VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ ZNALOSTÍ Z TEXTOVÝCH DAT V PROSTŘEDÍ JAZYKA PYTHON

KNOWLEDGE DISCOVERY FROM TEXT DATA IN THE PYTHON LANGUAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JÁN HOMOLA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. VLADIMÍR BARTÍK, Ph.D.**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2021/2022

Zadání bakalářské práce

Student: **Homola Ján**
Program: Informační technologie
Název: **Získávání znalostí z textových dat v prostředí jazyka Python**
Knowledge Discovery from Text Data in the Python Language
Kategorie: Data mining

Zadání:

1. Seznamte se s problematikou dolování v textu, včetně předzpracování těchto dat.
2. Prostudujte dostupné knihovny pro dolování v textu v jazyce Python a vyhledejte vhodné datasety pro tento účel.
3. Po dohodě s vedoucím navrhněte experimentální aplikaci provádějící předzpracování textových dat a některé zvolené úlohy dolování z textu.
4. Navrženou aplikaci implementujte, ověřte funkčnost a proveďte experimentální vyhodnocení.
5. Zhodnoťte dosažené výsledky a další možnosti pokračování tohoto projektu.

Literatura:

- Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Third Edition. Morgan Kaufmann Publishers, 2012, 703 p., ISBN 978-0-12-381479-1.
- Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, 2007.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Bartík Vladimír, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 20. října 2021

Abstrakt

Táto bakalárska práca sa zaoberá získavaním znalostí z textových dát, konkrétnejšie klasifikáciou textových recenzií užívateľov. Pomocou experimentov sa táto práca zameriava na metódy predspracovania textových dát a na porovnanie jednotlivých klasifikačných metód prostredníctvom vybraných dátových sád. Záverom práce je zhodnotenie dosiahnutých výsledkov experimentov, ktoré boli vykonané pomocou implementovanej aplikácie.

Abstract

This bachelor thesis deals with knowledge discovery from text data more specifically classification of text-based user reviews. Using experiments, this thesis focuses on methods for preprocessing text data and comparing different classification methods through selected datasets. The conclusion of the work is the evaluation of the achieved results of experiments that were performed using the implemented application.

Kľúčové slová

dolovanie z textových dát, predspracovanie textových dát, klasifikácia, klasfikačné metódy, strojové učenie, python

Keywords

text mining, text preprocessing, classification, classification methods, machine learning, python

Citácia

HOMOLA, Ján. *Získávání znalostí z textových dat v prostředí jazyka Python*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Bartík, Ph.D.

Získávání znalostí z textových dat v prostředí jazyka Python

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Vladimíra Bartíka, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Ján Homola
8. mája 2022

Poďakovanie

Rád by som poďakoval svojmu vedúcemu práce Ing. Vladimírovi Bartíkovi, Ph.D. za odbornú pomoc, cenné rady a čas, ktorý mi venoval pri vypracovaní tejto bakalárskej práce. Taktiež by som sa chcel poďakovať svojej priateľke a rodine za podporu počas písania bakalárskej práce.

Obsah

Kapitola 1

Úvod

V dnešnej dobe sú informačné technológie súčasťou nášho každodenného života, čo spôsobuje produkovanie obrovského objemu dát, s čím je úzko spätá potreba získať z týchto dát zaujímavé poznatky. Tieto poznaky môžu byť neskôr prospešné napríklad v rámci odhalenia podvodu, riadenia výroby alebo na udržanie si zákazníkov.

Táto bakalárska práca sa zaoberá získavaním znalostí z textových dát, konkrétnejšie sa jedná o klasifikáciu užívateľských recenzií. Vďaka takéjto automatizovanej analýze recenzií, môžu spoločnosti zistiť, čo verejnosť hovorí o ich produkte, ako produkt vníma cieľová skupina, ktoré prvky produktu je potrebné vylepšiť a taktiež zistiť, čo si zákazníci na danom produkte najviac oceňujú.

Cieľom tejto práce je vytvoriť aplikáciu, ktorá umožní porovnať vybrané klasifikačné metódy na dátových sadách s užívateľskými recenziami. Kapitola 2 sa zaoberá základnou teóriou v oblasti získavania znalostí z dát, ktorá popisuje proces získavania znalostí, druhy dolovacích dát a typické dolovacie úlohy. Následne kapitola 3 popisuje získavanie znalostí z textových dát, v rámci ktorej sú predstavené metódy predspracovania týchto textových dát spolu so spôsobmi prevodu textových dát do číselnej reprezentácie. V kapitole 4 je pozornosť zameraná na klasifikáciu textových dát spolu s využítím tohto typu dolovacej úlohy a postupným predstavením vybraných klasifikačných metód. Kapitola 5 popisuje návrh a implementáciu výslednej aplikácie určenej na klasifikáciu textových dát. Experimenty vykonané pomocou vytvorenej aplikácie sú zhrnuté v kapitole 6.

Kapitola 2

Získavanie znalostí z dát

V tejto kapitole sú výsvetlené zakladné pojmy, ktoré súvisia s problematikou získavania znalostí z dát. Predstavenie týchto zakladných pojmov, je kľučové pre pochopenie nasledujúcich kapitol.

Na začiatku kapitoly je predstavená definícia získavania znalostí a aké sú praktické využitia tohto druhu odvetia. Sekcia 2.2 sa zaoberá jednotlivými krokmi procesu získavania znalostí z dát. Následne sa v sekcii 2.3 zameriame na druhy zdrojových dát, ktoré používame k získavaniu znalostí. Za tradičné zdroje dát sú považované relačné databázy, dátové sklady a transakčné databázy. Okrem nich existuje mnoho ďalších typov dát, ktoré sú taktiež potenciálne užitočným zdrojom. Aj na tie sa bližšie pozrieme v tejto sekcii. Na záver tejto kapitoly sa v sekcií 2.4 budeme venovať základným typom dolovacích úloh.

2.1 Pojem získavanie znalostí

Získavanie znalostí z dát, taktiež známe pod anglickým názvom data mining. Tento odbor sa formoval na základe rýchleho vývoja informačných technológií, ktoré sa stali súčasťou bežného života. Toto spôsobilo výrazný nárast množstvá dát a nutnosť získať relevantné znalosti v týchto dátach. Tento proces môžeme definovať ako extrakciu modelov dát a vzorov z veľkého množstva dát. Tieto vzory a modely sú zaujímavé, potenciálne užitočné, skryté, netriviálne a predom neznáme, pričom predstavujú získané znalosti z dát [26].

Získané informácie sú skryté, to znamená, že nie sú v dátach explicitne uložené, ale musíme ich v dátach najsť netriviálnym postupom. Okrem toho by mali byť potenciálne užitočné a to napríklad v oblasti bankovníctva, marketingu, obchodu alebo pri získavaní znalostí z textu, tejto oblasti bude podrobnejšie venovaná kapitola 3.

2.2 Proces získavania znalostí

Proces získavanie znalosti z dát pozostáva z viacerých krokov, ktoré môžeme detailnejšie vidieť na obrázku 2.1. Počas tohto procesu zvyčajne dochádza k opakovaniu jednotlivých krokov za účelom získania čo najlepšieho výsledku. Ide o nasledujúcich sedem krokov [26]:

1. **Čistenie dát** – spočíva vo vyplnení prázdnych hodnôt, ktoré môžu byť doplnené manuálne, avšak tento spôsob je časovo veľmi náročný preto sa častejšie táto hodnota nahradí priemernou hodnotou alebo hodnotou s najväčším výskytom. Počas čistenia dát dochádza aj k vyhladeniu šumu, odstráneniu hodnôt, ktoré sú výrazne odlišné alebo nekonzistentné.

2. **Integrácia dát** – je spojenie dát z rôznych zdrojov za účelom zhromaždiť všetky dostupné dáta. V tomto kroku je potrebné riešiť problémy, ktoré integrácia dát prináša a to je problém redundantných dát. Ďalším problémom, ktorý sa naskytá je, ako rozlíšiť rovnaké entity, ktoré sa nachádzajú v rôznych dátových zdrojoch.

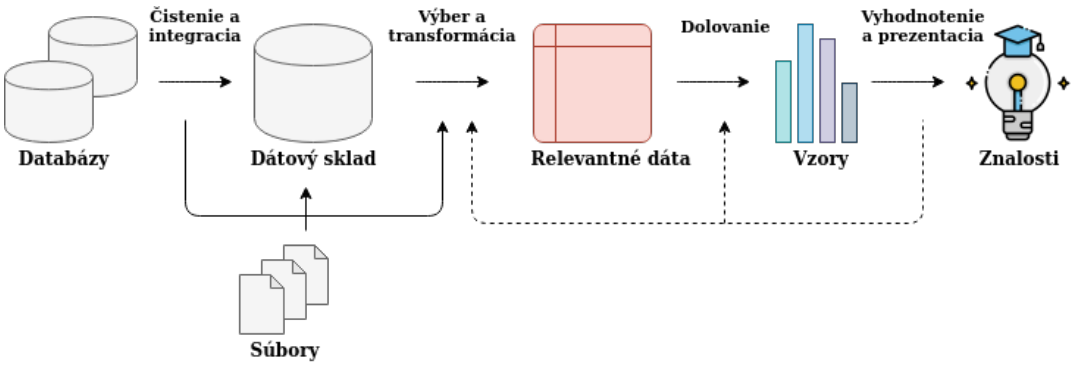
3. **Výber dát** – úlohou tohto kroku je vybrať a zredukovať dáta, ktoré sú potrebné pre riešenie dolovacej úlohy. Výstupom sú zredukované dáta, ktoré neovplyvnia výsledok dolovania.

4. **Transformácia dát** – slúži na transformáciu dát do takej formy, aby bolo možné na tieto dáta použiť vybranú dolovaciu metódu. V tomto kroku taktiež dochádza k zefektívneniu samotného dolovania a výsledné dáta môžu byť ľahšie pochopiteľné. Pre ľahšie porozumenie výsledných dát sa často používa generalizácia, ktorá zovšeobecňuje dáta. Niektoré dolovacie metódy často potrebujú dáta v normalizovanej podobe, tieto dáta sú transformované na intervale, ktorý je vopred známy, ako napríklad interval $<-1, 1>$ alebo $<0, 1>$. Tento druh transfromácie dát je napríklad dôležitý pri dolovacích metódach založených na neurónových sieťach.

5. **Dolovanie dát** – tento krok je základom celého procesu získavania znalostí, jeho výsledkom sú extrahované vzory alebo modely, ktoré boli získané na základe aplikácie inteligentnej metódy a konkrétneho algoritmu.

6. **Vyhodnotenie vzorov a modelov** – cieľom tohto kroku je identifikovať vzory, ktoré skutočne reprezentujú zaujímavé znalosti prostredníctvom miery užitočnosti. Pokiaľ miera užitočnosti nie je dostatočná, dochádza k ďalšej iterácií celého procesu.

7. **Prezentácia získaných znalostí** – úlohou posledného kroku tohto celého procesu je prezentovanie získaných znalostí v takej podobe, aby bolo možné ich ľahko a efektívne pochopiť zo strany užívateľa. Výsledok môže byť prezentovaní, napríklad pomocou tabuľky, pravidiel alebo grafov.



Obr. 2.1: Kroky procesu získavania znalosti, prevzaté z [26]

Prvé štyri kroky tohto procesu sú označované ako predspracovanie dát, ktoré dokáže zefektívniť samotný proces dolovania. V hlavnom kroku dolovania dát niekedy nastáva interakcia s databázou znalostí alebo s používateľom.

2.3 Druhy dát

Ziskavanie znalostí môže byť použité na dáta rôzneho druhu a to buď tranzientné dáta alebo také, ktoré sú v uložiskách uložené perzistentne. Medzi najčastejšie zdroje dát v súčastnosti sú relačné databázy. Okrem tohto druhu dát sú to taktiež dátové sklady, transakčné databázy, ale aj iné typy dátových zdrojov [9].

Relačné databázy

Relačné databázy sú jedny z najviac dostupných a bohatých zdrojov informácií, práve preto sú tieto zdroje dát najčastejšie používané pri dolovacích úlohách.

Jedná sa o kolekciu tabuliek, každá z nich má jedinečný názov a musí spĺňať pravidlá prvej normálnej formy. Jednotlivé tabuľky pozostávajú z atribútov (stĺpcov) a niekoľkých záznamov, ktoré reprezentujú riadky. Každý záznam v relačnej tabuľke je identifikovaný pomocou primárneho kľúča, ktorý pozostáva z neprázdnej množiny atribútov.

K dátam uloženým v relačnej databáze je možné pristupovať prostredníctvom databázových príkazov zapísaných typicky v jazyku SQL alebo pomocou grafického rozhrania. Zadaný príkaz je transformovaný do množiny relačných operácií ako je selekcia, projekcia alebo spojenie [9].

Dátové sklady

Dátový sklad je úložisko informácií, ktoré sú zozbierané z viacerých rozdielnych zdrojov dát a sú uložené pomocou jednotnej schémy obvykle na jednom mieste. Tieto dátové sklady vznikajú prostredníctvom čistenia dát, integrácie dát, transformácie dát, načítania dát a taktiež periodického aktualizovania dát. Údaje sa uchovávajú v dátovom sklade z historickej perspektívy, napríklad dáta za posledných 8 rokov a sú typicky sumarizované [26].

Dátový sklad je zvyčajne modelovaný multidimenzionálnou dátovou štruktúrou, známou ako **multidimenzionálna dátova kocka**. Každá dimenzia tejto dátovej kocky zodpovedá atribútu alebo množine atribútov databázovej schémy, kde každá bunka obsahuje údaje, ktoré sú agregované a vďaka tomu je vhodný pre vykonávanie OLAP (Online Analytical Processing) operácií. Dátový sklad prostredníctvom operácií OLAP poskytuje lepšiu podporu pre dátovú analýzu, než bežné relačné databázy, a preto ma väčší potenciál na získanie zaujímavých vzorov reprezentujúcich znalosti [26].

Transakčné databázy

Vo všeobecnosti je transakčná databaźa chápaná ako súbor, ktorého záznamy reprezentujú jednotlivé transakcie. Jedná sa o transakcie, ako napríklad nákup zákazníka alebo rezervácia leteniek. Takéto transakcie typicky pozostávajú z jedinečného identifikátoru transakcie a zoznamu položiek. Transakčné databázy môžu ešte zahŕňať ďalšie tabuľky, ktoré obsahujú informácie o zákazníkoch alebo o zakúpených produktoch.

Cieľom dolovacích úloh čerpajúcich z tohto druhu dát môže byť napríklad získánie množiny produktov, ktoré sú kupované spoločne [26].

Iné typy zdrojov dát

Pri dolovaní znalostí sa môžeme často stretnúť okrem už spomenutých zdrojov dát aj s ďalšími typmi, ako sú napríklad [26]:

- **Sekvenčné databázy** – tieto databázy uschovávajú postupnosti usporiadaných udalostí, rozhodujúce je poradie jednotlivých udalostí, takže čas nie je potrebné explicitne zaznamenávať.
- **Priestorové databázy** – uschovávajú dáta, ktoré poskytujú informácie a poznatky vzťahujúce sa k určitým miestam v priestore. Tieto databázy sú označované ako geografické databázy alebo geodatabázy.
- **Textové databázy** – obsahujúce štruktúrované dokumenty (knihy), neštruktúrované alebo čiastočné štruktúrované dokumenty (napr. JSON, XML). Dolovaním tohto typu dát je možné napríklad rozlíšiť bežnú správu od spamu. Dolovanie týchto dát je podrobnejšie popísaní v nasledujúcej kapitole 3.
- **Prúdy dát** – sú to dáta produkované súvisle. Medzi ich základnú vlastnosť patrí obrovský objem, ktorý je potenciálne nekonečný a dynamicky sa meniaci. Najčastejšie sa jedná o dáta pochádzajúce zo senzorov.
- **Multimediálne databázy** – databázy, ktoré uschovávajú audio a video dáta alebo fotografie. Niekedy sa aj textové dáta radia medzi multimediálne databázy.
- **Databázy časových radov** – obsahujú postupnosti hodnôt alebo udalostí, ktoré sa vzťahujú k meranému času. Príkladom takýchto dát môžu byť ceny jednotlivých akcií na burze.
- **Web** – heterogénna, obrovská databáza s veľkým potenciálom pre dolovanie, ktorá vďaka každodennému používaniu internetu rýchlo rastie. Medzi typické úlohy dolovania týchto zdrojov dát patrí automatické zhlukovanie alebo klasifikácia webových stránok.

2.4 Typy dolovacích úloh

V tejto podkapitole budú predstavené základné typy dolovacích úloh. Typ dolovacej úlohy reprezentuje, aké druhy vzorov a modelov dát sa snažíme zo zdrojových dát získať. Jedná sa o riešenie úlohy v kroku dolovania dát celého procesu získavania znalosti. Tieto dolovacie úlohy je vo všeobecnosti možné rozdeliť do dvoch základných kategórií [26]:

- **Deskriptívne** – cieľom týchto úloh je charakterizovať všeobecné vlastnosti analyzovaných dát, poprípade objaviť závislosť medzi týmito dátami. Predstaviteľmi tohto typu dolovacích úloh sú napríklad zhluková analýza alebo hľadanie asociačných pravidiel.
- **Prediktívne** – cieľom týchto úloh je predpovedať neznáme alebo budúce hodnoty atribútov a budúce správanie na základe analýzy aktuálnych dát. Medzi predstaviteľov tohto typu dolovacích úloh patrí napríklad klasifikácia alebo regresia.

Popis konceptu/triedy

Popis konceptu/triedy patrí medzi základné typy dolovacích úloh. Cieľom je popísať triedy alebo koncepty a to nejakým stručným, súhrnným a zároveň značne presným spôsobom. Tento spôsob nazývame **popis triedy/konceptu**, ktorý môžeme dosiahnuť prostredníctvom nasledujúcich spôsobov [26]:

- **Charakterizácia dát** – predstavuje sumarizáciu všeobecných charakteristík alebo vlastností analyzovanej triedy. Údaje zodpovedajúce triede špecifikovanej používateľom je možné typicky získať pomocou jednoduchého databázového dotazu. Príkladom triedy môže byť produkt, ktorého predaj stúpol o 17% za posledný polrok. Výstup chrakterizácie dát može byť reprezentovaný v rôzných formách ako napríklad stĺpcový, koláčový graf alebo dátová kocka [9].

- **Diskriminácia dát** – predstavuje porovnanie všeobecných vlastností cieľovej triedy so všeobecnými vlastnosťami jednej alebo viacerých rozdielnych tried. To znamená, že tentokrát hľadáme atribúty a ich hodnoty, v ktorých sú triedy najviac rozdielne. Dolovacia úloha by mohla byť napríklad, aký je rozdiel medzi produktmi, ktorých predaj za posledný polrok stúpol o 17 % od tých produktov, ktorých predaj klesol o 40 % [26].

Frekventované vzory a asociačné pravidlá

Medzi ďalšie typy dolovacích úloh patria frekventované vzory a asociačné pravidlá. Frekventované vzory, ako už názov napovedá, sú to vzory, ktoré sa často vyskytujú v dátach. Tieto vzory zahŕňajú množstvo druhov, ako sú napríklad frekventované množiny, frekventované podpostupnosti (frekventované sekvenčné vzory). Taktiež môžeme spomenúť aj iné podštruktúry, ako napríklad podstromy alebo podgrafy. Frekventovaná množina sa zvyčajne vzťahuje na množinu položiek, ktoré sú spolu často objavované [9].

Uvedieme si príklady jednotlivých frekventovaných vzorov. Ako príklad frekventovaných množín môžeme spomenúť futbalové kopačky a loptu, tieto produkty sú často v športových obchodoch kupované spoločne. Pri frekventovanéj podpostupnosti môže byť príkladom, kedy si zákazníci zvyčajne najkôr kúpia kameru a následne až pri niektorom z ďalších nákupov si dokúpia statív.

Prostredníctvom dolovania frekventovaných vzorov dochádza k objaveniu zaujímavých asociácií a korelácií. Počas objavovania týchto zaujímavých asociacií vznikajú *asociačné pravidlá*, tento proces objavovania sa nazýva *asociačná analýza*.
Príklad asociačného pravidla môže byť v tvare [26]:

$$kupuje(X, \textit{'kopačky'}) \Rightarrow kupuje(X, \textit{'lopta'})[podpora = 0.7\%, \textit{'spoľahlivosť'} = 59\%]$$

kde premena X reprezentuje daného zákazníka. Tento príklad asociačného pravidla hovorí, že zákaznící, ktorí si kúpia kopačky obvykle ich nákup obsahuje aj loptu. Na vyjadrenie významnosti príslušného frekventovaného vzoru v skúmaných dátach sa najčastejšie používajú dve miery *podpora* a *spoľahlivosť*. V tomto prípade podpora 0.7% vyjadruje v koľkých percentách zo všetkých skúmaných nakupov si zakáznik kúpil súčasne kopačky a loptu. Spoľahlivosť zase značí, že v 59% nákupov, pri ktorých si zákazník kúpil kopačky, si kúpil zároveň aj loptu. Tento príklad asociačného pravidlá obsahuje iba jeden predikát, ktorý je *kupuje* a preto ide o *jednodimenzionálne asociačné pravidlo*.

Avšak asociačné pravidla možu obsahovať aj viac ako jeden predikát, vtedy sa jedná o *multidimenzionálne asociačné pravidlo*. Príkladom tohto druhu asociačného pravidla je napríklad asociačné pravidlo obsahujúce tri predikáty, ktoré môže vyzerať takto [26]:

$$vek(X, \textit{'15..23'}) \wedge pohlavie(X, \textit{'muž'}) \Rightarrow kupuje(X, \textit{'playstation'})$$
$$[podpora = 1.7\%, \textit{'spoľahlivosť'} = 62\%]$$

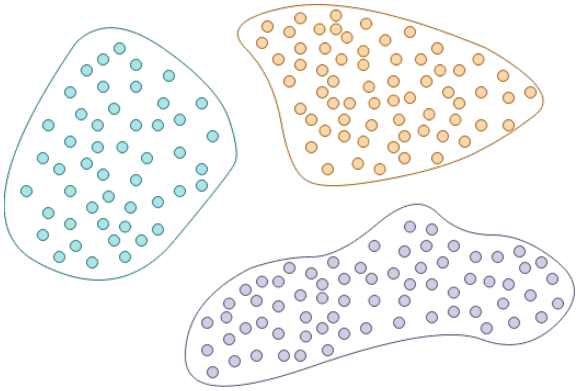
toto pravidlo hovorí 1.7% zo skúmaných zákazníkov, ktorí mali od 15 do 23 rokov a boli mužského pohlavia si kúpilo playstation. S pravdepodobnosťou 62% si zákazník mužského pohlavia v tejto vekovej kategórií kúpil playstation. Predikáty tohto multidimenzionálneho asociačného pravidla sú *vek*, *pohlavie* a *kupuje*

Kedže máme k dispozícií objektívne miery zaujímavosti, tak v prípade asociačných pravidiel sú zaujímavé iba vtedy, pokiaľ spĺňajú minimálnu hodnotu prahu spoľahlivosti a podpory. Tieto objektívne miery sa často kombinujú so subjektívnym posudením používateľa [26].

Zhluková analýza

Zhluková analýza je podobná klasifikácií v tom, že jednotlivé dátové objekty su priraďované do tried. Avšak rozdiel je v tom, že tieto triedy pri priraďovaní nie sú predom známe [4]. Zhlukovanie je možné použiť na generovanie cieľových tried pre analyzované dáta. Objekty sú zhlukované na základe princípu maximálnej podobnosti objektov rovnakej triedy a minimálnej podobnosti objektov rozdielnych tried. Takže triedy objektov sú tvorené tak, že objekty v rovnakej triede majú navzájom vysokú podobnosť a zároveň sú dosť odlišné od objektov v iných triedach. Takto vytvorené triedy sú nazývané zhluky [9]. Príklady aplikácie zhlukovej analýzy môžu byť [1]:

- **Segmentácia zákazníkov** – jedná sa o analýzu bežného správania podobných skupín zákazníkov. Na základe vytvorených skupín zákazníkov vieme následne napríklad odporučiť produkt, ktorý je pre danú skupinu zákazníkov potenciálne zaujímavý.
- **Vzťah k ďalším problémom v oblasti dolovania** – Zhlukovanie môže byť použité ako krok predspracovania dát pre ďalšie algoritmy určené pre dolovacie úlohy, ako je klasifikácia a charakterizácia, ktoré následne pracujú nad vytvorenými triedami.



Obr. 2.2: Príklad zhlukovej analýzy, ukazuje tri zhluky, prevzané z [26]

Klasifikácia a regresia

Ako už bolo spomenuté, klasifikácia a regresia patria medzi prediktívne dolovacie úlohy. Klasifikácia je proces vytvorenia modelu, ktorý popisuje a zároveň rozlišuje jednotlivé triedy dát. Celý proces klasifikácie pozostáva z troch krokov, a to trénovanie, testovanie a nakoniec aplikácia modelu. Vytvorený model je následne použitý k predikcii kategorických hodnôt. Tieto kategorické hodnoty sú neusporiadané a diskrétne. Zatiaľ čo regresia sa používa na

predikciu chybajúcich alebo nedostupných numerických hodnôt, ktoré sú spojitého charakteru. Medzi najčastejšiu štatistickú metódu používanú na numerickú predikciu je **regresná analýza** [9]. Klasifikácia a jej algoritmy sú podrobnejšie vysvetlené v kapitole 4.

Ostatné typy dolovacích úloh

Medzi ďalší typ dolovacej úlohy patrí **analýza odľahlých objektov**. U tohto druhu dolovacej úlohy predstavujú zaujímavé vzory nie také hodnoty, ktoré sa vyskytujú obvykle v analyzovaných dátach ale práve tie, ktoré sa výrazne odlišujú od ostatných. Takáto analýza môže odhaliť napríklad zneužitie kreditnej karty a to na základe rozpoznania neobvykle častého používania kreditnej karty, veľmi vysokých platieb a taktiež pokiaľ sú platby vykonávané z nezvyčajného miesta [26].

Na koniec sa pozrieme ešte na jeden typ dolovacej úlohy a to je **analýza evolúcie**. Tento typ dolovacej úlohy popisuje a modeluje trendy a pravidelnosti u takých dátových objektoch, ktoré sú časovo premenlivé. Jedným z príkladov takejto dolovacej úlohy je odhalenie pravidelnosti, trendu vo vývoji cien alebo akcií, a následne predpovedanie vývoja týchto cien, čo môže zefektívniť proces investovania [26].

Kapitola 3

Získavanie znalosti z textových dát

Ako už bolo spomenuté v predchadzajúcej kapitole jedným zo zdrojov dát pre dolovanie môžu byť textové dáta. Ide o získavanie znalostí z textových dát, známe pod anglickým názvom *text mining*.

Dolovanie textu môže byť definované ako znalostne náročný proces získavania vysoko kvalitných informácií z textových dát. Cieľom tohto procesu je objaviť nové, predom neznáme informácie a vzťahy z rôznych zdrojov prostredníctvom identifikácie a skúmania zaujímavých vzorov. V prípade dolovania textu sú však zdrojmi textové dáta a zaujímavé vzory sa nenachádzajú medzi databázovými záznamami ale v neštruktúrovaných textových dátach. Tieto textové zdroje môžu byť rôzne recenzie produktov, emaily, články, webové stránky a mnoho ďalších [5]. Medzi typické úlohy tohto druhu dolovania patrí napríklad klasifikácia textových dát, zhlukovanie textových dát a sumarizácia textových dát [18]. V nasledujúcej kapitole si bližšie predstavíme klasifikáciu textových dát, ktorou sa zaoberá praktická časť tejto práce.

Počas dolovania textových dát je potrebné použiť rôzne techniky predspracovania, aby bolo možné získať cenné poznatky z neštruktúrovaných textových dát. Taktiež je potrebná transformácia dát do takej podoby, ktorým môžu stroje porozumieť [5]. Predspracovanie textových dát bude dôkladnešie vysvetlené v sekcii 3.2.

3.1 Využitie

Získavanie znalostí z textových dát má široké uplatnenie, stretávame sa s ním v rôznych odvetviach. Kedže existuje mnoho príkladov využitia v reálnom živote, tak si v tejto sekcii spomenieme iba niektoré z nich [20] [23].

- **Služba starostlivosti o zákazníkov** – v rámci získavani znalostí z textových dát nadobúda v súčastnosti čoraz väčší význam. Vďaka pokročilým technológiam v dnešnej dobe existujú rôzne spôsoby, ktorými môže zákazník poskytnúť spätnú väzbu a to pomocou rozličných prostriedkov, ako sú napríklad zákaznícke prieskumy, online recenzie. Zákazníckymi recenziami sa zaoberá praktická časť tejto práce. Kombinácia spätnej väzby s nástrojmi na analýzu textu môže viesť k zvýšeniu spokojnosti zákazníkov a nárastu množstva skúseností s vysokou rýchlosťou riešenia problému.
- **Odhaľovanie podvodov** – využívajú obvykle poisťovne a finančné spoločnosti, pretože v týchto odvetiach sa väčšinou zhromažďujú všetky údaje v textovej forme, čo predstavuje obrovský potenciál pre získanie znalostí. Tieto inštitúcie sú následne

schopné rýchlo spracovať nároky a zabrániť podvodom pomocou výsledkov textovej analýzy s príslušnými štrukturovanými údajmi.

- **Filtrovanie spamov** – nežiadúce správy často slúžia ako vstupný bod pre hackerov na infikovanie počítačových systémov škodlivými softvérmi alebo ako reklamy. Textová analýza poskytuje metódu na filtrovanie a vymazanie takýchto správ, čím sa zvyšuje používateľská skúsenosť a minimalizuje riziko kybernetických útokov.
- **Risk managment** – bez ohľadu na odvetie je často nedostatočná analýza rizík hlavnou príčinou zlyhania, platí to hlavne vo finančnej sfére. Dolovanie textových dát v tomto odvetví môže dramaticky zmierniť riziko, zhromaždiť relevantné informácie z tisícok textových zdrojov, poskytnúť prepojenie informácií a zabezpečiť prístup k správnym infromáciam v správnom čase. Následne tento prístup k informáciam je možné uplatniť napríklad pri investovaní.
- **Odporúčanie reklám** – je pomerne nová a rastúca oblasť pre používanie dolovania textových dát. Pomocou dolovania textu môžu podniky pochopiť kontext na webovej stránke a umiestniť reklamy, ktoré sú relevantné pre infromácie obsiahnuté na webovej stránke, čo zvyšuje pravdepodobnosť úspechu reklamy. Napríklad reklama na kávovar bude lepšie fungovať na stránke o domácich spotrebičoch ako na stránke so športovým vybavením.

Ďalšie zaujímavé príklady použitia dolovania textových dát v bežnom živote je možné nájsť v oblasti medicíny, vo sfére bezpečnosti na sociálnych sieťach alebo taktiež v oblasti biznis inteligencie.

3.2 Predspracovanie textových dát

Predspracovanie textu je motóda na vyčistenie a zjednodušenie textových dát a ich transformácia do takej podoby, aby bolo možné jednoduchšie aplikovať na tieto dáta špecifickú dolovaciu úlohu. Hlavným problémom textových dát je to, že v ľudskom jazyku existuje mnoho spôsobov, ako povedať to isté. S týmto problémom sa musíme vysporiadať pretože stroje nedokážu porozumieť slovám ale len číslam, tekže musíme text previesť na čísla a to efektívnym spôsobom [2].

Na celkový úspech jednotlivých dolovacích úloh má proces predspracovania významný vplyv. Taktiež počas tohto procesu dochádza k redukcii vstupných dát a tým sa zrýchli samotný proces získavania znalostí. Napríklad pri dolovacích úlohách ako je klasifikácia a zhlukovanie je predspracovanie veľmi dôležite.

Predspracovanie pozostáva zvyčajne z krokov ako je odstránenie šumu, tokenizácia, prevod textu na malé písmená, odstránenie stopslov, hľadanie koreňa slov pomocou stematizácie alebo lematizácie [11] [14]. V nasledujúcej časti si ich detailnejšie popíšeme.

Odstránenie šumu

Odstránenie šumu spočíva v odstranení časti textov a znakov, ktoré nemajú význam pre danú dolovaciu úlohu a môžu narúšať výsledky jednotlivých dolovacích úloh. Tento krok predspracovania nie je vždy nevyhnutný a jeho použitie zavisí od druhu textových dát. Pokiaľ naše vstupné dáta pochádzajú napríklad z webových dokumentov je veľká pravdepodobnosť, že náš text bude zabalený do HTML značiek, ktoré je vhodné odstrániť [14].

Prevod textu na malé písmená

Tento krok prevodu textu na malé písmena sa často prehliada pritom patrí medzi najjednoduchší a veľmi efektívny krok predspracovania textu. Zabránime tak výskytu rôznych kópií rovnakých slov. Nie je nutné prevádzať písmená na malé, môžeme to urobiť aj opačne a previeť písmena na veľké.

Avšak u niektorých dolovacích úloh je dôležité zachovať veľké písmená, aby sme predišli strate informácie. Príkladom môže byť slovo *dom* čo reprezentuje miesto na bývanie a *DOM*, ktoré predstavuje skratku slova *Document Object Model*. Takže prevod na malé písmená je vo všeobecnosti užitočný krok, ale nemusí byť vhodný pre všetky úlohy [14].

Tokenizácia

Krok tokenizácie je definovaný ako proces rozdeľovania reťazcov slov na jednotlivé tokeny oddelené medzerami alebo iným znakom. Následne sú odstránené číselné hodnoty, špeciálne znaky, ako napríklad interpunkčné znamienka, a taktiež znaky ako sú zátvorky, spojovník a mnoho ďalších. Takto vytvorený list tokenov sa stáva vstupom pre ďalšie kroky predspracovania textu ako odstránenie stopslov, stematizácia alebo lematizácia [11].

Odstránenie stopslov

Stopslová patria medzi bežne používané slová v jazyku, ktoré majú iba gramatický význam, ako napríklad predložky *in*, *on*, *to* spojky *and*, *or*, *but* členy *a*, *an* neurčitý člen *the* a mnoho ďalších. Tieto takzvané stopslová sa obyčajne v texte vyskytujú s vysokou frekvenciou, príčom ich informačný zisk je veľmi malý a mohol by skresliť výsledok dolovacej úlohy [11].

Kedže sa jedná o veľmi frekventované slova obvykle vo všetkých textových zdrojoch, tak odstránením týchto slov dokážeme výrazne znížiť veľkosť vstupných dát. Takáto redukcia dát môže výrazne urýchliť proces získavania znalostí.

Stematizácia

V angličtine a mnohých ďalších jazykoch sa slová vyskytujú v texte vo viacerých tvaroch a to vďaka časovaniu alebo skloňovaniu. Ako príklad si môžeme uviesť anglické slová *played*, *playing* a *plays*. Stematizácia je proces počas, ktorého sú slová nahradzované ich koreňovou formou [14]. V našom prípade bude koreň slova *play*. Cieľom tohto procesu je zjednotenie slov, ktoré majú rozdielny tvar ale nesú rovnaký význam. Výsledný koreň slova v tomto prípade nemusí byť skutočným koreňom. Pomocou objavenia základného tvaru slova dochádza k výraznej redukcii počtu termov.
Počas procesu stematizácie môžu nastať dve hlavné chyby [10]:

- **over stemming** – ide o chybu, kedy sú dva slová s rozdielnym základom slova prevedené na rovnaký základ slova. Táto chyba je tiež známa pod názvom *false positive*.
- **under stemming** – ide o chybu, keď dva slová, ktoré majú rovnaký základ slova, nie sú prevedené na spoločný základ slova. Táto chyba je tiež označovaná ako *false negative*.

Existujú rôzne algoritmy na stematizáciu, medzi najviac využívaný algoritmus pre anglický jazyk je *Porter Stemmer*. Tento algoritmus je založený na odstraňovaní prípon, pomocou pevného zoznamu prípon a ďalších pravidiel anglického jazyka [18].

Lematizácia

Lemátizácia je veľmi podobná procesu stematizácie, ktorá ma taktiež za úlohu prevod slov do ich základného tvaru. Hlavným rozdielom je, že lematizácia sa to snaží vykonať správnym spôsobom, tým že využíva slovník a morfologickú analýzu slov za účelom odstránenia slovotvorných, pádových a iných predpon a prípon. Ďalej sa lematizácia odlišuje od stematizácie hlavne tým, že dokáže rozlišovať aj slová, ktoré majú rovnaký základ slova, ale majú rozdielny význam [3].

Ako príklad môžeme uviesť anglické slovo *better*, ktoré pomocou procesu lematizácie prevedieme do slovníkovej podoby *good*.

3.3 Prevod textu do vektorovej podoby

Na vykonanie akejkoľvek dolovacej úlohy ako napríklad klasifikácia, regresia alebo zhlukovanie je potrebné reprezetovať textové dáta pomocou čísel. Dôvodom je to, že mnohé algoritmy, ktoré sa využívajú pri jednotlivých dolavacích úlohách nie sú schopné spracovať textové dáta v ich surovej podobe [7]. Jedným zo spôsobov číselnej reprezentácie sú práve číselné vektory [22]. Máme rôzne spôsoby na prevod textových dát na čiselné vektory, ako napríklad Bag of Words, One–hot kódovanie alebo TF–IDF, ktoré si bližšie popíšeme nižšie [22] [7].

Bag of Words

Tento model je zredukovaná a zjednodušená reprezentácia textových dát, ktorá hovorí o frekvencií výskytu slov v textovom dokumente. Všetky informácie o poradí alebo štruktúre slov v texte sa nezohľadňuju. Model sa zoberá iba tým, či sa slová v dokumente vyskytujú a nie tým, kde sa v dokumente nachádzajú [12].

Prvým krokom pri aplikácií Bag of Words modelu je prechádzanie celej zbierky dokumentov a následné vytvorenie zoznamu unikátnych slov. V nasledujúcom kroku dochádza k opätovnému prechádzaniu celej zbierky dokumentov a určí sa frekvencia jednotlivých slov v danom dokumente [27].
Jednotlivé kroky si ukážeme na dvojici krátkych recenzií:

1. Veľmi pekná knižka.
2. Veľmi pekná knižka, ale veľmi dlhá.

V rámci tohto príkladu neberieme do úvahy odstránenie stop slov ani iné kroky predspracovania. Ako prvé sa vytvorí zoznam unikátnych slov, tieto slová sú: veľmi, pekná, knižka, ale, dlhá. Následne sa každému slovu priradí frekvencia výskytu v rámci danej recenzie. Tento proces je zobrazený v tabuľke 3.1.

	veľmi	pekná	knižka	ale	dlhá
Veľmi pekná knižka.	1	1	1	0	0
Veľmi pekná knižka, ale veľmi dlhá.	2	1	1	1	1

Tabuľka 3.1: Ukazuje vysledný vektor jednotlivých recenzií, prostredníctom modelu BoW

One-hot kódovanie

Jedná sa o jednu z jednoduchších metód na prevod textových dát do vektorovej podoby. Pri tejto metóde sa taktiež vytvorí zoznam unikátnych slov z celej zbierky dokumentov. Spočíva v tom, že každý dokument alebo časť textu je vektorovo reprezentovaná v tvare 1 a 0, kde 1 predstavuje, že dané slovo je zahrnuté v dokumente alebo časti textu a 0 značí, že toto slovo tam nie je zahrnuté. Jednou z nevýhod One-hot kódovania je, že veľkosť vektora sa rovná počtu jedinečných slov v celej zbierke dokumetov [7].

TF-IDF

Táto metóda je kombináciou frekvencie termov (TF) a inverznej frekvencie dokementov (IDF). Poskytuje mieru, ktorá zohľadňuje dôležitosť slova v závisloti od toho, ako často sa vyskytuje v dokumente a celej zbierke dokumentov. Váhu termínu (slova) t v dokumente d môžeme vyrátať pomocou vzorca 3.1.

$$W(d,t) = TF(d,t) * \log\left(\frac{N}{df(t)}\right) \tag{3.1}$$

Kde N je počet všetkých dokumentov v danej dátovej sade, $df(t)$ predstavuje počet dokumentov, ktoré obsahujú tento termín a ako už bolo spomenuté $TF(d,t)$ predstavuje frekvenciu termínu t v dokumente d [12].

Kapitola 4

Klasifikácia textových dát

Textová klasifikácia, tiež známa pod názvom kategorizácia textov, patrí medzi dolovacie úlohy, ako bolo spomenuté v predchádzajúcej kapitole. Proces tejto dolovacej úlohy spočíva v priradení preddefinovanej triedy k textovým dátam ako sú napríklad vety, odstavce alebo dokumenty na základe ich vlastností [15]. V kapitole 3 sme si vysvetlili, že tieto textové dáta je potrebné predspracovať a previesť do vektorovej reprezentácie. K riešeniu klasifikačných úloh používame zvyčajne algoritmy strojového učenia [11]. Niektoré z týchto algoritmov si predstavíme v sekcií 4.2.

Textové dáta sú veľmi bohaté zdroje informácií, takže textová klasifikácia má široké využitie v rôznych oblastiach [15]. Konkrétnymi príkladmi využitia textovej klasifikácie sa budeme zaoberať v sekcii 4.1.

4.1 Využitie

V dôsledku rýchlého vývoja informačných technológií sa textová klasifikácia používa celosvetovo v mnohých oblastiach, ako je medicína, spoločneské vedy, zdravodníctvo, marketing, právo, inžinierstvo a mnoho ďalších [12]. V tejto časti si bližšie popíšeme niekoľko úloh, ktoré používajú techniky klasifikácie textových dát, ako je napríklad analýza sentimentu, filtrovanie nežiaducich správ alebo pomoc v oblasti zákazníckej podpory [24], [11].

Analýza sentimentu

Analýza sentimentu je úloha, ktorá sa používa na určenie postoja, názoru, emócií vyjadrených osobou o kontrétnej téme, v textových dátach. Medzi tieto textové dáta môžu patriť napríklad recenzie produktov, filmov alebo správy [15]. Recenziám sa venuje čoraz väčšia pozornosť, pretože sú bohatým zdrojom pre marketingové tímy, sociológov, psychológov a mnoho ďalších, ktorí sa môžu zaoberať názormi a náladami verejnosti [8].

Úloha sa može zaoberáť binárnym problémom alebo problémom viacerých tried. Binárna analýza sentimentu klasifikuje textové dáta do dvoch tried a to pozitívnej a negatívnej triedy, zatiaľ čo viactriedna analýza sentimentu klasifikuje textové dáta do viacúrovňových intenzít emócií [15]. Binárnou analýzou a taktiež viactriednou analýzou sentimentu sa zaoberá praktická časť tejto práce, kde sú analyzované zákaznícke recenzie.

Rozhodovanie o sentimente je naročný problém, kvôli subjektívnosti, ktorá v podstate vyjadruje, čo si ľudia myslia [8].

Filtrovanie nežiadúcich správ

Ďalšou úlohou je filtrovanie nežiaducich správ, ktoré je jedným z príkladov binárnej klasifikácie. Pomocou tejto binárnej klasifikácie dokážeme zautomatizovať filtrovanie nežiaducich správ a tým používateľovi ušetríť čas, ktorý strávi odstraňovaním nepotrebných správ. Cieľ tejto úlohy spočíva v klasifikovaní každej správy do triedy nežiadúcich správ alebo do triedy bežných sprav [11].

Oblasť zákazníckej podpory

Klasifikácia textu môže pomôcť taktiež tímom zákazníckej podpory, čím dokážeme ušetriť drahocenný čas. Často sa používa na automatizáciu smerovania a triedenia tiketov. Klasifikácia textu umožnuje automaticky presmerovať požiadavku na zakaznícku podporu členovi tímu s konkrétnymi odbornými znalosťami danej témy. Napríklad ak zákaznik napíše otázku týkajúcu sa vrátenia peňazí, môžeme automaticky presmerovať túto požiadavku členovi tímu, ktorý má opravnenie na vrátenie peňazí a tým sa zabezpečí rýchlejšia a kvalitnejšia pomoc. Ďalším príkladom môže byť využitie detekcie jazyka a presmerovanie požiadavky na zákaznícku podporu, podľa druhu jazyka do príslušného tímu [16].

4.2 Klasifikačné metódy

Jedným z významých krokov v procese klasifikácie je výber správneho klasifikátora. Poznáme mnoho klasikačných metód, ktoré je možné použiť na textové dáta, ako napríklad K–Najbližších susedov, Naivný Bayesov klasifikátor, Support Vector Machines alebo taktiež Random Forest [12]. Táto práca sa zaoberá práve týmito metódami, ktoré budú v nasledujúcich podkapitolách bližšie vysvetlené.

4.2.1 K-najbližších susedov

Metóda K-najbližších susedov známa pod anglickým názvom *K–Nearest Neighbours* (ďalej ako KNN) je založená na podobnosti. Jej hlavnou myšlienkou je, že podobné položky patria do rovnakej triedy. Preto je potrebné len zozbierať a uložiť vhodný počet označených vzoriek patriacich do každej príslušnej skupiny, a týmto je tréningová fáza ukončená [27].

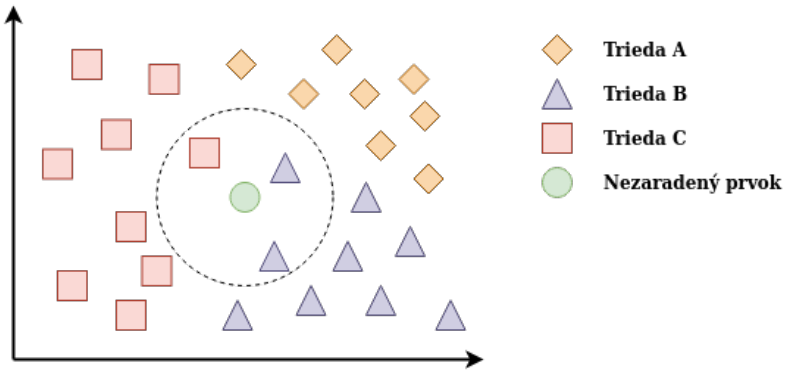
Počas procesu klasifikácie je potrebné určiť hodnotu K. Táto hodnota je faktorom, ktorý udáva požadovaný počet najpodobnejších dokumentov zo zbierky k nezaradenému dokumentu. Nezaradený dokument sa priradí do triedy s najvyšším výskytom [12]. Na rozdiel od trénovacej fázy je klasifikácia zvyčajne náročnejšia a často zložitejšia z dôvodu určovania podobnosti klasifikovanej položky vo vzťahu ku každej uloženej vzorke [27].

Jednou z typických metód na vyjadrenie numerickej podobnosti je Euklidovská vzdialenosť, ktorá je založená na výpočte vzdialenosti medzi dvojicami bodov. Vzorec 4.1 môžeme použiť na výpočet Euklidovskej vzdialenosti d_E medzi bodom A a bodom B s n rozmernými súradnicami [27].

$$d_E(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{4.1}$$

Obrázok 4.1 ukazuje myšlienku KNN, ale pre jednoduchosť je tento obrázok navrhnutý pomocou 2D dát, textové dáta bývaju väčšinou s väčším rozmerovým priestorom. V tomto

prípade $K = 3$, takže hľadáme tri najbližšie dokumenty za pomoci Euklidovskej vzdialenosti. Nezaradený prvok bude podľa metódy KNN priradený do triedy B, pretože sa v Euklidovskom priestore, kde $K = 3$, vyskytuje najčastejšie.



Obr. 4.1: Proces zaradenia prvku do triedy, kde $K = 3$, prevzaté a upravné z [12]

4.2.2 Naivný Bayesov klasifikátor

Naivný Bayesov klasifikátor patri medzi populárne motódy strojového učenia na klasifikáciu textových dát, hlavne vďaka svojej jednoduchosti a dobrým výsledkom dokonca aj v porovnaní s náročnejšími metódami. O tejto metóde hovoríme že je „naivná“ kvôli predpokladu nezávisloti medzi výskytom slov v dokumente. To umožnuje znížiť výpočetné náklady na nájdenie pravdepodobnosti, že dokument d_i patrí do určitej triedy c_j pomocou rovnice 4.2 [17]:

$$P\left(c_j \mid d_i\right) = \frac{P\left(c_j\right) P\left(d_i \mid c_j\right)}{P\left(d_i\right)} \tag{4.2}$$

V celkovej sade dokumentov D má každý dokument rovnakú pravdepodobnosť, takže $P(d_i)$ je rovnaké pre akékoľvek i. Keďže ide o konštantu, môžeme ju z predchádzajúcej rovnice vylúčiť. Predpokladajme, že dokument d_i je reprezentovaný vektorom príznakov $(f_1, f_2, \ldots, f_m)$. Prvok $P(d_i \mid c_j)$ vyjadruje, aká je pravdepodobnosť, že dokument, ktorý patrí do triedy c_j obsahuje prvky z vektora príznakov dokumentu d_i. Za predpokladu nezávislosti funkcií môžeme prvok $P(d_i \mid c_j)$ nahradiť za:

$$P(c_j \mid d_i) = P(c_j) \prod_{k=1}^{m} P(f_k \mid c_j) \tag{4.3}$$

Pravdepodobnosť jednotlivých tried môžeme vypočítať pomocou rovnice 4.4:

$$P(c_j) = \frac{n_j}{n} \tag{4.4}$$

kde n_j je počet dokumentov, ktoré patria do triedy c_j a n je počet všetkých dokumentov v celkovej sade D.

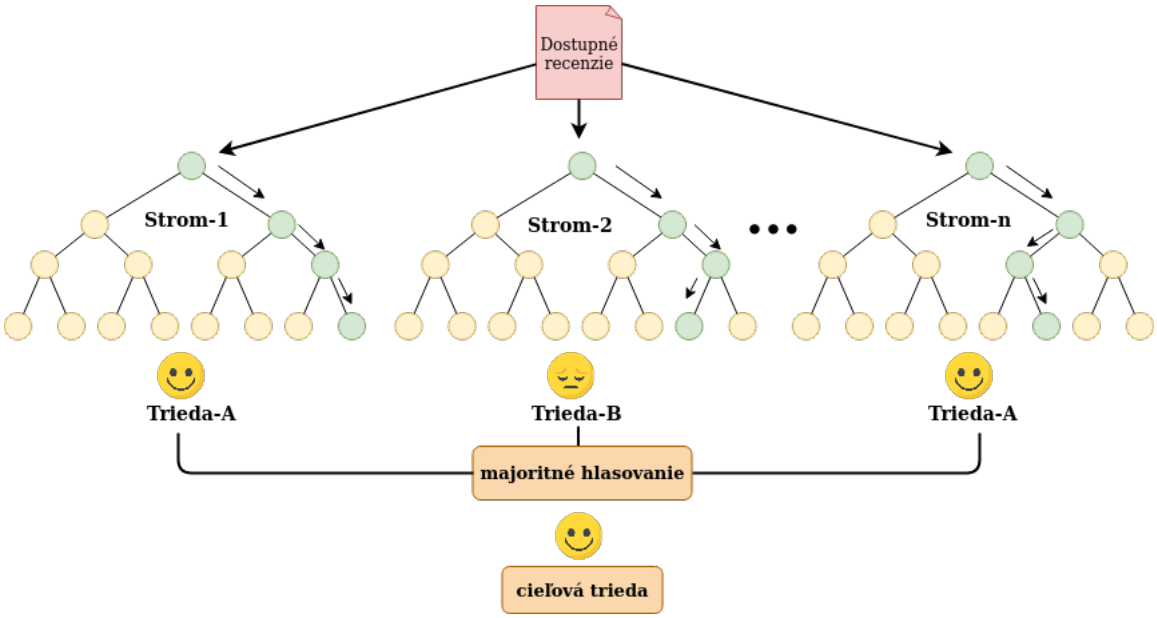
Následne pre každú triedu c_j vyrátame pravdepodobnostnú hodnotu $P(c_j \mid d_i)$. Dokument d_i bude zaradený do cieľovej triedy c_j, ktorá mala maximálnu hodnotu $P(c_j \mid d_i)$.

4.2.3 Random Forest

Random Forest je definovaný ako skupina rozhodovacích stromov, ktorých uzly sú definované v kroku predspracovania. Pomocou algoritmu rozhodovacích stromov sa vygenerujú rozhodovacie stromy. Random Forest teda pozostáva z týchto stromov, ktoré sa používajú na klasifikáciu nového objektu zo vstupného vektora. Každý vytvorený rozhodovací strom sa používa na klasifikáciu. Proces tejto metódy si môžeme bližšie vysvetliť pomocou jednotlivých krokov [21]:

1. Vyberieme K nahodných záznamov z tréningovej sady.
2. Vytvoríme rozhodovací strom z týchto K záznamov.
3. Pred opakovaním krokov 1 a 2 si zvolíme číslo N, toto číslo predstavuje počet stromov, ktoré chceme vytvoriť.
4. Najdite predpovede každého rozhodovacieho stromu a priraďte nový prvok do kategórie, ktorá mala najväčší výskyt.

Ako príklad si môžme uviesť všetky dostupné recenzie, z ktorých odoberieme n počet vzoriek a pre každú vzorku sa vytvorí vlastný rozhodovací strom. Každý rozhodovací strom vygeneruje výstup, ako je znázornené na obrázku 4.2. Finálne zaradenie recenzie sa určí podľa triedy, ktorá sa najčastejšie vyskytovala vo výstupoch jednotlivých stromov. Na obrázku 4.2 môžeme vidieť, že väčšina rozhodovacích stromov určuje recenziu ako pozitívnu, takže recenzia sa zaradí do triedy s pozitívnymi recenziami.

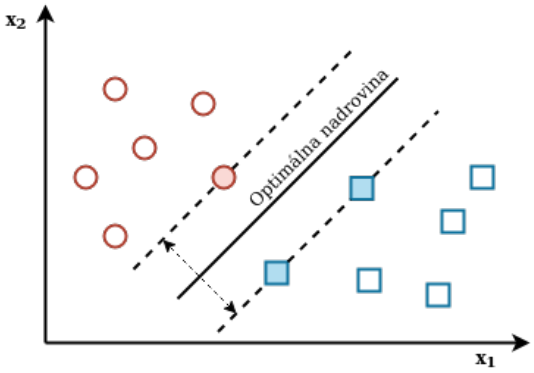


Obr. 4.2: Ukazuje proces zaradenia recenzie do cieľovej triedy, prevzaté a upravné z [19]

4.2.4 Support Vector Machines

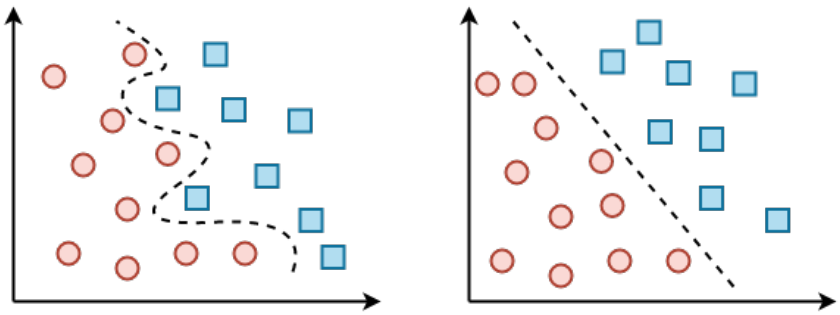
Support Vector Machines (SVM) je algoritmus strojového učenia s učiteľom, ktorý je v základnej podobe určení na binárnu klasifikáciu. Snaží sa nájst optimálnu hranicu, známu

pod názvom nadrovina, medzi rôznymi triedami. V základnej podobe sa táto metóda snaží nájsť čiaru, ktorá maximalizuje oddelenie medzi dvomi triedami v dvojrozmernom priestore, znázornené na obrazku 4.3. Takže vo všeobecnosti je cieľom tejto metódy nájsť nadrovinu, ktorá maximalizuje oddelenie medzi jednotlivými triedami v n rozmernom priestore. Body s minimálnou vzdialenosťou k nadrovine, najbližšie body, sa nazývajú podporné vektory (support vectors). Tieto podporné vektory sú na obrázku 4.3 zvýraznené [6].



Obr. 4.3: Zobrazuje SVM v dvojrozmernom priestore s dvoma triedami, prevzané z [6]

Vo vyšie uvedenom prípade sú vstupné dáta lineárne separovateľne. Problémy klasifikácie v reálnom svete však zvyčajne nemožno vyriešiť lineárnym oddelením. Tento prípad je znázornený na ľavej strane obrázka 4.4. SVM poskytuje riešenie tohto problému pomocou transformácie do iného, potenciálne vysokodimenzionálneho priestoru. Následne pretransformované dáta môžu byť v novom priestore lineárne oddeliteľné, ako možeme vidieť na pravej strane obrázka 4.4 [25].



Obr. 4.4: Zobrazuje princíp transformácie dát do vyššej dimenzie, prevzané z [25]

Viactriedna klasikácia pomocou SVM

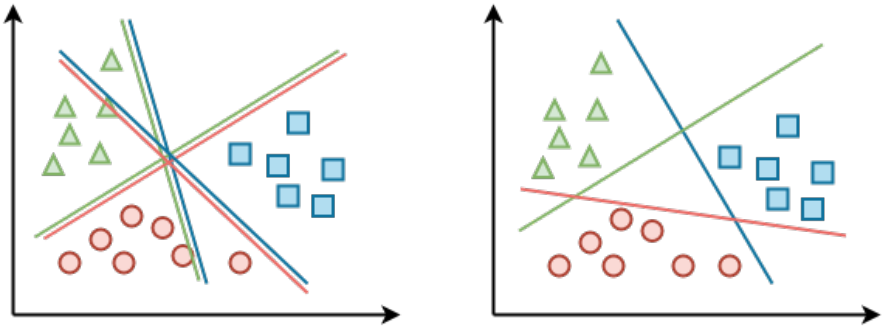
Vo svojom základnom type SVM nepodporuje viactriednu klasifikáciu, podporuje iba binárnu klasifikáciu, čiže rozdelenie vstupných dát do dvoch tried. Pre viactriednu klasifikáciu sa používa princíp rozdelenia problému viacnásobnej klasifikácie na viaceré binárne klasifikačné problémy. Medzi populárne metódy, ktoré sa používaju na viactriednu klasifikáciu pomocou SVM patria [6]:

- **One vs one** táto metóda využíva $\frac{m(m-1)}{2}$ binárných klasifikátorov, kde m predstavuje počet tried. Každý vytvorený klasifikátor sa snaží nájsť nadrovinu, ktorá oddeľuje

každé dve triedy, pričom zanedbáva body ďalších tried. Cieľová trieda sa určí pomocou väčšinového hlasovania.

- **One vs all** v tejto metóde sa snažíme nájst nadrovinu na oddelenie tried, využívame nato m binárnych klasifikátorv, kde m je počet tried. To znamená, že odlišujeme jednu triedu od ostatných. Máme dve skupiny, v tej prvej sú body práve jednej triedy, pričom v tej druhej sú všetky ostatné body. Táto metóda vyžaduje, aby každý model predpovedal skóre v triede. Index triedy s najväčším skóre sa potom použije na predpovedanie triedy.

Na obrázku 4.5 môžeme vidieť aplikáciu týchto dvoch metód pri klasifikácií do troch tried. Na ľavej strane je zobrazená *One vs one* metóda, pri ktorej potrebujeme nájsť nadrovinu na oddelenie každých dvoch tried, pričom zanedbávame body tretej triedy. Napríklad zelenomodrá čiara sa snaží maximalizovať oddelenie iba medzi zelenými a modrými bodmi, vôbec neberie do úvahy červené body. Pravá strana obrázka 4.5 zase ukázuje metódu *One vs all*, pri ktorej potrebujeme nadrovinu na oddelenie jednej triedy a všetkých ostatných. Napríklad modrá čiara sa snaží maximalizovať vzdialenosť medzi modrými bodmi a všetkými ostatnými bodmi naraz.



Obr. 4.5: Zobrazuje porovnanie viactriednej klasifikácie s troma triedami medzi metódou One vs one (na ľavej strane) a One vs all (na pravej strane), prevzané a upravené z [6]

Kapitola 5

Návrh a implementácia

Jednou z úloh tejto bakalárskej práce bolo vytvorenie experimentálnej aplikácie pomocou ktorej bude možné nad vybranou dátovou sadou vykonávať predspracovanie textových dát a následne aplikovať vybranú klasifikačnú metódu nad týmito dátami. Konkrétnejšie sa jedná o klasifikáciu zákazníckych recenzií. V tejto kapitole budú predstavené jednotlivé kroky vývoja aplikácie.

5.1 Špecifikácia požiadaviek

Výsledná aplikácia by mala obsahovať možnosť pridať rôzne dátové sady, ktoré sú vo vhodnej podobe. Tieto dátové sady musia pozostávať z anotovaných textových dát s priradenou cieľovou triedou, aby bolo možné vykonať trénovanie jednotlivých klasifikátorov.

Rovnako by aplikácia mala zahŕňať výber metód predspracovania textových dát, aby užívateľ mohol určiť, ktoré metódy predspracovania chce zahrnúť v rámci svojho experimentu. Takúto predspracovanú dátovu sadu je vhodné uložiť pre budúce používanie, pretože pri rozsiahlých dátových sadách je proces predspracovania textových dát časovo náročný.

Z dôvodu, že každá klasifikačná metóda dosahuje najlepšie výsledky na inom druh dát, je vhodné užívateľovi poskytnúť možnosť výberu klasifikačnéj metódy spolu s nastavením niektorých parametrov. Trénovanie klasifikátorov je väčšinou u rozsiahlejších dátových sád časovo náročné, z toho dôvodu je vhodné vytrénovaný klasifikátor uložiť.

Za účelom porovnania klasifikátorov je potrebné vyhodnotiť úspešnosť vybranej klasifikačnej metódy pri využití metód predspracovania textu, a to z hľadiska presnosti a časovej náročnosti. Ako posledná požiadavka na výslednú aplikáciu je možnosť vyskúšať vytvorený klasifikátor užívateľom prostredníctvom vlastného textu.

5.2 Použité technológie

Výsledná aplikácia bola implementovaná v jazyku Python pomocou knižníc a nástrojov, ktoré budú v tejto sekcii predstavené. Python je interpretovaný programovací jazyk, ktorý podporuje niekoľko programovacích paradigiem vrátane objektovo orientovaného programovania a funkcionálneho programovania. Hlavným dôvodom výberu tohto programovacieho jazyka je skutočnosť, že obsahuje množstvo knižníc pre jednoduchú prácu s rozsiahlymi dátami a možnosť ich vizualizácie, rôzne knižnice pre algoritmy strojového učenia a taktiež podporuje framework *Qt*, ktorý slúži na vytvorenie grafického užívateľského rozhrania.

Pandas[1]

Je voľne dostupná knižnica jazyka Python, ktorá je určená hlavne pre jednoduchú prácu s dátami. Táto knižnica poskytuje rôzne dátové štruktúry a operácie na manipuláciu s dátami, ako napríklad agregovanie alebo transformovanie. Veľkou výhodou je, že umožnuje načítať dáta z rôznych typov súborov.

NLTK[2]

Je skratka pre *Natural Language Toolkit* jedná sa o voľne dostupnú knižnicu jazyka Python, ktorá poskytuje nástroje pre spracovanie prirodzeného jazyka a textu. Poskytuje ľahko použiteľné rozhranie pre viac ako 50 korpusov a lexikálných zdrojov, ako je napríklad WordNet. Túto knižnicu je možné využiť napríklad pri tokenizácií, stematizácií, odstraňovaní stopslov a mnoho ďalších.

Scikit-learn (Sklearn)[3]

Patrí medzi jednu z najužitočnejších knižníc pre algoritmy strojového učenia v jazyku Python. Táto knižnica, ktorá je z veľkej časti napísana v Pythone a postavená na moduloch *NumPy*, *SciPy* a *Matplotlib* je voľne dostupná pod licenciou BSD.

Poskytuje výber učinných nástrojov pre strojové učenie vrátane klasifikácie, regresie, zhlukovania, redukcie rozmerov, predspracovanie dát, rôzne spôsoby rozdelenia dát a validáciu výsledkov.

Matplotlib[4]

Je komplexná knižnica jazyka Python, ktorá slúži na vytváranie statických, animovaných a interaktívnych vizualizácií. Jednou z výhod je, že poskytuje objektovo orientované API prostredníctvom ktorej je možné vkladanie grafov do samotnej aplikácie.

Seaborn[5]

Je vizualizačná knižnica pre grafické vykresľovanie v jazyku Python. Poskytuje rozhranie na vysokej úrovni pre kreslenie atraktívnej a informatívnej štatistickej grafiky. Táto knižnica je postavená nad knižnicou *Matplotlib* a je tiež úzko spätá s dátovými štruktúrami v knižnici *Pandas*.

PyQt5[6]

PyQt je knižnica, ktorá umožnouje používať framework *Qt*. Samotné *Qt* je napísané v jazyku C++, služí pre vývoj aplikacií na všetkých podporovaných platformách vrátane iOS a Android. Pri používaní tohto frameworku v jazyku Pyton môžeme výtvárať aplikácie oveľa rýchlejšie bez toho, aby sme obetovali rýchlosť C++.

[1] https://pandas.pydata.org/
[2] https://www.nltk.org/
[3] https://scikit-learn.org/stable/
[4] https://matplotlib.org/
[5] https://seaborn.pydata.org/
[6] https://pypi.org/project/PyQt5/

Qt Designer

Je nástroj *Qt* na navrhovanie a vytváranie grafikého používateľského rozhrania. Výstupom tohto nástroja je súbor s príponou `.ui`, jedná sa o špecialný XML formát.

5.3 Implementácia

5.3.1 Pridanie dátovej sady

Ako prvé je potrebné, aby si užívateľ zvolil dátovú sadu, ktorá bude následne použitá na trénovanie a testovanie vytvoreného klasifikátoru. Užívateľ ma na výber z troch dostupných dátových sád, ktoré sú bližšie popísane v kapitole 6.

Užívateľ ma taktiež možnosť pridať vlastnú dátovú sadu pomocou vyplnenia formulára. Formulár obsahuje samotný súbor a informácie o dátovej sade ako názov dátovej sady, názov stĺpca pre recenziu a takisto názov stĺpca pre hodnotenie. Počas spracovania týchto údajov ako prvé dochádza ku kontrole, či boli zadané všetky potrebne údaje. Následne dochádza ku kontrole, či sa jedná o podporovaný typ súboru, ako je `csv` a `json`. Taktiež dochádza ku kontrole existencie zadaných stĺpcov pomocou metódy `columns` z knižnice `Pandas`.

Pokiaľ sú všetky údaje správne, vytvorí sa inštancia triedy `Dataset` a následne sa uloží do databázy.

5.3.2 Predspracovanie dát

V ďalšom kroku si užívateľ vyberie, ktoré metódy predspracovania textových dát chce použiť. Následne sa vytvorí inštancia triedy `Preprocessing` a skontroluje sa, či náhodou už neexistujú takto predspracované dáta. Pokiaľ takto predspracované dáte ešte neexistujú, dochádza k spusteniu predspracovania dát.

Na začiatku sa načíta vybraná dátová sada pomocou metódy z knižnice `Pandas`, na základe typu súboru buď `read_csv` alebo `read_json`. Po načítaní dátovej sady dochádza k vykonaniu jednotlivých metód predspracovania. Počas metódy odstránenia šumu dochádza k odstraneniu HTML značiek, interpunkčných znamienok, čísel a napokon k prevedeniu textu na malé písmená. Na odstránenie stopslov bol použitý upravený zoznam stopslov z knižnice `NLTK`, z tohto zoznamu boli odstránené najmä slová so záporom, ako napríklad *not* alebo *don't*, ktoré by mohli ovplyvniť klasifikáciu recenzií. V rámci metód na hľadanie koreňa slov má uživateľ na výber zo stematizačných algoritmov z knižnice `NLTK`, ako je `PorterStemmer` a `SnowballStemmer` a takisto algoritmus `WordNetLematizer` na hľadanie koreňa slova pomocou metódy lematizácie. Na záver sa takto predspracované dáta uložia pomocou dátovej serializácie `pickle`.

5.3.3 Použitie klasifikačnej metódy

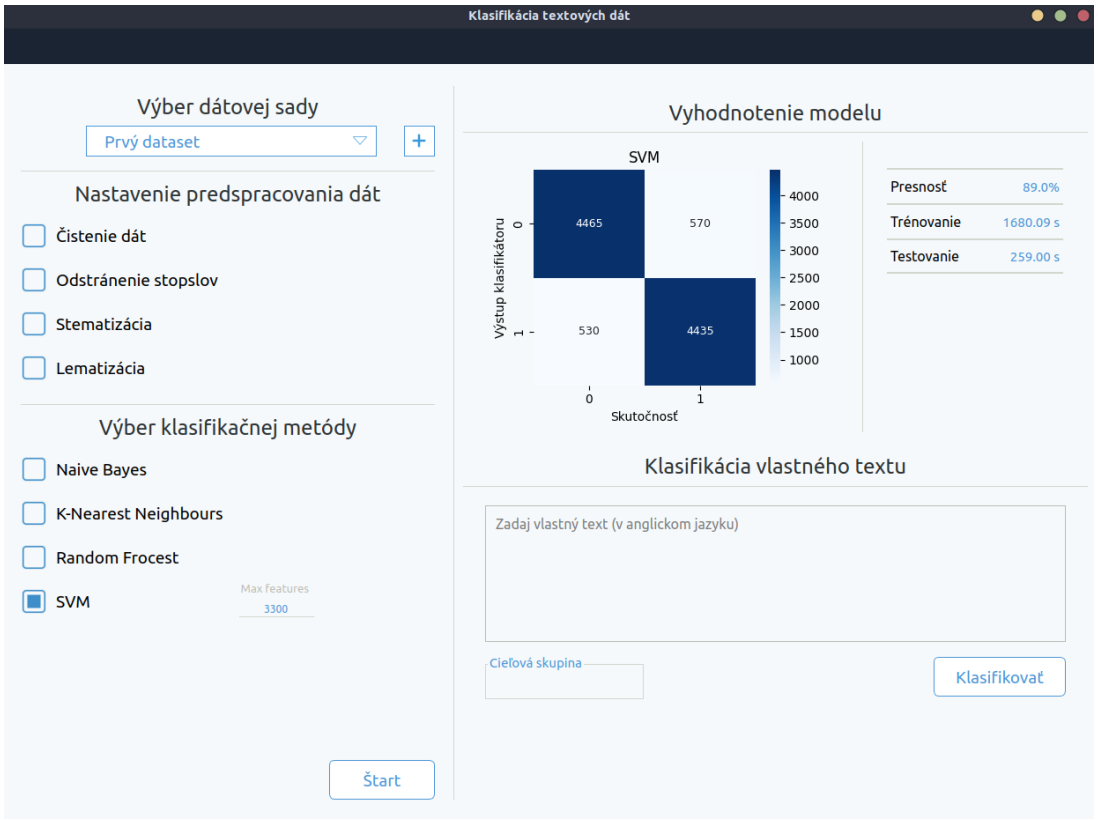
Užívateľ ma na výber zo štyroch klasifikačných metód, a to Naivný Bayesom klasifikátor, KNN, Random Forest a SVM, ktoré boli predstavené v kapitole 4. Po vybraní klasifikačnej metódy užívateľ zadá potrebné parametre k príslušnej klasifikačnej metóde a vytvorí sa inštancia triedy `Classifier`. Ako prvé sa skontroluje, či už neexistuje vytvorený klasifikátor k príslušnej dátovej sade so zadanými parametrami. Ak ešte takýto klasifikátor neexistuje, zaháji sa proces trénovania a testovania klasfikátora.

Pred samotným trénovaním a následným testovaním vybraného klasifikátora sa vytvorí inštancia triedy `CountVectorizer` z knižnice `sklearn` a pomocou metódy `transform_fit`

sa prevedú predspracované dáta do vektorovej podoby. Následne sa rozdelí predspracovaná dátová sada na trénovaciu a testovaciu časť pomocou funkcie `train_test_split` z knižnice `sklearn`. Jednotlivé klasifikátory sú trénované pomocou metód z tried `MultinominalNB`, `KNeighborsClassifier`, `RandomForestClassifier` a `SVC`, ktoré sú implementované v knižnici `sklearn`. Testovanie vytvoreného klasifikátora prebieha na testovacej časti dátovej sady pomocou metódy `predict` z knižnice `sklearn`. Na záver je vyhodnotená presnoť daného klasifikátora prostredníctvom funkcie `accuracy_score`, pre podrobnejšie vyhodnotenie úspešnosti sa používa matica zámen pomocou funckie `confusion_matrix`, obe tieto funkcie pochádzajú z knižnice `sklearn`.

5.3.4 Grafické užívateľské rozhranie

Grafické užívateľské rozhranie bolo vytvorené pomocou predstaveného nástroja *Qt Designer*. Na obrázku 5.1 je zobrazené grafické užívateľské rozhranie. Grafické rozhranie pozostáva zo vstupnej a výstupnej časti. V rámci vstupnej časti používateľ vyberie dátovú sadu, s ktorou chce pracovať, prípadne pomocou tlačidla `plus` pridá vlastnú dátovú sadu. Ďalej je potrebný výber metódy predspracovania textových dát a následne zvolenie klasifikačnej metódy s nastavením jednotlivých parametrov. Po stlačení tlačidla `Štart` užívateľ čaká na vyhodnotenie úspešnosti vybraného klasifikátora, ktoré sa zobrazí na pravej strane, ako môžeme vidieť na obrázku 5.1. Po vytvorení vybraného klasifikátora má užívateľ možnosť ďalšieho vstupu, tým je zadanie vlastnej recenzie na odskúšanie klasifikátora. Následne po zadaní vlastnej recenzie a stlačení tlačidla `Klasifikovať` sa zobrazí cieľova skupina danej recenzie.



Obr. 5.1: Grafické užívateľské rozhranie aplikácie.

Kapitola 6

Experimenty

Nasledujúca kapitola popisuje výsledky experimentov, ktoré boli zhotovené pomocou implementovanej aplikácie nad vybranými dátovými sadami. Hlavným cieľom týcho experimentov je porovnanie klasifikačných metód, ktoré boli spomenuté v kapitole 4 a taktiež zistiť ako vplýva predspracovanie textových dát na celkový výsledok klasifikačných metód.

Na začiatku tejto kapitoly si bližšie predstavíme vybrané dátové sady, na ktorých boli vybrané klasifikačné metódy trénované a následne aj otestované. V nasledujúcej podkapitole 6.2 sa pozrieme na porovnanie vybraných klasifikačných metód a to z hľadiska presnosti výsledkov a taktiež na základe doby strávenej pri trénovaní a testovaní . Ďalej sa v podkapitole 6.3 bližšie zameriame na to, ako môže predspracovanie dát ovplyvniť veľkosť vstupných dáta a výsledky klasifikačných metód. Na záver tejto kapitoly sa pozrieme na porovnanie metód hľadania základu slov.

6.1 Predstavenie dátových sád

V tejto podkapitole si predstavíme jednotlivé dátové sady pomocou, ktorých boli klasifikátory trénované, testované. Tieto dátové sady pozostávajú z textových recenzií, ktoré sú z rôznych oblastí a majú priradenú cieľovú triedu, do ktorej patria, aby bolo možné použiť vybrané klasifikačné metódy.

Filmové recenzie

Prvá dátová sada *Large Movie Review Dataset*[1] obsahuje filmové recenzie z internetovéj databázy IMDb, ktoré sú určené pre binárnu klasifikáciu sentimentu. Táto dátová sada pozostávajúca z filmových recenzií bola vytvorená počas práce [13]. Jedná sa o vyváženú dátovú sadu obsahujúcu 25 000 pozitívnych a 25 000 negatívnych recenzií.

Jednotlivé recenzie sú uložené v dvoch zložkách, prvá obsahuje textové súbory s pozitívnymi recenziami a druhá s negatívnymi recenziami. Pre jednoduchšie pracovanie s týmto datasetom bol vytvorený JSON súbor, ktorý obsahuje všetky recenzie a k ním prislušnú infromáciu, či sa jedná o pozitívnu alebo negatívnu recenziu.

[1] Dátová sada dostupná na: http://ai.stanford.edu/~amaas/data/sentiment/

Recenzie kníh

Táto menšia dátová sada *Multi-Domain Sentiment Dataset*[2] pozostáva z recenzií rôznych produktov na stránke Amazon, ako napríklad recenzie kníh, elektorniky alebo DVD. Pre túto prácu boli vybrané práve recenzie kníh, ktoré pozostávajú z 5 500 vyvážených záznamov zákazníckych recenzií. Na rozdiel od prvej dátovej sady, tieto recenzie nie su iba binárne ale obsahujú hodnotenie pomocou hviezdičiek (1–5), hodnotenia s troma hviezdičkami v tejto dátovej sade chýbajú. To znamená, že sú vhodné pre viactriednu klasifikáciu setimentu so štyrmi cieľovými triedami.

Rovnako ako prvú dátovú sadu, bolo potrebné aj túto dátovú sadu previesť do vhodnejšej formy, aby sa s ňou dalo rýchlejšie pracovať.

Recenzie aplikácií

Posledná použitá dátová sada *Dating Apps Reviews*[3] je výrazne rozsiahlejšia ako predchádzajúce dve a obsahuje až 527 000 recenzií od zákazníkov používajúcich aplikácie určené na hľadanie budúceho partnera. Taktiež táto dátová sada je určená na viactriednu klasifikáciu sentimentu s piatimi cieľovými triedami (1–5 hviezdičiek). Pre túto prácu bolo z rozsiahlej dátovej sady náhodne vybraných 7 000 vzoriek z každej triedy.

Táto dátova sada obsahuje všetky zákaznícke recenzie spolu s cieľovou triedou v jednom súbore *csv*, takže nie je potrebné ju špeciálne spracovávať, ako to bolo v prípade predchádzajúcich dátových sád.

6.2 Porovnanie klasifikačných metód

Jedným zo základných experimentov tejto práce je porovnanie jednotlivých klasifikačných metód, ktoré boli vysvetlené v teoretickej časti. Validácia výsledkov prebiehala pomocou *hold–out* metódy. Táto metóda je založená na rozdelení vstupných dát na trénovaciu a testovaciu sadu. Jednotlivé dátové sady, ktoré boli použité v tejto práci sú rozdelené v pomere 80/20, kde 80 % dát je použitých na trénovanie vybraného klasifikačného modelu a zvyšnych 20 % je určených na testovanie vytvoreného klasifikačného modelu.

U jednotlivých klasifikátorov boli parametre nastavené na hodnoty, s ktorými daný klasifikátor vykazoval najlepšie výsledky pre vybranú dátovú sadu (viď Príloha A). Najdenie najvhodnejšieho nastavenia parametrov prebiehalo opakovaným trénovaním a testovaním modelov s rozdielnym nastavením parametrov. Počas tohto experimentu neboli na vstupné dáta použité žiadne metódy predspracovania textových dát.

6.2.1 Presnosť

Prvá časť tohto experimentu je zameraná na porovnanie vybraných klasifikačných metód z hľadiska presnosti, to znamená do akej miery sú vytvorené klasifikátory schopné zaradiť vstupné dáta do cieľovej triedy.

Výsledky z testovania presnoti jednotlivých klasifikačných metód, ktoré boli aplikované na vybrané dátové sady sú zobrazné v tabuľke 6.1. U prvej dátovej sady, ktorá bola určená na binárnu klasifikáciu zákaznických recenzií filmov, vybrané klasifikačné metódy dosahovali najlepšie výsledky. Z tabuľky 6.1 je vidieť, že metóda Support Vector Machines (SVM)

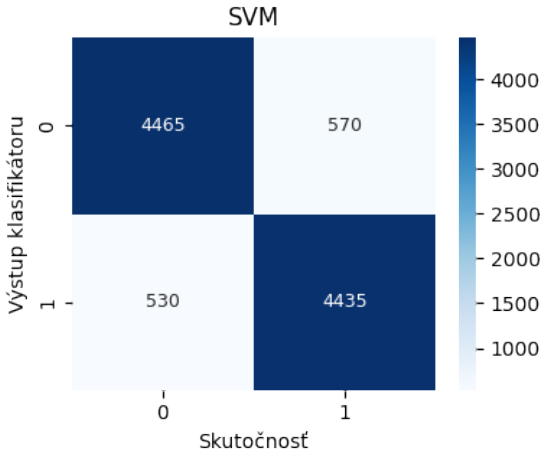
[2]Dátová sada dostupná na: https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[3]Dátová sada dostupná na: https://www.kaggle.com/datasets/sidharthkriplani/datingappreviews

	Naive Bayes	KNN	Random Forest	SVM
Filmové recenzie (1. dátová sada)	85.7 %	74.9 %	85.6 %	**89.0 %**
Recenzie kníh (2. dátová sada)	**56.7 %**	45.2 %	52.1 %	55.1 %
Recenzie aplikací (3. dátová sada)	46.5 %	40.8 %	45.9 %	**47.6 %**

Tabuľka 6.1: Klasifikačné metódy a ich presnosť pri jednotlivých dátových sadách

dosiahla najlepší výsledok u tejto prvej dátovej sady a to 89.0 %. Maticu zámen tohto klasifikátora ktorý dosiahol najlepšie výsledky môžeme vidieť na obrázku 6.1. Naivný Bayesov klasifikátor dosiahol druhé najlepšie výsledky s úspešnosťou 85.7 %, čím zaostáva približne o 3 % za metódou SVM. Skoro podobné výsledky dosiahla metóda Random Forest s úspešnosťou 85.6 % čím zaostávala iba o desatinu percenta za druhou zmienenou metódou. Metóda KNN dosiahla výrazne horšie výsledky ako predchádzajúce metódy u tejto dátovej sady a to 74.9 % úspešnosť, čo predstavuje rozdiel až necelých 15 % oproti metóde SVM.

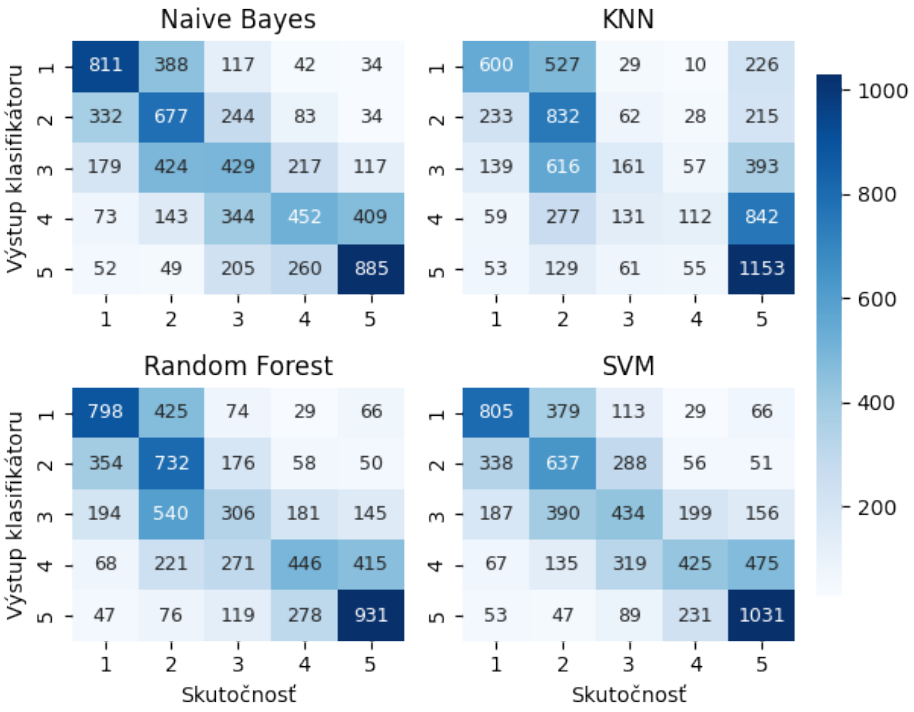


Obr. 6.1: Matica zámen klasifikátora SVM, ktorý dosiahol najlepšie výsledky a to 89.0 % pri prvnej dátovej sade

Druhá dátová sada určená na viactriednu klasifikáciu so štyrmi triedami dosiahla výrazne odlišné výsledky, čo môže byť spôsobené dvojnásobným počtom tried. Rovnako aj u tejto dátovej sady dosiahli najlepšie výsledky Naivný Bayesov klasifikátor a metóda SVM avšak v tomto prípade Naivný Bayesov klasifikátor dosiahol lepšie výsledky s úspešnosťou 56.7 % oproti SVM, ktoré dosiahlo 55.1 %. Metóda Random Forest s úspešnosťou 52.1 % zaostávala za metódou SVM o 3 %. Aj u tejto dátovej sady dosiahla metóda KNN s úspešnosťou 45.2 % najhoršie výsledky.

U vybraných klasifikačných metód posledná dátová sada dosiahla najhoršie výsledky, toto môže byť spôsobené práve tým, že sa jednalo o viactriednu klasifikáciu, kde išlo o klasifikáciu až do piatich tried. U téjto dátovej sady metóda SVM znova dosiahla najlepšie výsledky a to 47.6 % úspešnosť. Ostatné metódy zaostávaju iba o približne 2 % s výnimkou metódy KNN, ktorá aj u tejto dátovej sady dosiahla najhoršiu úšpešnosť 40.8 %.

U téjto dátovej sady sa bližšie pozrieme na výsledky vybraných metód z dôvodu že ich výsledky nedosiahli ani nadpolovičnú úspešnosť. Na obrázku 6.2 sú zobrazené matice zámen klasifikátorov, ktoré boli použité v tejto práci. Je vidieť, že jednotlivé klasifikátory sa vo väčšine prípadov trafili do správnej triedy alebo sa pomýlili iba v rozmedzi jednej tiredy. Výnimkou je však metóda KNN, ktorá sa pre túto dátovú sadu ukázala ako menej vhodná a nedokázala klasifikovať triedu s troma a štyrmi hviezdičkami v takej dobrej miere ako to dokázali ostatné metódy. Taktiež môžeme vidieť pri jednotlivých maticách zámen, že v niektorých prípadoch klasifikátory zaradili recenziu do úplne opačnej triedy, ako napríklad namiesto triedy, ktorá obsahuje recenzie s jednou hviezdičkou, zaradili túto recenziu práve do triedy s piatimi hviezdičkami. Toto môže byť spôsobené tým, že z textu recenzie nie je dostatočne jasné, do ktorej triedy náleží alebo napríklad kvôli tomu, že zákazníci často používajú v recenziách sarkazmus.



Obr. 6.2: Matice zámen jednotlivých klasifikátorov testovaných na poslednej dátovej sade so zákaznickými recenziami na aplikácie

6.2.2 Časová náročnosť

Druhá časť tohto experimentu sa zaoberá porovnaním vybraných klasifikačných metód z hľadiska časovej náročnosti na trénovanie a testovanie klasifikátoru. Počas tohto experimentu bol nad dátovými sadami každý klasifikátor 3 krát nezávisle trénovaní a testovaní. Tabuľka 6.2 popisuje výsledky priemerných hodnôt týchto časov z jednotlivých behov. Tieto výsledné hodnoty predstavujú iba orientačnú časovú náročnosť.

Na základe výsledkov z tohto experimentu môžeme povedať, že u zvolených dátových sád mala metóda KNN v rámci trénovanie najnižšiu časovú náročnosť, zatiaľ čo v rámci testovania sa umiestinala až na treťom mieste. Najlepšiu časovu náročnosť testovania dosiahol Naivný Bayesov klasifikátor.

	Trénovanie				Testovanie			
	Naive Bayes	KNN	Random Forest	SVM	Naive Bayes	KNN	Random Forest	SVM
Filmové recenzie (1. dátová sada)	0.38	0.09	1183.74	1680.09	0.12	64.69	8.36	259.00
Recenzie kníh (2. dátová sada)	0.04	0.01	53.35	67.25	0.02	1.01	0.57	14.01
Recenzie aplikacií (3. dátová sada)	0.21	0.05	786.84	2004.39	0.09	32.38	7.56	422.76

Tabuľka 6.2: Orientačná časová náročnosť jednotlivých klasifikačných metód v sekundách

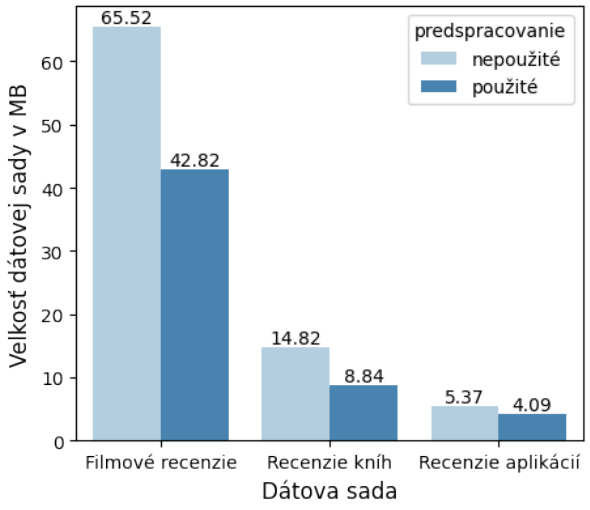
Z pohľadu časovej náročnosti na trénovanie a testovanie klasifikátoru dosiahla metóda SVM najhoršie výsledky. Takže na základe tohto experimentu môžeme povedať, že ak chceme dosiahnuť kvalitné výsledky čo u väčšiny vybraných dátových sád táto metóda SVM dosahuje (viz Tabuľka 6.1) musíme rátať s výrazne väčšiou časovou náročnosťou na trénovanie a testovanie. Pokiaľ by bol pre nás dôležitá rýchlosť klasifikácie a taktiež relatívne dobré výsledky je vhodnešie použiť Naivný Bayesov klasifikátor.

6.3 Vplyv predspracovania

Ďalší experiment sa zaoberá tým, aký vplyv majú jednotlivé motódy predspracovania, a to konkretnejšie na veľkosť vstupných dát a na presnoť klasifikačných metód. V tejto práci sú implementované metódy predspracovania, ako je odstránenie stopslov, hľadanie základu slov a čistenie dát, ktoré zahŕňa odstránenie šumu a prevod textu na malé písmená.

6.3.1 Veľkosť vstupných dát

V prvej časti tohto experimentu sa pozrieme na to, ako dokážu metódy predspracovania ovplyvniť veľkosť vybraných dátových sád. Počas tejto prvej časti experimentu bola použitá metóda čistenia dát, odstránenie stopslov a na hľadanie koreňa slova bola použitá lematizácia.



Obr. 6.3: Graf vplyvu predspracovania na veľkosť dátovej sady

Na obrázku 6.3 je vidieť, že predspracovanie výrazne znižuje veľkosť dátovej sady, hlavne u prvej dátovej sady, ktorá je najrozsiahlejšia. Takáto výrazná redukcia vstupných dát môže urýchliť celkový proces klasifikácie.

6.3.2 Presnosť klasifikáčných metód

Ďalšia časť tohto experimentu sa zaoberá vplyvom predspracovania na výslednú presnosť vybraných klasifikačných metód u prvej a druhej dátovej sady. Počas tohto experimentu boli postupne pozorované implementované metódy predspracovania. Ako prvá bola skúmaná metóda čistenia dát, ktorá pozostávala z odstránenia šumu a prevodu textu na malé písmená. Ďalšou pozorovanou metódou bolo odstránenie stopslov pomocou upraveného zoznamu stopslov. Následne bola skúmaná metóda hľadania slovného základu prostredníctvom lematizácie. Napokon boli odskúšané rôzne kombinácie tychto metód.

Tabuľka 6.3 zobrazuje do akej miery vplývajú jednotlivé metódy predspracovania textových dát u prvej dátovej sady na celkovú úspešnosť vybraných klasifikačných metód vzhľadom na ich úspešnosť bez využitia metód predspracovania.

	Naive Bayes	KNN	Random Forest	SVM
bez predspracovania	85.7 %	74.9 %	85.6 %	89.0 %
čistenie dát	85.7 %	**76.1 %**	85.5 %	89.1 %
odstránenie stopslov	85.8 %	**76.7 %**	85.6 %	88.8 %
lematizácia	85.5 %	**75.4 %**	85.5 %	89.0 %

Tabuľka 6.3: Vplyv predspracovania na presnosť vybraných klasifikátorov u prvej dátovej sady, ktorá obsahuje filmové recenzie

Na základe výsledkov experimentu vykonaného na prvej dátovej sade, ktoré sú zaznamenané v tabuľke 6.3 môžeme povedať, že použitie metód predspracovanie, ako je čistenie dát, odstránenie stopslov a lematizácia nemalo skoro žiadny vplyv na presnosť jednotlivých klasifikátorov, dosahujú skoro totožné výsledky. Výnimkou je však metóda KNN, u ktoréj použitie metód predspracovania malo pozitívny vplyv, a to pri všetkých použitých metódach.

Druhá dátová sada po použití metód predspracovania textových dát dosiahla zlepšenie výsledkov u viacerých vybraných klasifikačných metód (viď tabuľka 6.4). Čistenie dát,

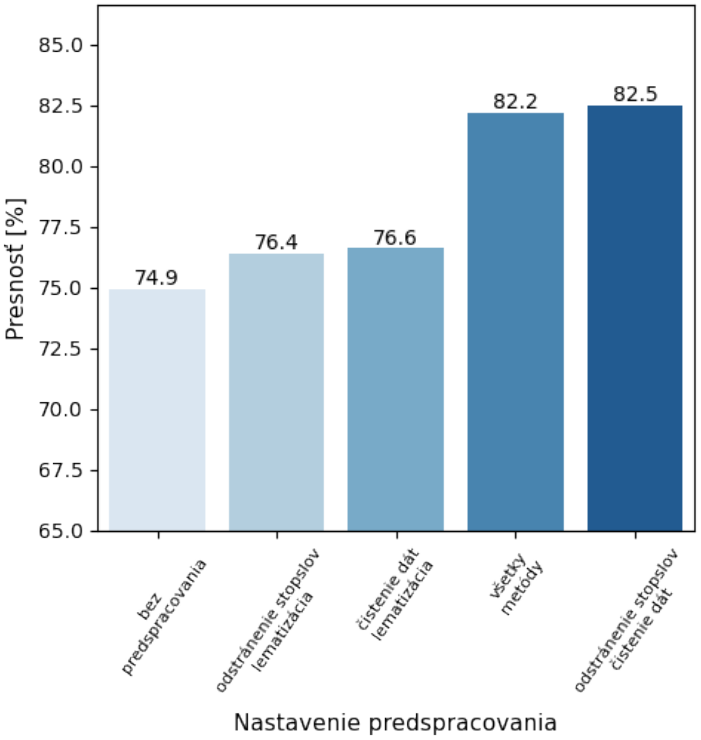
	Naive Bayes	KNN	Random Forest	SVM
bez predspracovania	56.7 %	45.2 %	52.1 %	55.1 %
čistenie dát	56.8 %	44.4 %	**54.3 %**	55.8 %
odstránenie stopslov	55.9 %	**51.0 %**	53.2 %	55.9 %
lematizácia	**57.0 %**	44.1 %	54.0 %	**56.1 %**

Tabuľka 6.4: Vplyv predspracovania na presnosť vybraných klasifikátorov u druhej dátovej sady, ktorá obsahuje recenzie kníh

ktoré zahŕňa odstránenie šumu a prevod textu na malé písmená malo pozitívny vplyv na presnosť klasifikačných metód s výnimkou KNN metódy, ktorá zaznamenala mierný pokles presnosti. U metódy Random Forest sa presnosť zvýšila najviac, a to o 2.2 %. Odstránenie stopslov dosiahlo zlepšenie presnoti u všetkých vybraných klasifikačných metód, avšak u

metódy KNN išlo o výrazne zlepšenie a to až 5.8 %. Lematizácia mala pozitívny vplyv najmä na klasifikačné metódy Random Forest a SVM, naopak na klasifikačnú metódu KNN nebola účinná. U tejto dátovej sady kombinácia metód predspracovania nepriniesla zlepšenie výsledkov.

Metódy predspracovania mali pri prvej dátovej sade pozitívny vplyv na klasifikačnú metódu KNN. Na základe toho boli taktiež vyskúšané rôzne kombinácie metód predspracovania. Výsledky sú zobrazené v grafe 6.4. Z tohto grafu je vidieť, že zvolené kombinácie metód predspracovania výrazne zvýšili úspešnosť tejto metódy. Ako najlepšia sa ukázala kombinácia čistenia dát a odstránenia stopslov s úspešnosťou 82.5 %, čo predstavuje až 7.6-percentné zlepšienie výsledku.



Obr. 6.4: Vplyv predspracovania na presnosť metódy KNN u prvej dátovej sady

Pomocou tohto experimentu, ktorý bol vykonaný na prvej a druhej dátovej sade môžeme povedať, že metódy predspracovania nemajú výrazný vplyv na presnosť Naivného Bayesovho klasifikátora. U ostatných klasifikačných metód vo väčšine prípadov predspracovanie textových dát prinieslo zlepšenie presnosti klasifikácie.

6.4 Vplyv výberu metódy na hľadanie základu slov

Posledný experiment sa zaoberá porovnaním metód na hľadanie základu slov, ktoré boli v tejto práci implementované. Na tento experiment boli vybrané klasifikačné metódy ako KNN, Random Forest a SVM. Naivný Bayesov klasifikátor bol počas tohto experimentu vynechaný, pretože nevykazoval výrazné zlepšenie u vybraných dátových sád po použití metódy na hľadanie koreňa slov. V rámci metódy stematizácie boli pozorované dva algortimy

a to *PorterStemmer* a *SnowballStemmer*, u metódy lematizácie bol použitý *WordNetLemmatizer*.

Tabuľka 6.5 zobrazuje vplyv výberu metódy hľadania koreňa slova na celkovú presnosť klasifikácie vykonanej na prvej dátovej sade. Na prvý pohľad je zrejmé, že výber metódy hľadania koreňa slova nemá výrazný vplyv na presnoť klasifikácie. Popritom môžeme zhodnotiť, že pri klasifikačnej metóde KNN, algoritmy stematizácie dosahovali o niečo lepšie výsledky, než u klasifikačných metód Random Forest a SVM.

	KNN	Random Forest	SVM
lematizácia (WordNetLemmatizer)	75.4 %	85.5 %	89.0 %
stematizácia (PorterStemmer)	76.2 %	85.5 %	89.1 %
stematizácia (SnowballStemmer)	76.3 %	85.4 %	88.8 %

Tabuľka 6.5: Vplyv výberu metódy hľadania základu slov u prvej dátovej sady (Filmové recenzie)

Výsledky experimentu vykonaného na druhej dátovej sade, ktoré sú zaznamená v tabuľke 6.6 dosiahli podobné výsledky ako u prvej dátovej sady. Navyše sa u tejto dátovej sady ukázalo, že presnosť klasifikačných metód nie je výrazne ovplyvnená výberom metódy hľadania koreňa slov.

	KNN	Random Forest	SVM
lematizácia (WordNetLemmatizer)	44.1 %	54.0 %	56.1 %
stematizácia (PorterStemmer)	44.1 %	53.0 %	57.0 %
stematizácia (SnowballStemmer)	44.4 %	53.9 %	57.0 %

Tabuľka 6.6: Vplyv výberu metódy hľadania základu slov u druhej dátovej sady (Recenzie kníh)

V rámci tohto experimentu u poslednej dátovej sady neprinieslo pozorovanie vplyvu výberu metódy hľadania základu slov žiadne výrazne zmeny, preto nie je bližšie spomenuté.

Na základe výsledkov tohto experimentu môžeme povedať, že výber metódy hľadania základu slov nemá výrazný vplyv na presnoť klasifikácie u jednotlivých klasifikačných metód. Obvykle sa rozdiel presnoti klasifikácie pohyboval v rámci jedného percenta.

Kapitola 7

Záver

Cieľom tejto bakalárskej práce bolo oboznámiť sa s problematikou dolovania textových dát, vrátane predspracovania týchto dát. Následne na vybranej dolovacej úlohe ukázať postup počas dolovania z textových dát. Pre túto prácu bola vybraná klasifikácia textových dát, konkrétne sa jednalo o zákaznícke recenzie.

Prvá časť práce sa zaoberá teoretickým vstupom do problematiky získavania znalostí z dát vo všeobecnosti. V tejto časti práce je uvedené aj to, z akých krokov pozostáva proces získavania znalostí, aké typy zdrojov a dolovacích úloh poznáme. V ďalšej časti je predstavené získavanie znalostí z textových dát, vrátane metód predspracovania textových dát a taktiež spôsob reprezentácie textových dát. Záver teoretickej časti je venovaný klasifikácii textových dát a jednotlivým klasifikačným metódam, ako je KNN, Naivný Bayesov klasifikátor, Random Forest a SVM, ktoré boli pri tejto práci použité.

Druhá časť tejto práce sa venuje návrhu a implementácii aplikácie, ktorá je využítá pri experimentoch. Následne sú predstavné dátové sady, na ktorých boli jednotlivé experimenty vykonávané. Experimenty boli rozdelené na dve časti, prvá časť je určená na porovnanie vybraných klasifikačných metód z hľadiska presnosti a časovej náročnosti. Druhá časť sa zaoberá vplyvom metód predspracovania na presnosť klasifikácie.

Túto prácu by bolo možné v budúcnosti rozšíriť o nové klasifikačné metódy, ktoré sú založené na neurónových sieťach. Ďalším rozšírením môže byť pridanie viacerých dolovacích úloh, ako napríklad zhlukovanie textových dát.

Literatúra

[1] Aggarwal, C. C. *Data Mining: The Textbook*. 1. vyd. Springer, 2015. ISBN 978-3-319-14141-1.

[2] Agrawal, R. *Must Known Techniques for text preprocessing in NLP* [online]. Jún 2021 [cit. 2022-04-11]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/.

[3] Balodi, T. *What is Stemming and Lemmatization in NLP?* [online]. Júl 2020 [cit. 2022-04-13]. Dostupné z: https://www.analyticssteps.com/blogs/what-stemming-and-lemmatization-nlp.

[4] Dunham, M. H. *Data Mining: Introductory and Advanced topics*. 1. vyd. Pearson Education, 2006. ISBN 0-13-088892-3.

[5] Feldman, R. a Sanger, J. *The Text Mining Handbook*. 1. vyd. Cambridge University Press, 2007. ISBN 978-0-521-83657-9.

[6] Goyal, C. *Multiclass Classification Using SVM* [online]. Máj 2021 [cit. 2022-04-23]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/05/multiclass-classification-using-svm/.

[7] Goyal, C. *Part 5: Step by Step Guide to Master NLP – Word Embedding and Text Vectorization* [online]. Jún 2021 [cit. 2022-04-20]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches/.

[8] Haddi, E., Liu, X. a Shi, Y. The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*. 2013, zv. 17, s. 26–32. ISSN 1877-0509.

[9] Han, J., Kamber, M. a Pei, J. *Data Mining: Concepts and Techniques*. 3. vyd. Morgan Kaufmann, 2012. ISBN 978-0-12-381479-1.

[10] Jivani, A. G. et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl.* 2011, zv. 2, č. 6, s. 1930–1938.

[11] Jo, T. *Text Mining: Concepts, Implementation,and Big Data Challenge*. 1. vyd. Springer, 2019. ISBN 978-3-319-91814-3.

[12] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. et al. Text classification algorithms: A survey. *Information*. Multidisciplinary Digital Publishing Institute. 2019, zv. 10, č. 4, s. 150.

[13] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. et al. Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, s. 142–150 [cit. 2022-04-23]. Dostupné z: http://www.aclweb.org/anthology/P11-1015.

[14] Mayo, M. *A General Approach to Preprocessing Text Data* [online]. December 2017 [cit. 2022-04-11]. Dostupné z: https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html.

[15] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. et al. Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*. ACM New York, NY, USA. 2021, zv. 54, č. 3, s. 1–40.

[16] MonkeyLearn. *Text Classification: What it is And Why it Matters* [online]. 2022 [cit. 2022-04-19]. Dostupné z: https://monkeylearn.com/text-classification/.

[17] Montejo Ráez, A. *Automatic Text Categorization of Documents in the High Energy Physics Domain*. Dizertačná práca. Dostupné z: http://cds.cern.ch/record/932677.

[18] Paralič, J., Furdík, K., Tutoky, G., Bednár, P., Sarnovský, M. et al. *Dolovanie znalostí z textov*. 1. vyd. Equilibria, s.r.o., 2010. ISBN 978-80-89284-62-7.

[19] R, S. E. *Understanding Random Forest* [online]. Jún 2021 [cit. 2022-04-21]. Dostupné z: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

[20] Rai, A. *What is Text Mining: Techniques and Applications* [online]. Jún 2019 [cit. 2022-04-09]. Dostupné z: https://www.upgrad.com/blog/what-is-text-mining-techniques-and-applications/.

[21] Shah, K., Patel, H., Sanghvi, D. a Shah, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*. Springer. 2020, zv. 5, č. 1, s. 1–16.

[22] Stein, R. A., Jaques, P. A. a Valiati, J. F. An analysis of hierarchical text classification using word embeddings. *Information Sciences*. 2019, zv. 471, s. 216–232. ISSN 0020-0255.

[23] Team, E. *10 Text Mining Examples* [online]. Apríl 2020 [cit. 2022-04-09]. Dostupné z: https://www.expert.ai/blog/10-text-mining-examples/.

[24] Wonderflow. *How Consumer Insights Teams Can Leverage Text Mining Techniques* [online]. Júl 2019 [cit. 2022-04-18]. Dostupné z: https://www.wonderflow.ai/blog/how-consumer-insights-teams-can-leverage-text-mining-techniques.

[25] Xun, W., Yu, W., Champion, A., Fu, X. a Xuan, D. Detecting Worms via Mining Dynamic Program Execution. *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm*. Október 2007, s. 412 – 421.

[26] Zendulka, J., Bartík, V., Lukáš, R. a Rudolfová, I. *Získávání znalostí z databází, Studijní opora*. 2006.

[27] Žižka, J., Dařena, F. a Svoboda, A. *Text mining with machine learning: principles and techniques.* 1. vyd. CRC Press, 2019. ISBN 978-1-138-60182-6.

Príloha A

Nastavenie klasifikačných metód

Tabuľka A.1 zobrazuje nastavenie jednotlivých metód, ktoré boli použité pri experimentoch v tejto bakalárskej práci.

	Naive Bayes	**KNN**		**Random Forest**		**SVM**
	Max. počet príznakov	Max. počet príznakov	Počet susedov	Max. počet príznakov	Počet stromov	Max. počet príznakov
Filmové recenzie (1. dátová sada)	4000	4000	1400	3300	1200	3300
Recenzie kníh (2. dátová sada)	4300	5300	500	5300	800	5200
Recenzie aplikacií (3. dátová sada)	4600	4600	2900	4300	900	4500

Tabuľka A.1: Nastavenie jednotlivých metód, počas experimentov

Príloha B

Obsah priloženého pamäťového média

Priložené pamäťové médium obsahuje nasledujúce súbory a adresáre:

- `src.zip` – adresár obsahujúci zdrojové súbory k aplikácii
- `xhomol27.pdf` – text bakalárskej práce
- `xhomol27.zip` – adresár obsahujúci zdrojové súbory k textovej časti bakalárskej práce
- `README.md` – súbor s návodom spustenia aplikácie