

EKONOMICKÁ UNIVERZITA V BRATISLAVE

Fakulta Hospodárskej informatiky

Evidenčné číslo: 103006/B/2025/3 6146 8353 7750 1956

**Rozvoj inteligentného systému pre riadenie zásob v maloobchode
s využitím jazyka Python**

Bakalárska práca

2025

Soňa Pazdernatá

EKONOMICKÁ UNIVERZITA V BRATISLAVE
Fakulta Hospodárskej informatiky

**Rozvoj inteligentného systému pre riadenie zásob v maloobchode
s využitím jazyka Python**

Bakalárska práca

Študijný program: Hospodárska informatika
Študijný odbor: Ekonómia a manažment
Školiace pracovisko: Katedra matematiky a aktuárstva
Vedúci záverečnej práce: Ing. Andrej Bednařík

Bratislava 2025

Soňa Pazdernatá



Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Soňa Pazdernatá

Študijný program: hospodárska informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)

Študijný odbor: ekonómia a manažment

Typ záverečnej práce: Bakalárska záverečná práca

Jazyk záverečnej práce: slovenský

Sekundárny jazyk: anglický

Názov: Rozvoj inteligentného systému pre riadenie zásob v maloobchode s využitím jazyka Python.

Anotácia: Táto práca sa zameriava na vývoj inteligentného systému pre riadenie zásob v maloobchode s využitím jazyka Python. Cieľom je vytvoriť efektívny nástroj, ktorý pomôže maloobchodníkom optimalizovať ich zásoby, minimalizovať riziko nedostatku a prebytku tovaru. Práca sa sústreďí na analýzu historických dát o predajoch, predikciu budúceho dopytu a optimalizáciu objednávkového cyklu s použitím knižníc Pandas, scikit-learn a NumPy. Hlavným cieľom je poskytnúť maloobchodníkom efektívny nástroj na lepšie hospodárenie so zdrojmi a zvýšenie konkurencieschopnosti a ziskovosti.

Vedúci: Ing. Andrej Bednařík
Oponent: Mgr. František Slaninka, PhD.
Katedra: KMA FHI - Katedra matematiky a aktuárstva
Vedúci katedry: doc. Ing. Michal Páleš, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 09.03.2024

Dátum schválenia: 11.03.2024

doc. Ing. Martin Mišút, CSc.
osoba zodpovedná za realizáciu študijného programu

Čestné vyhlásenie

Česne vyhlasujem, že bakalársku prácu som vypracovala samostatne pod vedením vedúceho záverečnej práce a využitím literatúry uvedenej v zozname.

Dátum 5.1.2025

.....
(podpis študenta)

Pod'akovanie

Moje pod'akovanie by som chcela venovať Ing. Andrejovi Bednaříkovi za odbornú pomoc, cenné rady, trpezlivosť a odporúčania pri písaní mojej záverečnej práci.

ABSTRAKT

PAZDERNATÁ, Soňa: *Rozvoj inteligentného systému pre riadenie zásob v maloobchode s využitím jazyka Python*. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra matematiky a akuárstva. -Vedúci záverečnej práce: Ing. Andrej Bednařík – Bratislava: FHI EU, 2025, počet strán 59.

Cieľom práce je vyvinúť inteligentný systém pre riadenie zásob v maloobchode s využitím programovacieho jazyka Python, ktorý prostredníctvom analýzy historických predajných dát a predikcie budúceho dopytu pomôže maloobchodníkom efektívnejšie optimalizovať zásoby, minimalizovať riziko nedostatku a prebytku tovaru, a tým zvýšiť ich konkurencieschopnosť a ziskovosť. Údaje sme získali z upraveného datasetu retail_sales.csv, ktorý bol vytvorený kombináciou verejne dostupných informácií a umelej inteligencie, tak aby simuloval realistické správanie predaja v maloobchodnom prostredí. Výsledkom riešenia práce je systém, ktorý umožňuje predpovedať dopyt na obdobie 12 mesiacov. Na základe odhadnutého dopytu vypočítať veľkosť ekonomickej objednávky, bod objednania a bezpečnostnú zásobu, čo sú dôležité metriky na optimalizáciu zásob. Pridaná hodnota práce je v aplikácii metód strojového učenia pre zefektívnenie rozhodovania v oblasti riadenia zásob, čo prispieva k lepšiemu hospodáreniu so zdrojmi a zvýšeniu konkurencieschopnosti maloobchodníkov.

Kľúčové slová: riadenie zásob, predikcia dopytu, strojové učenie, EOQ, XGBoost, Python

ABSTRACT

PAZDERNATÁ, Soňa: *Development of an intelligent inventory management system for retail using Python programming language.* – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Mathematics and Actuarial Science. – Supervisor of the final thesis: Ing. Andrej Bednařík – Bratislava: FHI EU, 2025, number of pages 59.

The aim of this final thesis is to develop an intelligent inventory management system for retail using Python programming language. By analyzing historical sales data and forecasting future demand, the system helps retailers optimize their inventory more effectively, minimize the risk of stockouts and overstocking, thus increasing their competitiveness and profitability. The data used for this thesis was obtained from a modified dataset named retail_sales.csv, which was created by combining publicly available information and artificial intelligence in order to simulate realistic sales behavior in a retail environment. The outcome of the thesis is a system that enables demand forecasting for a 12-month period. Based on the estimated demand, it calculates the economic order quantity, reorder point and safety stocks, which are key metrics for inventory optimization. The added value of the thesis lies in the application of machine learning methods to improve decision making in inventory management, which contributes to better resource utilization and increased competitiveness of retailers.

Key words: inventory management, demand forecasting, machine learning, EOQ, XGBoost, Python

Obsah

Zoznam obrázkov	9
Úvod	11
1. Súčasný stav riešenej problematiky doma a v zahraničí	12
1.1. Riadenie zásob.....	12
Príznaky zlého riadenia zásob	13
1.2. Metódy riadenia zásob.....	13
1.2.1. <i>ABC analýza</i>	13
1.2.2. <i>JIT</i>	14
1.2.3. <i>EOQ</i>	15
2. Predikcia dopytu.....	19
2.1. Predikcia využitím časových radov	19
2.2. Strojové učenie (Machine learning)	20
2.2.1. <i>Regresia</i>	22
2.2.2. <i>Klasifikácia</i>	22
2.2.3. <i>Modely strojového učenia</i>	22
3. Cieľ práce, metodika práce a metódy skúmania.....	25
4. Výsledky práce	27
4.1 Popis a úprava dát.....	27
4.2 Popis kódu	29
4.2.1 <i>Importovanie knižníc</i>	29
4.2.2 <i>Funkcia na načítanie a agregáciu údajov</i>	29
4.2.3 <i>Funkcia na doplnenie prázdnych mesiacov</i>	30
4.2.4 <i>Funkcia na predspracovanie dát</i>	31
4.2.5 <i>Optimalizácia hyperparametrov modelu XGBoost</i>	31
4.2.6 <i>Tréning, predikcia, výpočet optimalizačných metrik a metrik hodnotenia modelu</i> ...	33
4.2.7 <i>Spúšťacia časť programu a zobrazenie výsledkov</i>	35

4.3	Výsledky optimalizácie zásob a presnosť modelu	37
4.4	Výsledky predikcie dopytu jednotlivých produktov	39
5.	Diskusia	54
	Záver	56
	Zoznam použitej literatúry	58

Zoznam obrázkov

Obrázok 1 Pôvodný dataset	27
Obrázok 2 Upravený dataset.....	28
Obrázok 3 Importované knižnice.....	29
Obrázok 4 Načítanie a agregácia dát	30
Obrázok 5 Doplnenie prázdnych mesiacov	30
Obrázok 6 Predspracovanie dát	31
Obrázok 7 Optimalizácia hyperparametrov	32
Obrázok 8 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť 1.....	34
Obrázok 9 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť 2.....	34
Obrázok 10 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť 3.....	35
Obrázok 11 Spúšťacia časť programu a vykreslenie stĺpcového diagramu	36
Obrázok 12 Vykreslenie grafu priebehu dopyty pre jednotlivé produkty.....	36
Obrázok 13 Výsledky optimalizačných metrík pre jednotlivé produkty a chyby metrík.....	37
Obrázok 14 Graf s výsledkami optimalizačných metrík pre jednotlivé produkty.....	39
Obrázok 15 Predikcia dopytu pre korektor.....	40
Obrázok 16 detailná predikcia pre korektor	40
Obrázok 17 Predikcia dopytu pre produkt laptop.....	41
Obrázok 18 Detailná predikcia pre laptop	41
Obrázok 19 Predikcia dopytu pre produkt letné šaty.....	42
Obrázok 20 Detailná predikcia pre letné šaty.....	42
Obrázok 21 Predikcia dopytu pre produkt linka.....	43
Obrázok 22 Detailná predikcia pre linku.....	43
Obrázok 23 Predikcia pre produkt opaľovací krém.....	44
Obrázok 24 Detail predikcie pre opaľovací krém.....	44
Obrázok 25 Predikcia pre produkt parfém.....	45
Obrázok 26 Detail predikcie pre parfém	45
Obrázok 27 Predikcia pre produkt plet'ový krém	46
Obrázok 28 Detailná predikcia pre plet'ový krém	46
Obrázok 29 Predikcia pre produkt rúž.....	47

Obrázok 30 Detailná predikcia pre rúž	47
Obrázok 31 Predikcia pre produkt smart hodinky	48
Obrázok 32 Detailná predikcia pre smart hodinky	48
Obrázok 33 Predikcia pre produkt tablet	49
Obrázok 34 Detailná predikcia pre tablet	49
Obrázok 35 Predikcia pre produkt tenisky	50
Obrázok 36 Predikcie pre produkt Tričko	50
Obrázok 37 Detailná predikcia pre tenisky	50
Obrázok 38 Detailná predikcia pre tričko	51
Obrázok 39 Predikcie pre produkt wireless earbuds	51
Obrázok 40 Predikcie pre produkt zimná bunda	52
Obrázok 41 Detailná predikcia pre wireless earbuds	52
Obrázok 42 Detailná predikcia pre zimná bunda	52
Obrázok 43 Predikcia pre produkt šampón	53
Obrázok 44 Detailná predikcia pre šampón.....	53

Úvod

Efektívne riadenie zásob je v dnešnej dobe nevyhnutnou súčasťou každého úspešného maloobchodu. V prostredí, ktoré sa vyznačuje dynamikou, rýchlo meniacim sa dopytom a správaním zákazníkov a zároveň prítomnosťou silnej konkurencie, je správne načasovanie a množstvo objednaného tovaru veľmi dôležité. Zásoby sú síce nevyhnutné pre každý maloobchod, avšak sú spojené s vysokými nákladmi pre podnik (Chopra & Meindl, 2007). Neefektívne riadenie zásob vedie k zvýšeným nákladom na skladovanie, viazanie kapitálu, ale pri ich nedostatku predstavuje zameškané príležitosti predaja, znižuje úroveň spokojnosti spotrebiteľov a môže negatívne ovplyvniť reputáciu podniku (Shenoy & Rosas, 2017). Cieľom optimalizácie zásob je snaha o nájdenie rovnováhy medzi dvoma protichodnými požiadavkami a to, minimalizáciou nákladov a maximalizáciou uspokojenia potrieb zákazníkov (Slack et al., 2010). Tradičné metódy riadenia zásob, ktoré boli v minulosti vysoko efektívne, dnes už nemusia byť postačujúce. V dôsledku rozvoja moderných technológií a nárastu výpočtových kapacít sa čoraz viac pristupuje k využívaniu strojového učenia v oblasti riadenia zásob. (Pandey et al., 2025) Algoritmy strojového učenia dokážu rýchlo pracovať s veľkými množstvami dát, učiť sa z historických vzorcov, zvládajú aj komplexnejšie nelineárne vzory a prispôbiť sa aj menej predvídateľnému správaniu zákazníkov (Zhou, 2021). Ich flexibilita, schopnosť sa učiť a prispôbiť vzorom v dátach ich robí vhodným nástrojom na riešenie úloh v dynamicky meniacich sa trhových podmienkach (Rebala et al., 2019).

Táto bakalárska práca sa zameriava na vývoj inteligentného systému pre riadenie zásob v maloobchode s využitím moderných technológií a algoritmov strojového učenia implementovaných v jazyku Python. Navrhnutý systém využíva pokročilé metódy predikcie na odhad dopytu na 12 mesiacov pomocou modelu strojového učenia XGBoost, ktorý je výkonný regresný algoritmus. Taktiež na základe predikcií počíta kľúčové metriky pre optimalizáciu zásob, ktoré v tejto práci sú ekonomická veľkosť objednávky, bod objednania a úroveň bezpečnostnej zásoby. Prínos práce je vytvorenie praktického a efektívneho nástroja pre maloobchody, ktorý môže slúžiť na zlepšenie rozhodovania v oblasti zásobovania. Na základe presnejších odhadov dopytu do budúcnosti a vypočítaných metrík prispieva k lepšej informovanosti v danej problematike. Kvalita predikčného modelu je vyhodnocovaná pomocou štatistických metrík hodnotenia modelu, čo zaručuje kontrolu spoľahlivosti systému pri reálnom nasadení v podniku.

1. Súčasný stav riešenej problematiky doma a v zahraničí

Riadenie zásob je významné pre každý podnik, ale najmä pre maloobchody, ktorých skladovanie a distribúcia produktov je ich neoddeliteľná súčasť. V súčasnosti sa doma aj v zahraničí využívajú rôzne prístupy a technológie na riadenie zásob. Tradičné metódy, ako napríklad ABC analýza, Just-In-Time (JIT) a EOQ (Economic Order Quantity), sú stále široko používané pre ich jednoduchosť a transparentnosť (Mohammed et al., 2024). V posledných rokoch sa začínajú viac presadzovať moderné spôsoby, ktoré využívajú pokročilé štatistické modely a algoritmy strojového učenia, ktoré umožňujú presnejšiu predikciu dopytu.

Využívanie pokročilých metód analýzy údajov umožňuje podnikom dosiahnuť výraznú konkurenčnú výhodu, rýchlejšie reagovať na zmeny a lepšie sa prispôbiť volatilitě trhu (Dorota-Owczarek, 2024). Pre lepšie pochopenie tejto problematiky je dôležité si vysvetliť niekoľko pojmov.

1.1. Riadenie zásob

Riadenie zásob je proces sledovania a kontroly stavu zásob, ako aj zabezpečenia ich primeraného doplnovania s cieľom uspokojiť požiadavky zákazníkov. Stanovenie vhodnej úrovne zásob je rozhodujúce, pretože zásoby viažu dostupné finančné prostriedky a ovplyvňujú výkonnosť podniku. Príliš veľa zásob znižuje pracovný kapitál a ovplyvňuje likviditu spoločnosti. Naopak, mať príliš málo zásob vedie k vypredaniu zásob a zmeškaným predajným príležitostiam, čo vedie k nižšiemu zisku. Preto by sa manažment mal zamerať na hľadanie rovnováhy pri udržiavaní optimálneho stavu zásob, ktorý zabezpečí vysokú spokojnosť zákazníkov, minimalizuje riziko nedostatku produktov, a zároveň umožní dosiahnuť čo najnižšie náklady na skladovanie (Priniotakis & Argyropoulos, 2018).

Najčastejšie vyskytujúce sa problémy v riadení zásob:

- **Nadmerná zásoba:** nastáva vtedy keď podnik drží viac zásob na sklade ako dokáže predat' v primeranom čase. To spôsobuje viazanie kapitálu a zvyšuje náklady na skladovanie. Rizikom je aj to, že zásoby budú zastarané a v najhoršom prípade sa môžu pokaziť (Sunol, 2024).
- **Nedostatočná zásoba:** nastáva vtedy, keď podnik nemá dostatočné zásoby, čo vedie k premeškaniu predajných príležitostí a nespokojnosti zákazníkov. Nespokojní

zákazníci môžu negatívne ovplyvniť budúce predaje. Najčastejšie sa nedostatočná zásoba objavuje počas obdobia vrcholového predaja napríklad počas sviatkov (Alina, 2024).

- **Slabá prognóza dopytu:** nepresné predpovede môžu viesť k nadbytku alebo nedostatku. Bez spoľahlivých údajov a metód prognózovania môžu podniky mať problémy s udržiavaním optimálnej úrovne zásob, ktorá by zaistila dostupnosť zásob pre zákazníkov (Alina, 2024).
- **Neefektívne procesy dopĺňania zásob:** Bez dynamických procesov a jasného prehľadu o úrovni zásob môžu podniky čeliť oneskoreniam pri dopĺňaní zásob, čo vedie k vypredaniu zásob a zmeškaným predajným príležitostiam. Navyše, manuálne procesy môžu byť náchylné na ľudskú chybu, čo vedie k nepresným záznamom zásob a následne nesprávnemu riadeniu zásob (Sunol, 2024).

Príznaky zlého riadenia zásob:

Medzi príznaky zlého riadenia zásob patrí napríklad:

- stúpajúce množstvo nevybavených objednávok
- zvyšujúci sa viazaný kapitál v zásobách, zatiaľ čo množstvo nevybavených objednávok neklesá
- rastúca frekvencia zrušených objednávok
- častá obmena zákazníkov
- príliš veľa zastaraných položiek na sklade
- pravidelne nedostatočná skladovacia kapacita (Dupaľ, 2018).

1.2. Metódy riadenia zásob

Modely riadenia zásob predstavujú kľúčový prvok optimalizácie dodávateľského reťazca, pričom ich hlavným cieľom je minimalizácia celkových nákladov spojených s objednávaním, skladovaním a potenciálnymi stratami z nedostatku tovaru.

1.2.1. ABC analýza

ABC analýza je metóda riadenia zásob, ktorá klasifikuje položky na základe ich dôležitosti a hodnoty, aby sa určilo, koľko úsilia a kontroly je potrebné na ich efektívne

riadenie. Princíp tejto analýzy vychádza z „80–20 pravidla“ alebo Pareto analýzy. Tento koncept vychádza z objavu Vilfreda Pareta, že 20% populácie vlastní 80% majetku, čo sa dá preniesť aj na riadenie zásob – 20% položiek často vyžaduje 80% pozornosti, zatiaľ čo zvyšných 80 % položiek potrebuje iba 20% kontroly.

ABC analýza delí zásoby do troch kategórií:

- **A-položky:** Predstavujú najhodnotnejšie položky, ktoré si vyžadujú špeciálnu starostlivosť. Pre tieto položky manažéri občasne využívajú automatizovaný systém, no rozhodnutia o ich riadení si vyžadujú dôkladné preskúmanie.
- **B-položky:** Položky strednej hodnoty, ktoré sa spravidla riadia bežnými štandardnými opatreniami. Väčšinou sa kontrolujú pomocou automatizovaného systému.
- **C-položky:** Najlacnejšie položky, ktoré sú skladom vo veľkom množstve a vyžadujú minimálnu kontrolu, často sú riadené postupmi prispôbenými konkrétnej situácii.

Cieľom je, aby sa zdroje sústredili na položky, ktoré sú pre podnik najdôležitejšie. Aj keď C-položky nie sú hodnotné, môže byť užitočné udržiavať ich na sklade kvôli ich skrytému významu. Napríklad, lacné súčiastky, ako sú náhradné diely, môžu byť dôležité pre plynulosť výroby, alebo drobné súčasti, ktoré sú potrebné na predaj hlavného produktu, napríklad stierače pre nové vozidlá. C-položky môžu byť tiež dôležité pre uspokojenie zákazníkov a môžu generovať vysoké zisky v pomere k ich nízkym nákladom.

Niektoré organizácie pridávajú k trom základným kategóriám aj štvrtú kategóriu:

- **D-položky:** Zásoby, ktoré sa používajú zriedka alebo sú „mŕtve“ a často sa uvažuje o ich vyradení zo systému.

ABC analýza tak umožňuje efektívnejšie riadenie zásob tým, že klasifikuje položky podľa ich hodnoty a dôležitosti a umožňuje podnikom optimalizovať kontrolné opatrenia pre jednotlivé kategórie položiek (Slack et al., 2010).

1.2.2. JIT

Systém zásob Just-in-time („práve včas“) je manažérska stratégia, ktorá zosúladí objednávkami surovín priamo od dodávateľov s výrobnými plánmi. „Práve včas“ znamená, že podniky dostanú tovar čo najbližšie k tomu kedy ho budú potrebovať. Podniky, ktoré využívajú túto zásobovaciu stratégiu zvyšujú efektívnosť a znižujú odpad prijímaním tovaru

len vtedy ak ho potrebujú na výrobu, čo znižuje náklady na skladovanie. Táto metóda vyžaduje presne odhadnutý dopyt od výrobcov. Maloobchodníci, reštaurácie, technologická výroba a výroba automobilov sú príkladmi odvetví, v ktorých je tento systém efektívny (Banton, 2025).

1.2.3. EOQ

Optimálna veľkosť objednávky alebo inak aj EOQ (Economic Order Quantity) je jeden z najzákladnejších a najjednoduchších modelov riadenia zásob v podniku. Model odpovedá na otázku „koľko by sme mali objednať?“, aby súčet nákladov na objednávanie a skladovanie bol minimálny (Shenoy & Rosas, 2017). Vznikol už pred niekoľkými desaťročiami, avšak dodnes sa využíva pre jeho jednoduchosť a komplexitu. Aby základný model bol využiteľný, treba vychádzať z niekoľkých predpokladov napríklad:

- konštantný a známy dopyt po produkte
- konštantné obstarávacie náklady na jednotku
- rovnomerné čerpanie zásob zo skladu
- konštantné a známe náklady na objednávanie a prepravu
- obstarávacia lehota je konštantná
- celá objednávka prichádza v jednej zásielke
- nie je možné sa dostať do deficitu

Odvodenie EOQ:

- Počet objednávok za rok:

$$N = \frac{D}{Q}, \quad (1)$$

kde D predstavuje dopyt za rok a Q je veľkosť ekonomickej objednávky.

- Ročné objednávkové náklady:

$$\frac{D}{Q} C_0, \quad (2)$$

kde D je ročný dopyt, Q je veľkosť ekonomickej objednávky a C_0 sú náklady na objednanie jednej objednávky.

- Ročné skladovacie náklady:

$$\frac{Q}{2} iC, \quad (3)$$

kde $\frac{Q}{2}$ je priemerná zásoba počas roka, je iC sú ročné náklady na skladovanie jednej jednotky.

- Ročné náklady na nákup tovaru:

$$D \times C, \quad (4)$$

kde D je ročný dopyt a C cena jednotky tovaru.

- Súčet celkových nákladov za rok:

$$TIC = \frac{D}{Q} C_0 + \frac{Q}{2} iC + D * C \quad (5)$$

Na nájdenie optimálneho Q^* minimalizujeme $TIC(Q)$. Diferencovaním podľa Q :

$$\frac{d(TIC)}{d(Q)} == \frac{D}{Q} C_0 + \frac{Q}{2} iC + D * C = 0 \quad (6)$$

Úpravou a zjednodušením získame:

$$Q^2 = \frac{2DC_0}{iC} \quad (7)$$

Vzorec na výpočet základného modelu EOQ:

$$Q^* = \sqrt{\frac{2DC_0}{iC}}, \quad (8)$$

kde Q^* je veľkosť ekonomickej objednávky, D je dopyt (počet jednotiek za určité časové obdobie), C_0 sú náklady na objednanie pre jednu objednávku (dopravné, administratívne náklady,...), iC sú náklady na skladovanie jednej jednotky (za určité časové obdobie) (Shenoy & Rosas, 2017).

Ďalšie modely EOQ:

Okrem základného modelu EOQ, ktorý je postavený na mnohých predpokladoch existujú aj rozšírené varianty modelu, ktoré tieto predpoklady čiastočne eliminujú. Vďaka svojej robustnosti a možnosti úprav je často využívaný vo firmách dodnes na optimalizáciu zásob. Takéto modely napríklad sú:

- **model zásob EOQ s možnosťou vyčerpania zásob** - zohľadňuje náklady spojené s oneskoreným dodaním zásob
- **model s zásob s diskontom** - zohľadňuje množstevné zľavy od dodávateľa nad určité objednané množstvo (Shenoy & Rosas, 2017).

Výhody EOQ:

Minimalizácia nákladov na skladovanie

- EOQ pomáha znižovať náklady spojené s držaním zásob tým, že určí optimálne množstvo objednávky, čím sa minimalizuje nadbytočné skladované množstvo

Minimalizácia objednávkových nákladov

- Optimalizáciou veľkosti objednávky sa znižuje počet objednávok a tým aj náklady spojené s ich realizáciou, čo prispieva k efektívnejšiemu hospodáreniu s financiami.

Zvýšená efektivita

- EOQ poskytuje presné odporúčania pre optimálne úrovne zásob a načasovanie opätovného objednávanie, čo zjednodušuje rozhodovací proces a prispieva k lepšiemu riadeniu dodávateľského reťazca.

Zlepšenie správy zásob pre konkrétny podnik

- Použitím EOQ modelu získava podnik presný údaj o tom, kedy a koľko objednať na základe ich vlastných podmienok, v ktorých podnik pôsobí, vďaka tomu môžu optimalizovať svoju stratégiu pre riadenie zásob.

Zníženie rizika nedostatku zásob

- Vypočítané optimálne objednávkové množstvo pomáha predchádzať nedostatku zásob, čo zvyšuje spoľahlivosť dodávok a spokojnosť zákazníkov.

Možnosť využiť množstevné zľavy

- Pri správnom plánovaní objednávok môžu firmy profitovať z diskontov, ktoré dodávatelia poskytujú pri väčších objednávkach, čo môže viesť k dodatočným úsporám.

Nevýhody základného modelu EOQ:

Komplexné matematické výpočty

- EOQ model je založený na zložitejších algebrických výpočtoch, čo môže byť problémom najmä pre malé podniky bez pokročilých matematických znalostí alebo bez prístupu k sofistikovaným analytickým nástrojom.

Predpoklady a obmedzenia modelu

- Model má viaceré predpoklady ako napríklad konštantný dopyt, okamžitú dodáciu dobu a fixné náklady, čo nie vždy zodpovedá realite, najmä v prípade sezónnych alebo ekonomických výkyvov. Tieto predpoklady môžu viesť k odchýlkam v optimálnom množstve objednávky.

Obmedzená aplikovateľnosť pre viaceré produkty

- Základný EOQ model je primárne určený pre jedno produktové systémy, čo znamená, že pri správe portfólia viacerých produktov môže byť potrebné hľadať alebo implementovať rozšírené varianty modelu (Forrest, 2025).

2. Predikcia dopytu

Dopyt je formálne definovaný ako to, čo zákazníci chcú, kedy to chcú a koľko toho chcú. Je to neobmedzený uhol pohľadu, napríklad zákazníci môžu chcieť tovar aj teraz keď nie je na sklade. Predikcia dopytu predstavuje dôležitý proces v oblasti riadenia zásob, ktorý umožňuje organizáciám efektívne plánovať nákup, výrobu a distribúciu produktov. Je založená na analýze historických dát, trhových trendov a iných relevantných faktorov s cieľom predpovedať budúce požiadavky zákazníkov (Vandeput, 2023).

Metódy predikcie dopytu možno rozdeliť na:

- **Kvalitatívne metódy** - sa spoliehajú na odborný úsudok a subjektívne hodnotenia, keď sú historické údaje nedostupné alebo nie je možné ich efektívne použiť. Výhodou týchto metód je rýchla aplikácia a nízke náklady, avšak sú náchylné na chyby spôsobené zaujatím a nedostatočnou objektivitou (W. Chase, 2013).
- **Kvantitatívne metódy** - využívajú historické časové rady, matematické a štatistické modely na extrapoláciu budúcich hodnôt. Kvantitatívne prístupy sú systematické, objektívne a umožňujú modelovať vzory ako trend, sezónnosť a cyklickosť (W. Chase, 2013).

2.1. Predikcia využitím časových radov

Jedna z najpoužívanejších kvantitatívnych metód pri predikcii dopytu je využitie modelov časových radov. Časový rad je súbor pozorovaní x_t , kde každé z nich je zaznamenané v čase t (Brockwell & Davis, 2016). Modely časových radov analyzujú na historické vzory dopytu ako trend a sezónnosť a premietajú ich do budúcnosti. Aby to dokázali väčšinou ich rozkladajú na tri pod zložky:

- level - priemerná hodnota okolo ktorej sa dopyt v čase mení,
- trend - predstavuje konzistentnú zmenu dopytu z jedného časového obdobia na ďalšie,
- sezónnosť - sezónne faktory určujú ako je dopyt rozdelený v opakujúcom sa časovom období ako deň, týždeň alebo rok (Vandeput 2023).

ARIMA - AutoRegressive Integrated Moving Average

Model ARIMA je označený ako ARIMA(p,d,q), kde p je počet autoregresii, teda počet oneskorených hodnôt zaradených do modelu. Hodnota d reprezentuje počet diferencovaní potrebných na dosiahnutie stacionarity. Počet členov pohyblivého priemeru predstavuje q . Pričom všetky parametre musia byť nezáporné (Fattah et al. 2018). Štatistické modely ako ARIMA poskytujú efektívne riešenia pri predikcii časových radov, avšak ich schopnosť zachytiť komplexnejšie a nelineárne vzťahy je obmedzená. Preto sa na predikciu čoraz častejšie využíva strojové učenie, ktoré dokázu zachytiť aj zložitejšie vzory a dosahujú väčšiu presnosť. Model ARIMA je robustný a jeden z najvyužívanejších štatistických nástrojov na analýzu a predikciu časových radov. Model sa skladá z troch hlavných komponentov:

- Autoregresia (AR) - vyjadruje lineárnu závislosť aktuálnej hodnoty premennej a predošlých (oneskorených) hodnôt
- Integrácia (I) - slúži na dosiahnutie stacionarity časových radov, čo je kľúčová charakteristika pohľadu štatistickej analýzy časových radov. Dosahuje sa pomocou diferencovania prvého rádu, ktoré väčšinou spočíva odčítaním predchádzajúcej hodnoty od aktuálnej.
- Pohyblivý priemer (MA) - vyjadruje lineárnu závislosť aktuálnej hodnoty chyby a jednej alebo viacerých chýb z predchádzajúcich pozorovaní

2.2. Strojové učenie (Machine learning)

Strojové učenie je technika, ktorá zlepšuje výkon systému učením zo skúseností prostredníctvom výpočtových metód. V počítačových systémoch sa skúsenosť prejavuje vo forme údajov a hlavným cieľom strojového učenia je vyvinúť algoritmy, ktoré na základe týchto dát konštruujú modely. Poskytnutím údajov zo skúseností učiacemu algoritmu získame model, ktorý je schopný robiť predikcie. Ak považujeme informatiku za disciplínu zameranú na algoritmy, potom strojové učenie predstavuje oblasť venovanú algoritmom, ktoré sa učia zo skúseností (Zhou, 2021).

Strojové učenie má tri podkategórie:

1. Učenie s učiteľom (Supervised Learning) - V učení s učiteľom je „stroju“ poskytnutý dataset inak povedané súbor dátových bodov, spolu so správnymi odpoveďami k dátovým bodom. Učiaci algoritmus dostáva rozsiahly súbor s dátovými bodmi

a odpoveďami, tzv. označený dataset. Algoritmus sa musí naučiť kľúčové charakteristiky pre každý dátový bod v datasete na určenie odpovedi. Aby keď dostane nový dátový bod, vzhľadom na kľúčové charakteristiky, bol schopný predikovať správny výsledok alebo odpoveď. Napríklad, keď dáme programu tisíce obrázkov so správnymi odpoveďami (napríklad mačka, pes, auto, lietadlo). „Stroj“ sa „naučí“ rozpoznávať charakteristiky v obrázkoch a rozoznať psa od mačky a lietadla. (Rebala et al., 2019)

2. Učenie bez učiteľa (Unsupervised Learning) - Pri učení bez učiteľa je „stroju“ (algoritmu) poskytnutý dataset bez žiadnych správnych odpovedí. Vzhľadom na obrovské množstvo údajov by algoritmus mal byť schopný nájsť vzory a štruktúru, trendy podobnosti. Na základy týchto trendov podobnosti algoritmus vytvorí klustre alebo skupiny podobných prvkov a zisťuje podobnosť medzi novým prvkom a existujúcou skupinou. Napríklad algoritmu dáme dáta o vzorcoch nákupného správania zákazníkov v potravinách. Časom dokáže identifikovať nákupné vzore ako napríklad tí čo nakupujú školské tašky, kúpia aj farebné ceruzky a tí čo kúpia pivo kúpia aj čipsy. „Stroj“ nechápe čo tieto objekty presne sú. Avšak dokáže nájsť medzi nimi spojenie. (Rebala et al., 2019)
3. Učenie posilňovaním (reinforced learning) - Ide o prístup, ktorý trénuje „stroje“ pomocou metódy pokus omyl tým, že zavádza systém odmien. To motivuje vybrať najlepšiu možnú akciu. Zväčša sa využíva na hranie hier, šachu a iných (Rebala et al., 2019).

V strojovom učení je dôležité nájsť rovnováhu medzi presnosťou modelu na tréningových dátach a jeho schopnosťou presne predikovať nové, neznáme dáta. Tento problém je úzko spojený s pojmami overfitting (preučenie) a underfitting (podučenie), ktoré výrazne ovplyvňujú schopnosť presnosti predikcie modelu. Overfitting nastáva, ak je model príliš prispôsobený tréningovým dátam. Výsledkom je, že dosahuje výborné výsledky pri tréningových dátach ale zlyháva pri predikcii, pretože si pamätá údaje namiesto toho, aby sa z nich naučil vzory. Väčšinou keď model trpí preučení nastáva to lebo bolo použitých priveľa parametrov v porovnaní s množstvom pozorovaní. Underfitting je opačným problémom, kde model používa príliš málo parametrov na zachytenie zložitosti dát. Vedie to k tomu, že model nedokáže zachytiť ani základné vzory v dátach. Dosahuje slabé výsledky už počas tréningovania a ešte horšie pri predikcii (Massaron & Boschetti, 2016). Je veľmi

dôležité sa vyhnúť týmto extrémom pri parametrizácii modelu, aby model dosahoval presné a využiteľné výsledky.

2.2.1. Regresia

Regresná analýza je štatistická metóda, ktorá sa využíva aj v strojovom učení. Dovoľuje vytvoriť model na základe označeného datasetu (ako napríklad historické údaje pre akcie), ktorý následne využíva na predikciu budúcich hodnôt. Preto sa radí pod kategóriu učenie s učiteľom. Regresia je spôsob učenia sa z existujúcich údajov (môžu byť historické) v snahe nájsť medzi nimi vzťahy. Techniky regresnej analýzy pomáhajú modelovať medzi rôznymi premennými v dátach s cieľom predpovedať hodnotu konkrétnej premennej, ktorá nás zaujíma.

Existujú dva základné typy regresie:

- Lineárna regresia – predpovedá spojitú hodnotu, napríklad cenu akcie alebo dopyt,.
- Logistická regresia – predpovedá pravdepodobnosť, že sa niečo stane, napríklad či akcia prekročí určitú hodnotu (Rebala et al., 2019).

2.2.2. Klasifikácia

Klasifikácia predstavuje problém identifikácie kategórie, do ktorej patrí daný vstup spomedzi množiny možných kategórií. Množina kategórií je zvyčajne označená vopred a modely sa všeobecne učia z tréningových údajov. Klasifikačné modely môžu byť vytvorené pomocou jednoduchých prahových hodnôt, regresných metód alebo iných metód, ako sú rozhodovacie stromy, random forest, neurónové siete, a ďalšie. Klasifikácia patrí medzi úlohy učenia s učiteľom, kde sa model trénuje na správne označených dátach. Model, ktorý sa naučil z tréningových dát identifikáciu kategórie alebo triedu vstupného prvku, sa nazýva klasifikátor. Klasifikátor môže byť binárny alebo viactriedny. Binárny klasifikátor určuje, či vstup patrí do jednej alebo viac kategórií. Viactriedny klasifikátor identifikuje vstup do jednej z viacerých kategórií (Rebala et al., 2019).

2.2.3. Modely strojového učenia

Rozhodovacie stromy:

Rozhodovacie stromy predstavujú efektívny a intuitívny model strojového učenia, ktorý sa využíva pri riešení klasifikačných alebo regresných problémov. Predpovedajú výsledok na

základe postupného kladenia otázok o vlastnostiach objektov. Každý strom pozostáva z uzlov, ktoré predstavujú otázky (alebo premenné) a hrán, ktoré predstavujú odpovede (alebo hodnoty). Na konci sa nachádzajú listy (koncevé uzly) , ktoré predstavujú výsledok rozhodovacieho procesu (Rebala et al., 2019).

Random forest :

Random Forest je kombinovaný model pozostávajúci zo súboru rozhodovacích stromov, vďaka čomu je jeho predikcia spoľahlivejšia a presnejšia. Využíva techniku náhodného výberu trénovacej podmnožiny dát (tzv. bagging), na základe ktorej sú vytvárané rôzne rozhodovacie stromy. Vďaka náhodnému výberu údajov pre každý strom sa znižuje riziko preučenia modelu, keďže jednotlivé rozhodovacie stromy sú samostatne veľmi náchylné na preučenie. Výsledná predikcia sa v prípade klasifikácie určuje hlasovaním a v prípade regresie spriemerovaním odpovedí všetkých stromov (Rebala et al., 2019).

XGBoost:

Extreme Gradient Boost alebo v skratke XGBoost je kombinovaný model, ktorý využíva vysoko efektívne a optimalizované techniky gradientného posilňovania rozhodovacích stromov. Na rozdiel od klasických kombinovaných modelov, ktoré stavajú svoje stromy paralelne, ako napríklad Random Forest, XGBoost ich stavia postupne. XGBoost stavia každý nový strom až potom, ako sa naučí na chybách predošlého jednoduchšieho stromu, s cieľom minimalizovať chyby. Oproti ostatným boosting modelom má viaceré vylepšené vlastnosti, vďaka ktorým sa dostáva na vrchol. Jedna z vlastností je regulačný parameter, ktorý pomáha s komplexnosťou a zabraňuje modelu v preučení sa. Ďalšia významná vlastnosť zahŕňa paralelné výpočty, čo umožňuje modelu byť rýchlejší. XGBoost má taktiež zabudované vlastné spracovanie neúplných dáta a preto ak nejaké hodnoty chýbajú v datasete nie je potrebné ich odstraňovať, pretože si s nimi vie poradiť bez toho, aby ovplyvnili jeho kvalitu. Vďaka týmto a ďalším vlastnostiam patrí XGBoost medzi najvýkonnejšie a najvyužívanejšie modely pre regresiu a klasifikáciu aj pri práci s veľkými a robustnými datasetmi (Brownlee 2019).

LSTM - Long Short Term Memory:

Long Short Term Memory alebo v skratke LSTM je model, ktorý využíva špeciálny typ rekurentných neurónových sietí. Je navrhnutý tak, aby sa dokázal učiť dlhodobé závislosti v dátach. Táto vlastnosť je veľmi dôležitá pri modelovaní časových radov, kde je potrebné uchovávať informácie o predošliých stavoch alebo hodnotách počas dlhších období bez straty

presnosti. Kľúčom k fungovaniu LSTM sú tzv. brány a bunky, ktoré predstavujú špeciálny mechanizmus, ktorý riadi tok informácií v rámci siete. Bunky predstavujú hlavný pamäťový tok, ktorý uchováva informácie. Každá brána rozhoduje o tom, aké informácie ponechá a odošle ďalej alebo aké zabudne. Proces je riadený tromi typmi brán:

- Vstupná brána (input gate) - určuje, ktoré nové informácie sa uložia do vnútornej pamäte bunky
- Zabúdacia brána (forget gate) – rozhoduje, ktoré informácie budú vymazané z buniek
- Výstupná brána (output gate)- kontroluje, ktorá časť informácie z bunky bude odovzdaná na výstup

Vnútorň stav bunky sa aktualizuje kombináciou operácií sčítania a násobenia aby sa zachovala stabilita a sieť sa dokázala učiť aj z dávnych udalostí v sekvencii. Vďaka týmto vlastnostiam LSTM siete dokážu uchovávať dôležité informácie aj cez veľké množstvo krokov. Preto sú ideálne na predikciu časových radov (ako napríklad dopyt), spracovanie sekvenčných dát a všeobecne v úlohách, kde je dôležitá dlhodobá pamäť (Lazzeri, 2020).

3. Cieľ práce, metodika práce a metódy skúmania

Cieľom práce je vyvinúť inteligentný systém pre riadenie zásob v maloobchode s využitím programovacieho jazyka Python. Systém prostredníctvom analýzy historických predajných údajov a predikcie budúceho dopytu pomôže maloobchodníkom efektívnejšie optimalizovať zásoby, minimalizovať riziko nedostatku a prebytku tovaru, a tým zvýšiť ich konkurencieschopnosť a ziskovosť. Čiastkové ciele zahŕňajú spracovanie predajných údajov, tvorbu sezónnych príznakov a oneskorených hodnôt, implementácia predikčného modelu, overenie kvality predikcie modelu, výpočet optimalizačných metrík pre riadenie zásob a vizualizácia výsledkov.

Spracované informácie v tejto bakalárskej práci pochádzajú z literatúry týkajúcej sa riadenia zásob, strojového učenia a algoritmov strojového učenia v jazyku Python. Na spracovanie teoretickej časti boli použité knihy, vedecké články a pomocou metódy selekcie vybrané dôležité informácie z literatúry, prevažne zo zahraničných zdrojov.

Pri vývoji algoritmu sme najprv začali prípravou dát. Dáta využité na túto prácu pochádzajú z verejného zdroja, stránky kaggle.com a boli upravené pomocou umelej inteligencie aby boli vhodnejšie na vytvorenie a testovanie algoritmu pre predikciu dopytu na 12 mesiacov s výpočtom optimalizačných metrík na riadenie zásob. Dáta predstavujú predaje produktu v daný deň. Vývoj systému sme začali načítaním a agregovaním dát pomocou knižnice Pandas v Pythone, ktorá umožňuje načítavanie súborov v tabuľkovej forme a prácu s nimi. Dáta boli agregované na mesačnú úroveň nakoľko predikcia je na mesačnej báze. Následne dáta boli skontrolované a doplnené nulovými hodnotami v prípade, že nastala situácia, kde sa nejaký produkt v daný mesiac nepredal. Potom dáta prechádzali predspracovaním, ktoré v tomto prípade zahŕňa najprv logaritmickú transformáciu predaného množstva, ktorá zabezpečuje zníženie vplyvov extrémnych hodnôt. Následne vytvorenie cyklických premenných, ktoré sa využívajú v strojovom učení pre pomoci modelom pochopiť cyklickosť mesiacov napríklad, že po decembri nasleduje január. Ako posledné vytvárame oneskorené premenné na zachytenie autokorelácie, sezónnosti a na zlepšenie výkonu predikcie modelu. Ďalším krokom je nastavenie alebo inak povedané aj ladenie modelu pomocou optimalizácie hypermaterov. Hyperparametre sú parametre modelu, ktoré sa nastavujú pred jeho samotným tréningom. Ich správne nastavenie je kľúčové pretože ovplyvňuje výkonnosť modelu. Počet stromov je parameter, ktorý určuje koľko rozhodovacích stromov bude vytvorených. Vyšší počet stromov môže viesť k zlepšeniu

presnosti predikcie, no zároveň zvyšuje výpočtovú náročnosť. Naopak nižší počet stromov môže viesť k podučeniu modelu. Rýchlosť učenia určuje akou mierou sa každý nový strom podieľa na zlepšení predikcie. Nižšia hodnota vedie k lepším predikciám pretože sa model učí pomalšie ale je potrebný väčší počet stromov. Príliš vysoká hodnota môže spôsobiť preučenie sa modelu. Maximálna hĺbka stromu určuje počet úrovní, ktoré môže mať jeden rozhodovací strom. Nízka hĺbka, znamená plytšie stromy, ktoré môžu mať nedostatočnú kapacitu zachytiť komplexnejšie vzťahy v dátach. Hlboké stromy dokážu zachytiť aj detailnejšie vzory, ale sú náchylné na preučenie sa. Minimálny počet pozorovaní v liste predstavuje minimálny počet pozorovaní, ktoré musia byť prítomné v koncovom liste rozhodovacieho stromu. Ide o regulačný parameter, ktorý pomáha predísť k pretrénovaniu modelu. Optimalizácia hyperparametrov prebieha pomocou funkcie `RandomizedSearchCV`, ktorá náhodne vyberá kombinácie hyperparametrov v definovanom počte iterácií. V tejto práci zvolený model bol `XGBoost`, ktorý je veľmi populárny pre predikcie dopytu pomocou regresie.. Funkcia je súčasťou knižnice `skicit-learn`, ktorá je určená pre strojové učenie v Pythone. Následne prebieha samotné tréningovanie modelu, počas ktorého sa hodnotí aj výkonnosť modelu pre jednotlivé produkty. Na hodnotenie boli použité štatistické metriky chybovosti MAE, RMSE a MAPE. MAE (Mean Absolute Error) je priemer absolútnych hodnôt rozdielov medzi predikciami a skutočnými hodnotami. RMSE (Root Mean Square Error) je odmocnina z priemeru druhých mocnín rozdielov medzi predikciou a skutočnými hodnotami. MAPE (Mean Absolute Percentage Error) je priemer percentuálnych rozdielov medzi predikovanou a skutočnou hodnotou. Potom prebieha postupná predikcia dopytu, pričom odhadnuté hodnoty sa využili ako vstupy pre nasledujúce mesiace.

Na záver bola na základe vzorca pre klasický model EOQ vypočítaná optimálna veľkosť objednávky, bod objednania a bezpečnostná zásoba. Tieto metriky sú dôležité pre riadenie zásob, pretože umožňujú efektívnejšie hospodárenie so zdrojmi. Výsledné dáta sú zobrazené pomocou tabuľky ako výpis na konzole. Okrem toho dáta boli vizuálne zobrazené pomocou grafov. Pre optimalizačné metriky bol zvolený stĺpcový graf a na zobrazenie priebehu historických predajov a predikovaných hodnôt bol využitý časovo závislý graf. Na ich vykreslenie bola použitá knižnica `matplotlib` v Pythone, ktorá slúži najmä na tvorbu diagramov a grafov v 2D. Algoritmus bol vytvorený v jazyku Python (verzia 3.13.2), ktorý je často v praxi využívaný na strojové učenie a prácu s dátami. Pri vývoji sme použili vývojové prostredie od Microsoftu s názvom Visual Studio Code, ktoré je veľmi populárne na vývoj a programovanie.

4. Výsledky práce

V tejto kapitole sa bližšie poverujeme riešeniu zadanej problematiky, ktorou je vývoj inteligentného systému na optimalizáciu zásob pre maloobchodníkov pomocou analýzy historických údajov o predajoch. Implementácia a testovanie systému prebieha na dátach simulujúcich predaje v maloobchode, avšak nereprezentujú žiadny konkrétny reálny podnik. Potom si analyzujeme výsledky tohto systému, ktoré zahŕňajú vypočítané optimalizačné metriky zásob, štatistické metriky hodnotenia modelu a vizuálnu analýzu predikcií.

4.1 Popis a úprava dát

Dataset, ktorý bol využitý na túto prácu, pochádza zo stránky kaggle.com, jeho názov je retail-sales-dataset. Tento dataset bol vytvorený, aby čo najviac reflektoval skutočné predaje. Obsahuje polia: Transaction ID, Date, Customer ID, Gender, Age, Product Category, Quantity, Price per Unit, Total Amount, ako je možné vidieť nižšie na obrázku číslo 1. Dokopy obsahuje 1001 záznamov o predajoch, pričom všetky pochádzajú z roku 2023. Tento dataset bol dobrý ako východisko, pre dáta potrebné na riešenie zadanej problematiky, avšak v pôvodnej podobe nebol úplne vhodný na priame použitie pri riešení zadanej problematiky.

Obrázok 1 Pôvodný dataset

	A	B	C	D	E	F	G	H	I
1	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit	Total Amount
2	1	11/24/2023	CUST001	Male	34	Beauty	3	50	150
3	2	2/27/2023	CUST002	Female	26	Clothing	2	500	1000
4	3	1/13/2023	CUST003	Male	50	Electronics	1	30	30
5	4	5/21/2023	CUST004	Male	37	Clothing	1	500	500
6	5	5/6/2023	CUST005	Male	30	Beauty	2	50	100
7	6	4/25/2023	CUST006	Female	45	Beauty	1	30	30
8	7	3/13/2023	CUST007	Male	46	Clothing	2	25	50
9	8	2/22/2023	CUST008	Male	30	Electronics	4	25	100
10	9	12/13/2023	CUST009	Male	63	Electronics	2	300	600
11	10	10/7/2023	CUST010	Female	52	Clothing	4	50	200
12	11	2/14/2023	CUST011	Male	23	Clothing	2	50	100
13	12	10/30/2023	CUST012	Male	35	Beauty	3	25	75
14	13	8/5/2023	CUST013	Male	22	Electronics	3	500	1500
15	14	1/17/2023	CUST014	Male	64	Clothing	4	30	120
16	15	1/16/2023	CUST015	Female	42	Electronics	4	500	2000
17	16	2/17/2023	CUST016	Male	19	Clothing	3	500	1500
18	17	4/22/2023	CUST017	Female	27	Clothing	4	25	100
19	18	4/30/2023	CUST018	Female	47	Electronics	2	25	50
20	19	9/16/2023	CUST019	Female	62	Clothing	2	25	50

Zdroj: vlastné spracovanie

Dataset bolo potrebné modifikovať tak, aby neobsahoval len údaje o kategórii produktov, ale konkrétne produkty. Z každej z troch kategórií, Beauty, Electronics, Clothing,

bolo vybraných 5 reprezentatívnych produktov, čím vznikol výsledný dataset obsahujúci 15 produktov. Pre účely predikcie dopytu bolo nevyhnuté, aby dataset pokrýval dostatočne dlhé časové obdobie, ktoré umožňuje identifikáciu trendov a sezónnych výkyvov. Z tohto dôvodu bol rozšírený z jedného kalendárneho roku na obdobie od 2019 do 2024. Úpravy boli realizované s dôrazom na to, aby údaje čo najvernejšie reprezentovali skutočné predajné dáta v maloobchodnom prostredí a zároveň obsahovali sezónne výkyvy, ktoré sú pre danú problematiku kľúčové. Tieto kroky zabezpečili využiteľnosť datasetu pri modelovaní dopytu a zvýšili jeho analytickú hodnotu. Vzhľadom na rozsah úpravy dát a ich náročnosti boli na generovanie údajov použité nástroje generatívnej umelej inteligencie. Pomocou generatívnej AI boli najprv všetky kategórie vymenené za konkrétne produkty. A následne pomocou AI boli údaje rozšírené na dlhšie časové obdobie pričom dôraz bol kladený na realistickú simuláciu predajného správania v maloobchodnom prostredí. Po vygenerovaní údajov bol celý dataset ručne zrevidovaný s cieľom zvýšiť konzistenciu a vierohodnosť údajov. Na obrázku číslo 2 je možné vidieť náhľad na tento upravený dataset. Demografické údaje (pohlavie a vek) boli odstránené vzhľadom, keďže neboli relevantné pre riešenie problematiku.

Obrázok 2 Upravený dataset

	A	B	C	D	E	F
1	Date	Product	Quantity	Price per Unit	Total Amount	
2	1/1/2018	Smart hodinky	6	254.8	1528.8	
3	1/1/2018	Tričko	5	17.83	89.15	
4	1/1/2018	Opalovací krém	4	21.23	84.92	
5	1/2/2018	Pleťový krém	9	21.23	191.07	
6	1/2/2018	Rúž	7	10.19	71.33	
7	1/2/2018	Šampón	15	12.74	191.1	
8	1/2/2018	Opalovací krém	1	21.23	21.23	
9	1/2/2018	Zimná Bunda	10	93.42	934.2	
10	1/3/2018	Pleťový krém	8	21.23	169.84	
11	1/3/2018	Tenisky	5	67.95	339.75	
12	1/3/2018	Parfém	2	67.95	135.9	
13	1/3/2018	Tablet	2	501.96	1003.92	
14	1/3/2018	Wireless Earbuds	5	127.4	637	
15	1/4/2018	Opalovací krém	3	21.23	63.69	
16	1/4/2018	Wireless Earbuds	6	127.4	764.4	
17	1/4/2018	Tenisky	3	67.95	203.85	
18	1/5/2018	Parfém	4	67.95	271.8	
19	1/5/2018	Zimná Bunda	11	93.42	1027.62	
20	1/5/2018	Laptop	4	1274.02	5096.08	

Zdroj: vlastné spracovanie

4.2 Popis kódu

4.2.1 Importovanie knižníc

Na obrázku číslo 3 je začiatok kódu, kde prebieha import všetkých knižníc, ktoré sú potrebné na funkčnosť celého systému. Knižnica Pandas slúži na manipuláciu s údajmi v tabuľkových formátoch. Numpy je knižnica na prácu s poľami a numerickými operáciami a na vedecké výpočty. Matplotlib je knižnica na vizualizáciu 2D grafov. Scikit-learn knižnica je základnou knižnicou pre strojové učenie, z ktorej využívame RandomizedSearchCV a TimeSeriesSplit, ktoré si vysvetlíme v ďalších častiach kódu. Na záver importujeme knižnicu math, z ktorej si vyberáme funkciu sqrt, ktorá slúži na odmocniny.

Obrázok 3 Importované knižnice

```
1 import pandas as pd
2 import numpy as np
3 import xgboost as xgb
4 import matplotlib.pyplot as plt
5 from sklearn.model_selection import RandomizedSearchCV, TimeSeriesSplit
6 from sklearn.metrics import mean_absolute_error, mean_absolute_percentage_error
7 from math import sqrt
8
```

zdroj: vlastné spracovanie

4.2.2 Funkcia na načítanie a agregáciu údajov

Na obrázku 4 je funkcia, ktorá slúži na načítanie datasetu zo súboru vo formáte CSV a jeho následnú transformáciu a agregáciu na mesačnej úrovni. Funkcia pozostáva z niekoľkých krokov:

1. Načítanie údajov do data frame (df) pomocou metódy z knižnice pandas s volaním `pd.read_csv()`, kde v zátvorkách je vstupný parameter, premenná `file_path` v ktorej neskôr je uložená cesta k csv súboru. Vďaka tomuto program vie načítať dáta a pracuje s nimi vo vytvorenom dataframe.
2. Konverzia dátumu pomocou funkcie `pd.to_datetime()` konvertuje stĺpec `Date` v dataframe na dátumový typ (`datetime`). Tento krok je nevyhnutný na ďalšiu prácu s časovými údajmi, napríklad triediť, zoskupovať alebo filtrovať podľa dní, mesiacov a rokov.

3. Agregácia údajov na mesačnej úrovni a zároveň podľa produktu. Pomocou funkcií `df.groupby()` a `pd.Grouper()`. Funkcia `pd.Grouper` agreguje dáta na mesačnej úrovni a `df.groupby` podľa produktu. Pričom sa hneď počíta súčet predaného množstva (`Quantity`) pre každú kombináciu produktu a mesiaca.
4. Výstup pomocou funkcie `return df` vráti výsledný dataframe po všetkých úpravách.

Obrázok 4 Načítanie a agregácia dát

```

10 # 1) Načítanie a agregácia dát na mesačnú úroveň
11 def load_and_aggregate_data(file_path):
12     df = pd.read_csv(file_path)
13     df["Date"] = pd.to_datetime(df["Date"])
14     df = df.groupby([pd.Grouper(key="Date", freq="MS"), "Product"], as_index=False)["Quantity"].sum()
15     return df
16

```

Zdroj: vlastné spracovanie

4.2.3 Funkcia na doplnenie prázdnych mesiacov

Keďže všetky produkty sa nepredávajú rovnako, môže sa stať, že sa nejaký produkt v danom mesiac vôbec nepredal. Po agregovaní dát sa preto niektoré produkty v danom mesiaci vôbec nezobrazia. V takom prípade by to mohlo veľmi výrazne ovplyvniť analýzu hlavne pri časových radoch, preto je potrebné doplniť tieto chýbajúce produkty a priradiť im nulovú hodnotu k predanému množstvu. Túto úpravu zabezpečuje uvedená časť kódu na obrázku 5, ktorá nájde všetky možné kombinácie mesiacov a produktov v dataframe pomocou funkcie `all_combinations`. Ak nejaká kombinácia chýba automaticky ju doplní a priradí jej hodnotu `Quantity = 0`.

Obrázok 5 Doplnenie prázdnych mesiacov

```

17 # 2) Doplnenie chýbajúcich mesiacov nulovou hodnotou
18 def fill_missing_months(df):
19     product_categories = df["Product"].unique()
20     date_range = pd.date_range(df["Date"].min(), df["Date"].max(), freq="MS")
21     all_combinations = pd.MultiIndex.from_product(
22         [date_range, product_categories],
23         names=["Date", "Product"]
24     )
25     df = df.set_index(["Date", "Product"]).reindex(all_combinations, fill_value=0).reset_index()
26     return df
27

```

zdroj: vlastné spracovanie

4.2.4 Funkcia na predspracovanie dát

Funkcia zobrazená na obrázku 6 slúži na predspracovanie dát tak, aby boli vhodne pripravené na modelovanie pomocou algoritmov strojového učenia. Vykonáva niekoľko transformácií, ktoré zabezpečujú lepšiu analýzu a modelovanie. Najprv pomocou funkcie `sin_month` a `cos_month` sa cyklicky kódujú mesiace na zachytenie ich cyklického charakteru v roku. Tým sa zabezpečí, že mesiace ako napríklad december a január sú v modeli považované za blízke časové obdobia. Ďalším krokom je aplikácia logaritmickéj transformácie pomocou funkcie `np.log1p(df["Quantity"])`, ktorá zmiernuje vplyv extrémnych hodnôt na analýzu a ďalšiu predikciu. Táto transformácia je bežne využívaná pri modelovaní časových radov. V nasledujúcom kroku program vytvára `lag_features`, teda oneskorené hodnoty dopytu (`Quantity`), ktoré predstavujú dopyt v predchádzajúcich obdobiach. Konkrétne sa vytvárajú oneskorenia o 1, 2, 3, 6, 9 a 12 mesiacov. Tieto príznaky sú vysvetľujúce premenné a veľmi dôležité pri modelovaní. Umožňujú algoritmu využiť historický vývoj dopytu na predikciu budúceho správania. Na záver záznamy s chýbajúcimi hodnotami, ktoré mohli vzniknúť pri vytváraní oneskorení sú odstránené a výsledný dataframe je vrátený ako výstup.

Obrázok 6 Predspracovanie dát

```
28 # 3) Predspracovanie dát: log-transformácia, sezónne príznaky a lag features
29 def preprocess_data(df):
30     df["Month"] = df["Date"].dt.month
31     df["Year"] = df["Date"].dt.year
32     df["sin_month"] = np.sin(2 * np.pi * df["Month"] / 12)
33     df["cos_month"] = np.cos(2 * np.pi * df["Month"] / 12)
34     # Log-transformácia
35     df["Quantity"] = np.log1p(df["Quantity"])
36     lags = [1, 2, 3, 6, 9, 12]
37     df = df.sort_values(["Product", "Date"])
38     # lag features z log-transformovaných hodnôt
39     for lag in lags:
40         df[f"Quantity_Lag_{lag}"] = df.groupby("Product")["Quantity"].shift(lag)
41     df.dropna(inplace=True)
42     return df
```

Zdroj: vlastné spracovanie

4.2.5 Optimalizácia hyperparametrov modelu XGBoost

Funkcia na obrázku číslo 7 zabezpečuje optimalizáciu hyperparametrov modelu XGBoost a výber najlepšej konfigurácie pre predikciu časového radu konkrétneho produktu. Hyperparametre sú nastaviteľné hodnoty, ktoré sa definujú ešte pred samotným tréningom

modelu a majú zásadný vplyv na jeho učenie a výkonnosť. Po samotnom vytvorení funkčného modelu je dôležité vyskúšať viacero ich nastavení pred určením, z ktorých nastavení sa bude vyberať pre záverečné predikcie. V premennej `param_dist` sú uvedené jednotlivé hodnoty hyperparametrov, ktoré majú byť testované. Ide o:

- počet stromov (`n_estimators`),
- rýchlosť učenia (`learning_rate`),
- maximálnu hĺbku stromu (`max_depth`),
- minimálny počet pozorovaní na list (`min_child_weight`).

Model sa inicializuje pomocou `xgb.XGBRegressor()`, pričom je nastavený na regresnú úlohu (`objective='reg:squarederror'`). Následne `RandomizedSearchCV` vyberie náhodne kombinácie z `param_dist`. Je nastavený na 50 iterácií (`n_iter=50`), čo znamená, že náhodne vyberie 50 kombinácií, ktoré testuje. Výkonnosť modelov je vyhodnocovaná pomocou negatívnej hodnoty priemernej absolútnej chyby (`neg_mean_absolute_error`), ktorá sa v `scikit-learn` používa pre minimalizáciu. Na zabezpečenie konzistencie a reprodukovateľnosti výsledkov sa využíva nastavenie `random_state=42`. Po natrénovaní modelov na tréningových dátach funkcia `random_search.fit(X_train, y_train)` vyberie a vráti model s najlepšimi hyperparametrami, ktorý dosiahol najnižšiu priemernú absolútnu chybu.

Obrázok 7 Optimalizácia hyperparametrov

```
44 # 4) Optimalizácia hyperparametrov
45 def optimize_hyperparameters(X_train, y_train):
46     param_dist = {
47         'n_estimators': [ 100, 50, 60, 80, 90],
48         'learning_rate': [0.1, 0.2, 0.5],
49         'max_depth': [3, 5, 7],
50         'min_child_weight': [ 1, 2],
51     }
52
53     model = xgb.XGBRegressor(objective='reg:squarederror', random_state=42, verbosity=0)
54     random_search = RandomizedSearchCV(
55         estimator=model,
56         param_distributions=param_dist,
57         n_iter=50,
58         scoring='neg_mean_absolute_error',
59         verbose=0,
60         n_jobs=-1,
61         random_state=42
62     )
63     random_search.fit(X_train, y_train)
64     return random_search.best_estimator
```

Zdroj: vlastné spracovanie

4.2.6 Tréning, predikcia, výpočet optimalizačných metrik a metrik hodnotenia modelu

Funkcia na obrázkoch 8, 9 a 10 predstavuje najdôležitejšiu časť kódu, pretože zabezpečuje kompletný proces tréningu modelu, jeho validácie, tvorby predikcií, výpočtu metrik hodnotenia modelu a výpočtu optimalizačných metrik pre riadenie zásob.

Najprv sa pre každý produkt extrahujú dáta a overí sa, či je k dispozícii aspoň 24 záznamov. Ak nie, program upozorní používateľa výpisom na konzolu: „Produkt {Product} nemá dost' dát na predikciu!“ a daný produkt preskočí. Potom sa vyberú vstupné premenné, ktoré sú sezónne príznaky a oneskorené premenné. Pre každý produkt sa následne vytvorí model, ktorý je najprv optimalizovaný pomocou funkcie `optimize_hyperparameters()`, ktorá bola vysvetlená vyššie. Potom sa natrénuje na historických dátach. Následne sa použije časová krížová validácia pomocou funkcie `TimeSeriesSplit`, popri ktorej sa počítajú metriky hodnotenia modelu MSE, RMSE, MAPE. Po úspešnej validácii sa model natrénuje na celých dostupných dátach a použije sa na predpoveď dopytu. Model postupne predikuje dopyt na 12 mesiacov dopredu, pričom pri každom kroku využíva predchádzajúce predikcie ako vstup. Na záver sa pre každý produkt vypočítajú optimalizačné metriky pre riadenie zásob. Prvá je veľkosť optimálnej objednávky (EOQ), kde sa využíva klasický model, ktorého vzorec je:

$$EOQ = \sqrt{\frac{2 * \text{ročný dopyt} * \text{náklady na objednanie}}{\text{náklady na skladovanie}}},$$

kde ročný dopyt je počítaný ako priemerný mesačný dopyt vynásobený 12 aby spĺňal predpoklad klasického modelu o rovnomernom dopyte. Náklady na objednanie majú konštantnú hodnotu 25 a náklady na skladovanie (jednej jednoty na rok) majú hodnotu 3. Ďalšia metrika je bezpečnostná zásoba, ktorá určuje rezervné množstvo zásob potrebné na pokrytie nečakaných výkyvov v dopyte.

$$\text{Bezpečnostná zásoba} = \text{bezpečnostný faktor} * \sigma * \sqrt{\text{dodacia lehota}},$$

kde bezpečnostný faktor je 1,65 pre 95% spoľahlivosť, σ je štandardná odchýlka denného dopytu a dodacia lehota, ktorá je nastavená v tomto kóde na 7 dní. Posledná metrika je bod objednania, ktorá predstavuje úroveň zásob kedy je potrebné uskutočniť novú objednávku. Vzorec na výpočet je:

$$\text{Bod objednania} = \text{Priemerný denný dopyt} * \text{dodacia lehota} + \text{bezpečnostná rezerva}$$

Obrázok 8 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť 1

```
68 # 5) Tréning modelu, predikcia a výpočet opimalizačných metrík
69 def train_and_forecast(df):
70     results = []
71     forecast_dict = {}
72
73     for product in df["Product"].unique():
74         product_df = df[df["Product"] == product].copy()
75         product_df.sort_values("Date", inplace=True)
76
77         if len(product_df) < 24:
78             print(f"Produkt '{product}' nemá dost dát na predikciu!")
79             continue
80
81         features = ["sin_month", "cos_month"] + [f"Quantity_Lag_{lag}" for lag in [1, 2, 3, 6, 9, 12]]
82         X = product_df[features]
83         y = product_df["Quantity"]
84
85         tscv = TimeSeriesSplit(n_splits=5)
86         splits = list(tscv.split(X))
87         last_train_idx, last_test_idx = splits[-1]
88
89         X_train_cv, X_test_cv = X.iloc[last_train_idx], X.iloc[last_test_idx]
90         y_train_cv, y_test_cv = y.iloc[last_train_idx], y.iloc[last_test_idx]
91
92         model = optimize_hyperparameters(X_train_cv, y_train_cv)
93         model.fit(X_train_cv, y_train_cv)
94
95         y_pred_cv = model.predict(X_test_cv)
96         y_pred_cv_exp = np.expm1(y_pred_cv)
97         y_test_actual = np.expm1(y_test_cv)
98
99
100         mae = mean_absolute_error(y_test_actual, y_pred_cv_exp)
101         rmse = sqrt(np.mean((y_test_actual - y_pred_cv_exp)**2))
102         mape = mean_absolute_percentage_error(y_test_actual, y_pred_cv_exp)
103
104         model.fit(X, y)
```

Zdroj: vlastné spracovanie

Obrázok 9 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť 2

```
106 # Postupná predikcia na 12 mesiacov dopredu
107 last_date = product_df["Date"].iloc[-1]
108 forecast_dates = pd.date_range(start=last_date + pd.DateOffset(months=1), periods=12, freq='MS')
109 history = product_df["Quantity"].tolist() # log-transformované hodnoty
110 forecast_values = []
111 for forecast_date in forecast_dates:
112     month = forecast_date.month
113     sin_month = np.sin(2 * np.pi * month / 12)
114     cos_month = np.cos(2 * np.pi * month / 12)
115     try:
116         lag_1 = history[-1]
117         lag_2 = history[-2]
118         lag_3 = history[-3]
119         lag_6 = history[-6]
120         lag_9 = history[-9]
121         lag_12 = history[-12]
122     except IndexError:
123         break
124     input_features = pd.DataFrame([[sin_month, cos_month, lag_1, lag_2, lag_3, lag_6, lag_9, lag_12]], columns=features)
125     pred_log = model.predict(input_features)[0]
126     pred_val = np.expm1(pred_log) #revezria hodnot z log-transformacie
127     forecast_values.append(pred_val)
128     history.append(np.log1p(pred_val))
129
130 if len(forecast_values) < 6:
131     continue
132
133 avg_forecast = np.mean(forecast_values)
```

Obrázok 10 Tréning , predikcia , výpočet metrík hodnotenia modelu a metrík optimalizácie časť

```
137 #premenné na výpočet optimalizačných metrík
138 ordering_cost = 25.0
139 holding_cost = 3.0
140 lead_time = 7
141 safety_factor = 1.65
142
143 annual_demand = avg_forecast * 12
144 avg_daily_demand = avg_forecast / 30
145 actual_quantities = np.expm1(product_df["Quantity"])
146 std_daily_demand = actual_quantities.std() / np.sqrt(30)
147
148 #výpočet optimalizačných metrík
149 eoq = np.sqrt((2 * annual_demand * ordering_cost) / holding_cost)
150 safety_stock = safety_factor * std_daily_demand * np.sqrt(lead_time)
151 reorder_point = avg_daily_demand * lead_time + safety_stock
152
153 results.append({
154     "Produkt": product,
155     "Predikovaný dopyt (priemer 12M)": round(avg_forecast, 0),
156     "EOQ (optimálna objednávka)": round(eoq, 2),
157     "Bod objednania": round(reorder_point, 2),
158     "Bezpečnostná zásoba": round(safety_stock, 2),
159     "MAE ": round(mae, 2),
160     "RMSE ": round(rmse, 2),
161     "MAPE ": round(mape, 2)
162 })
163
164 forecast_dict[product] = {
165     "dates": forecast_dates,
166     "forecasts": forecast_values
167 }
168
169 return pd.DataFrame(results), forecast_dict
```

Zdroj: vlastné spracovanie

4.2.7 Spúšťačia časť programu a zobrazenie výsledkov

Záverečná časť kódu, zobrazená na obrázkoch 11 a 12, zabezpečuje spustenie všetkých funkcií, ako aj výpis a vizualizáciu výsledkov. Do premennej `filepath` sa priraduje cesta k súboru v csv formáte, čo je v tomto prípade dataset `retail_sales.csv`. Týmto spôsobom možno jednoducho meniť vstupný súbor, ktorého dáta chceme využiť na predikciu a výpočet metrík pre optimalizáciu riadenia zásob. Najprv sa načítajú dáta volaním funkcie `load_and_aggregate_data()`. Táto funkcia zároveň zabezpečí aj sa agregáciu predajov jednotlivých produktov za jednotlivé mesiace. Nasleduje funkcia `fill_missing_month()`, ktorá doplní jednotlivé chýbajúce mesiace nulovou hodnotou predaja. A poslednou funkciou na spracovanie je `preprocess_data()`, ktorá vykoná logaritmickú transformáciu a vytvorí vysvetľujúce premenné. Nakoniec sa volá funkcia `train_and_forecast()`, ktorá zabezpečí samotné tréningovanie modelov s predikciou dopytu na 12 mesiacov a výpočet optimalizačných metrík. Po dokončení nasleduje výpis výsledkov do konzoly a vizualizácia pomocou knižnice `Matplotlib`. Vykreslia sa dva typy

grafu. Prvý stĺpcový graf zobrazuje výsledky metrík pre optimalizáciu zásob pre každý produkt (obrázok 13). Potom nasleduje postupné zobrazovanie časovo závislého grafu pre dopyt, ktorý zobrazujú skutočný historický dopyt s predikciou modelu na 12 mesiacov pre konkrétny produkt (obrázok 14).

Obrázok 11 Spúšťacia časť programu a vykreslenie stĺpcového diagramu

```

167 if __name__ == "__main__":
168     file_path = "retail_sales.csv"
169
170     df = load_and_aggregate_data(file_path)
171     df = fill_missing_months(df)
172     df = preprocess_data(df)
173
174     df_results, forecast_dict = train_and_forecast(df)
175     print("Výsledné optimalizačné metriky:")
176     print(df_results)
177
178     print("\nDetailné predikcie na 12 mesiacov:")
179     for product, data in forecast_dict.items():
180         print(f"\nProdukt: {product}")
181         for date, forecast in zip(data["dates"], data["forecasts"]):
182             print(f"    {date.strftime('%Y-%m-%d')}: {forecast:.2f}")
183
184     # Bar plot s optimalizačnými metrikami
185     if not df_results.empty:
186         fig, ax = plt.subplots(figsize=(12,6))
187         index = np.arange(len(df_results))
188         bar_width = 0.2
189
190         ax.bar(index, df_results["Predikovaný dopyt (priemer 12M)"], bar_width, label='Predikovaný 12M priemer')
191         ax.bar(index + bar_width, df_results['EOQ (optimálna objednávka)'], bar_width, label='EOQ')
192         ax.bar(index + 2*bar_width, df_results['Bod objednania'], bar_width, label='Bod objednania')
193         ax.bar(index + 3*bar_width, df_results['Bezpečnostná zásoba'], bar_width, label='Bezpečnostná rezerva')
194
195         ax.set_xticks(index + 1.5*bar_width)
196         ax.set_xticklabels(df_results['Produkt'], rotation=45, ha='right')
197         ax.legend()
198         ax.set_title('Optimalizácia zásob ☐ Prehľad metrik')
199         plt.tight_layout()
200         plt.show()

```

Zdroj: vlastné spracovanie

Obrázok 12 Vykreslenie grafu priebehu dopytu pre jednotlivé produkty

```

202 # Grafy dopytu
203 products = list(forecast_dict.keys())
204 for product in products:
205     fig, ax = plt.subplots(figsize=(10, 4))
206     # historické dáta pre produkt - pôvodné (transformované) dáta
207     product_df = df[df["Product"] == product].copy().sort_values("Date")
208     # skutočný dopyt (prevod späť z log-transformácie)
209     actual_dates = product_df["Date"]
210     actual_values = np.expml(product_df["Quantity"])
211     # odhadnutý dopyt
212     forecast_dates = forecast_dict[product]["dates"]
213     forecast_values = forecast_dict[product]["forecasts"]
214
215     ax.plot(actual_dates, actual_values, marker='o', color='blue', label='Skutočný dopyt')
216     ax.plot(forecast_dates, forecast_values, marker='o', color='green', linestyle='--', label='12M predikcia')
217     ax.set_title(f'Dopyt pre produkt: {product}')
218     ax.set_xlabel('Dátum')
219     ax.set_ylabel('Počet jednotiek')
220     ax.grid(True)
221     ax.legend()
222     for tick in ax.get_xticklabels():
223         tick.set_rotation(45)
224     plt.tight_layout()
225     plt.show()

```

Zdroj: vlastné spracovanie

4.3 Výsledky optimalizácie zásob a presnosť modelu

Vyhodnotenie presnosti modelu:

Z obrázka č.13 možno pozorovať, že predikčné modely pre jednotlivé produkty dosiahli pomerne uspokojivé výsledky na základe hodnôt metrík modelu. Len päť produktov dosiahlo chybné predikcie, čo znamená že mali MAPE vyššie ako 30%. Tieto výsledky mohli byť spôsobené kvalitou dát, keďže boli generované umelou inteligenciou, modelom sa nemuselo podariť v nich nájsť vzor. Podrobnejšie výsledky:

- MAE (Mean Absolute Error) sa pohybuje v rozmedzí od 6,30 do 34,94 , pričom najnižšiu hodnotu dosahuje laptop a najvyššiu korektor.
- RMSE (Root Mean Squared Error) je v rozmedzí od 8,25 do 41,31, pričom najnižšiu hodnotu má rúž a najvyššiu opäť korektor.
- MAPE (Mean Absolute Percentage Error) sa nachádza v rozmedzí od 9% do 39%, pričom najnižšiu hodnotu dosahuje šampón a najvyššiu opäť korektor.

Keďže jednotlivé metriky nehodnotia chyby rovnako preto sa líši poradie jednotlivých produktov vzhľadom na jednotlivé chyby.

Obrázok 13 Výsledky optimalizačných metrík pre jednotlivé produkty a chyby metrík

	Produkt	Predikovaný dopyt (priemer 12M)	EOQ (optimálna objednávka)	Bod objednania	Bezpečnostná zásoba	MAE	RMSE	MAPE
0	Korektor	126.0	158.600006	67.10	37.76	34.94	41.31	0.39
1	Laptop	53.0	103.230003	25.14	12.71	6.30	8.25	0.13
2	Letné šaty	79.0	125.779999	77.99	59.53	27.63	34.97	0.28
3	Linka	54.0	104.269997	23.81	11.13	10.68	13.41	0.23
4	Opalovací krém	55.0	105.239998	72.32	59.40	15.02	20.22	0.45
5	Parfém	50.0	99.910004	22.92	11.27	12.67	15.51	0.26
6	Pletový krém	68.0	116.459999	33.41	17.59	13.82	16.06	0.25
7	Rúž	52.0	101.529999	22.41	10.39	7.79	8.73	0.16
8	Smart hodinky	62.0	111.139999	46.96	32.55	22.96	28.04	0.29
9	Tablet	50.0	99.860001	22.76	11.13	12.18	13.29	0.24
10	Tenisky	70.0	118.160004	51.91	35.62	28.40	39.37	0.33
11	Tričko	61.0	110.519997	48.76	34.50	18.15	28.75	0.26
12	Wireless Earbuds	58.0	107.720001	39.85	26.31	10.97	13.22	0.31
13	Zimná Bunda	52.0	101.769997	56.10	44.02	15.04	23.89	0.38
14	Šampón	124.0	157.320007	38.43	9.56	11.26	14.57	0.09

Zdroj: vlastné spracovanie

Optimalizácia zásob:

Na základe predikovaného dopytu na obdobie 12 mesiacov boli pre každý produkt vypočítané tri základné optimalizačné metriky zásob, ktoré umožňujú efektívne plánovanie zásob.

1. EOQ (Veľkosť ekonomickej objednávky)

EOQ predstavuje optimálne množstvo, ktoré by mal podnik objednať naraz, aby minimalizoval celkové náklady spojené s objednávaním a skladovaním zásob.

Vzhľadom na použitie základného EOQ modelu s nemennými parametrami:

- Náklady na objednávku: 25 €
- Náklady na skladovanie: 3 € / jednotku ročne

sa rozdiely medzi produktmi odvíjajú výlučne od očakávaného ročného dopytu. Produkty s vyšším odhadovaným dopytom budú mať vyššie EOQ, pretože sa oplatí objednávať vo väčšom množstve, čím sa znižuje frekvencia objednávok.

Napríklad, korektor dosahuje najvyšší priemerný predikovaný dopyt 126 ks a jeho EOQ je aj najvyššie (≈ 159). Taktiež tablet a parfém majú najnižší priemerný predikovaný dopyt 50 ks a ich EOQ je preto najnižší (≈ 100).

2. Bod objednania

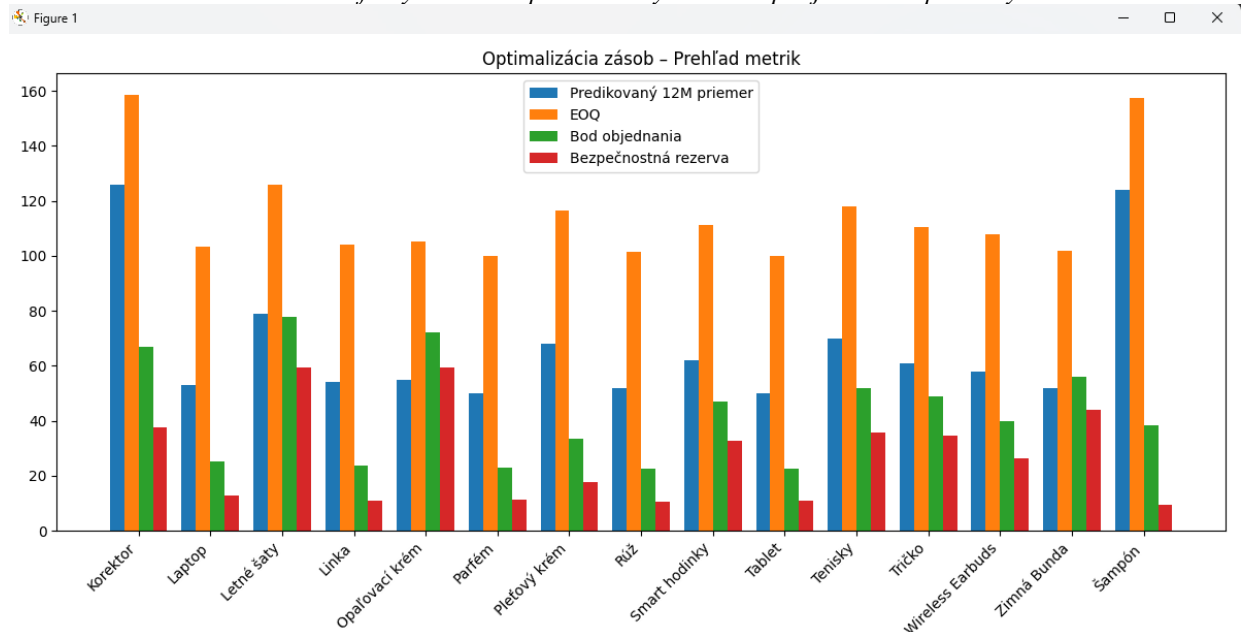
Bod objednania predstavuje kritickú hranicu zásob produktu, pri ktorej by mal podnik zadať novú objednávku, aby sa predišlo výpadku tovaru. Je to dôležitý prvok riadenia zásob, ktorý zabezpečuje, že nové produkty dorazia do skladu včas, ešte predtým, ako sa súčasné zásoby minú. Pre produkt linka to znamená, že potreba zadania novej objednávky nastane, keď zásoby tohto produktu klesnú na úroveň 24 ks. Výsledky ukazujú že produkty s vyšším dopytom alebo s vyššou variabilitou majú vyšší bod objednania. Napr. korektor má priemerný odhadovaný dopyt 126 ks a jeho bod objednania je (po zaokrúhlení) 67 ks. Avšak šampón, ktorý má priemerný odhadovaný dopyt 124 ks má bod objednania (po zaokrúhlení) 38 ks. To znamená, že korektor má ďaleko vyššiu variabilitu dopytu a naopak šampón má stabilnejší dopyt.

3. Bezpečnostná zásoba

Bezpečnostná zásoba je rezervné množstvo, ktoré chráni pred nečakanými výkyvmi v dopyte počas dodacej lehoty. Výsledky ukazujú, že bezpečnostná rezerva sa odvíja od variability dopytu a nie veľkosti dopytu. Preto produkty, ktoré majú stabilnejší dopyt majú nižšiu bezpečnostnú rezervu. Najviac stabilný dopyt má šampón, ktorého bezpečnostná zásoba je ≈ 10 ks a najvyššiu hodnotu majú letné šaty ≈ 78 ks, čo znamená že ich dopyt má najvyššiu variabilitu.

Na obrázku číslo 13 je zobrazený výpis z konzoly, kde sú pre každý produkt vypísané hodnoty metrík pre optimalizáciu zásob. Graf na obrázku 14 zobrazuje grafické znázornenie všetkých optimalizačných metrík pre jednotlivé produkty.

Obrázok 14 Graf s výsledkami optimalizačných metrík pre jednotlivé produkty

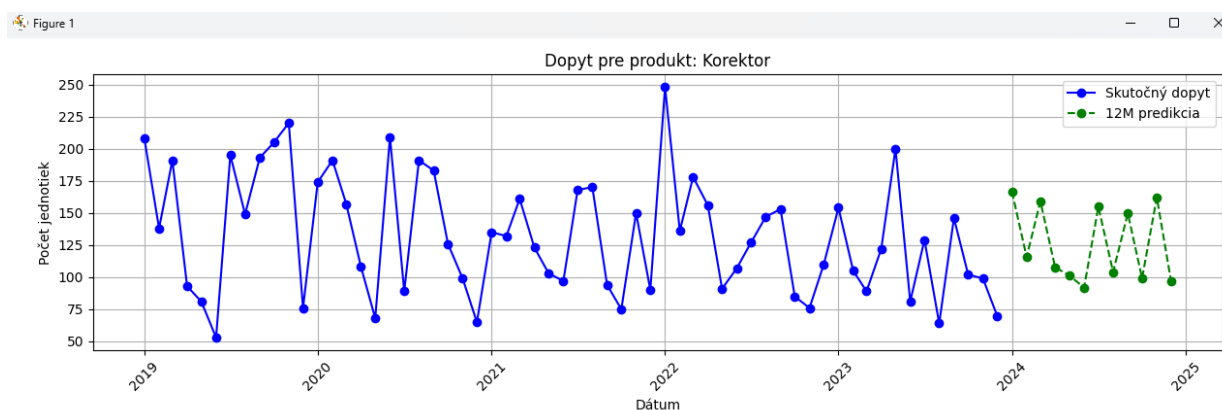


Zdroj: vlastné spracovanie

4.4 Výsledky predikcie dopytu jednotlivých produktov

Ako bolo dokázané v predchádzajúcej časti, model dosahuje pomerne uspokojivú presnosť predikcie pre jednotlivé produkty podľa metrík hodnotenia modelu. V tejto časti sa zameriame na priebeh historického a predikovaného dopytu pri jednotlivých produktoch, pomocou vizuálnych výstupov programu. Súčasťou výsledkov, ktoré program poskytuje sú detailné tabuľky predikcii pre každý produkt na každý mesiac. Program zároveň generuje grafy, ktoré zobrazujú historické predaje a predikcie na nasledujúcich 12 mesiacov. Modrá farba predstavuje skutočný historický dopyt a zelená predikciu na 12 mesiacov.

Obrázok 15 Predikcia dopytu pre produkt korektor



Zdroj: vlastné spracovanie

Obrázok 16 detailná predikcia pre korektor

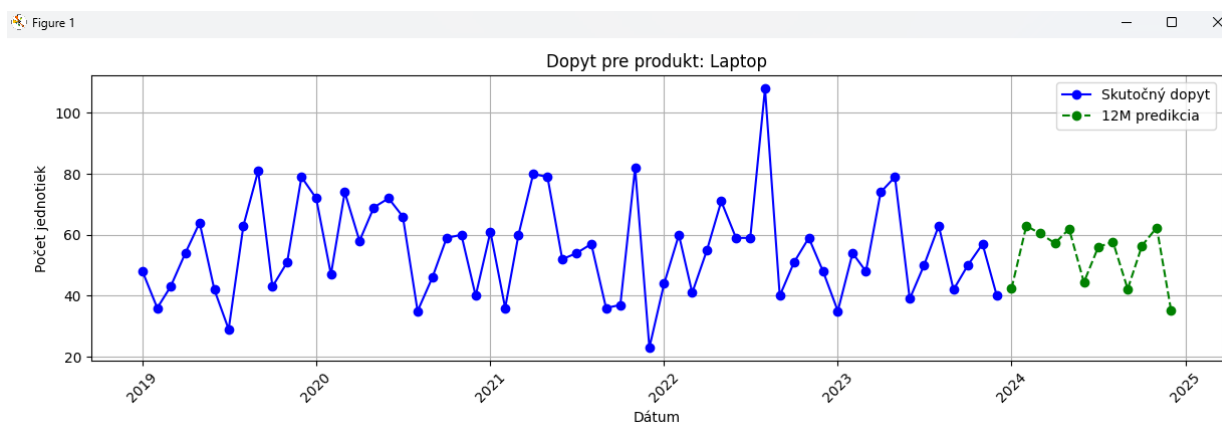
Detailné predikcie na 12 mesiacov:

```
Produkt: Korektor
2024-01-01: 166.79
2024-02-01: 115.72
2024-03-01: 159.03
2024-04-01: 107.72
2024-05-01: 101.28
2024-06-01: 91.89
2024-07-01: 154.86
2024-08-01: 104.00
2024-09-01: 149.65
2024-10-01: 99.17
2024-11-01: 162.12
2024-12-01: 96.90
```

Zdroj: vlastné spracovanie

Z prvého grafu na obrázku číslo 15 možno sledovať priebeh historického dopytu a predikciu pre produkt korektor. Graf zobrazuje, že historický dopyt sa vyznačuje vysokou variabilitou bez jasného sezónneho vzoru. Avšak predikcia dopytu je vyrovnanejšia oproti skutočným predaným hodnotám v minulosti. Ide o konzervatívny prístup modelu, ktorý znamená, že model nezachytil žiadny opakujúci sa vzor a preto sa snaží držať priemeru historických hodnôt. V dôsledku tohto produkt korektor dosahuje najvyššie hodnoty metrik hodnotenia modelu. Na obrázku číslo 16 zobrazuje výpis na konzole pre konkrétne predikované hodnoty pre korektor.

Obrázok 17 Predikcia dopytu pre produkt laptop



Zdroj: vlastné spracovanie

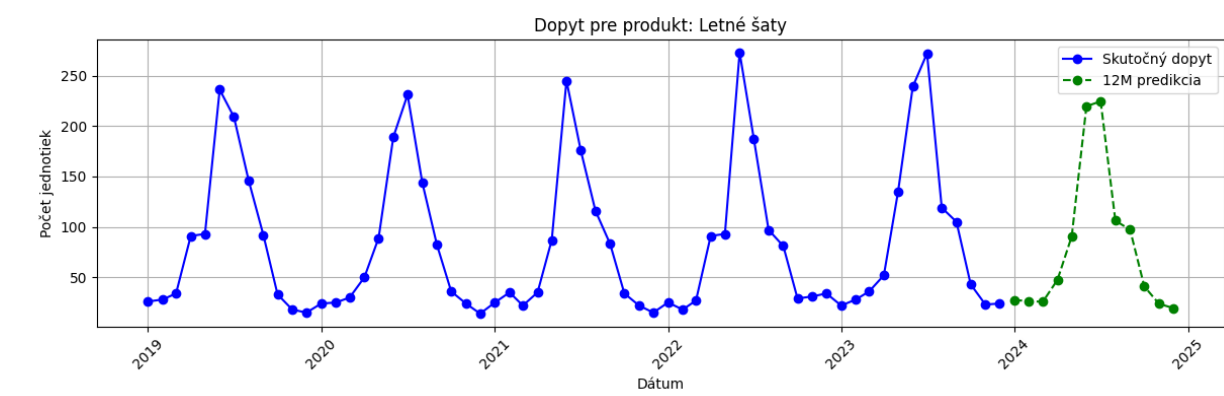
Obrázok 18 Detailná predikcia pre laptop

Produkt: Laptop
2024-01-01: 42.38
2024-02-01: 62.94
2024-03-01: 60.45
2024-04-01: 57.25
2024-05-01: 61.95
2024-06-01: 44.42
2024-07-01: 55.96
2024-08-01: 57.73
2024-09-01: 42.22
2024-10-01: 56.45
2024-11-01: 62.40
2024-12-01: 35.30

Zdroj: vlastné spracovanie

Na obrázku číslo 17 je znázornený graf historického a predikovaného dopytu pre produkt laptop. Nastáva podobná situácia, kde predikcia pôsobí plocho oproti skutočnému dopytu, ktorý vykazuje viaceré výkyvy. Model opäť zvolil konzervatívny prístup. Tento prístup môže byť dôsledkom nepravidelného vývoja dopytu, pri ktorom model nedokáže identifikovať opakujúce sa vzory v dátach. Predikcia je síce stabilnejšia, čo z pohľadu riadenia zásob môže byť bezpečnejší, ale menej presný prístup. Napriek tomu model pri produkte laptop dosiahol najnižšie hodnoty hodnotenia modelu pri MAE a MAPE. Konkrétne číselné hodnoty pre predikciu produktu laptop vidíme na obrázku č. 18.

Obrázok 19 Predikcia dopytu pre produkt letné šaty



Zdroj: vlastné spracovanie

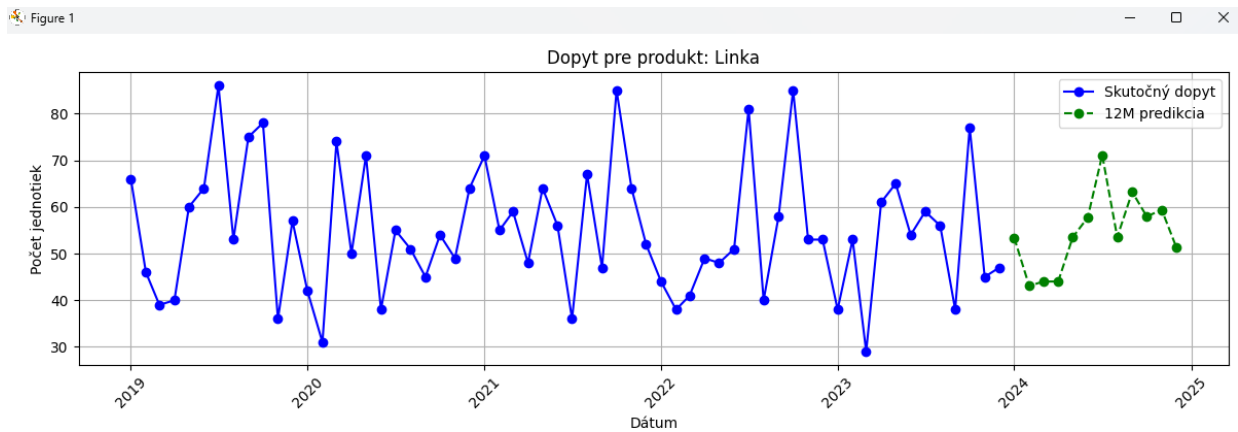
Obrázok 20 Detailná predikcia pre letné šaty

Produkt: Letné šaty
2024-01-01: 27.44
2024-02-01: 26.46
2024-03-01: 25.68
2024-04-01: 47.56
2024-05-01: 90.57
2024-06-01: 219.59
2024-07-01: 224.46
2024-08-01: 106.18
2024-09-01: 97.12
2024-10-01: 40.92
2024-11-01: 23.69
2024-12-01: 19.52

Zdroj: vlastné spracovanie

Naopak z obrázka číslo 19 možno pozorovať priebeh historického dopytu pre produkt letné šaty, ktorý má výraznú letnú sezónnosť. Graf ukazuje, že model dokázal túto sezónnosť s vysokou presnosťou. Predikcia správne načasovala sezónnosť a odhadla aj veľkosť špičky. Táto skutočnosť naznačuje, že model je spoľahlivý nástroj pre plánovanie zásob s pravidelnou sezónnosťou. Obrázok č. 20 zobrazuje konkrétne číselné hodnoty týchto predikcií pre jednotlivé mesiace, ktoré boli vypísané na konzole.

Obrázok 21 Predikcia dopytu pre produkt linka



Zdroj: vlastné spracovanie

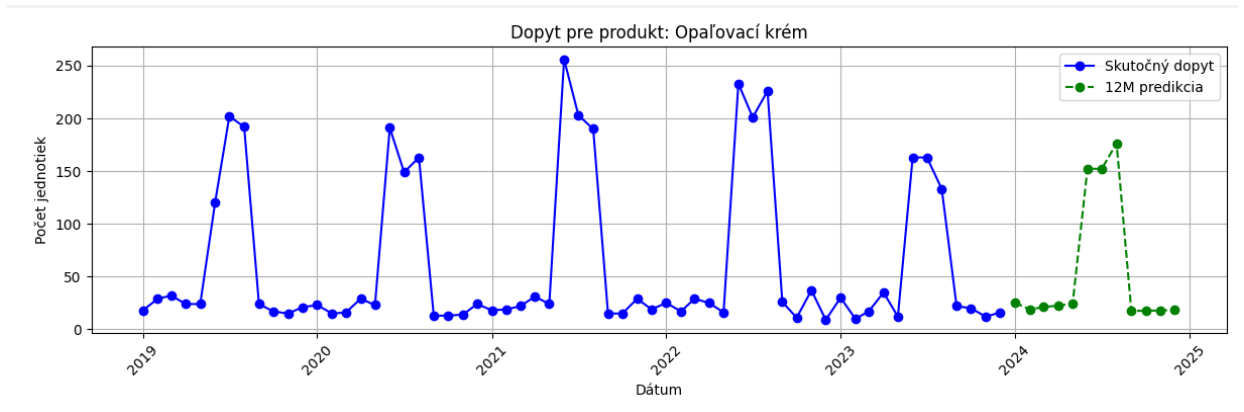
Obrázok 22 Detailná predikcia pre linku

Produkt: Linka	
2024-01-01:	53.39
2024-02-01:	43.08
2024-03-01:	44.00
2024-04-01:	44.04
2024-05-01:	53.51
2024-06-01:	57.74
2024-07-01:	71.12
2024-08-01:	53.53
2024-09-01:	63.21
2024-10-01:	57.95
2024-11-01:	59.39
2024-12-01:	51.40

Zdroj: vlastné spracovanie

Graf na obrázku č. 21 zobrazuje priebeh historického dopytu a predikcie pre produkt linka. V historických dátach je výrazná variabilita dopytu, zatiaľ čo predikcia je plochejšia. Avšak z predikcie je jasné, že model identifikoval nejaký vzor aj vo vysoko variabilných dátach a pokúsil sa ho premietnuť do budúcnosti v konzistentnejšie podobe. Detail číselných hodnôt pre predikciu produktu linka sú zobrazené na obrázku číslo 22.

Obrázok 23 Predikcia pre produkt opaľovací krém



Zdroj: vlastné spracovanie

Obrázok 24 Detail predikcie pre opaľovací krém

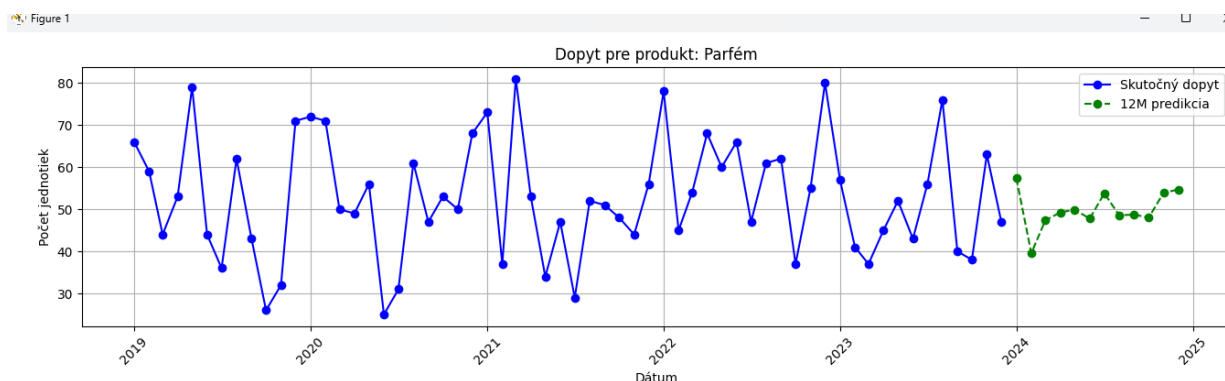
Produkt: Opaľovací krém

```
2024-01-01: 25.40
2024-02-01: 18.93
2024-03-01: 21.10
2024-04-01: 22.58
2024-05-01: 24.27
2024-06-01: 152.24
2024-07-01: 152.24
2024-08-01: 176.33
2024-09-01: 17.29
2024-10-01: 17.77
2024-11-01: 17.77
2024-12-01: 18.61
```

Zdroj: vlastné spracovanie

Obrázok č. 23 znázorňuje predikciu dopytu pre produkt opaľovací krém, ktorý má jednoznačnú letnú sezónnosť. Z grafu je zrejmé, že model opäť úspešne zachytil sezónny vzor, čo potvrdzuje jeho schopnosť identifikovať pravidelné sezónne výkyvy. Avšak mimo hlavnej sezóny sú hodnoty vyhladené, čo je spôsobené nepravidelnými výkyvmi, v ktorých nenašiel model vzor a zvolil opäť konzervatívny prístup. Detailné číselné hodnoty tejto predikcie možno pozorovať na výpise z konzoly na obrázku č. 24.

Obrázok 25 Predikcia pre produkt parfém



Zdroj: vlastné spracovanie

Obrázok 26 Detail predikcie pre parfém

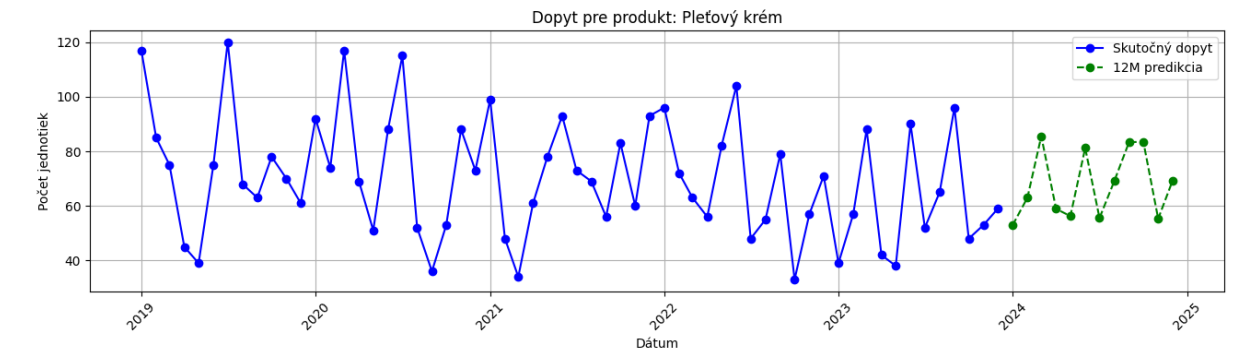
Produkt: Parfém

2024-01-01:	57.40
2024-02-01:	39.55
2024-03-01:	47.36
2024-04-01:	49.26
2024-05-01:	49.90
2024-06-01:	47.77
2024-07-01:	53.79
2024-08-01:	48.56
2024-09-01:	48.74
2024-10-01:	48.01
2024-11-01:	53.95
2024-12-01:	54.67

Zdroj: vlastné spracovanie

V prípade produktu parfém je skutočný dopyt menej sezónny a viac variabilný, ako možno vidieť na obrázku číslo 25. Z predikcie je možné pozorovať, že model sa snažil zachytiť isté výkyvy ale zvolil konzervatívnejší prístup a držal sa v blízkosti okolo priemeru. V takomto prípade s vysokými výkyvmi by bolo vhodné zvážiť iné faktory, ktoré by mohli takto výrazne ovplyvňovať dopyt, aby sa mohla zlepšiť presnosť predikcie. Detail výpisu číselných hodnôt na konzole je zobrazený na obrázku číslo 26.

Obrázok 27 Predikcia pre produkt pleťový krém



Zdroj: vlastné spracovanie

Obrázok 28 Detailná predikcia pre pleťový krém

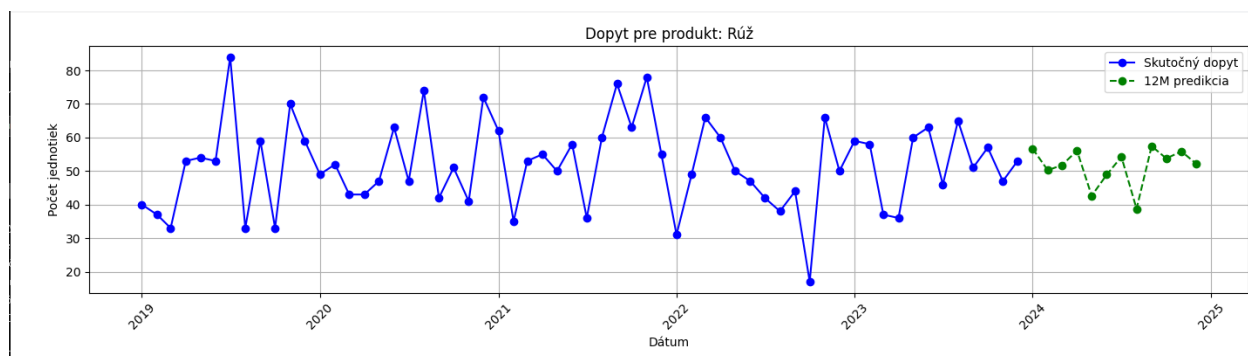
Produkt: Pleťový krém

2024-01-01:	52.96
2024-02-01:	63.01
2024-03-01:	85.52
2024-04-01:	58.90
2024-05-01:	56.25
2024-06-01:	81.21
2024-07-01:	55.54
2024-08-01:	69.09
2024-09-01:	83.39
2024-10-01:	83.44
2024-11-01:	55.29
2024-12-01:	69.17

Zdroj: vlastné spracovanie

Pri produkte pleťový krém na obrázku číslo 27 možno pozorovať, že má výrazne a opakujúce výkyvy v skutočnom dopyte. Model sa snaží tieto výkyvy zachytiť, avšak predikcia pôsobí vyrovnanejšie oproti historickým údajom. Tento výsledok naznačuje, že algoritmus rozpoznal určité vzory, ale nepriradil im dostatočnú váhu v predikcii. Obrázok číslo 28 zobrazuje konkrétne hodnoty predikcie pre pleťový krém, ktoré program vypísal na konzolu.

Obrázok 29 Predikcia pre produkt produkt rúž



Zdroj: vlastné spracovanie

Obrázok 30 Detailná predikcia pre rúž

Produkt: Rúž

2024-01-01: 56.65

2024-02-01: 50.24

2024-03-01: 51.76

2024-04-01: 56.06

2024-05-01: 42.63

2024-06-01: 48.92

2024-07-01: 54.24

2024-08-01: 38.75

2024-09-01: 57.43

2024-10-01: 53.75

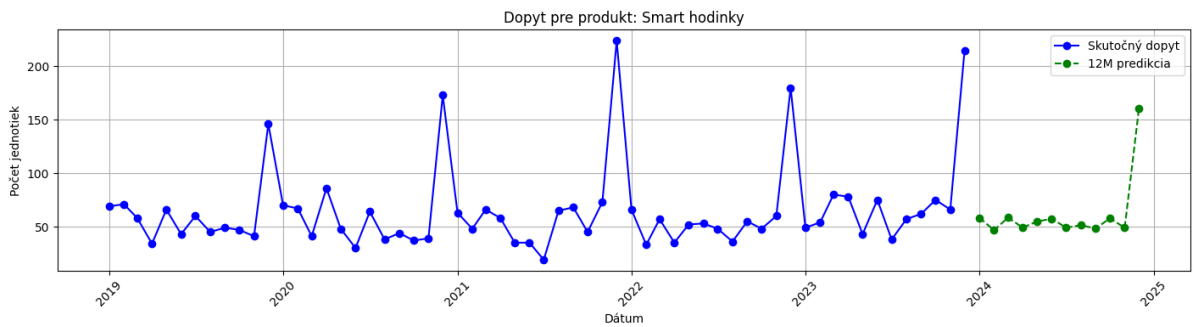
2024-11-01: 55.86

2024-12-01: 52.23

Zdroj: vlastné spracovanie

Obrázok č. 29 znázorňuje predikciu pre produkt rúž, ktorý v minulosti sa vyznačoval miernou variabilitou dopytu. Algoritmus identifikoval niektoré vzory a tie sa snažil premietnuť do predikcie. Možno pozorovať, že zachytil aj správne úroveň dopytu. Avšak jeho konzervatívne nastavenie spôsobuje plochosť predikcie. Jednotlivé číselné hodnoty predikcie pre produkt rúž z výpisu konzoly možno pozorovať na obrázku číslo 30.

Obrázok 31 Predikcia pre produkt smart hodinky



Zdroj: vlastné spracovanie

Obrázok 32 Detailná predikcia pre smart hodinky

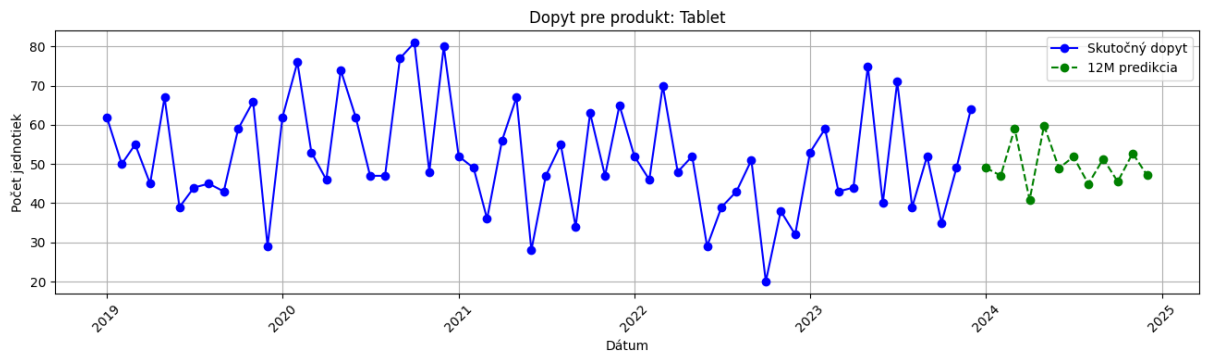
Produkt: Smart hodinky

2024-01-01:	58.34
2024-02-01:	46.64
2024-03-01:	58.46
2024-04-01:	49.13
2024-05-01:	54.79
2024-06-01:	57.43
2024-07-01:	49.04
2024-08-01:	51.38
2024-09-01:	48.18
2024-10-01:	57.78
2024-11-01:	48.99
2024-12-01:	160.92

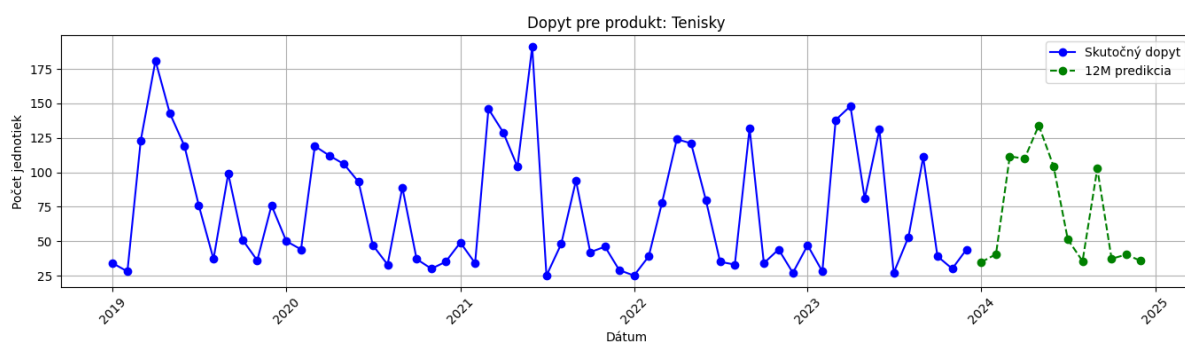
Zdroj: vlastné spracovanie

Produkt smart hodinky má výrazný nárast predaja v decembri (decembrová sezónnosť), ktorú model zachytil správne, čo možno pozorovať na obrázku číslo 31. Potvrďuje to, že model zvláda rozpoznať sezónne špičky. Obrázok č. 32 zobrazuje konkrétne hodnoty predikovaného dopytu pre jednotlivé mesiace, ktoré boli vypísané na konzole.

Obrázok 33 Predikcia pre produkt tablet

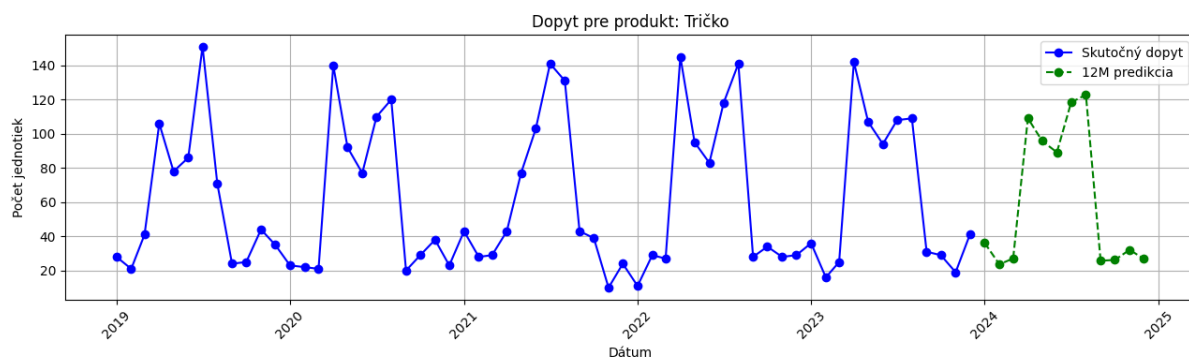


Obrázok 35 Predikcia pre produkt tenisky



Zdroj: vlastné spracovanie

Obrázok 36 Predikcie pre produkt Tričko



Zdroj: vlastné spracovanie

Obrázok 37 Detailná predikcia pre tenisky

Produkt: Tenisky

2024-01-01:	34.77
2024-02-01:	40.53
2024-03-01:	111.55
2024-04-01:	109.84
2024-05-01:	134.01
2024-06-01:	104.40
2024-07-01:	51.22
2024-08-01:	35.52
2024-09-01:	102.70
2024-10-01:	37.23
2024-11-01:	40.28
2024-12-01:	35.71

Zdroj: vlastné spracovanie

Obrázok 38 Detailná predikcia pre tričko

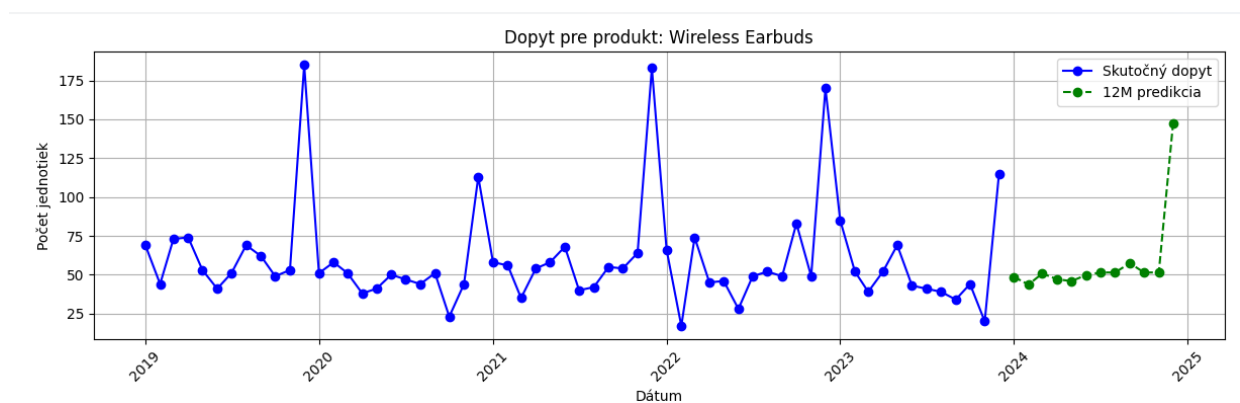
Produkt: Tričko

2024-01-01: 36.16
2024-02-01: 23.64
2024-03-01: 27.06
2024-04-01: 109.06
2024-05-01: 95.82
2024-06-01: 89.15
2024-07-01: 118.36
2024-08-01: 122.82
2024-09-01: 25.74
2024-10-01: 26.21
2024-11-01: 32.15
2024-12-01: 26.78

Zdroj: vlastné spracovanie

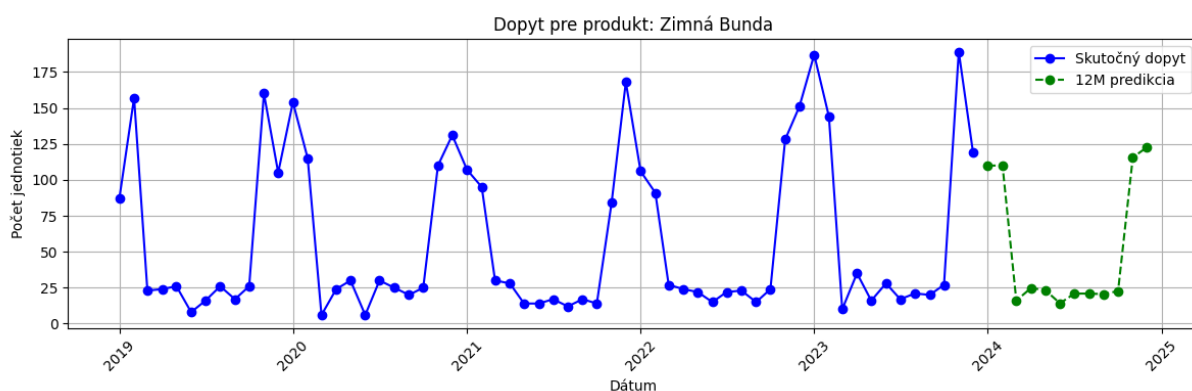
Pri produktoch tenisky (obrázok č. 35) a tričko (obrázok č. 36) možno pozorovať, že modely dobre zachytili sezónnosť, ktorá sa vyskytuje na jar a leto. V prípade tenisiek sa modelu podarilo identifikovať aj ďalšie vzory dopytu okrem samotnej sezónnej špičky. Tričko má výraznejšiu sezónnosť a dosahuje vyššie hodnoty dopytu, ktoré model dokázal reflektovať. Okrem toho model nepodhodnotil ani výkyvy v ostatných nesezónnych mesiacoch na rozdiel od predikcie pri pleťovom kréme. Obrázok číslo 37 zobrazuje konkrétne hodnoty predikovaného dopytu pre tenisky a na obrázku č. 38 sú zobrazené konkrétne hodnoty predikcie pre tričko, ktoré boli vypísané na konzole.

Obrázok 39 Predikcie pre produkt wireless earbuds



Zdroj: vlastné spracovanie

Obrázok 40 Predikcie pre produkt zimná bunda



Zdroj: vlastné spracovanie

Obrázok 41 Detailná predikcia pre wireless earbuds

Produkt: Wireless Earbuds

2024-01-01: 48.60
2024-02-01: 43.66
2024-03-01: 50.78
2024-04-01: 47.08
2024-05-01: 45.87
2024-06-01: 49.71
2024-07-01: 51.43
2024-08-01: 51.43
2024-09-01: 57.39
2024-10-01: 51.43
2024-11-01: 51.43
2024-12-01: 147.34

Zdroj: vlastné spracovanie

Obrázok 42 Detailná predikcia pre zimná bunda

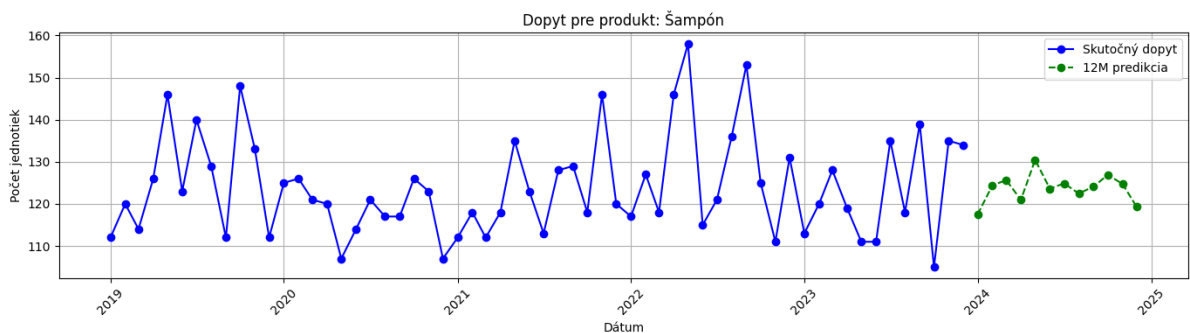
Produkt: Zimná Bunda

2024-01-01: 109.92
2024-02-01: 109.92
2024-03-01: 16.02
2024-04-01: 24.82
2024-05-01: 23.21
2024-06-01: 14.14
2024-07-01: 20.91
2024-08-01: 20.91
2024-09-01: 20.54
2024-10-01: 22.69
2024-11-01: 115.60
2024-12-01: 122.71

Zdroj: vlastné spracovanie

Na obrázkoch č. 39 a 40 možno vidieť predikciu pre wireless earbuds a zimná bunda. Model opäť presne zachytil sezónnosť. Avšak predikcia pre produkt zimná bunda je síce správne načasovaná, ale pôsobí ďaleko vyrovnanejšie ako v historických dátach. Model podhodnotil veľkosť sezónnych špičiek. Na obrázkoch číslo 41 a 42 môžeme pozorovať výpis z konzoly pre detailnú predikciu pre produkt wireless earbuds a zimná bunda.

Obrázok 43 Predikcia pre produkt šampón



Zdroj: vlastné spracovanie

Obrázok 44 Detailná predikcia pre šampón

Produkt: Šampón

2024-01-01:	117.50
2024-02-01:	124.41
2024-03-01:	125.65
2024-04-01:	120.97
2024-05-01:	130.46
2024-06-01:	123.62
2024-07-01:	124.87
2024-08-01:	122.47
2024-09-01:	124.10
2024-10-01:	126.91
2024-11-01:	124.74
2024-12-01:	119.27

Zdroj: vlastné spracovanie

Obrázok 43 znázorňuje produkt šampón, ktorý nevykazuje sezónnosť a jeho historický dopyt je výrazne variabilný. V posledných dvoch rokoch možno pozorovať, že trend vykazuje mierny pokles. Model túto skutočnosť zachytil, preto sú predikované hodnoty nižšie než skutočné. Znamená to, že model dokáže reagovať na zmenu trendu v dlhšom časovom horizonte. Napriek tomu, že model dosahuje najnižšiu hodnotu MAPE, je zjavné, že nedokázal správne zachytiť variabilitu. Obrázok č. 44 zobrazuje konkrétne hodnoty

predikovaného dopytu pre jednotlivé mesiace, ktoré boli vypísané na konzole pre produkt šampón.

5. Diskusia

Cieľom tejto práce bolo vyvinúť inteligentný systém pre riadenie zásob v maloobchode, ktorý na základe historických predajných dát dokáže predikovať budúci dopyt a optimalizovať objednávkový cyklus. Na základe prezentovaných výsledkov možno konštatovať, že vytvorený program sa preukazuje ako potenciálne efektívny nástroj pre maloobchodníkov. Systém podporuje zlepšenie hospodárenia so zdrojmi, zvyšovanie konkurencieschopnosti a ziskovosti, keďže poskytuje relevantné a prakticky využiteľné výstupy v oblasti riadenia zásob. Systém dosiahol uspokojivú úroveň presnosti, pričom najlepšie výsledky boli zaznamenané pri produktoch so stabilným alebo pravidelne sezónnym dopytom. Naopak, pri produktoch s vyššou variabilitou alebo nepravidelnými výkyvmi boli odchýlky v predikcii výraznejšie.

Z pohľadu optimalizácie zásob systém úspešne aplikoval výstupy predikcie na výpočet kľúčových logistických metrík – EOQ, bod objednania a bezpečnostná zásoba. Výsledky ukazujú, že model dokáže reagovať nielen na veľkosť dopytu, ale aj na jeho variabilitu, čo je dôležité pre reálne plánovanie v maloobchodnom prostredí.

Je potrebné zdôrazniť, že model pracuje s niekoľkými obmedzeniami. Jedným z hlavných je, že model bol trénovaný na dátach, ktoré len majú simulovať realitu, ale nepochádzajú z konkrétneho maloobchodu. V dôsledku toho sú v dátach veľmi časté nepravidelné vzory, ktoré má model problém zachytiť. Ďalším obmedzením je, že model využíva na výpočet EOQ základný model, ktorý pracuje konštantnými parametrami nákladov, hoci v praxi bývajú dynamickejšie. Model takisto nezohľadňuje obmedzenia ako je skladová kapacita, minimálne objednávacie množstvá alebo oneskorenia v dodávke.

Z podrobných analýz grafov sa ukázalo, že existujú oblasti v modeli, ktoré by bolo vhodné do budúcnosti zlepšiť. Model pracoval len s časovými faktormi a nebral do úvahy žiadne externé faktory, ako sú napr. sviatky, marketingové kampane, výpredaje, atď. Všetky tieto faktory významne ovplyvňujú dopyt a preto by bolo dobré, aby sa do modelu zahrnuli. Prínosné by mohlo byť aj zahrnutie výslovné trendovej premennej aby ho model dokázal lepšie zachytiť. Taktiež by bolo vhodné zváženie hybridného prístupu pre produkty pre rôzne

variability dopytu (jednoduchší pre stabilný dopyt, komplexnejší pre vysoko variabilný dopyt). Pri postupnom ladení modelu sme zistili, že vizuálne predikcie a štatistické metriky hodnotenia nemusia vždy korešpondovať. Preto najdôležitejším krokom do budúcnosti je nevyhnutné otestovať model na dátach z konkrétneho maloobchodu a prispôbiť ho ich špecifickým podmienkam a potrebám.

Záver

Hlavným cieľom tejto práce bolo vyvinúť inteligentný systém pre riadenie zásob v maloobchode s využitím programovacieho jazyka Python, ktorý prostredníctvom analýzy historických predajných dát a predikcie budúceho dopytu pomôže maloobchodníkom efektívnejšie optimalizovať zásoby, minimalizovať riziko nedostatku aj prebytku tovaru a tým zvýšiť ich konkurencieschopnosť a ziskovosť. V úvodnej časti práce boli definované a priblížené jednotlivé pojmy nevyhnutné na pochopenie problematiky riadenia zásob a využívaných metód v tejto oblasti. Následne bola pozornosť venovaná predikcii dopytu a princípom strojového učenia.

V nasledujúcich častiach sme aplikovali tieto poznatky pri tvorbe systému na predikciu dopytu s výpočtom metrík pre optimalizáciu zásob a vizualizáciu výsledkov. Vytvorený systém bol implementovaný v jazyku Python s využitím jeho knižníc na prácu s dátami, vizualizáciu dát a strojové učenie. Na základe historických údajov o predaji bol vytvorený predikčný model pre jednotlivé produkty. Zvolený model bol XGBoost, ktorý je regresný algoritmus častokrát práve využívaný na predikciu dopytu. Model bol použitý na odhad na dopytu na obdobie 12 mesiacov pomocou postupnej predikcie kde odhadnutá hodnota bola využitá na výpočet pre ďalšie mesiace. Predikované hodnoty boli následne využité na výpočet optimalizačných zásobovacích metrík, ktoré v tejto práci zahŕňajú základný model EOQ (veľkosť ekonomickej objednávky), bod objednania a bezpečnostná rezerva. Presnosť modelov pri predikcii bola hodnotená pomocou štatistických metrík RMSE, MSE a MAPE, ktoré poskytujú dôležité informácie o presnosti predikcie. Na záver sme vizualizovali výsledky pomocou grafov. Prvý bol stĺpcový graf pre všetky optimalizačné metriky a časovo závislý graf znázorňujúci priebeh historických predajov a predikovaný dopyt pre jednotlivé produkty. Pri bližšej analýze výsledkov predikcie sme zistili, že ani nízke metriky hodnotenia modelu neznamenajú vždy najpresnejšiu predikciu. Ukázalo sa, že vyvinutý systém dokáže najlepšie pracovať s predajmi, ktoré majú stabilný alebo sezónny dopyt. Napriek tomu sa model ukázal ako užitočný nástroj, ktorý poskytuje relevantné informácie pre praktické využitie v riadení zásob. Tým sme dosiahli splnenie hlavného cieľa bakalárskej práce.

V závere sme diskusiou poukázali na možnosti zlepšenia vytvoreného systému. Ako prvé bolo navrhnuté pridanie vysvetľujúcich premenných, ktoré by mohli zahŕňať sviatky, marketingové kampane a ďalšie. Aby mohli modelu lepšie zachytiť variabilitu dopytu. Ďalšie

odporúčanie bolo pridanie výslovne trendovej premennej do modelu. Avšak najdôležitejším krokom je otestovanie systému na reálnych dátach konkrétneho podniku, keďže v tejto bakalárskej práci bol testovaný len na dátach, ktoré by mali reprezentovať realitu.

Zoznam použitej literatúry

Alina. (2024, November 4). Most common inventory management problems in 2024 (and their solutions) | Retalon. *Retalon*. <https://retalon.com/blog/inventory-management-problems>

Banton, C. (2025, February 22). *Just-in-Time (JIT): Definition, example, pros, and cons*. Investopedia. <https://www.investopedia.com/terms/j/jit.asp>

Brockwell, P. J., & Davis, R. A. (2016). Introduction to time series and forecasting. In *Springer texts in statistics*. <https://doi.org/10.1007/978-3-319-29854-2>

Brownlee, J. (2019). Xgboost with python. *Machine Learning Mastery*.

Dorota-Owczarek. (2024, January 14). Revolutionize Your Stock Levels with Machine Learning Inventory Management. *nexocode*. <https://nexocode.com/blog/posts/inventory-optimization-machine-learning/>

Dupaľ, A., CSc. (2018). *LOGISTIKA*. Spint 2 s.r.o.

Fattah, Jamal & Ezzine, Latifa & Aman, Zineb & Moussami, Haj & Lachhab, Abdeslam. (2018). Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management*. 10. 184797901880867. 10.1177/1847979018808673.

Forrest, B. (2025, January 7). Advantages of Economic Order Quantity in minimizing costs. *AMZPREP*. <https://amzprep.com/economic-order-quantity/>

Chopra, S., & Meindl, P. (2007). *Supply chain management: Strategy, Planning, and Operation*.

Lazzeri, F. (2020). *Machine Learning for Time Series Forecasting with Python*. John Wiley & Sons.

Massaron, L., & Boschetti, A. (2016). *Regression Analysis with Python*. Packt Publishing Ltd.

Mohammed, I., Ullah, A., & Ruqeishi, A. L. (2024). INVENTORY MANAGEMENT: METHODS, APPROACHES, BENEFITS AND CHALLENGES. *International Journal of Social Sciences and Management*, 7(4). <https://doi.org/10.37602/IJSSMR.2024.7402>

Pandey, A., Manna Aradha, & Tanvir, M. (2025). Optimizing Inventory Management Using Machine Learning A Case Study in the Retail Sector. *International Journal of Advanced Research*, 19(2).

- Priniotakis, G., & Argyropoulos, P. (2018). Inventory management concepts and techniques. *IOP Conference Series Materials Science and Engineering*, 459, 012060. <https://doi.org/10.1088/1757-899x/459/1/012060>
- Rebala, G., Ravi, A., & Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer.
- Shenoy, D., & Rosas, R. (2017). Problems & Solutions in Inventory Management. *Springer eBooks*. <https://doi.org/10.1007/978-3-319-65696-0>
- Slack, N., Chambers, S., & Johnston, R. (2010). *Operations Management*. Pearson Education.
- Sunol, H. (2024, September 26). *Inventory control in Warehouses: Key strategies to Boost Profits*. Cyzerg. <https://cyzerg.com/blog/inventory-control-6-ways-to-maximize-profitability-minimize-cost/>
- Vandeput, N. (2023). *Demand forecasting best practices*. Manning.
- W. Chase, C. (2013). *Demand-Driven Forecasting : A structured approach to forecasting* (2nd ed.). John Wiley & Sons, Incorporated.
- Zhou, Z. (2021). Machine learning. In *Springer eBooks*. <https://doi.org/10.1007/978-981-15-1967-3>