý je jednoduchá a môže byť vykonaná rôznymi spôsobmi. Popis rôznych transformácií nájdeme v mnohých publikáciách [8].

Pri klasifikácií do tried, spájame hodnoty príslušností do jednotlivých kategórií pomocou agregačných funkcií. Zákazník získa zľavu alebo nie, zamestnanec dostane odmenu alebo nie pri pravidlových systémoch a pri fuzzy logike dostaneme variabilnejšie odpovede.

Každý zo spomínaných príkladov nám predstavuje situáciu, kedy je objekt opísaný vlastnými kritériami, parametrami a hodnotou do akej miery ich spĺňa. Našim cieľom je spojiť ich do výslednej hodnoty, na čo nám slúži mnoho agregačných funkcií. To, akú

agregačnú funkciu zvolíme, záleží na danej situácií, aké hodnoty agregujeme a aký očakávame výsledok. Napríklad, aritmetický priemer je neutrálnou agregačnou funkciou a použijeme ho, ak chceme, aby všetky hodnoty z domény boli do výsledku zarátané, čiže je plnekompenzačnou agregačnou funkciou, neeliminuje na základe žiadnej hodnoty z domény. Používa sa napríklad pri výpočte priemeru zo známok, hľadaní priemerného veku. Ak agregujeme hodnoty viacerých atribútov, pričom jedna hodnota je 0 a druhá 1, výsledok je 0,5, preto je aritmetický priemer pne kompenzačná neutrálna agregačná funkcia [15].

Výsledok bude medzi najvyššou a najnižšou vstupnou hodnotou pre všetky stredohodnotové funkcie. V prípade geometrického priemeru výsledok je nižší alebo rovný ako pre aritmetický priemer a v prípade výskytu hodnoty 0 výsledok je 0. Táto funkcia manažuje prípady, keď všetky atomické požiadavky musia byť splnené a vyššie hodnoty čiastočne kompenzujú nižšie.

V situácií výbere astronauta, by sme funkciu aritmetický priemer nemohli použiť, pretože, ak všetky kritéria spĺňa perfektne, ale stačí jedno kritérium nedostatočne, nemôže skúšky prejsť. Pri funkcií aritmetický priemer by sa jedna nedostatočná hodnota kompenzovala ďalšími dobrými a výsledok by nezodpovedal realite. V takýchto príkladoch potrebujeme funkciu, ktorá nám zabezpečí, aby všetky atomické požiadavky boli aspoň čiastočne splnené a aj keď je iba jedna hodnota z celej domény nízka, výsledok znižuje.

Opačný príklad máme pri diagnostike pacienta. Každá choroba je opísaná množstvom príznakov, ktoré danú chorobu charakterizujú. Ak pacient nedisponuje všetkými z nich, neznamená, že dané ochorenie nemá. Vtedy stačí ak aj jeden príznak sa objavuje vo veľkej intenzite a ostatné v nízkej, aby sme boli na pozore, pretože pacient môže potrebovať liečbu. Teda aj keď iba jedna hodnota z domény je vysoká a ostatné nízke, potrebujeme zvýrazniť výsledok na vyššiu hodnotu.

Ďalej sa môžeme dostať do situácie, kde výsledok nízkych hodnôt sa znižuje, vysokých hodnôt sa zvyšuje a pri kombinácií dostávame strednú hodnotu. Typickým príkladom môže byť situácia, keď nám nízke hodnoty znamenajú negatívne informácie a vysoké hodnoty pozitívne. Preto pri nízkych, negatívnych hodnotách chceme znížiť a naopak pri vysokých, pozitívnych, chceme výsledok zvýrazniť. Pri zmiešaných informáciách aj pozitívnych aj negatívnych, potrebujeme vypočítať strednú hodnotu. To môžeme ilustrovať na príklade klasifikačného priestoru na obrázku 2 . Napríklad, ak máme dva atribúty, ktoré opisujú správanie zákazníkov, vysoké hodnoty oboch atribútov zvýraznia perspektívnosť zákazníka, nízke hodnoty zvýraznia jeho neperspektívnosť a mix vysokých a nízkych hodnôt znamenajú viac-menej perspektívneho zákazníka. Ďalším príkladom môže

byť rozhodnutie či investovať alebo nie. Existujú rôzne vplyvy trhu, ktoré nám predstavujú klasifikačné atribúty, na základe ktorých sa rozhodujeme [8]. Ako sme videli, rôzne príklady, vyžadujú rôzne agregačné funkcie. Niekedy vyžaduje aby sa hodnoty spriemerovali, inokedy aby sa posilnili v prospech danej triedy, daného rozhodnutia, inokedy naopak, aby hodnoty výsledok znížili. Ďalej si ich formálne popíšeme.

1.4.1 Strednohodnotové agregačné funkcie:

Agregačná funkcia sa správa ako strednohodnotová, ak pre každé x platí:

$$\min(\mathbf{x}) \leq f(\mathbf{x}) \leq \max(\mathbf{x}) \tag{2}$$

Funkcie patriace do kategórie strednohodnotových funkcií vracajú výstupnú hodnotu, ktorá je väčšia ako najmenšia hodnota vstupnej domény a menšia, ako je jej najväčšia hodnota.

Príklady strednohodnotových funkcií:

1. **Priemery** (aritmetický - $M_W(\mathbf{x})$, geometrický - $G_W(\mathbf{x})$, harmonický- $H_W(\mathbf{x})$, kvadratický $Qw(\mathbf{x})$, vážený)

Vzťah medzi priemermi:

$$H_W(\mathbf{x}) \leq G_W(\mathbf{x}) \leq M_W(\mathbf{x}) \leq Qw(\mathbf{x}) \tag{3}$$

kde $\mathbf{x} = x_1, x_2,..., x_n$ – predstavuje vstupný vektor.

Pri stednohodnotových funkciách taktiež platí dualita, výsledky duálnych funkcií by mali opačné usporiadanie v nerovnici ako v (3).

Aritmetický priemer:

$$M_w(\boldsymbol{x}) = \sum_{i=1}^{n} \left(x_i * \frac{1}{n} \right) \tag{4}$$

Je to jediná logicky neutrálna agregačná funkcia, pretože je sám sebe duálnou funkciou, plne kompenzačná, monotónna, spojitá, symetrická, asociatívna, idempotentná a stabilná pre lineárne transformácie.

Geometrický priemer:

$$G_w(\boldsymbol{x}) = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} \tag{5}$$

Geometrický priemer je konjunktívne polarizovaná funkcia, dáva výsledky menšie ako aritmetický priemer, ale nie je konjunktívna funkcia. Je to spojitá, symetrická funkcia, nemá neutrálny element, má 0 ako anihilátor. Jeho duálna funkcia, dosahuje výsledky medzi aritmetickým priemerom a maximálnou hodnotou vstupného vektora a jej anihilátor je 1.

Kvadratický priemer:

$$G_w(\boldsymbol{x}) = \left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} \tag{6}$$

Kvadratický priemer je disjunktívne polarizovaný, podobne ako duálna funkcia ku geometrickému priemeru, ale nemá anihilátor. Je to striktne rastúca, idempotentná, nemá neutrálny element a spojitá. Výsledok kvadratického priemeru bude vždy nižší ako maximálna hodnota vstupného vektora, pretože neobsahuje anihilátor.

2. Medián

Hodnotu mediánu získame tak, že usporiadame hodnoty vstupnej domény od najmenšej po najväčšiu a vyberieme stredný element. Ak sa v strede nachádzajú dva elementy, vypočítame ich aritmetický priemer. Vlastnosti mediánu zahŕňajú limitujúce hodnoty, monotónnosť, symetriu, idempotenciu a taktiež má kompenzačnú vlastnosť.

3. Quasi-aritmetický priemer

Všetky priemery patria do triedy quasi-aritmetický priemer. Matematická formulácia [1]:

$$M_f(\boldsymbol{x}) = f^{-1}\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right] \tag{7}$$

Je generalizačnou strednohodnotovou funkciou. Zmenou funkcie f (pomocou aditívneho generátora f) môžeme generovať rôzne priemery, pričom funkcia f je striktne monotónna a pokračujúca:

- Keď f (t) = t dostaneme aritmetický priemer,
- Keď f (t) = - log t dostaneme geometrický priemer,
- Keď f (t) = t^{-1} dostaneme harmonický priemer,
- Keď f (t) = t^2 dostaneme kvadratický priemer.

1.4.2 Konjunktívne a disjunktívne agregačné funkcie

Agregačná funkcia sa správa ako konjunktívna, ak pre každé x platí:

$$f(\mathbf{x}) \leq \min(\mathbf{x}) = \min(x_1, x_2, \ldots, x_n) \quad (8)$$

Agregačná funkcie sa správa ako disjunktívna, ak pre každé x platí:

$$f(\mathbf{x}) \geq \max(\mathbf{x}) = \max(x_1, x_2, \ldots, x_n) \quad (9)$$

Pre všetky konjunktívne agregačné funkcie platí, že ich výsledná hodnota je menšia alebo rovnaká ako minimum vstupných hodnôt a naopak pre všetky disjunktívne agregačné funkcie platí, že ich výsledná hodnota bude väčšia alebo rovná ako maximum vstupných hodnôt.

Typickými konjunktívnymi funkciami je skupina triangulárnych noriem (t-normiem) a typickými disjunktívnymi funkciami je skupina triangulárnych konormiem (t-konoriem) a na základe ich vlastností si aj jednotlivé tieto triedy popíšeme.

Súčasnú podobu t-noriem a t-konoriem popísali Schweizer a Sklar [42]. Predstavujú nám zovšeobecnenie dvojhodnotovej boolean logiky na viachodnotovú logiku na intervale [0,1]. T-normy nám zovšeobecňujú *„and"* operátor a t-konormy *„or"* operátor.

T-normy aj t-konormy sú komutatívne, monotónne, asociatívne a majú neutrálny prvok, ktorým sa líšia. T-normy majú neutrálny prvok číslo 1 a t-konormy majú neutrálny prvok 0.

Príklady t-normiem:

1. Minimum: $T_M(\mathbf{x}) = \min(\mathbf{x})$
2. Produkt: $T_P(\mathbf{x}) = \prod_{i=1}^{n} x_i$
3. Lukasiewicz: $T_L(\mathbf{x}) = \max(0, \sum_{i=1}^{n} x_i - (n-1))$
4. Drastický súčin :$T_D(\mathbf{x}) = \begin{cases} x_i & , ak\ x_j = 1\ pre\ \forall j \neq i \\ 0 & , inak \end{cases}$

Príklady t-konoriem:

1. Maximum: $S_M(\mathbf{x}) = \max(\mathbf{x})$
2. Duálny produkt (pravdepodobnostný súčet) : $S_P(\mathbf{x}) = 1 - \prod_{i=1}^{n}(1 - x_i)$
3. Duálny k Lukasiewicz: $S_L(\mathbf{x}) = \min(1, \sum_{i=1}^{n} x_i)$
4. Drastický súčet: $S_D(\mathbf{x}) = \begin{cases} x_i & ,ak\ x_j = 0\ pre\ \forall j \neq i \\ 1 & ,inak \end{cases}$

T-normy a t-konormy sú navzájom duálne funkcie, čo je veľmi dôležité pri ich definovaní a ich použiteľnosti. Navzájom duálne funkcie majú rovnaké vlastnosti. Minimum je najväčšou t-normou a maximum je najmenšou t-konormou a zároveň sú jediné idempotentné funkcie, hraničné funkcie so stredohodnnotovými funkciami (vzorce 2,8,9). Drastický súčin je najmenšou t-normou a drastický súčet je najväčšou t-konormou. V praxi sa veľmi nevyužívajú, ale sú zaujímavé z matematického hľadiska ako funkcie, ktoré ohraničujú t-normy a t-konormy.

1.4.3 Zmiešané agregačné funkcie

V niektorých situáciách vyžadujeme, aby boli posilnené vysoké hodnoty a naopak znižovaný výsledok agregovaním nízkych hodnôt. Samotné t-normy a t-konormy nám neposkytujú tento obojsmerný kompenzačný efekt, napr. jedna slabá hodnota pri použití t-konoriem nám nezníži výsledok a v prípade vysokých a nízkych hodnôt výsledok nie je stredná hodnota. Pri zmiešaných agregačných funkciách sa výsledok mení podľa vstupov. To, ako sa bude funkcia správať, teda aké výsledky nám ponúkne, sa odvíja od vstupných dát.

Trieda zmiešaných agregačných funkcií zahŕňa rôzne agregačné funkcie, medzi ktoré patria uninormy, kompenzačné T-S funkcie, ST-OWA, symetrické súhrny a rôzne generátory funkcií.

Základné vlastnosti zmiešaných agregačných funkcií:

1. Rôzne správanie na jednotlivých intervaloch vstupných dát:
   - $[0,e]^2$ – konjunktívne správanie – vlastnosti ako t-normy
   - $[e,1]^2$ – disjunktívne správanie – vlastnosti ako t-konormy
   - na zostávajúcom intervale [0,e] x [e,1] sa funkcia správa ako stredohonotová agregačná funkcia.
2. Absorbujúci element

3. Dualita
4. Komutatívnosť
5. Idempotencia – iba pre neutrálny element
6. Striktná monotónnosť
7. Kompenzačný efekt

Príklad zmiešaných agregačných funkcií – Uninormy

Uninorma pre dve vstupné hodnoty je funkcia dvoch premenných U : $[0,1]^2 \rightarrow [0,1]$ [8]. Fodor, Yager a Rybanov [17] predstavili skupiny agregačných funkcií, ktoré sa správajú ako konjunktívne pri nízkych vstupných hodnotách, ako disjunktívne pri vysokých vstupných hodnotách a ako strednohodnotové, ak sú vstupné hodnoty nízke aj vysoké. Tieto funkcie obsahujú neutrálny element, ktorý nám tvorí hranicu medzi vysokými a nízkymi hodnotami, ktorý leží na otvorenom intervale]0,1[. Ak sa neutrálny element rovná 1, funkcia má vlastnosti t-noriem a ak 0, tak má vlastnosti t-konoriem [8].

$$f(\boldsymbol{x}) = \frac{\prod_{i=1}^{n} x_i}{\prod_{i=1}^{n} x_i + \prod_{i=1}^{n} 1 - x_i} \tag{10}$$

Vlastnosti funkcie (10) na základe vstupného intervalu:

- konjunktúrna na intervale $\left[0; \frac{1}{2}\right]^n$
- disjunktívna na intervale $\left[\frac{1}{2}; 1\right]^n$
- strednohodnotá na zostávajúcom intervale.

Funkcia je asociatívna, symetrická s neutrálnym elementom $\frac{1}{2}$ a nespojitá na hraniciach intervalu $[0; 1]^n$. V praxi však môže byť problematický pevne stanovaný neutrálny element bez možnosti prispôsobenia.

Existujú aj rôzne triedy generalizovaných agregačných funkcií - napríklad:

$$F(x, y) = x^a * y^b \tag{11}$$

- KEĎ a = b = 1, POTOM je funkcia súčinová t-norma,
- KEĎ a = b = 1/2, POTOM geometrický priemer
- KEĎ a < 1 A b < 1, a + b = 1, POTOM váhový geometrický priemer
- KEĎ a > 1 A b > 1, POTOM konjunktívna funkcia, ktorá nie je t-norma. Výsledok je medzi 0 a min ale neplatia všetky axiómy pre t-normy.

Existuje veľké množstvo agregačných funkcií. Sú kategorizované do štyroch skupín, v rámci ktorých sú rôzne podskupiny, ako napríklad, trojuholníkové normy, konormy, priemery, Choquet a Sugeno integrály.

Otázkou je, ako vybrať vhodnú agregačnú funkciu pre špecifickú aplikáciu. Je jedna agregačná funkcia dostatočná alebo by sme mali použiť rôzne agregačné funkcie na rôznych častiach aplikácie.

Ďalšími vlastnosťami, ktoré treba brať do úvahy pri vyberaní vhodnej agregačnej funkcie je jednoduchosť, efektívnosť, jednoduchosť vysvetlenia, atď. Neexistujú žiadne všeobecné pravidlá výberu a je to na tvorcovi systému. V nasledujúcom vyjadrení sa sústredíme na prvé dve kritéria: byť v súlade s významovo dôležitými vlastnosťami agregačných funkcií a dosahovanie chceného výstupu z požadovaných vstupov.

Formálne zapísaný problém výberu: Spokojnosť blížiaca sa k rovnosti $f(\mathbf{x}_k) \approx y_k$ sa zvyčajne premieňa na minimalizačný problém.

Min//r// - funkcia f spĺňa *P*1,*P*2,

//r// - odchýlka

r – *vektor odlišností medzi čakanou hodnotou a vypočítanou hodnotou*

P1, P2 - vlastnosti

Formálne sme popísali jeden z mnoho existujúcich spôsobov ako optimalizovať výber funkcie. Z uvedeného vyplýva, že úloha klasifikácie môže byť riešená množstvom odlišných algoritmov a funkcií. Rôzne výsledky totiž dosiahneme kombináciou modelov s odlišnou štruktúrou, odlišným spôsobom definovania parametrov, rôznymi hodnotiacimi funkciami alebo špecifickou prácou s dátami. V rámci prehľadnosti a zrozumiteľnosti potom výsledný algoritmus najčastejšie identifikujeme práve pomocou jednoznačne definovaných komponentov. Pri riešení konkrétnej úlohy klasifikácie nie je únosné testovať všetky existujúce spôsoby. Škálu vhodných algoritmov a funkcií najčastejšie obmedzujeme na základe požiadaviek doménového experta. To môže súčasne naznačiť komponenty vhodné na riešenie úlohy. V ďalšom kroku si vysvetlíme tvorbu ordinálnych súčtov a ich aplikáciu na rôzne modelové príklady.

2 Cieľ práce, metodika práce a metódy skúmania

Matematické modely nám už od nepamäti slúžili na opis prírodných javov. Matematický model je zovšeobecnenie reálneho javu a slúži nám ako šablóna pre podobné prípady. Problémom je veľká zložitosť reality, preto pri teoretickom opise veľa faktorov neberieme do úvahy, čo nám zabezpečuje jednoduchšie pochopenie reality a následnú aplikáciu. V našej práci sa však snažíme čo najbližšie priblížiť ľudskému rozhodovaniu, aby sa človek mohol spoľahnúť na systémy v akejkoľvek oblasti.

Našim cieľom je spracovať požiadavky používateľov pomocou matematického modelu a vytvoriť tak klasifikačný model transparentný pre každého používateľa. Téma klasifikácie je dnes horúcou témou hlavne kvôli možnostiam využitia umelej inteligencie a jej aplikácie, strojové učenie. Avšak tieto systémy sú technicky náročné a pre koncového používateľa mnohokrát nepochopiteľné. Preto našim hlavným cieľom je vytvorenie klasifikačného modelu, ktorý bude čo naviac zobrazovať realitu a zároveň bude jednoznačne vysvetliteľný a interpretovateľný.

Popri hlavnom cieli sme si taktiež definovali súvisiace čiastkové ciele:

- spravodlivá klasifikácia objektov,
- spracovanie požiadaviek definovaných lingvistickými výrazmi pomocou agregačných funkcií,
- vytvorenie klasifikačného modelu pomocou ordinálnych súčtov,
- teoretický návrh klasifikačného modelu na spôsobe interaktívneho strojového učenia.

Pri plnení nášho cieľa využijeme poznatky z oblasti agregačných funkcií a ordinálnych súčtov. V našej práci chceme poukázať na zložitosť existujúcich systémov a tým aj ich ťažkú aplikovateľnosť v kritických odvetviach. Vysvetliteľnosť systémov sa stáva dôležitou témou v oblasti technológií a hlavne umelej inteligencie a strojového učenia, kde časť kontroly nechávame na stroji. Jedným zo 6 priorít Komisie Európskej Únie na obdobie 2019-2024 je pripravenosť Európy na digitálny vek, ktoré zahŕňa aj opatrenia k UI, a to podporu excelentnosti a budovanie dôvery v oblasti UI [16]. Naša práca sa zameriava na inovatívnu klasifikáciu, ktorá sa ukázala ako perspektívna podpora pre UI. Výstupom našej práce chceme dosiahnuť parametrizovatelný model, ktorý môžeme meniť na základe požiadaviek zákazníka. V praktickej časti experimentujeme s dátami, aby sme porovnali výstupy jednotlivých modelov, interpretovali ich správanie a dosiahnuté výsledky.

Na naplnenie našich cieľov využijeme nasledujúce metódy:

- metóda analýzy – celú problematiku sme si rozdelili na menšie časti, ktoré sme podrobne skúmali. Skúmali sme rôzne zdroje, venovali sme sa jednotlivým klasifikačným metódam, ako aj ich aplikovaniu v UI. Matematicky sme si opísali základné agregačné funkcie a v ďalšej časti si ukážeme ich spájanie na ordinálne súčty a tak vytovrenie nového klasifikačného modelu. Skúmali sme charakteristické znaky jednotlivých faktorov, vzťahy medzi poznávanými faktami a snažili sa nájsť pričinné súvislosti a kauzálne zákonitosti,
- metóda komparácie – pri porovnaní výsledkov z rôznych klasifikačných modelov, kedy využijeme agregačné funckie rôznych vlastností,
- metóda interpretácie – pri použití grafov a tabuliek na zobrazenie získaných výsledkov,
- metóda dedukcie – pri opise výsledkov dosiahnutých rôznymi aplikovanými modelmi,
- metóda syntézy – pri zovšeobecnení výsledkov z konkrétneho príkladu klasifikácie,
- testovanie – overili sme si očakávané správanie funkcií ich aplikovaním na príklady.

Na vytvorenie modelu sme použili programovací jazyk python. Prostredie, v ktorom sme model tvorili, sa nazýva Jupyter notebook. Je to open-source webová aplikácia, ktorá upožňuje vytvorenie a zdielanie zdrojových kódov, dokumentácií, vizulácií, rovníc a textu. Je taktiež vhodný na vytváranie numerických simulácií, štatistických modelov, na dátové vizualizície, strojové učenie a oveľa viac.

Dáta použité na experimentovanie s klasifikáciou sme získali z verejne dostupnej databázy, the UCI Machine Learning Repository[18], ktorá obsahuje kolekciu rôznych setov dát, ktoré ponúka pre komunity na empirické analýzy algoritmov strojového učenia. V našej práci sme konkrétne využili dátový set, South German Credit Data Set, dáta Nemeckej banky o klientoch, ktorí si podali žiadosť o úver. Dáta sú stručne popísané, avšak pre naše podmienky doplníme dáta o vymyslené modelové situácie a budeme pozorovať, ako sa zmení výsledok.

3 Výsledky práce

Výsledky práce, ako aj konkrétny postup, ktorým sme ich dosiahli, sú opísané v nasledujúcej časti. Najskôr si predstavíme ordinálne súčty, teoreticky popíšeme ich tvorbu a uvedieme si konkrétne príklady klasifikácie pri použití rôznych agregačných funkcií. Na príkladoch si vysvetlíme ich vlastnosti a ako ich môžeme aplikovať do reálnych situácií. Potom si zoberieme konkrétnu sadu dát, uvedieme si modelové príklady a aplikujeme na nich náš vytvorený model. Na záver si interpretujeme naše výsledky a navrhneme možné rozšírenie klasifikačného modelu o nové vlastnosti.

3.1 Predpoklady klasifikácie pomocou zmiešaných agregačných funkcií

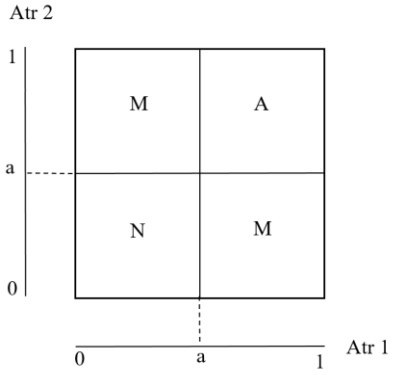
Ako sme si v teoretickej časti uviedli, na klasifikáciu údajov existuje mnoho metód, predstavili sme si ich výhody aj nevýhody. Zamerali sme sa hlavne na v súčasnosti najznámejšie metódy klasifikácie, ako sú pravidlové systémy a neurónové siete. Neurónové siete ako klasifikačný nástroj v UI nám prinášajú dobré výsledky, ale nevieme vysvetliť, ako sme sa k výsledkom dostali. Naopak pravidlové systémy nám poskytujú jasné dôvody, ale ich komplexnosťou sa ich presnosť znižuje. V našej práci vytvoríme preto klasifikačný model využitím ordinálnych súčtov, ktoré sú jasne definované a preto dosiahnutie výsledkov je vysvetliteľné na základe vlastností funkcií a taktiež nám otvárajú možnosť využiť rôzne agregačné funkcie a tým pokryť rôzne situácie vyžadujúce klasifikáciu.

V našej práci sa zaoberáme klasifikáciou, ktorá nám rozdeľuje vstupné dáta na tie, ktoré spĺňajú naše podmienky vyjadrené atribútmi a tie, ktoré nespĺňajú. Výsledný model klasifikácie nám však nedáva striktné rozdelenie vstupných dát, ale ich príslušnosť do kategórie, čo nám zabezpečí zachytenie vágnosti.

Predpokladom, aby sme vstupné dáta mohli klasifikovať pomocou ordinálnych súčtov je splnenie vlastností agregačných funkcií (1), ktoré sme uviedli v kapitole 1.4. Nevyhnutnou podmienkou je, aby sme vstupné údaje transformovali na interval [0,1]. V našej práci sme využili lineárnu transformáciu:

$$x = \begin{cases} 0 & pre\ atr1\ \leq minimum \\ \frac{atr1 - D_L}{D_H - D_L} & pre\ atr1\ \in (D_L, D_H) \\ 1 & pre\ atr1\ \geq maximum \end{cases} \tag{12}$$

Klasifikáciu pomocou dvoch atribútov môžeme vidieť na obrázku 4. V grafe môžeme vidieť pomenovanie tried na *N* (nie), ktorá požaduje vyjadrenie konjuktívnymi funkciami, trieda *A* (áno), vyjadrená disjunktívnymi funkciami a trieda *M* (možno), vyjadrená strednohodnotovými funkciami.



Obrázok 4 Klasifikačný priestor trojhodnotovej logiky pre dva atribúty
Zdroj : Vlastné spracovanie

Využitie ordinálnych súčtov nám zabezpečí, aby vysoké hodnoty posilňovali naše výsledky ku kategórii *A* a naopak, nízke stupne príslušnosti znižovali výsledok a približovali sa ku *N*. Zmiešané vysoké aj nízke hodnoty nám spadajú do kategórie možno *(M)* [9].

Ak by sme využili t-normy (t-konormy) na celom intervale, dosiahli by sme nežiadúce výsledky. Na ilustratívnom príklade si ukážeme prečo a ako môžeme využitím ordinálnych súčtov týmto nevýhodám predísť. V príklade sa budeme sústrediť iba na disjunktívne agregačné funkcie, ale vďaka dualite môžeme rovnakou logikou zhodnotiť aj konjunktívne funkcie. Ako sme si už aj v prvej kapitole pri opise vlastností disjunktívnych funkcií spomenuli, výsledok po aplikovaní disjunktívnej agregačnej funkcie je väčší ako maximum vstupných hodnôt. Inak môžeme povedať, že výsledok je zvýšený vďaka použitiu disjunktívnej funkcie. V niektorých prípadoch však takýto jav zvýšenia výsledku nie je žiadaný. Ako príklad si môžeme uviesť klasické odporúčacie systémy, ktoré sa v súčasnosti masovo využívajú. Odporúčacie systémy navrhujú pre používateľa podobné produkty podľa jeho predchádzajúcich záujmov. Výber vhodných produktov je realizovaný na základe niekoľkých vopred daných kritérií. Matematicky môžeme danú situáciu zapísať nasledovne: Máme vstupujúci vektor **x**, ktorý nám predstavuje stupne príslušnosti jednotlivých alternatív i-tej vlastnosti. Našim cieľom je agregovať stupne príslušnosti a na základe výsledku zhodnotiť, na koľko je daný produkt relevantný pre konkrétneho zákazníka. Čím viac vlastnosti spĺňa daný prvok, tým je relevantnejší. Produkty sú potom zoradené a zobrazené

zákazníkovi. Na základe opísanej situácie môžeme povedať, že vlastnosti funkcie, ktoré pri agregácií jednotlivých stupňov príslušnosti vyžadujeme, sú nasledovné:

- Funkcia musí byť kontinuálna – musí zohľadniť, ak vyššia príslušnosť, potom vyššia výsledná hodnota
- Funkcia má neutrálny element 0 - ak jednu z vlastností nespĺňa produkt vôbec, neznamená, že je menej relevantný, ale nemení nám výsledok
- Funkcia je symetrická

Na základe týchto predpokladov môžeme povedať, že spomínané vlastnosti nám spĺňajú disjunktívne funkcie. Avšak ukážme si konkrétny výpočet:

- Pre Lukasiewiczovu t-konormu:

Vstupný vektor: x = (0,1;0,1;0,1;..;0,1) pre n = 10

$$S_L(x) = min(1, \sum_{i=1}^{10} 0{,}1) = 1$$

Ako vidíme, výsledok je 1, a teda produkt s daným vektorom príslušností by sme zákazníkovi odporučili ako najlepšiu možnosť.

- Pre pravdepodobnostný súčet (duálna funkcia k súčinovej t-norme)

Vstupný vektor: x = (0,3;0,3;..0,3) pre n = 10

$$S_P(x) = 1 - \prod_{i=1}^{10} (1 - 0{,}3) = 0{,}97$$

Z výsledku taktiež môžeme vidieť, že produkt by bol vysoko odporúčaný. Na príkladoch pozorujeme, že aj pri veľmi nízkej splniteľnosti atribútov, sa nám výsledná hodnota po agregácií zvyšuje. V našom príklade nám táto vlastnosť vytvára nežiadúci efekt. Ak by jeden produkt spĺňal vlastnosti s väčšou intenzitou, ale napríklad len v jednom alebo dvoch atribútoch, produkt by náš systém neodporučil alebo by ho odporučil s menšou váhou ako produkt, ktorý spĺňa vlastností síce viac, ale s malou intenzitou. Pričom zákazník kúpi radšej produkt s výraznejšou jednou alebo dvomi vlastnosťami, ako produkt, ktorý dobre nespĺňa ani jednu vlastnosť. Tento jav môžeme vidieť vo všetkých nami doposiaľ spomínaných funkciách. Jedinou výnimkou je funkcia maximum pre t-konormy (minimum pre t-normy). Avšak maximum (minimum) nám neposkytuje žiadne ovplyvnenie výslednej hodnoty vstupnými. Výsledná je len tá, ktorá v jednej vlastnosti je najvýraznejšia, čo tiež nie je postačujúce.

Preto v našom konkrétnom príklade, ale aj iných a podobných situáciách sa vyžíva spôsob posilnenie intenzity príslušnosti na správnom mieste (noble reinforcement) [8] [7]

[46], kedy vysoké hodnoty posilňujú náš výsledok, pričom nízke hodnoty ho znižujú. Na základe zistení sme si preto zadefinovali hranicu medzi vysokými a nízkymi hodnotami z intervalu]0,1[. Na obrázku 4 vidíme túto hranicu ako parameter a. Avšak teraz sa otvára diskusia, ako najvhodnejšie ju určíme. Na hranicu vplýva mnoho faktorov, ako napríklad, čo považujeme za vysoký stupeň príslušnosti, či skôr produkty nadobúdajú vysoké alebo nízke hodnoty, aké je ich variabilita a pod. Na zjednodušenie sme na našich príkladoch túto hranicu vyjadrili ako presné číslo. Pri komplexnejších klasifikáciách sa môže hranica vyjadriť pomocou fuzzy čísla alebo hranice medzi skupinami ako fuzzy množiny. Obrázok 4 zobrazuje rozdelenie klasifikačného priestoru ostrými hranicami, kde hranicu tvorí parameter a, ktorý je súčasne neutrálny element. Matematicky vyjadrené, vstupný interval [0,1] si rozdelíme na subintervaly [0,a] a [a,1], na ktorých sem aplikovali funkcie s rôznym správaním. Na to nám slúžia ordinálne súčty a gama operátor.

3.2 Klasifikácia pomocou gama operátora

Gama operátor je operátor, ktorý spája správanie t-noriem a t-konoriem. Má kompenzačný charakter a teda svojím správaním je porovnateľný aj ľudskému rozhodovaniu. Gama operátor je exponenciálnou kombináciou dvoch konkrétnych funkcií – súčinovej t-normy a jej duálnej funkcie, t-konormy pravdepodobnostný súčet [8]. Gama operátor bol predstavený v publikácií od autorov Zimmermann and Zysno [46] ako konkrétny príklad aplikovania dvojice duálnych funkcií t-normy a t-konormy. Neskôr sa operátor generalizoval na T-S operátory, kde môžeme dosadiť aj inú dvojicu duálnych funkcií [8] [39].

Všeobecný matematický zápis je nasledovný:

$$\gamma_A(x,y) = \left(\prod_{i=1}^{n} x_i\right)^{(1-\gamma)} * \left(1 - \prod_{i=1}^{n}(1-x_i)\right)^{\gamma} \tag{13}$$

Za podmienok:

- $\gamma \in [0,1]$
- n je dĺžka vstupného vektora **x**.

Iný tvar gama operátora [25]:

$$\gamma_A(x,y) = (1-\gamma)\prod_{i=1}^{n} x_i + \gamma\left(1 - \prod_{i=1}^{n}(1-x_i)\right) \tag{14}$$

kde γ predstavuje neutrálny element.

Pri výpočte pomocou gama operátorov nám stačí uviesť iba jeden parameter, gamu, na obrázku 4 je zobrazený ako a. Hodnotu gamy získame od experta alebo môžeme určiť pomocou strojového učenia zo vstupných dát. Ak ale v konkrétnom príklade, kedy máme všetky vstupné hodnoty rovné 0,5 a chceme, aby aj výsledná hodnota sa rovnala 0,5, parameter gama sa musí rovnať 0,5. Rovnako, ak by sme mali rovnomerné rozdelenie vstupných údajov alebo by sme vyžadovali, aby všetky parametre mali rovnaký vplyv na výsledok, aj v takom prípade sa musí parameter gama rovnať 0,5. Za týchto podmienok stratíme flexibilitu v určení parametra gama a teda aj flexibilitu výpočtu.

Ďalšími vlastnosťami, ktoré gama operátor nenadobúda je idempotentnosť. Tu zostáva otvorená otázka, či ju vyžadujeme alebo nie.

3.3 Klasifikácia pomocou ordinálnych súčtov

Ordinálne súčty sú silným nástrojom, ktorý nám umožňuje vytvoriť nové t-normy/ t-konormy zo širokej škály už existujúcich. Najskôr si všeobecne popíšeme ako sa ordinálne súčty vytvárajú a potom si ukážeme konkrétne príklady.

Tvorbu, ako aj príklady si ukážeme použitím iba dvoch parametrov, pre jednoduchšiu ilustráciu grafov, ako aj ich výsledkov.

Doménu vstupných hodnôt pre agregačné funkcie definujeme ako jednotkový štvorec $[0,1]^2$, ako vidíme na obrázku 5. Na diagonále tohto štvorca definujeme menšie štvorce, subintervaly $[a_i, b_i]^2$, ktoré budú definované vlastnou funkciou, t-normou. Zvyšný priestor definujeme pomocou t-normy, minimum [8]. Výsledná funkcia je taktiež t–normou, ktorú voláme ordinálny súčet.

Definícia: nech $(T_i)_{i=1...K}$ je rodina t-noriem a $]a_i, b_i[^2$ je rodina neprázdnych, navzájom nezávislých otvorených subintervalov z [0,1]. Funkcia $T : [0,1]^2 \rightarrow [0,1]$ definovaná

$$T(x,y) = \begin{cases} a_i + (b_i - a_i) * T_i\left(\frac{x-a_i}{b_i-a_i}, \frac{y-a_i}{b_i-a_i}\right) & , (x,y) \in]a_i - b_i[^2 \\ min(x,y) & , inak \end{cases} \qquad (15)$$

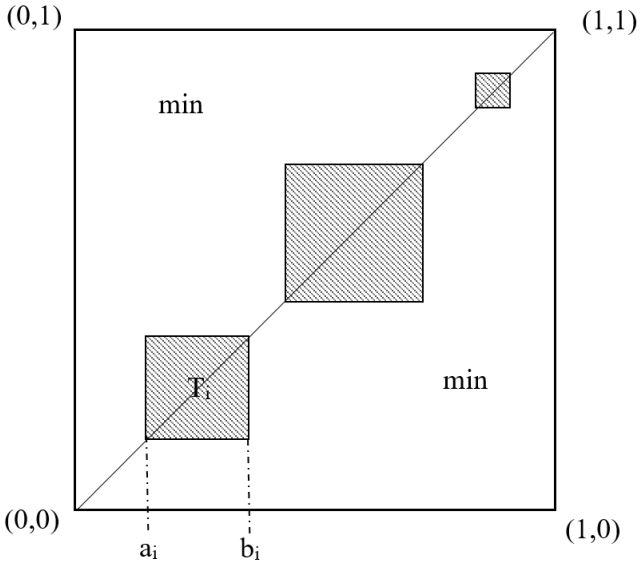
je potom t-norma nazývaná ordinálny súčet sčítancov $< a_i, b_i, T_i >$, $i = 1, ..., K$.

Ordinálne súčty predstavujú nadstavbu pre t-normy a t-konormy. Predstavujú nové funkcie, ktoré sa skladajú z kombinácie už existujúcich. Každá t-norma je triviálny ordinálny súčet iba s jedným sčítancom.

Zovšeobecnenie ordinálnych súčtov sme si popísali iba na t-normách, avšak pre duálnosť funkcií ordinálne súčty s využitím t-konorm sa vypočítajú analogicky.

Vďaka rôznorodosti funkcií nám ordinálne súčty ponúkajú cestu, ako si vytvoriť vlastné funkcie správajúce sa podľa našich požadovaných vlastností.

Ako sme spomínali, najskôr si analyzujeme rôzne možnosti ordinálnych súčtov, čo nám prinášajú, aké vlastnosti, na základe čoho budeme vedieť vybrať vhodnú funkciu pre konkrétne definované požiadavky. Na ilustráciu si uvedieme konkrétne príklady klasifikácie. Na demonštráciu využijeme vždy rovnaké dáta, aby sme výsledky ľahko mohli porovnať.



Obrázok 5 Vytvorenie ordinálneho súčtu pomocou t-noriem

Zdroj : Vlastné spracovanie

De Beats a Mesiar predstavili iný tvar ordinálneho súčtu [35]:

a) $B_1, B_2 : [0,1]^2 \rightarrow [0,1]$

$$A(x,y) = a \,.\, B_1\left(1 \wedge \frac{x}{a}, 1 \wedge \frac{y}{a}\right) + \ (1 - a).\ B_2(0 \vee \frac{x-a}{1-a}\ , 0 \vee \frac{y-a}{1-a}) \qquad (16)$$

b) $A_1 : [0,a]^2 \rightarrow [0,a]$, $A_2 : [a,1]^2 \rightarrow [a,1]$

$$A(x,y) = A_1\ (a \wedge x, a \wedge y\ ) + \ A_2\ (a \vee x, a \vee y) - a \qquad (17)$$

V našich ďalších výpočtoch sme pracovali s tvarom ordinálneho súčtu (17). Pre zjednodušenie rátame iba s dvomi parametrami x a y, parameter a nám predstavuje neutrálny element a hraničné hodnoty sú 0 a 1. V (16) nám B_1 predstavuje konjunktívnu funkciu a B_2

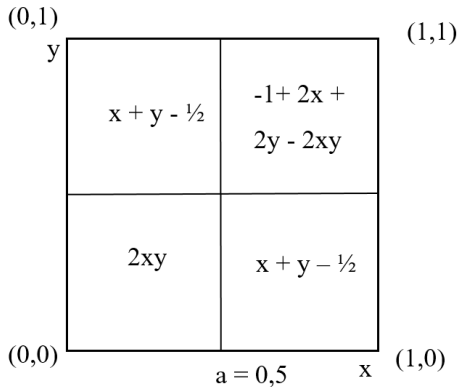
disjunktívnu, rovnako ako A v (17) je konjunktívna na intervale $[0, a]^2$ a disjunktívna na intervale $[a,1]^2$.

3.4 Rôzne funkcie dosadené do ordinálnych súčtov

V nasledujúcej časti si ukážeme správanie ordinálnych súčtov pri použití rôznych funkcií. Na demonštráciu použijeme funkcie s rôznymi vlastnosťami, aby sme poukázali na variabilitu dosiahnutých výsledkov. Na zachovanie požiadaviek z obrázka 4, pre triedu N sme uvažovali s konjunktívnou funkciou, pre triedu A s disjunktívnou funkciou a pre triedu M so stredonohodnotovou funkciou. Na základe jednotlivých vlastností môžeme experimentálne určiť najvhodnejšiu funkciu na klasifikáciu údajov pri konkrétnych príkladoch. Z exaktne definovaných funkcií vieme zdôvodniť správanie modelu a teda vysvetliť klasifikáciu.

3.4.1 Produkt a pravdepodobnostný súčet

Vlastnosti ordinálneho súčtu vyjadreného striktnými funkciami produkt a pravdepodobnostný súčet si zobrazíme graficky na obrázku 6.



Obrázok 6 Grafická interpretácia ordinálneho súčtu pre t-normu produkt, t-konormu pravdepodobnostný súčet a aritmetický priemer

Zdroj : Vlastné spracovanie

Na začiatku si overíme splniteľnosť podmienok agregačných funkcií a následne si vysvetlíme funkcie použité na výpočet agregácie.

Kvôli splniteľnosti podmienok (1) agregačných funkcií, naše hraničné hodnoty sú 0, 1, pozri obrázok 6. Ak je jedna z hodnôt 0 a druhá je menšia ako neutrálny element, tak výsledok je 0. Rovnaká zákonitosť platí pre výsledok 1. Ako neutrálny element sme si

v našom príklade zvolili hodnotu 0,5. Hodnota parametra sa môže meniť podľa vstupných dát a očakávaných výsledkov, ale pre naše ilustratívne príklady predpokladáme rovnomerné rozdelenie vstupných hodnôt na intervale [0,1] a taktiež chceme zachovať rovnakú proporcionalitu medzi nízkymi a vysokými hodnotami, preto $a = 0,5$.

Ako sme už spomenuli, na výpočet konjunktívneho správania použijeme súčinovú t-normu a disjunktívneho jeho duálnu funkciu. Avšak, aby sme dosiahli správne výsledky pre stredné hodnoty, musíme vzorce upraviť. Teda, aby pri vstupnom vektore x(0,5,..0,5) vyjadrujúcom splniteľnosť parametrov, bola výsledná hodnota taktiež 0,5, musíme pôvodné konjunktívne a disjunktívne funkcie správne matematicky zmeniť. Pôvodné funkcie, ako sme si v prvej kapitole uviedli, majú charakter znižovania (zvyšovania pri disjunktívnych funkciách) výstupu, čo nie je pri neutrálnom elemente správne. Neutrálny element nám nemôže zmeniť výsledok. Ak by sme zachovali pôvodnú funkciu súčinovej t-normy, výsledok bude menší ako neutrálny element. Pozrime si spomínanú situáciu na príklade:

Súčinová t-norma: $T_P(x, y) = x * y$

$T_P(0,5; 0,5) = 0,25$

Upravený vzorec pre súčinovú t-normu: $A_1(x, y) = 2xy$ (18)

$A_1(0,5; 0,5) = 0,25 * 2 = 0,5$

Pravdepodobnostný súčet: $S_P(x, y) = x + y – x * y$

$S_P(0,5; 0,5) = 0,75$

Upravený vzorec pre pravdepodobnostný súčet: $A_2(x, y) = -1 + 2x + 2y - 2xy$ (19)

$A_2(0,5; 0,5) = 0,5$

Strednohodnotovú funkcia si vyjadríme pomocou aritmetického priemeru. Aritmetický priemer je jediná hodnotovo neutrálna funkcia. Výpočet:

$A_M(x, y) = A_1(x, a) + A_2(a, y) – a$, kde $a = ½$

$A_M(x, y) = A_1(x, ½) + A_2(½, y) – ½ = x + y - ½$ (20)

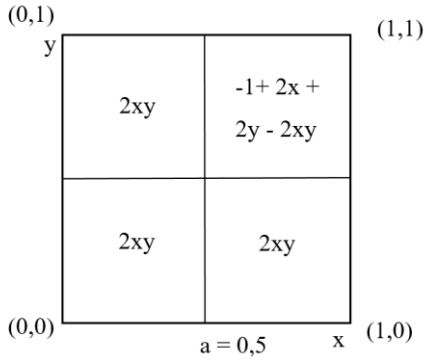
Tabuľka 1 Výsledok klasifikácie, keď a = 0,5, A_1 je t-norma produkt a A_2 je t-konorma pravdepodobnostný súčet, stednohodnotová funkcia je vyjadrená aritmetickým priemerom

ENTITA	ATR1	ATR2	X	y	VÝSLEDOK
E1	20	50	0,2	0,5	0,2
E2	20	20	0,2	0,2	0,08
E3	20	40	0,2	0,4	0,16
E4	30	90	0,3	0,9	0,7
E5	80	20	0,8	0,2	0,5
E6	80	80	0,8	0,8	0,92
E7	80	90	0,8	0,9	0,96
E8	100	0	1	0	0,5
E9	100	100	1	1	1
E10	0	0	0	0	0
E11	60	10	0,6	0,1	0,2
E12	50	50	0,5	0,5	0,5

Zdroj : Vlastné spracovanie

Na všetkých príkladoch z tabuľky 1 môžeme vidieť kompenzačné vlastnosti použitých funkcií. Príklady E1 – E3 nám zobrazujú konjunktívne správanie, E4, E5 strednohodnotové a E6, E7 disjunktívne správanie. Na príkladoch E5, E8 a E12 vidíme, že strednohodnotová funkcia sa správa ako aritmetický priemer, pretože obe hodnoty rovnako vplývajú na výsledok. Avšak pri výsledkoch E4 a E11 pozorujeme, že výsledok nie je aritmetický priemer, ale v E4 vysoká hodnota zvyšuje výsledok a naopak v E11 nízka hodnota znižuje výsledok.

Alternatívou pre tento typ ordinálneho súčtu môže byť zmena strednohodnotovej funkcie, napríklad na geometrický priemer. Stále nám zostávajú predpoklady, ako aj v predchádzajúcom príklade, máme 2 vstupné parametre, neutrálny element je ½, t-norma aj t-konotma zostáva rovnaká. Preto sa nám výsledky menia iba pri vstupných hodnotách, kedy je jedna hodnota nízka a jedna vysoká. Grafickú interpretáciu môžeme vidieť na obrázku 7.



Obrázok 7 Grafická interpretácia ordinálneho súčtu pre t-normu produkt, t-konormu pravdepoodbnostný súčet a geometrický priemer

Zdroj : Vlastné spracovanie

Výpočet pre strednohodnotovú funkciu – geometrický priemer:

Vzorec pre geometrický priemer dostaneme aplikovaním quasi-artimetického priemeru ako strednohodnotovú funkciu [25], podkapitola 1.4.1.

Vyjadrenie strednohodnotovej funkcie pre oridinálny súčet:

$$A_M(x, y) = A_1(x, a) + A_2(a, y) - a \text{ , kde } a = ½ \quad (21)$$

Vyjadrenie strednohodnotovej funkcie pomocu quasi-aritmetického priemeru pre ordinálny súčet:

$$A_Q(x,y) = g^{-1}\left(g\left(A_1\left(x,\frac{1}{2}\right)\right) + g\left(A_2\left(\frac{1}{2},y\right)\right) - g\left(\frac{1}{2}\right)\right) \quad (22)$$

Keď g(t)= -log t, potom dostaneme geometrický priemer.

$$A_G(x,y) = \frac{A_1\left(x,\frac{1}{2}\right).A_2\left(\frac{1}{2},y\right)}{\frac{1}{2}} = 2xy \quad (23)$$

Výsledky z tabuľky 2 oproti tabuľke 1 sa menia iba v prípade, ak jeden zo vstupných atribútov je väčší ako neutrálny element a jeden menší. Hlavný rozdiel môžeme vidieť v príklade E8, kde očakávame strednohodnotevé správanie, ale výsledok je 0. Je to preto, lebo 0 je anihilátor pre geometrický priemer a naopak 1 je anihilátor pre jeho duálnu funkciu. Na základe našich výpočtov môžeme povedať, ak by sme vyžadovali v klasifikácií plnú kompenzáciu pri jednoznačnom splnení alebo nesplnení podmienky, použitím geometrického priemeru vieme túto skutočnosť zachytiť.

Tabuľka 2 Výsledok klasifikácie, keď a = 0, A_1 je produkt, A_2 je pravdepodobnostný súčet a ako strednohodnotová funkcia je použitý aritmetický priemer

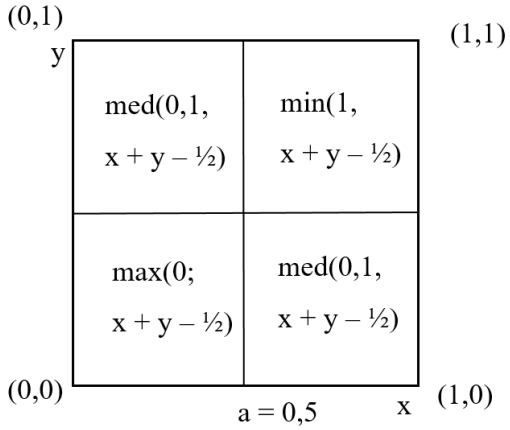
ENTITA	ATR1	ATR2	X	y	výsledok
E1	20	50	0,2	0,5	0,2
E2	20	20	0,2	0,2	0,08
E3	20	40	0,2	0,4	0,16
E4	30	90	0,3	0,9	0,48
E5	80	20	0,8	0,2	0,32
E6	80	80	0,8	0,8	0,8
E7	80	90	0,8	0,9	0,9
E8	100	0	1	0	0
E9	100	100	1	1	1
E10	0	0	0	0	0
E11	60	10	0,6	0,1	0,1
E12	50	50	0,5	0,5	0,5

Zdroj : Vlastné spracovanie

Na výpočet správania „možno“ môžeme použiť ktorúkoľvek strednohodnotovú funkciu. Niektoré z nich, ako aj ich vlastnosti, sme opísali v kapitole 1.4.1. Jednou z nich, ktorú sme si aj uviedli, je Quasi-aritmetický priemer, ktorá nám predstavuje rodinu priemerov. Tento tvar nám prináša väčšiu flexibilitu pri tvorbe ordinálneho súčtu [8]. Vďaka jej vlastnostiam môžeme funkciu priemer generovať podľa zmeny aditívneho generátora g a tak nám prináša väčšiu flexibilitu pri určovaní strednohodnotovej funkcie.

3.4.2 Lukasiewiczova t-norma a t-konorma

Ako ďalšiu obmenu vo funkcií ordinálneho súčtu zmeníme disjunktívnu a konjunktívnu funkciu. Nilpotentné funkcie, Lukasiewiczova t-norma a t-konorma nám prinášajú ďalšie zaujímavé vlastnosti. V prechádzajúcom príklade, ak sme mali vo vektore vstupných hodnôt všetky nízke hodnoty, výsledná hodnota bola nízka, ale stále nie 0. Pri Lukasiewiczovej t-norme môže byť výsledok 0. Ak vstupný vektor obsahuje iba nízke hodnoty, pre používateľa to znamená , že táto možnosť nie je atraktívna. Rovnako to platí aj pri duálnej funkcií, keď vstupný vektor obsahuje vysoké hodnoty, výsledok agregácie nám dá 1. Podobne ako v predchádzajúcom príklade pôvodné funkcie musíme upraviť a ako neutrálny element taktiež rátame s hodnotou 0,5. Použité funkcie sú zobrazené na obrázku 8 a ich správanie si vysvetlíme na dátach v tabuľke 3.



Obrázok 8 Grafická interpretácia ordinálneho súčtu pre Lukasiewicovu t-normu, Lukaziewiczovu t-konormu a aritmetický priemer

Zdroj : Vlastné spracovanie

- Použité funkcie:

Lukasiewiczova t-norma: $T_L(x , y) = \max(0 , x+ y - 1)$

$A_1(x, y) = \max(0 , x+ y – ½)$ (24)

Lukasiewiczova t-konorma: $S_L(x , y) = \min(1 , x + y)$

$A_2(x, y) = \min(1 , x + y – ½)$ (25)

Strednohodnotová funkcia: $A_M(x, y) = \text{med}(0 , 1 , x + y – ½)$ (26)

Na príklade E6, E7 z tabuľky 3 vidíme, že vysoké vstupujúce hodnoty nám dajú výsledok 1 a naopak v príklade E2 výrazne nízke hodnoty nám dajú výsledok 0. Taktiež môžeme vidieť proporcionalitu medzi ostatnými hodnotami. Výsledky v príkladoch E4,E5,E8,E11,E12 sa zhodujú s výsledkami v tabuľke 1. V oboch príkladoch sme využili aritmetický priemer ako strednohodnotovú funkciu, preto sa nám aj výsledky zhodujú. Taktiež môžeme použiť aj iné strednohodnotové funkcie, ako napríklad geometrický priemer, ktorý nám poskytuje rovnaké vlastnosti, ako sme si si uviedli v tabuľke 2.

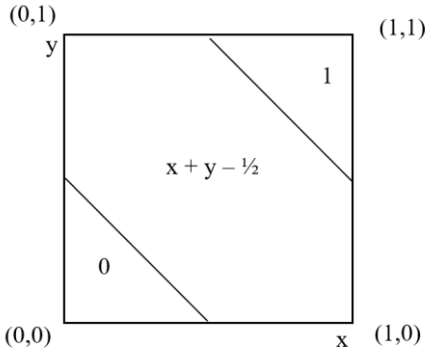
Z grafu môžeme vidieť, že zápis správania na jednotlivých intervaloch môžeme zaznačiť aj jednou funkciou. Pozri obrázok 9.

Tabuľka 3 Výsledok klasifikácie, keď a = 0,5, A_1 je Lukasiewiczova t-norma a A_2 je Lukasiewiczova t-konorma

ENTITA	ATR1	ATR2	X	y	VÝSLEDOK
E1	20	50	0,2	0,5	0,2
E2	20	20	0,2	0,2	0
E3	20	40	0,2	0,4	0,1
E4	30	90	0,3	0,9	0,7
E5	80	20	0,8	0,2	0,5
E6	80	80	0,8	0,8	1
E7	80	90	0,8	0,9	1
E8	100	0	1	0	0,5
E9	100	100	1	1	1
E10	0	0	0	0	0
E11	60	10	0,6	0,1	0,2
E12	50	50	0,5	0,5	0,5

Zdroj : Vlastné spracovanie

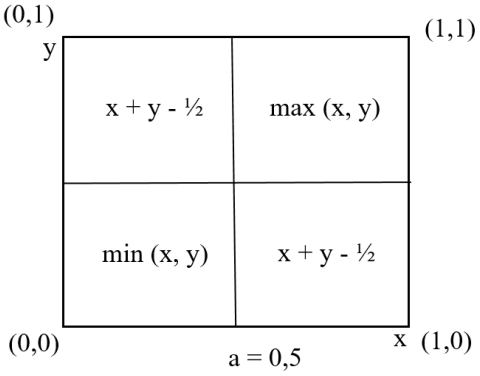
Ak by sme na výpočet použili funkciu z obrázka 9, výsledky by sa zhodovali s výsledkami v tabuľke 3.



Obrázok 9 Zjednodušená grafická interpretácia ordinálneho súčtu pre Lukasiewiczovu t-normu a t-konormu
Zdroj : Vlastné spracovanie

3.4.3 T-norma minimum a t-konorma maximum

Ďalšie funkcie, ktoré môžeme pri tvorbe ordinálnych súčtov použiť sú minimum a maximum. Sú to idempotentné funkcie a nemajú kompenzačný charakter, čo môže byť nežiadúce. Grafická interpretácia je zobrazená na obrázku 10 a príklady v tabuľke 4.



Obrázok 10 Grafická interpretácie ordinálneho súčtu pre t-normu minimum, t-konormu maximum a aritmetický priemer
Zdroj : Vlastné spracovanie

Ako môžeme vidieť na príkladoch E1 – E3, výstupná hodnota po agregácií je vo všetkých troch prípadoch rovnaká, 0,2, aj keď vstupné hodnoty druhého atribútu sa menia. Ordinálny súčet pri využití t-normy minimum nezachytáva kompenzáciu druhého parametra. V prípadoch E4, E5, E8, E11, E12 vidíme správanie strednohodnotovej funkcie, kde sme opäť použili aritmetický priemer.

Tabuľka 4 Výsledok klasifikácie, keď a = 0,5, A_1 je t-norma minimum a A_2 je t-konorma maximum

ENTITA	ATR1	ATR2	x	y	VÝSLEDOK
E1	20	50	0,2	0,5	0,2
E2	20	20	0,2	0,2	0,2
E3	20	40	0,2	0,4	0,2
E4	30	90	0,3	0,9	0,7
E5	80	20	0,8	0,2	0,5
E6	80	80	0,8	0,8	0,8
E7	80	90	0,8	0,9	0,9
E8	100	0	1	0	0,5
E9	100	100	1	1	1
E10	0	0	0	0	0
E11	60	10	0,6	0,1	0,1
E12	50	50	0,5	0,5	0,5

Zdroj : Vlastné spracovanie

Funkcie, ktoré sme si predstavili majú rôzne vlastnosti, z čoho nám vyplýva, že nám pokryjú rôzne prípady klasifikácie údajov. Vlastnosti, ktoré nám pokrývajú funkcie sú napríklad striktnosť, nilpotentnosť, idempotentnosť. Naviac, striktné a nilpotentné funkcie vieme vytvoriť pomocou aditívnych operátorov, čo znamená, že môžeme priradiť funkciu na základe dát, ktoré nám predstavujú doménu údajov na trénovanie.

Jediná logicky neutrálna agregačná funkcia je aritmetický priemer, ostatné sú konjunktívne alebo disjunktívne polarizované. Ako sme aj videli na príkladoch, strednohodnotová funkcia nemusí byť vždy rovnaká, ale taktiež môžeme vybrať z rôznych možností, podľa potreby výsledkov. Ak využijeme strtednohodnotovú funkciu aritmetický priemer, výsledné hodnoty vstupného vektora stupňov príslušností budú rovnaké pre všetky typy ordinálnych súčtov. Ako môžeme vidieť v tabuľkách 1, 3, 4, aj keď sme použili rôzne funkcie pre konjuknktívne a disjunktívne správanie, výsledok pri strednohodnotovom správaní je rovnaký. Okrem aritmetického priemeru sme si demonštrovali použitie aj inej strednohodnotovej funkcie, geometrický priemer. Geometrický priemer nám dáva nižšie výsledky ako aritmetický priemer a zároveň jeho anihilátor je 0. Taktiež pozorujeme, že vo všetkých prípadoch pri vstupnom vektore (0,5; 0,5) je vždy výsledok 0,5. Na základe týchto teoretických poznatkov, ktoré sú popísané v rôznych publikáciách, ako aj ich dôkazy, môžeme formalizovať výsledky a vysvetliť výber použitej funkcie [7] [12] [25].

Pri definovaní výpočtu klasifikácie pomocou ordinálnych súčtov je potrebné vybrať vhodnú funkciu konjunkcie a disjunkcie a taktiež parameter a, čo nám otvára široké spektrum možností. Vhodnú funkciu volíme na základe definovaní klasifikačného priestoru

používateľom. Vďaka lingvistickému opisu a vlastností funkcií, budeme vedieť presne vysvetliť dôvod klasifikácie jednotlivých prvokov. V nasledujúcej časti si ukážeme vytvorenie konkrétneho klasifikačného modelu na základe definovaných požiadaviek a aplikujeme ho na dáta, ktorých výslednú klasifikáciu potom interpretujeme.

3.5 Klasifikácia žiadateľov o úver v banke

V predchádzajúcej kapitole sme si na príkladoch opísali vlastnosti funkcií a teraz si predstavíme set dát, na ktorom si definujeme niekoľko modelových príkladov a riešenie, ako výsledný klasifikačný model vyzerá. Spomínané dáta nám predstavujú dáta klientov nemeckej banky, na základe ktorých klientom schvália alebo neschvália úver. Dátový set obsahuje 20 popisných atribútov, z toho 14 kategorických, 3 ordinálne a 3 binárne. Všetky atribúty sú ohodnotené [18], preto je dátový set použiteľný aj pre vysvetliteľnú UI. Chýbajú však definované podmienky klasifikovania a preto si ich sami určíme. Všetky hodnoty atribútov sú číselne vyjadrené a platí pravidlo, čím vyššie číslo, tým nám hodnota atribútu prináša menšie riziko a teda klient je pre nás perspektívnejší.

Naše modelové príklady predpokladajú s nasledujúcim scenárom: Banka chce používať model klasifikácie ako poradný nástroj či schváliť klientovi úver alebo nie. Pomocou modelu vidíme, ktoré požiadavky klient spĺňa alebo nespĺňa a s akou príslušnosťou. Na základe výsledkov modelu potom zamestnanec môže posúdiť závažnosť nesplnenia požiadaviek a rozhodnúť sa podľa výsledkov klasifikácie o poskytnutí úveru. Ak úver nie je klientovi schválený, na základe systému zistíme dôvody a interpretujeme ich klientovi.

Postup vytvorenia klasifikačného modelu:

A) Úprava dát – aby sme splnili podmienky (1) agregačných funkcií
B) Agregácia atomických atribútov do jednej hodnoty, ak je potrebná
C) Voľba vhodného modelu, aby sme splnili podmienky klasifikácie objektov do tried zadané používateľom
D) Klasifikácia a interpretácia výsledkov

Krok A) Úprava dát

Kvôli splneniu nevyhnutných podmienok agregačných funkcií, musíme vstupné dáta transformovať na interval [0,1]. Všetky atribúty popisujúce našich klientov sú numerické,

preto transformácia nie je problém. Aby sme mohli klasifikáciu vysvetliť, potrebujeme stanovené pravidlá od experta, tie však dáta, ani štúdia, ktorá ich popisuje neposkytuje, preto si sami uvedieme na príkladoch modelové situácie.

Na úpravu dát sme použili lineárnu transformáciu, ako sme aj vyššie spomenuli s určením maximálnej a minimálnej hodnoty splnenia kritéria. Transformáciou sa nebudeme hlbšie zaoberať, pretože to nie je náplň našej práce, ale pre potreby klasifikácie sme to museli spraviť. Algoritmus na úpravu údajov na interval [0,1] sme použili rovnaký pri všetkých modelových príkladoch.

Banka si uchováva nasledujúce informácie o žiadateľoch:

1. stav účtu – stav účtu klienta v banke (kategorická)
2. soba – dĺžka poskytnutia úveru v mesiacoch (ordinálna)
3. história úverov – história dodržiavania predchádzajúcich alebo súčasných úverových zmlúv (kategorická)
4. účel – dôvod potreby úveru (kategorická)
5. hodnota – výška úveru (ordinálna)
6. úspory – úspory klienta (kategorická)
7. dĺžka zamestnania – doba, ako dlho je klienta zamestnaný u súčasného zamestnávateľa (kategorické)
8. výška splátok - splátky úveru, ako percento disponibilného príjmu dlžníka (kategorické)
9. osobný status – kombinácia: pohlavie a rodinný stav (kategorická)
10. veritelia – ďalší veriteľ úveru, ak existuje (kategorická)
11. dĺžka bývania – doba, koľko klient žije v súčasnom ubytovaní (kategorická)
12. majetok – klientov najviac hodnotný majetok (kategorická)
13. vek (ordinálna)
14. iné druhy úverov - úvery od iných poskytovateľov (kategorická)
15. bývanie – typ ubytovania, v ktorom klient v súčasnosti žije (kategorická)
16. počet úverov – počet úverov vrátane súčasného (kategorická)
17. práca – druh práce (kategorická)
18. závislí ľudia – počet ľudí, ktorí finančne závisia od klienta (binárna)
19. telefón – či klient vlastní telefón na svoje meno (binárna)
20. zahraničný pracovník – či klient pracuje v zahraničí (binárna)

Pri spojitých hodnotách atribútov, ktorými sú doba, hodnota a vek sme si určili maximálnu a minimálnu hodnotu a upravili sme dáta lineárnou transformáciou, vzorec (12). Tabuľka 5 nám predstavuje ohraničujúce hodnoty pre 3 spojité atribúty.

Tabuľka 5 Parametre transformačnej funkcie

Nízka hodnota pre atribút DOBA	6
Vysoká hodnota pre atribút DOBA	60
Nízka hodnota pre atribút HODNOTA	1000
Vysoká hodnota pre atribút HODNOTA	10000
Nízka hodnota pre atribút VEK	30
Vysoká hodnota pre atribút VEK	40

Zdroj : Vlastné spracovanie

Kategorické hodnoty atribútov sme transformovali tak, že najnižšie ohodnotenie atribútu zodpovedá 0, najvyššie 1 a medzistupne proporčne zodpovedajú hodnoteniu z pôvodných dát. Príklad transformácie atribútu história úverov môžete vidieť v tabuľke 6.

Tabuľka 6 Transformácia kategorickej premennej atribútu HIST_UVEROV

HIST_UVEROV OHODNOTENIE	TRANSFORMACIA OHODNOTENIA
0 (meškanie so splátkami)	0
1 (kritický účet/iné úvery)	0,25
2 (žiadne úvery)	0,5
3 (všetko splatené)	0,75
4 (predchádzajúci úver v tejto banke + všetko splatené)	1

Zdroj : Vlastné spracovanie

Tabuľka 7 Hodnoty všetkých atribútov vybraných klientov

ID_KLIENTOV	72	81	238	786	790	814
STAV_UCTU	1	1	1	0.5	0.5	0.5
DOBA	0.556	0.556	0	0	0.167	0.056
HIST_UVEROV	0.5	1	0.5	0.25	0	0.25
UCEL	0.3	0.3	0.2	0	0	0.1
HODNOTA	0.013	0.024	0	0	0.008	0.042
USPORY	0.6	1	0.2	0.4	0.2	0.2
DLZKA_ZAM	1	1	0.4	0.4	0.4	1
VYSKA_SPLATOK	1	1	0.25	0.25	0.5	0.5
STATUS	0.75	0.75	0.5	0.5	0.5	0.5
VERITELIA	0.333	0.333	0.333	0.333	0.333	0.333
DLZKA_BYVANIA	1	0.5	0.75	0.25	0.25	1
MAJETOK	0.75	0.75	0.25	0.5	0.25	1
VEK	0.9	1	0	0.2	0	1
I_DRUHY_UVEROV	1	1	1	0.667	1	0.333
BYVANIE	0.667	0.667	0.333	0.667	0.333	1
POCET_UVEROV	0.25	0.25	0.25	0.25	0.5	0.25
PRACA	0.75	0.75	0.5	0.5	0.25	1
ZAVISLI_LUDIA	1	1	0.25	1	1	0.25
TELEFON	0.25	1	0.25	0.25	0.25	1
ZAHR_PRAC	1	1	1	1	1	1

Poznámka: Tabuľka je konvertovaná kvôli jej veľkosti: Prvý stĺpec predstavuje opis hodnôt jednotlivých klientov. Každý ďalší stĺpec predstavuje hodnoty jedného klienta s ID v prvom riadku.

Zdroj : Vlastné spracovanie

Krok B) Agregácia atomických atribútov do jednej hodnoty

V našich modelových príkladoch sme klasifikovali na základe dvoch komponovaných parametrov, ktorými sú KLIENT a UVER. Opisné atribúty sme rozdelili na dve skupiny a komponovali sme ich jednotlivé hodnoty do jednej výslednej:

- Osobné informácie (10 atribútov) - KLIENT
- Informácie o úvere (10 atribútov) – UVER

Klasifikačný parameter KLIENT je zložený z atribútov dĺžka zamestnania, osobný status, dĺžka bývania, majetok, vek, bývanie, práca, závislí ľudia, telefón a zahraničný pracovník. A parameter UVER spája atribúty stav účtu, doba, história úverov, účel, hodnota, úspory, výška splátok, veritelia, iné druhy úverov a počet úverov.

Na základe týchto dvoch skupín atribútov (KLIENT a UVER) sme klasifikovali našich klientov do 3 tried: *áno, nie* alebo *možno*. K agregácií atribútov do jednotlivých skupín sme použili aritmetický priemer, pretože nemáme žiadne iné informácie ako atribúty spolu agregovať a aritmetický priemer (20) je neutrálnou funkciou.

Prvý modelový príklad

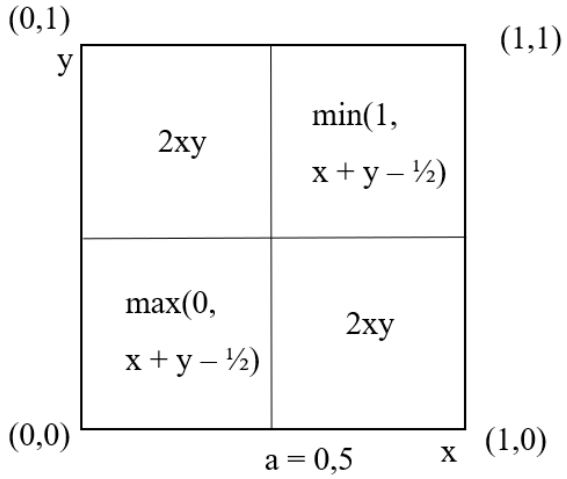
Požiadavky na schvaľovanie úverov:

Ak sú obe skupiny atribútov splnené s vysokou mierov, úver určite schválime. Ak sú obe s nízkou mierou, určite úver neschválime. Ak je jedna vysoká a druha nízka, skôr sa prikláňame k neschváleniu úveru, pričom chceme zachovať, aby podobní klienti mali podobné výsledky.

Krok C) Definovanie modelu klasifikácie klientov do 3 tried

Použité agregačné funkcie v klasifikačnom modeli:

- trieda *N*: konjunktívna funkcia - Lukasiewiczova t-norma (24),
- trieda *A*: disjunktívna funkcia - Lukasiewiczova t-konorma (25),
- trieda *M*: strednohodnotová funkica - geometrický priemer (23).



Obrázok 11 Klasifikačný priestor prvého modelového príkladu

Zdroj : Vlastné spracovanie

Na základe existujúcich informácií sme použili ako konjunktívnu funkciu Lukasiewiczovu t-normu, ako disjunktívnu jeho duálnu funkciu, Lukasiewiczovu t-konormu a ako strednohodnotovú funkciu, geometrický priemer. Banka nepotrebuje, aby klient spĺňal obe skupiny parametrov na 100%, ale stačí ak sú tesne pod hranicou 100%, preto sme zvolili Lukasiewiczovu t-konormu a rovnako to platí aj pri nízkych hodnotách, kde sme zvolili Lukasiewiczovu t-normu, ktoré sú nilpotentné funkcie. Banka je opatrná, keď je jedna skupina atriútov slabá a druhá dobrá, preto sme použili geometrický priemer, ktorý znižuje výsledky. Klasifikačný priestor a použité funkcie na klasifikáciu nám zobrazuje obrázok 11. Tabuľka 8 nám zobrazuje vybrané záznamy klasifikácie.

Tabuľka 8 Výsledky prvého modelového príkladu, pri použití Lukasiewiczovej t-norme, t-konorme a geometrického priemeru

ID	UVER	KLIENT	VÝSLEDOK
847	0.607	0.95	1
997	0.686	0.875	1
81	0.646	0.842	0.99
179	0.701	0.792	0.99
72	0.555	0.807	0.86
814	0.256	0.875	0.45
238	0.373	0.423	0.3
931	0.273	0.552	0.3
210	0.409	0.382	0.29
786	0.265	0.527	0.28
790	0.321	0.423	0.24

Zdroj : Vlastné spracovanie

Krok D) Interpretácia výsledkov:

Klient s ID 790 splnil obe skupiny atribútov s nízkou intenzitou, preto jeho výsledná klasifikácia sa znižuje. Klient s ID 847 aj s ID 997 splnil obe podmienky s vysokou intenzitou, a preto jednoznačne získajú úver. Klient s ID 814 splnil skupinu atribútov predstavujúcich úver s nízkou intenzitou a druhú skupinu atribútov predstavujúcich klienta splnil s vysokou intenzitou, výsledná klasifikácia sa nám však znižuje. Dôvody, prečo klient skôr nedostane úver si môžeme konkrétne vymenovať pre klienta s ID 786. V tabuľke 7 môžeme vidieť jednotlivé hodnoty atribútov. DOBA, na ktorú klient žiada úver je veľmi krátka a HODNOTA, ktorú žiada je veľmi nízka, čo je pre banku nevýhodné. Preto hodnota ÚVER je nízka. Hodnotu trochu zvýšili atribúty ako STAV_UCTU, ktorý je priemerný, úspory, ktoré sú tiež nízke, ale nie nulové. Pri atribútoch o osobných informáciách môžeme vidieť, že niektoré sú priemerné napríklad BYVANIE (je v prenájme) a ostatné nízke, ako napríklad VEK (je to mladý človek).

Druhý modelový príklad

Požiadavky na schvaľovanie úverov:

Ak obe skupiny atribútov klienta budú splnené s nízkou intenzitou, tak sa znižuje možnosť získať úver. Ak sú hodnoty nízke a aspoň jeden z nich je 0, potom úver jednoznačne neschválime. Ak sú hodnoty príslušností skupín atribútov vysoké, tak úver skôr schválime. Ak je aspoň jeden z komponovaných parametrov splnený na 100 %, tak úver jednoznačne schválime. Ak je jedna skupina vlastností splnená s vysokou a druhú s nízkou

intenzitou, čo zahŕňa aj 100% splnenie (1) alebo nesplnenie(0), tak výsledná hodnota je nejednoznačná a rozhodne sa až zamestnanec po pohovore s klientom. Podmienky klasifikácie sa sprísnili na základe vonkajších vplyvov, preto na tých klientov, ktorí majú veľmi vysoké skóre, ale nie 1, sa zamestnanec na preverenie osobne pozrie do podkladov a tí s priemerným skóre sú pozvaní na dodatočný pohovor.

Krok C) Definovanie modelu klasifikácie klientov do 3 tried:

Použité agregačné funkcie v klasifikačnom modeli:

- trieda *N*: konjunktívna funkcia - súčinová t-norma (18),
- trieda *A*: disjunktívna funkcia - súčinová t-konorma (19),
- trieda *M*: strednohodnotová funkica - aritmetický priemer (20).

Správanie na jednotlivých klasifikačných triedach modelu sa nám mení. V konjunktívnej aj disjunktívnej časti požadujeme striktné správanie a preto použijeme súčinovú t-normu a t-konormu. Ako strednohodnotovú funkciu aplikujeme aritmetický priemer, aby sme zachovali neutralitu ovplyvnenia výsledku jednotlivými skupinami atribútov.

Tabuľka 9 Výsledky druhého modelového príkladu, pri použití súčinovej t-norme, t-konorme a aritmetického priemeru

ID	UVER	KLIENT	VÝSLEDOK
847	0.607	0.95	0.96
997	0.686	0.875	0.92
81	0.646	0.842	0.89
179	0.701	0.792	0.88
72	0.555	0.807	0.83
814	0.256	0.875	0.63
238	0.373	0.423	0.33
931	0.273	0.552	0.32
210	0.409	0.382	0.31
786	0.265	0.527	0.29
790	0.321	0.423	0.27

Zdroj : Vlastné spracovanie

Krok D) Interpretácia výsledkov

Ako vidíme z tabuľky 9, klient s ID 847 už jednoznačne nie je adept na úver, pretože ani jeden z parametrov nespĺňa v plnej výške, avšak jeho zaradenie je stále vysoké. Naopak klient s ID 814 vidíme, že si polepšil, pretože nízka intenzita prvého atribútu neznižuje

celkový výsledok, ale oba parametre sa rovnako zohľadňujú vo výsledku. V tomto modelovom príklade si môžeme popísať klienta s ID 72, aby sme zistili, prečo sa výsledok prikláňa skôr k schváleniu úveru. Po agregácií atribútov do skupiny ÚVER je výsledná hodnota skôr vysoká. STAV_UCTU je výborný (klient nemá žiadne dlhy). Hodnota pôžičky je síce nízka, ale je na dlhší čas a splátky sú pre banku vyhovujúce. Druhú skupinu atribútov spĺňa ešte vo vyššej miere, čo nám zvyšuje výsledok. Je to celkom mladý človek bývajúci v podnájme, ale nemá rodinu a má dlhodobé zamestnanie. Hodnoty jednotlivých atribútov môžete vidieť v tabuľke 7.

Tretí modelový príklad:

Požiadavky na schvaľovanie úverov:

Ak sú obe skupiny atribútov splnené s nízkou intenzitou, tak určite klient nedostane úver. Ak sú oba vysoké, tak skôr úver schválime, ale s istotou, iba ak je jeden zo skupín atribútov je splnený na 100%.

Krok C) Definovanie modelu klasifikácie klientov do 3 tried:

Použité agregačné funkcie v klasifikačnom modeli:

- trieda N: konjunktívna funkcia - Lukasiewiczova t-norma (24),
- trieda A: disjunktívna funkcia - pravdepodobnostný súčet (19),
- trieda M: strednohodnotová funkica - aritmetický priemer (20).

Ako môžeme vidieť z požiadaviek, podmienky na agregovanie nízkych a vysokých hodnôt nemusia znieť rovnako a nemusia mať rovnaké vlastnosti. Kombinujeme klasifikovanie z predchádzajúcich dvoch príkladov. Nepoužijeme preto na seba duálne funkcie, ale ako konjunktívnu funkciu použijeme nilpotentnú, Lukasiewiczovu t-normu, ako disjunktívnu funkciu použijeme striktnú, pravdepodobnostný súčet a ako strednohodnotovú funkciu použijeme aritmetický priemer.

Výsledky klasifikácie tretieho modelového príkladu môžeme vidieť v tabuľke 10.

Krok D) Interpretácia výsledkov:

Pri porovnaní výsledkov môžeme pozorovať, že klienti s ID 847, 997, 81, 179, 72, 814, 931 majú rovnaké skóre ako v tabuľke 7 a klienti s ID 238, 210, 786, 790 majú rovnaké

skóre ako pri výsledkoch v prvom modeli, tabuľka 6. Na agregáciu dát v jednotlivých prípadoch sme použili rovnaké funkcie, preto sa nám výsledky zhodujú.

Tabuľka 10 Výsledky klasifikácie tretieho modelového príkladu, pri použití Lukasiewiczovej t-normy, pravdepodobnostného súčtu a aritmetického priemeru.

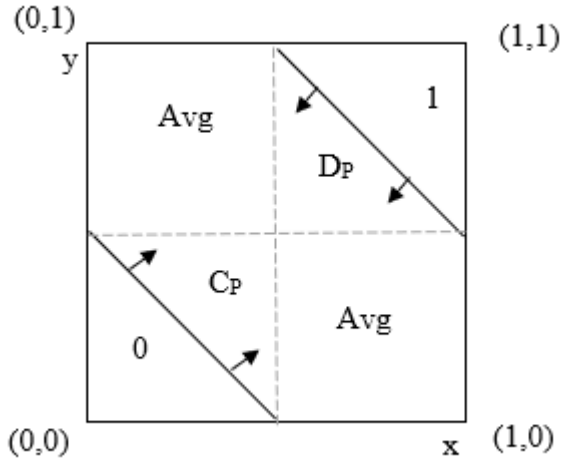
ID	UVER	KLIENT	VÝSLEDOK
847	0.607	0.95	0.96
997	0.686	0.875	0.92
81	0.646	0.842	0.89
179	0.701	0.792	0.88
72	0.555	0.807	0.83
814	0.256	0.875	0.63
931	0.273	0.552	0.33
238	0.373	0.423	0.3
210	0.409	0.382	0.29
786	0.265	0.527	0.29
790	0.321	0.423	0.24

Zdroj : Vlastné spracovanie

Ako vidíme vo všetkých troch tabuľkách 8, 9, 10, výsledky nie sú veľmi polarizované na hraničné hodnoty 1 a 0, pretože sme extrémne hodnoty agregovaním do skupín parametrov aritmetickým priemerom zmiernili. Model nám preto poskytuje v mnohých prípadoch nejednoznačnosť, prikláňa sa skôr ku schváleniu alebo neschváleniu, ale nedáva nám jednoznačne výsledky, čo nemusí byť žiadúce, záleží na požiadavkách klienta. Ak by sme požadovali viac polarizované hodnoty, môžeme hranice posunúť a tak odstrániť veľkú nejednoznačnosť.

Predstavme si modelovú situáciu na našom prvom príklade. Zadanie klasifikácie znie rovnako, preto použijeme Lukasiewiczovu t-normu a t-konormu a ako strednohodnotovú funkciu použijeme aritmetický priemer, kde posunieme hranicu, aby sme dostali nižšie výsledky, ako aj pôvodné zadanie vyžadovalo. Uvedenú situáciu môžeme vidieť na obrázku 12. Model predstavuje pôvodný klasifikačný priestor a šípky nám znázorňujú požadované zmeny modelu.

Na splnenie požiadaviek musíme upraviť matematický model. Existuje mnoho funkcií s určitým správaním, ale taktiež existujú aj parametrické funkcie, ktoré menia správanie na základe zmeny parametra. Poznáme napríklad Yager, Frank, Schweizer a Sklar, Dombi [8] [27]. Všeobecne tieto skupiny funkcií obsahujú bežne používané t-normy aj t-konormy, ako aj ich hraničné hodnoty.



Obrázok 12 Zmena klasifikačného priestoru po zmene parametra klasifikačných funkcií

Zdroj : Vlastné spracovanie

Predstavíme si vlastnosti rodiny funkcií Schweizer a Sklar [42].

Rodina t-noriem je definovaná nasledovne:

$$T_\lambda^S(x,y) = \begin{cases} \min(x,y) & pre\ \lambda = -\infty \\ T_P(x,y) & pre\ \lambda = 0 \\ T_D(x,y) & pre\ \lambda = \infty \\ \left(\max\left(x^\lambda + y^\lambda - 1, 0\right)\right)^{\frac{1}{\lambda}} & inak \end{cases} \tag{27}$$

Rodina t-konoriem je definovaná nasledovne:

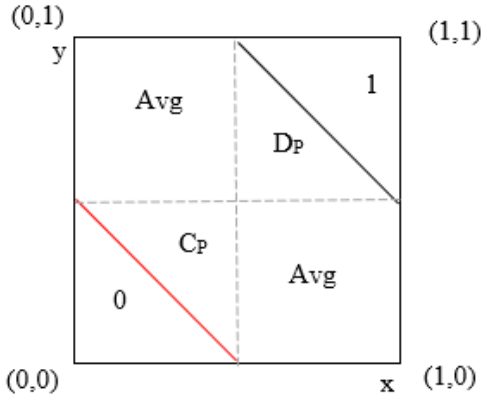
$$S_\lambda^S(x,y) = \begin{cases} \max(x,y) & pre\ \lambda = -\infty \\ T_P(x,y) & pre\ \lambda = 0 \\ T_D(x,y) & pre\ \lambda = \infty \\ 1 - \left(\min\left((1-x)^\lambda + (1-y)^\lambda - 1, 0\right)\right)^{\frac{1}{\lambda}} & inak \end{cases} \tag{28}$$

Pôvodnú rodinu funkcií s konjunktívnym správaním musíme upraviť na interval [0;0,5] a rodinu funkcií s disjunktívnym správaním na interval [0,5;1], aby sme ich mohli použiť v ordinálnych súčtoch. Zmena parametra pre konjunktívnu funkciu ovplyvňuje hranicu vyznačenú červenou na obrázku 13.

Konjunktívna funkcia: parametrizovaná Lukasiewiczova t-norma upravená pre použitie v ordinálnom súčte:

$$T_L(x, y) = (\max(0, x^\lambda + y^\lambda - 0{,}5^\lambda))^{\frac{1}{\lambda}} \quad (29)$$

- ak λ = 1 – pôvodná funkcia – Lukasiewiczova t-norma (24)
- ak λ > 1 – viac reštriktívna
- ak λ < 1 – menej reštriktívna



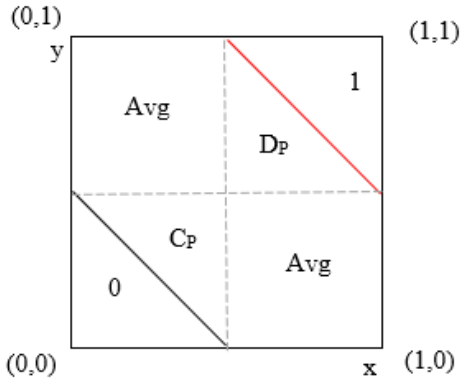
Obrázok 13 Zmena klasifikačného priestoru po zmene parametra pre konjunktúrnu funkciu

Zdroj : Vlastné spracovanie

Disjunktívna funkcia: parametrizovaná Lukasiewiczova t-konorma upravená pre použitie v ordinálnom súčte:

$$S_L(x, y) = (\min(1, x^\lambda + y^\lambda - 0{,}5^\lambda))^{\frac{1}{\lambda}} \quad (30)$$

- ak λ = 1 – pôvodná funkcia - Lukasiewiczova t-konorma
- ak λ > 1 – viac reštriktívna
- ak λ < 1 – menej reštriktívna



Obrázok 14 Zmena klasifikačného priestoru po zmene parametra pre disjunktívnu funkciu

Zdroj : Vlastné spracovanie

Zmena parametra v disjnuktívnej funkcií ovplyvňuje hranicu vyznačenú červenou na obrázku 14.

Strednohodnotová funkcia môže byť vyjadrená ako power priemer (31) (generalizovaná funkcia, ktorá vyjadruje aj kvadratický priemer) [8]. Na základe parametra dostaneme aritmetický alebo geometrický priemer, taktiež minimum a maximum.

$$A\ (x, y) = (0{,}5x^r + 0{,}5y^r)^{\frac{1}{r}} \quad , -\infty \leq r \ \leq \ \infty \tag{31}$$

Strednohodnotová funkcia: parametrizovaná funkcia power priemer upravená na ordinálny súčet:

$$\mathrm{A(x, y)} = (x^r + y^r - 0{,}5^r)^{\frac{1}{r}} \tag{32}$$

- ak r =1 – originálna funkcia – aritmetický priemer (20)
- ak r > 1 – menej reštriktívna, viac sa prikláňa k triede A
- ak r < 1 – viac reštriktívna, viac sa prikláňa k triede N

Na základe obrázku 12 vidíme, že konjunktívne správanie chceme zmeniť na viac reštriktívne, disjunktívne správanie na menej reštriktívne a strednohodnotové správanie je pesimistické. Preto hranicu pre triedu *N* sme posunuli vyššie, teda prísnejšie a hranicu triedy *A* nižšie. Na základe požiadaviek sme si určili hodnoty parametrov λ a *r*:

- Konjunktívna funkcia: viac reštriktívna → λ > 1→ λ = 3
- Strednohodnotová funkcia: viac reštriktívna → λ < 1 → λ = -1
- Disjunktívna funkcia: menej reštriktívna → r < 1 → r = -1

Tabuľka 11 Výsledky prvého modelového príkladu po zmene parametrov klasifikačných funkcií

ID	UVER	KLIENT	VÝSLEDOK
847	0.607	0.95	1
997	0.686	0.875	1
81	0.646	0.842	1
179	0.701	0.792	1
72	0.555	0.807	0.961
814	0.256	0.875	0.329
931	0.273	0.552	0.288
238	0.373	0.423	0.142
210	0.409	0.382	0
786	0.265	0.527	0.272
790	0.321	0.423	0

Zdroj : Vlastné spracovanie

Výsledky klasifikácie prvého modelového príkladu s upravenými hranicami môžeme vidieť v tabuľke 11. Môžeme pozorovať, že výsledky sú viac polarizované a preto je jednoduchšie pre zamestnanca banky určiť, kto získa úver, kto nie a na posúdenie nejasností zostane menej klientov.

Na vytvorenie klasifikačných modelov sme použili programovací jazyk python. Python je moderný programovací jazyk, jeho autorom je Guido van Rossum (1989) a jeho popularita stále rastie. Je univerzálny, vhodný na vytvorenie aplikácií na analýzu dát [38], spracovanie médií a preto sme ho použili aj my na vytvoreneie nášho parametrizovateľného klasifikačného modelu. Model môže byť potom ľahšie rozšíriteľný na aplikáciu analýzy dát s použitím strojového učenia. Použité dáta sú uložené v csv súboroch, ktoré môžeme ľahko pomocou python knižnice os načítať a ďalej manipulovať s dátami. Výsledky Klasifikácie taktiež zapíšeme do pôvodného csv súboru a tak môžu byť distribuované v rôznych platformách, ktoré súbor csv podporujú. Napríklad v kancelárskom nástroji MS Excel, ktorý je užívateľovi jednoducho prístupný a ľahko sa v ňom výsledky ďalej spracovávajú alebo interpretujú [11]. Python okrem iného nám poskytuje výhody rýchleho spracovania aj veľkého objemu dát a jednoduchej syntaxe programovania.

Ako môžeme na jednotlivých výstupoch modelov pozorovať, klasifikovanie niektorých klientov sa veľmi nemenilo. Napríklad klient s ID 72 nadobúdal hodnoty {0,86; 0,83; 0.961}. Ale naopak, napríklad klientovi s ID 814 sa príslušnosť zaradenia v klasifikačnom modeli celkom výrazne menila. Výsledky klasifikácie nadobúdali hodnoty {0,45; 0,63; 0.329}. Tu môžeme vidieť, aké sú veľmi dôležité úvodné klasifikačné podmienky. Zmenou postoja experta sa nám mení aj klasifikačný model a preto aj jeho výsledky. Definícia klasifikačného modelu klientom je veľmi kľúčová aj kvôli interpretovateľnosti výsledkov. Takto môžeme ľahko vysvetliť, prečo sa nám hodnota menila, ktoré atribúty najviac spôsobili zmeny a pod.

Ako môžeme z modelových príkladov pozorovať, rôznorodosť funkcií nám pokrýva mnoho praktických príkladov. Vďaka rôznym vlastnostiam funkcií vieme rôzne opisy reality používateľov ľahko zachytiť. Teda približujeme sa k realite, neskresľujeme ju. Vieme matematicky vyjadriť aj vágnosť a dostať výsledky, ktoré vieme interpretovať. Modelové príklady sme si ukázali na dátach poradného systému pre banky. Avšak model môžeme aplikovať aj do prostredia medicíny, kde nám jednotlivé atribúty predstavujú symptómy chorôb a klasifikačný model nám môže pomôcť pri diagnostikovaní choroby [23]. Taktiež môžeme model použiť aj na motiváciu zákazníkov, kde môžeme zákazníkov s podobnými

zvykmi podobne motivovať [33]. Alebo ako odporúčací systém napríklad na realitnom trhu, či hľadanie dovolenky a odporučenie tej najlepšej možnosti, ktorá bude spĺňať naše požiadavky.

V našej práci sme vytvorili modelové situácie klasifikácie a pozorovali sme, ako sa menili výslekdy pri zmene funkcií a parametrov. Ďalším krokom môže byť porovnanie výsledkov klasifikcáie s očakávanými výsledkami. Vytvorenie aplikácie na optimalizáciu klasifikačného priestoru, by mohla byť nadstavba na naše výsledky inovatívnej klasifikácie. Keď sme na základe používateľových požiadaviek zistili, že na klasifikáciu je vhodná nilpotentná funkcia, zvolili sme Lukasiewiczovu t-normu. Ale tu nám zostáva otázka, či je tou najvhodnejšou funkciou. Vďaka požiadavkám vysvetlenými výrazmi prirodzeného jazyka aplikujeme na model iba funkcie, ktoré spĺňajú tieto požiadavky. Po zvolení vhodnej kategórie funkcií by sme mohli jej parametre naučiť na základe existujúcich príkladov a dát, pre ktoré vieme výsledok. Ako sme si uviedli pri rodine funkcií Schweizer a Sklar, zmenou parametra dostávame rôzne správanie modelu. Ale určenie vhodného parametra je pre experta takmer nemožná úloha. Na základe už existujúcich vstupno-výstupných dát, by sme vedeli optimalizovať parameter, aby sa výsledky čo najbližšie zhodovali s realitou.

Porovnaním výsledkov s očakávanými pomocou automatizácie získame najvhodnejšie hodnoty parametrov s najmenšou odchýlkou na využitie v danej úlohe klasifikácii dát. Opísaný postup je iba návrh na vytvorenie klasifikačného modelu pomocou strojového učenia pre budúcu prácu.

4 Diskusia

Klasifikačný model vytvorený pomocou ordinálnych súčtov nám ponúka nový pohľad na klasifikáciu. Keďže poznáme konkrétny algoritmus klasifikácie a vieme na základe používateľových podmienok aj povedať prečo, teda ho vieme vysvetliť aj interpretovať, dostávame model vysvetliteľnej („glass-box“) klasifikácie. Vieme vysvetliť, znamená, že poznáme agregačnú funkciu na klasifikovanie a vieme odôvodniť aj jej vlastnosti. Vieme interpretovať, povedať, prečo z daných vstupných parametrov sme dostali daný výstup. Model nám taktiež nedáva výsledky striktnej klasifikácie, ale vyjadruje príslušnosť do kategórie. Vďaka aplikovaniu teórie fuzzy množín a fuzzy logiky preto vieme zachytiť vágnosť údajov a interpretovať výsledky. V našom modeli sme predpokladali definovanie vstupných podmienok expertom, ktorý pozná danú problematiku a vie popísať, aké výsledky chce dosiahnuť. Vďaka ordinálnym súčtom sme vedeli zachytiť nepresné vyjadrenie podmienok a taktiež vieme zachytiť rôzne požiadavky používateľa vďaka variabilnosti modelu.

Výhody klasifikácie pomocou ordinálnych súčtov:

- Zachytenie vágnosti - Klasifikačný priestor nám zachytáva klasifikáciu do 3 tried: *áno*, *nie*, *možno*. Avšak výsledok modelu nám nevyjadruje exaktné zaradenie prvku, ale jeho príslušnosť. Teda vieme povedať, ku ktorej možnosti prvok viac inklinuje a s akou intenzitou.
- Možnosť použitia menej atribútov - Zvyšovaním atribútov sa prehľadnosť klasifikačného modelu znižuje. Široká škála agregačných funkcií nám zmenší ich množinu na skupiny atribútov (komponované parametre).
- Spoľahlivosť – Vieme odôvodniť výsledky klasifikácie, aj popísať postup, ako sme sa k nim dostali.
- Lingvistické výrazy - Na opis vstupných podmienok môže používateľ použiť krátke opisné vety, nečíselné výrazy. Pre expertov sú jednoduchšie definovateľné, taktiež nám zahrňujú vágnosť informácií, nefixujeme sa na presné čísla a sú bližšie aj koncovému používateľovi [10]. Vďaka nim vieme výsledky ľahšie interpretovať a sú pochopiteľnejšie. My sme si využitie lingvistických výrazov ukázali iba na definovaní vstupných podmienok, ale ich použitie môže byť oveľa väčšie, napríklad pri definovaní atribútov, ich vzťahov a ich štruktúry pre lepšie pochopenie transformácie na interval [0,1].

V našej praktickej úlohe, kde sme experimentovali s dátami, sme zvolili podľa opisu požiadaviek napríklad striktnú súčinovú t-normu, či nilpotentnú funkciu Lukasiewiczovu t-normu. Avšak otázkou zostáva či by iná striktná, či nilpotentná funkcia neklasifikovala dáta lepšie. Táto dedukcia a možnosť parametrizovať model nám otvára možnosť využitia strojového učenia na nájdenie tej najvhodnejšie funkcie a jej parametrov, na základe vstupno-výstupných dát.

Zmenu výsledkov v modeli po zmene parametrov sme si ukázali na rodine funkcií Schweizer a Sklar. Na základe naučeného parametra *lambda* - λ (27) (28) zo vstupno-výstupných dát sa môžeme odvolať na podstatu funkcií klasifikácie (striktná, nilpotentná, min, nekontinuálna). Taktiež konjunktívna a disjunktívna funkcia nemusia byť duálne, pretože ak by lamba bola väčšia ako 0, tak potom dostaneme čistú t-normu a naopak t-konormu. Teda parameter nám vysvetlí, prečo sa funkcia správa ako sa správa a prečo nám poskytla dané odpovede. Pre človeka je takmer nemožné určiť parameter, ale ak mame dáta, pre ktoré výsledok poznáme alebo používateľ nám povedal, aký má byť výsledok, potom môžeme strojovým učením nastaviť parameter *lambda* - λ, ktorý ma hodnotu 1 pre Lukasiewiczovu t-normu.

Aj na vyjadrenie strednohodnotovej funkcie môžeme použiť parametrickú funkciu, ako sme si ukázali na príklade power priemer. Parameter *r* (30) určíme z dát. Spomínané skutočnosti nás priviedli k transparentnej a vysvetliteľnej klasifikácií, kde aj vstupné parametre sú flexibilné, s inklinovaním k triede *áno* alebo *nie*.

Podobne môžeme určiť parametre aj pre iné agregačné funkcie. Vďaka týmto predpokladom by sme získali vysvetliteľný klasifikačný model strojového učenia. V práci sme si popísali niektoré najznámejšie algoritmy využívané v strojovom učení, ktoré však sú náročné na vstupný objem dát a sú takzvané „čierne skrinky“, nevieme jednoznačne vysvetliť, ako prišli k záverom. Pri učení klasifikačného modelu pomocou ordinálnych súčtov, by sme taktiež potrebovali vstupné dáta a nepredišli by sme nevýhodám, ktoré sme popísali v kapitole 1.2 a 1.3. Preto potrebujeme interakciu človeka (human-in-the-loop) [23], aby voľba funkcie bola jasná alebo boli pokryté aj príklady, kde nám chýbajú dáta. Extrémne hodnoty atribútov a anomálie nám rieši transformácia dát [3]. Vytvorenie takéhoto nástroja umelej inteligencie, kde vyžadujeme interakciu človeka nazývame interaktívne strojové učenie (iAI – interactive AI) [22]. V našom konkrétnom príklade pri vytvorení interaktívnej UI klasifikácie by experti určili subdoménu funkcií na základe požiadaviek a potom podľa dát, by sa vybrala tá najvhodnejšia. Taktiež by nám mohli určiť či má byť klasifikácia viac alebo menej reštriktívna, na základe čoho by sme vedeli akým smerom (na kladné alebo

záporná hodnoty) máme učiť parametre funkcií. Vytvorenie takého modelu je návrhom na pokračovanie na základe našich teoretických zistení.

Ďalšou otázkou zostáva vhodná transformácia na interval [0,1]. Ako sme si už aj naznačili, využitie lingvistických súhrnov tu nadobúda nový zmysel, kedy na základe popisu expertom aplikujeme vhodnú transformačnú funkciu. V našej práci sme si len pokryli lineárnu transformáciu. Ďalším krokom analýzy by bolo, ako transformovať iné dátové typy ako obrázky alebo text, nerovnomerné rozdelenie dát, nepresné informácie, napríklad vyjadrené pomocou fuzzy čísiel alebo ako by sme transformovali dáta, ak vysoká a nízke hodnoty sú pre nás dôležité (nadobúdajú hodnotu 1) a stredné hodnoty sú menej dôležité (nadobúdajú hodnotu 0).

Veľmi zaujímavou témou môže byť automatické generovanie lingvistických súhrnov z výsledkov klasifikácie [43].

Ako sme videli aj v našom príklade klasifikácie žiadateľov o úver, klienti boli popísaní mnohými atribútmi. Doteraz sme sa bavili o klasifikácií iba na základe dvoch atribútov. Avšak jednou z vlastností agregačných funkcií je asociatívnosť, ako sme si uviedli v kapitole 1.4. Teda agregačné funkcie môžeme použiť na agregáciu n ($n \in N$) atribútov a preto by sme mohli klasifikáciu ľubovoľne rozširovať.

Atribúty klientov v banke sme však pre prehľadnosť zgrupovali do logických kategórií, na základe ktorých sme až klientov klasifikovali. Tu sa nám tiež otvára priestor na otázky, ako správne agregovať atribúty, aby najlepšie zachytili realitu. Klasifikácia môže byť rozšírená o dôkladnejší opis vstupných atribútov. Napríklad požiadavky, ktorý atribút ma vyššiu váhu, ktoré medzi sebou súvisia alebo ktorý atribút je odvodený od druhého. Príklad opisu atribútov:

Ak ATR1 je nízky a (ATR2 a ak je možné aj ATR3 sú vysoké) a (väčšina podmienok z {ATR4..ATRn} je vysoko splnených) potom je závažnosť vysoká.

Taktiež zaujímavé môžu byť situácie kolízie atomických podmienok, napríklad súčasný výskyt symptómov A, B, C je menej významy ako výskyt A,G. Takúto situáciu zachytávajú Choquet integrály [7].

ZÁVER

V našej práci sme vytvorili model klasifikácie do troch tried: *áno*, *nie* a *možno* s flexibilnými hranicami pomocou ordinálnych súčtov konjunktívnych a disjunktívnych funkcií, s možnosťou prikloniť sa na stranu *áno* alebo *nie*. Takýto postoj klasifikácie sa využije, ak nemáme k dispozícií dostatočný objem vstupných dát s označením aj ich výstupu (na učenie z dát) alebo ak nemáme AK-POTOM pravidlá.

Na začiatku sme porovnali už existujúce klasifikačné metódy. Algoritmy strojového učenia nám prinášajú vysokú presnosť, ale nevieme, ako sa k výsledkom dostali. Pravidlové systémy sú jasne vysvetliteľné modely, ale na druhej strane komplexnosťou sa stávajú menej presné. Nami navrhnutý model nám prináša nový prístup ku klasifikácii s vysvetliteľným výstupom.

Náš vytvorený model využíva agregačné funkcie, ordinálne súčty, na klasifikáciu dát. Klasifikácia pomocou ordinálnych súčtov vyžaduje najskôr informácie od experta na popis klasifikačného problému, ako sa klasifikujú vysoké hodnoty vstupných atribútov, ako sa klasifikujú nízke hodnoty atribútov a poprípade ako sa klasifikuje kombinácia vysokých a nízkych hodnôt. Rozšírený model môže obsahovať aj opis agregácie parametrov alebo opis transformácie dát na interval [0,1]. Na základe informácií sa zostaví konkrétny klasifikačný model, ktorý nám po načítaní vstupných dát vypočíta výstup.

Vďaka transparentnosti modelu je klasifikácia vysvetliteľná a preto je aj ľahšie interpretovateľná koncovým používateľom. Dôvera v systémy je veľmi užitočná, hlavne ak ich používame v kritických oblastiach, ako je medicína, kedy ide o ľudské životy, alebo pri autonómnych dopravných prostriedkoch. Technológie sú súčasťou nášho každodenného života, pomáhajú nám rozhodovať sa, preto je veľmi dôležité, aby sme sa vedeli na ich rozhodnutie spoľahnúť.

Vytvorený klasifikačný model sme aplikovali na konkrétny dátový set a pozorovali sme výsledky. Na modelových príkladoch sme si ukázali vytvorenie klasifikačných algoritmov na mieru podľa požiadaviek, demonštrovali sme, ako sa môže klasifikácia meniť zmenou funkcií či parametrov a ukázali sme si, ako ďalej vieme model rozširovať, aby sme nestratili jeho základné vlastnosti.

Parametrický model klasifikácie nám otvára možnosť využiť strojové učenie na získanie najvhodnejšej funkcie na klasifikáciu a jej parametrov. V našej práci sme teoreticky opísali možnosť využitia nášho modelu v aplikácii umelej inteligencie, s možnosťou vysvetlenia dosiahnutých výsledkov a ich interpretovania.

Zoznam použitej literatúry

[1] ACZÉL, János (ed.). *Lectures on functional equations and their applications*. Academic press, 1966.

[2] ADOMAVICIUS, Gediminas; TUZHILIN, Alexander. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 2005, 17.6: 734-749.

[3] AGGARWAL, Charu C. *Data mining: the textbook*. Springer, 2015.

[4] ALONSO, Jose M.; CASTIELLO, Ciro; MENCAR, Corrado. Interpretability of fuzzy systems: Current research trends and prospects. *Springer handbook of computational intelligence*, 2015, 219-237.

[5] ASIRI, Sidath. *Machine Learning Classifiers*. [elektronický zdroj] Towards Data Science. 2018. [cit. 25.10.2020] Dostupné na: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623

[6] BALÁŽ, Vladimír, et al. Štrukturálne závislosti národných inovačných systémov: modelovanie nelineárnej dynamiky inštitúcií pomocou neurónových sietí. *Ekonomický časopis*, 2011, 59.01: 3-28.

[7] BELIAKOV, Gleb; CALVO, Tomasa. Construction of aggregation operators with noble reinforcement. *IEEE transactions on fuzzy systems*, 2007, 15.6: 1209-1218.

[8] BELIAKOV, Gleb, et al. *Aggregation functions: A guide for practitioners*. Heidelberg: Springer, 2007.

[9] BLOCH, Isabelle. Information combination operators for data fusion: A comparative review with classification. *IEEE Transactions on systems, man, and cybernetics-Part A: systems and humans*, 1996, 26.1: 52-67.

[10] BOUCHON-MEUNIER, Bernadette, et al. (ed.). *Technologies for Constructing Intelligent Systems 1: Tasks*. Physica, 2013.

[11] Bříza, V. *Excel 2017 podrobný průvodce*. Praha: Grada Publishing, a.s., 2017. 232 s. ISBN 978-80-247-1965-8.

[12] CALVO, Tomasa, et al. Aggregation operators: properties, classes and construction methods. In: *Aggregation operators*. Physica, Heidelberg, 2002. p. 3-104.

[13] DETYMIECKI, Marcin (2001). *Fundamentals on Aggregation Operators*. [rigorózna práca]. Berkeley:University of California. 2001. 39 s.

[14] DUBOIS, Didier; PRADE, Henri. On the use of aggregation operations in information fusion processes. *Fuzzy sets and systems*, 2004, 142.1: 143-161.

[15] DUJMOVIC, Jozo. *Soft Computing Evaluation Logic: The LSP Decision Method and Its Applications*. John Wiley & Sons, 2018.

[16] Európska komisia: *Biela kniha o umelej inteligencii – európsky prístup k excelentnosti dôvere* [elektronický zdroj]. Brusel, 2020, online. [cit. 7. 2. 2021]. Dostupné na: https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_sk.pdf

[17] FODOR, János C.; YAGER, Ronald R.; RYBALOV, Alexander. Structure of uninorms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1997, 5.04: 411-427.

[18] GRÖMPING, Ulrike. *South German Credit Data: Correcting a Widely Used Data Set.* Report 4/2019, Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin.

[19] GUPTA, Madan M.; QI, J11043360726. Theory of T-norms and fuzzy inference methods. *Fuzzy sets and systems*, 1991, 40.3: 431-450.

[20] HAN, Jiawei. Research frontiers in advanced data mining technologies and applications. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Berlin, Heidelberg, 2007. p. 1-5.

[21] HARDESTY, Larry. *Explained: Neural Networks*. [Online] MIT News. 2017. [cit. 25.10.2020] Dostupné na: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[22] HOLZINGER, Andreas, et al. Interactive machine learning: experimental evidence for the human in the algorithmic loop. *Applied Intelligence*, 2019, 49.7: 2401-2414.

[23] HOLZINGER, Andreas. Interactive machine learning for health informatics: when do we need the human-in-the-loop?. *Brain Informatics*, 2016, 3.2: 119-131.

[24] HUDEC, Miroslav. *Fuzzzy logika pre hospodársku informatiku*. Bratislava: Vydavateľstvo Ekonóm – Bratislava.2015. 220 s. ISBN 978-80-225-4100-8

[25] HUDEC, Miroslav - MINÁRIKOVÁ, Erika - MESIAR, Radko - SARANTI, Anna - HOLZINGER, Andreas. *Classification by ordinal sums of conjunctive and disjunctive functions for explainable AI and interpretable machine learning solutions*. *Knowledge-Based Systems*, 2021, 220: 106916.

[26] JEMALA, Marek, a kol. Globalizácie ako sústava procesov vytvárania vyspelejšieho, ale rizikového sveta. In *Ekonomický časopis*, 2008, roč. 56, č. 9, s. 925-942

[27] KLEMENT, Erich Peter; MESIAR, Radko; PAP, Endre. *Triangular norms*. Springer Science & Business Media, 2013.

[28] LINARDATOS, Pantelis - PAPASTEFANOPOULOS, Vasilis - KOTSIANTIS, Sotiris. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 2021, 23.1: 18.

[29] MAGDALENA, Luis. Fuzzy rule-based systems. *Springer Handbook of Computational Intelligence*. Springer, Berlin, Heidelberg, 2015. 203-218.

[30] MAGLOGIANNIS, Ilias G. (ed.). *Emerging artificial intelligence applications in computer engineering: real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies*. Ios Press, 2007.

[31] MAŘÍK, Vladimír - ŠTĚPÁNKOVÁ, Olga - LAŽANSKÝ, Jiří. *Umnělá inteligence (1)*. Praha: Vydavateľstvo Akademia vied českej republiky. 1993. 264 s. ISBN 80-200-0496-3

[32] MAŘÍK, Vladimír - ŠTĚPÁNKOVÁ, Olga - LAŽANSKÝ, Jiří. *Umnělá inteligence (4*). Praha: Vydavateľstvo Akademia vied českej republiky. 2003. 474 s. ISBN 80-200-1044-0

[33] MEIER, Andreas; WERRO, Nicolas. A fuzzy classification model for online customers. *Informatica*, 2007, 31.2.

[34] MEIER, Andreas, et al. Using a fuzzy classification query language for customer relationship management. In: *Proceedings of the 31st international conference on Very large data bases*. 2005. p. 1089-1096.

[35] DE BAETS, Bernard; MESIAR, Radko. Ordinal sums of aggregation operators. In: *Technologies for Constructing Intelligent Systems 2*. Physica, Heidelberg, 2002. p. 137-147.

[36] MINÁRIKOVÁ, Erika. *Klasifikácia údajov s prostredí MS Excel* [bakalárska práca]. Bratislava: Ekonomická univerzita v Bratislave [s. n.], 2019. 44s.

[37] NOVÁK, Vilém. *Fuzzy množiny a jejich aplikace.* 1. vyd. Praha: SNTL – Nakladtelství technické literatury, 1986. 278 s.

[38] PHILLIPS, Dusty. *Python 3 object-oriented programming: build robust and maintainable software with object-oriented design patterns in Python 3.8*. Packt Publishing Ltd, 2018.

[39] PRADERA, Ana; TRILLAS, Enric; CALVO, Tomasa. A general class of triangular norm-based aggregation operators: quasi-linear T–S operators. *International Journal of Approximate Reasoning*, 2002, 30.1: 57-72.

[40] ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 1958, 65.6: 386.

[41] SAMINGER, Susanne. On ordinal sums of triangular norms on bounded lattices. *Fuzzy Sets and Systems*, 2006, 157.10: 1403-1416.

[42] SCHWEIZER, Berthold - SKLAR, Abe,: Associative functions and triangle inequalities, *Publicationes Mathematicae Debrecen 8*, 1961, 169-186.

[43] VUČETIĆ, Miljan; HUDEC, Miroslav; BOŽILOVIĆ, Boško. Fuzzy functional dependencies and linguistic interpretations employed in knowledge discovery tasks from relational databases. *Engineering Applications of Artificial Intelligence*, 2020, 88: 103395.

[44] WRIGHT, Craig S. *The IT regulatory and standards compliance handbook: How to survive information systems audit and assessments*. Elsevier, 2008.

[45] YAGER, Ronald R. Noble reinforcement in disjunctive aggregation operators. *IEEE Transactions on Fuzzy Systems*, 2003, 11.6: 754-767.

[46] ZIMMERMANN, H.-J.; ZYSNO, Peter. Decisions and evaluations by hierarchical aggregation of information. *Fuzzy sets and systems*, 1983, 10.1-3: 243-260.