Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Diplomová práce

Analýza sentimentu zpravodajských textů z prostředí zemědělství

Bc. Vojtěch Mikeš

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Vojtěch Mikeš

Informatika

Název práce

Analýza sentimentu zpravodajských textů z prostředí zemědělství

Název anglicky

Sentiment analysis of agricultural news texts

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku analýzy sentimentu. Hlavním cíle práce je analyzovat sentiment textů z prostředí zemědělství. Dílčí cíle práce jsou: Analyzovat vhodné metody pro realizaci experimentu, Připravit data pro experiment, Realizovat experiment, Formulovat závěry z experimentu.

Metodika

Teoretická část diplomové práce bude založena na analýze literárních zdrojů.

V praktické části bude provedena analýza sentimentu textů získaných z webového portálu agris.cz. V první kroku bude vybrána vhodná metoda pro zpracování. Následně proběhne příprava dat a definování vhodné oblasti zájmu pro následující analýzy. V dalším kroku bude provedena analýza sentimentu a formulovány závěry.

Doporučený rozsah práce

60–80 stran

Klíčová slova

sentiment, analýza, agris, zemědělství, texty, novinky, umělá inteligence, ai

---

Doporučené zdroje informací

Bayesian Analysis in Natural Language Processing : Second Edition. 2019. ISBN:9781681735283
Linguistics for the Age of AI. 2021. ISBN:9780262363136
Machine Learning Algorithms : Popular Algorithms for Data Science and Machine Learning, 2nd Edition. 2018. ISBN:9781789347999

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Michal Stočes, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 27. 9. 2022

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 21. 02. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Analýza sentimentu zpravodajských textů z prostředí zemědělství" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne: 21.3.2023 _____

Poděkování

Rád bych touto cestou poděkoval Ing. Michalovi Stočesovi, PhD. za vedení mé práce a za nesmírně užitečné rady, které mi ochotně poskytoval v průběhu zpracovávání mé diplomové práce. Dále bych chtěl poděkovat svým rodičům a ostatním členům rodiny za jejich podporu za dobu studia. V neposlední řadě bych rád poděkoval všem svým přátelům, kteří mi věřili a stáli při mně.

Analýza sentimentu zpravodajských ze zemědělství

Abstrakt

Práce se zabývá analýzou sentimentu zpravodajských textů ze zemědělství. Tato data jsou získána z portálu Agris.cz (ČZU a MZČR, 2022). Analýza sentimentu se zabývá zpracováním surových dat z webového portálu do takové formy, která je vhodná pro vypracování analýzy sentimentu poskytnutých dat. Jedná se o očištění dat od nežádoucích částí, tokenizaci a vektorizaci dat. V této části jsou řešeny všechny úlohy a vyskytující se problémy týkající se datového zpracování přirozeného jazyka počítačem. Dále se práce zabývá výběrem, vyhodnocením kvality a trénováním modelu, který je následně použit k samotnému vyhodnocení sentimentu. V poslední části práce dochází k interpretaci výsledků analýzy sentimentu a testování ovlivňování sentimentu externími vlivy. Tato analýza je uvedena v diskusi práce, protože není součástí cílů práce.

Klíčová slova: sentiment, analýza, agris, zemědělství, texty, novinky, umělá inteligence, ai

Sentiment analysis of agricultural news texts

Abstract

The thesis deals with sentiment analysis of news texts from agriculture. This data is obtained from the Agris.cz portal (ČZU a MZČR, 2022). Sentiment analysis deals with processing the raw data from the web portal into a form that is suitable for developing a sentiment analysis of the provided data. It involves cleaning the data from unwanted parts, tokenization, and vectorization of data. In this section, all the tasks and occurring problems related to natural language data processing by computer are addressed. Further, the paper deals with the selection, quality evaluation and training of the model which is then used for actual sentiment evaluation. The last part of the paper concludes by interpreting the results of the sentiment analysis and testing of sentiment being influenced by external influences. This analysis is presented in the discussion of the thesis as it is not part of the objectives of the thesis.

Keywords: analysis, agris, agriculture, sentiment, texts, news, artificial intelligence, ai

Obsah

1 Úvod

Lidským jazykem, nezáleží zde na konkrétní řeči, se dá vyjádřit mnoho informací, které mohou mít vysokou cenu pro různá odvětví různých průmyslů. Avšak zpracovávání těchto dat není zdaleka tak jednoduché, jak se může zdát, protože vyžaduje vysoké úsilí člověka a čas.

Je tedy logické že se zpracovávání lidského jazyka lidé pokouší přesunout ze svých beder na bedra algoritmů, které by jim pomohli tyto problémy řešit a automatizovat. O efektivní řešení NLP problémů se snažíme od dob, co byli vynalezeny první výpočetní technika. Jedním z těchto problémů je analýzy sentimentu, kterou se tato práce zaobírá.

Jedno z prvotních úskalí, které je potřeba vyřešit je zpracování dat, dokumentů, které budou sloužit jako základ pro vyhodnocování sentimentu. Tyto data se mohou skládat z celých knih, článků, odstavců až po jednotlivé věty a nakonec slova. Dokumenty je třeba zpracovat takovým způsobem, který je vhodný pro použití v jednom z možných klasifikačních algoritmů sentimentu, které se následně starají o ohodnocení textu. Úprava tradičně probíhá pomocí prověřeného matematického aparátu, který umožňuje pozměnit text z podoby, která je srozumitelná člověku na podobu, které rozumí klasifikační algoritmus.

Existuje velká množina algoritmů a statistických metod, jak tyto problémy řešit. Některé jsou na výpočet jednodušší a hodí se na menší dokumenty, jiné, složitější jsou dobré pro hledání vztahů mezi jednotlivými částmi dokumentů nebo dokonce mezi dokumenty samotnými. Výběr těchto algoritmů může záviset na několika faktorech jako je již zmíněná výpočetní složitost nebo způsob implementace až po přesnost algoritmu, která ovšem vždy bude suboptimální.

Tato práce se v první části zaobírá představením některých z nic a jejich aplikací na reálné problémy a návrh jejich řešení s demonstrací užitečnosti představených metod. V druhé části se práce zaobírá aplikací představených metod na reálných datech, a to na datech ze zemědělského portálu agri.cz.

Analyzovanými daty jsou souhrny článků od roku 2004 po rok 2022 v anglickém jazyce. Jedná se o data z různých zemědělských a potravinářství.

2 Teoretické východisko práce

První zárodky algoritmů a pokusů pro zpracovávání lidského jazyka se začali objevovat už na začátku padesátých let dvacátého století. Hlavní motiv byl zprvu vytvoření počítačem ovládaných překladačů, které by byli schopné překládat celé věty. Tyto pokusy se do jisté míry dařilo realizovat, avšak výkonnost nebyla jakkoli závratná (Nadkarni, a další, 2011).

Dalším požadavkem, který se začal čím dál tím častěji objevovat okolo roku 1980 byla možnost rozpoznávání textů dle jejich slovních významů a na základě těchto významů schopnost takové texty kategorizovat. Byla tedy vyžadována analýza sentimentu, kterou by mohl provádět počítač (Nadkarni, a další, 2011).

2.1 Analýza sentimentu

Analýza sentimentu je proces, který se zaobírá analyzováním autorských textů a pokouší se zjistit jaký autor zaujímá emoční postoj k danému textu. Postoj se nachází na intervalu od negativního sentimentu po kladný sentiment. Tento proces může být velmi užitečný například pro potřeby rozhodování které děláme na základě sesbíraných dat v organizaci, nebo státní správě a dalších oblastech (Feldman, 2013).

Lidé mají tendence nechat se ovlivňovat na základě zpráv, které se jim dostanou do dosahu a které konzumují na denní bázi. Tyto informace jsou ve velké míře zpracovávány tím způsobem, že jsou rovnou přejímány jako pravda a velké množství lidí, zejména střední a nižších vrstev nebere v potaz možnost, že by zprávy mohli být falešné nebo záměrné zkreslené (Tandoc, 2019).

Hlavně záměrné zkreslení zpráv, tedy zakomponování některé negativní, nebo pozitivní emoce může mít vliv na chování člověka. Ať už se bude jednat o jeho mírnění anebo radikalizaci v některých situacích (Tandoc, 2019). Právě analýza sentimentu může odhalit, které zprávy jsou jakým způsobem ovlivněné a jaké je například rozložení těchto zpráv v rámci celého souboru textů, nebo jediného textového dokumentu.

2.2 Nástroje pro zpracovávání analýzy sentimentu

Pro zpracování analýzy sentimentu existují různé nástroje, ať už se jedná o ucelená softwarová řešení, které mohou být distribuovány jako klasické desktop aplikace, nebo SaaS

řešení, nebo o programovací jazyky a k nim dostupné již předpřipravené knihovny s API a frameworky.

Volba nástroje pro zpracování analýzy může působit banálně, ale jedná o velmi důležité rozhodnutí, protože ne každý nástroj disponuje potřebnými možnostmi, které jsou nutné pro správné vyhodnocení dat. Mezi nástroje pro zpracování analýzy sentimentu nejsou řazeny pouze takové, které jsou určeny pro modelování, resp. vyhodnocování modelů, ale také takové nástroje, které jsou používané k přípravě dat, která jsou nutnou součástí samotného provedení analýzy sentimentu.

2.2.1 Nástroje pro přípravu dat

Nástroje pro prvotní přípravu dat, která zahrnuje očištění dat od nepotřebných částí, která nejsou pro analýzu podstatná jsou povětšinou nástroje pro data mining nebo jiné nástroje, které zvládají pracovat s formáty ve kterých jsou zkoumaná data dostupná.

Nepotřebné části souborů dat mohou například být části dat, která nesou informace o tom kde na webu byl analyzovaný text uložen, jaké obsahoval obrázky, v jakém je jazyce apod. Samozřejmě záleží na dalších postupech vyhodnocování, která data v souboru nechat a která odstranit. Tyto postupy silně závisí na tom, jaké informace se snažíme z textu získat.

Datové soubory, které jsou k dispozici ve většině uznávají nepsanou konvenci používaní takového formátu souborů, jehož pravidlem je oddělování hodnot předem definovaným oddělovačem. Například jeden z velmi známých a používaných datových souborů pro analýzu sentimentu vytvořená na Stanfordově univerzitě používá jako oddělovač svislou čáru (|) mezi daty a netisknutelný znak „\n“ pro oddělení řádků (Socher, a další, 2013).

Níže v textu je uveden neúplný výčet nástrojů možných naivních nástrojů pro zpracování dat. Pro jednodušší a méně objemné datové soubory lze použít například Microsoft Excel, nebo jeho mutace od jiných společností (OpenOffice, Google Sheets apod.).

Pro složitější datové soubory, které potřebují větší zásah do úpravy dat, nebo kde je jejich struktura nekonzistentní a je nutné je upravit tyto softwarové produkty už nepomohou. Je nutné zvolit programy přímo určené na zpracovávání dat. Jako příklad může být použit například SPSS Modeler od společnosti IBM. Tento software pro data mining má v sobě

integrované nástroje pro práci se složitými datovými strukturami a zároveň v něm lze provádět základní i velmi pokročilé datové analýzy.

Poslední jsou uvedeny programy, které jsou konstruovány na míru datovým souborům, které zpracovávají. Tyto programy se mohou teoreticky psát v každém výpočetně úplném jazyce, který disponuje slušným vstupně výstupním interface a podporuje bitový streaming pro rychlejší, resp. pro asynchronní zpracovávání dat. Jako příklad může být například Java, C#, C++, R, Python. V této problematice je nejrozšířenější jazyk Python. Python také disponuje řadou knihoven, které zrychlují a zlehčují návrh a implementaci algoritmu. Mezi takové knihovny se řadí například oblíbená Pandas, TensorFlow, SciKit-Learn apod.

2.3 Aplikace analýzy sentimentu v praxi

Analýzu sentimentu lze využít v mnoha různých oborech. Může se jednat o byznys, politiku, finančnictví, nebo třeba analýzu chování a názory obyvatel nějakého území (D'Andrea, a další, 2015), (Wang, a další, 2013).

Každý z těchto oborů sleduje odlišná data, ale většina z nich má společný zdroj těchto dat. Zdrojem dat je povětšinou internet (D'Andrea, a další, 2015). Sběr dat na internetu probíhá pomocí crawlerů, kteří jsou naprogramování pro sběr určitých dat, která se vážou k sledovanému zájmu těch, kteří data sbírají a následně je vyhodnocují (D'Andrea, a další, 2015).

Pokud vezmeme v úvahu například obor byznysu můžeme sledovat, jak zákazníci reagují na naše produkty a do jaké míry jsou s nimi spokojení (Wang, a další, 2013). Tyto data nám mohou říct mnoho o našem produktu v kontextu, jak ho používá určitě velká masa lidí (závisí na vzorku dat). Další možností, jak může analýza sentimentu pomoci našemu podnikání může být sledování rovnou celé značky, kterou podnik reprezentuje. Tento typ dat lze sledovat například z komentářů/recenzí pod výrobky, nebo z diskusí na sociálních fórech.

Sbírání dat se nemusí vždy týkat jen produktů nebo značek. Lze sledovat téměř cokoliv. Může se jednat o celé značky, úzkou produktovou řadu, ale také například podniky, které se nezabývají nabízením produktů, ale službami. Můžeme sledovat kvalitu restaurací, nebo jiných podniků nabízející podobné služby.

Další možností je sběr dat nejen našeho podniku, ale také těch konkurenčních a následné analyzování a porovnávání těchto dat, ze kterých můžeme vytvořit určité politiky nebo strategie, jak se našim konkurentům vyrovnat nebo jak se naopak udržet před nimi (Qiu, a další, 2010).

Jednou z velice důležitou oblastí, kde se hojně využívá analýza sentimentu je politika, resp. politické kampaně. Sběrem dat, která poskytují voliči na sociálních sítích můžeme sestavit relativně přesný obraz kandidátů a zjistit, jak si stojí v očích voličů a na tomto základě můžeme vypracovat strategii, kterou se bude kampaň ubírat. Tohoto můžeme dosáhnout například porovnáváním sesbíraných dat v čase a zjišťovat, jak si vedli nové strategie oproti těm starým (D'Andrea, a další, 2015).

V oblasti finančnictví se sběr dat a analýza sentimentu zaměřuje především na finanční reporty například z prostředí akciových trhů. Tímto způsobem bylo například odhalena (Shumaker, a další, 2012) korelace mezi sentimentem finančních zpráv z Arizonského finančního textového systému (AZFinText) a některými akciemi. Zprávy z AZFinTextu nepřímo ovlivňovali cenu vybraných akcií.

2.4 Analýza sentimentu pomocí strojového učení

Kvůli velké nepřesnosti dosavadních metod zpracovávání analýz sentimentu pomocí počítačů vzniká tlak na použití algoritmů určených pro vytváření výpočetního přístupu nazvaného strojové učení. Učení těchto algoritmů probíhá pomocí obrovských datových balíků, které byli předem ohodnoceny (pokud se jedná o učení pod dohledem) anebo ohodnoceny nejsou a algoritmus se učí z dat sám (učení bez dohledu) (Nadkarni, a další, 2011).

Jedná se tedy o klasifikační (nebo regresní) úlohu, ve které do algoritmu po jednotlivých epochách posíláme vstupní data (v tomto případě ohodnocený testovací textový korpus) a necháme algoritmus se učit. Po naučení algoritmu na požadovanou úspěšnost rozpoznávání sentimentu z trénovacího souboru lze na vstup zasílat jakákoli data stejné struktury a algoritmus bude schopen určit sentiment dat s takovou pravděpodobností na jakou byla síť vycvičena.

Analýza sentimentu sebou ale nese i několik sub-problému, které se týkají hlavně zpracování dat, kterými algoritmy učíme a posléze i „krmíme".

Před začátkem učení algoritmu je tedy nutná data upravit do podoby vhodné pro učení algoritmu a následné vyhodnocování sentimentu. Některé problémy, které mohou nastat jsou specifické k případu a povaze dat na která se snažíme výsledný model nasadit. Například pokud bychom zpracovávali data z lékařského prostředí, nemocniční zprávy, recenze pacientů apod. je potřeba upravit datový korpus tak, aby učený algoritmus byl schopen rozpoznávat například různé tituly lékařů, nebo často lékaři používané zkratky (Nadkarni, a další, 2011).

Výstupy analýzy povětšinou bývají ve formě ohodnocení analyzovaných textů na předem definované, neměnné stupnici. V praxi se většinou používá tříbodová, nebo dvoubodová stupnice s hodnotami: kladný, neutrální a záporný sentiment (kladná a záporná v případě dvoubodové). Tato stupnice může obsahovat více položek, avšak s tím se zvedá složitost a prostor pro špatnou klasifikaci.

2.5 Proces zpracování analýzy sentimentu.

Proces analýzy sentimentu pomocí počítačových algoritmů se skládá z několika po sobě jdoucích kroků, které mají neměnné pořadí. Výčet kroků v procesu je následovný: zpracování dat, transformace dat, analýza sentimentu, vyhodnocení a interpretace výsledků (McShane, a další, 2021).

2.5.1 Zpracování dat

Zpracování dat se pro použití v statisticky pravděpodobnostních klasifikátorech v podstatě neodlišuje se od klasické sémantické analýzy. Cílem přípravy je „očistit“ zkoumaná data od částí, které nejsou pro analýzu podstatné. Definice toho, co je důležité a co ne musí být dodržena napříč celým analyzovaným souborem, protože by potom došlo k nekonzistenci dat a znemožnění spolehlivé interpretace dat pro klasifikátor (McShane, a další, 2021).

2.5.2 Transformace dat

V části procesu, která se zabývá transformací dat se v případě klasické lingvistické sémantické analýzy používá metoda TMRs (někdy se používá i označení ARM) tato metoda přiřazuje vybraným částem textů z předchozího kroku určitý význam, který dále hraje roli při samotné analýze sentimentu.

Metoda ARM byla vyvinuta týmem pod vedením Fei Liu v 2015 avšak první příklad byl zveřejněn doktorkou Laurou Baranescu, která první příklad uvedla již v roce 2013. Metoda pracuje s daty takovým způsobem, kterým se datům snaží přidělit směr a shrnovat je do vrcholů ze kterých se tvoří acyklický graf. Kvůli složitosti algoritmu metoda počítá s řešením pomocí sub grafů. Velikost grafu, resp. sub grafu závisí na složitosti a objemu textu (Dohare, a další, 2017).

Tato metoda je samozřejmě dostupná i v algoritmickém řešení pro použití počítačem ale pro většinu analýz sentimentu se nepoužívá z důvodu vysoké náročnosti na výpočet (Nadkarni, a další, 2011).

V oboru výpočetní lingvistiky, která se zabývá řešením NLP problémů, a tedy i analýzou sentimentu se používají stejné metody, jaké se používají pro obecný textový data mining. Jedná se o tokenizaci a klíčování dat. Tokenizace dat znamená rozdělení souvislých vět nebo textových shluků na jednotlivá slova. Těmto jednotlivým slovům se říká tokeny.

Ukázka tokenizace v anglickém jazyce je vidět na obrázku 1. Tokeny jsou používány pro pokročilé metody zpracování dat, které jsou určené již pro matematické modely jako je například model Bag-Of-Words apod. přiřazující jednotlivým tokenům váhy. Tyto váhy jsou následně použity pro trénování klasifikačních modelů v analýze sentimentu. Proces klíčování je používám k odstranění některých předpon a přípon které umožní definovat výčet kořenů slov, ze kterých se dále pokouší zjistit smysl slov (IBM, 2020).

```
import nltk

text_sample = "Czech University of Life Sciences Prague"

tokens = nltk.word_tokenize(text_sample, 'english')

#VYSTUP = ['Czech', 'University', 'of', 'Life', 'Sciences', 'Prague']
```

Obrázek 1 - Příklad tokenizace, zdroj: autor

Metoda klíčování je pro analýzu sentimentu někde nevýhodná právě kvůli odstraňování předpon a přípon. Problém vzniká hlavně u jazyků, které nejsou angličtina a používají předpony pro změnu intenzity významu slova (superlativy).

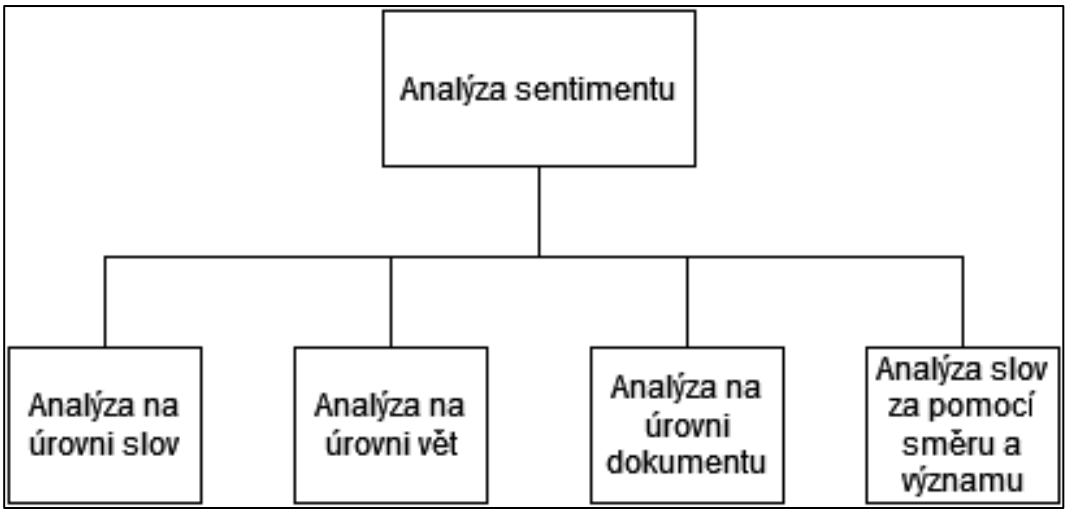
2.5.3 Analýza sentimentu

Samostatná analýza sentimentu následně probíhá vyhodnocením matematických modelů, které vznikly z přípravy dat (trénování klasifikátoru), která je popsána výše. O vyhodnocení modelů se nejčastěji starají již zmíněné modely strojového učení, které byli naučeny na cvičebních souborech a jejich přesnost byla ověřena na verifikačním souboru. Mezi tyto modely se řadí například klasifikátory, které jsou zmíněny v kapitolách dále v textu.

2.5.4 Interpretace výsledků

Po vyhodnocení vstupních dat následuje část procesu, která se zabývá interpretací výsledků. Atribut klasifikátoru, který ovlivňuje věrohodnost výsledků je jeho přesnost, tedy s jakou precizností je schopen rozpoznat úroveň sentimentu ze vstupního textu a následně tento sentiment ohodnotit na určené stupnici. Výsledky jsou párované. K příslušnému textu a k němu hodnota, která byla textu přidělena matematickým modelem vyjadřuje míru sentimentu (Kharde, a další, 2016).

Analýzu sentimentu lze rozdělit do několika skupin podle toho, jak uskutečníme rozdělení dat, která chceme analyzovat. Rozdělením dat se rozumí úroveň granulity, kterou zvolíme pro analyzovaný datový soubor. Podle Khardeho a Sonawaneho (Kharde, a další, 2016) lze rozdělit data do čtyřech základních skupin. Skupiny jsou zobrazeny na obrázku 2 níže.



Obrázek 2 - Druhy analýz sentimentu, zdroj: (Kharde, a další, 2016)

Úroveň rozdělení dat na úrovni dokumentu funguje na principu označení celého dokumentu určitou úrovní sentimentu, ten může být buďto pozitivní nebo negativní. Běžným přístupem, jak klasifikovat sentiment celého dokumentu je nejdříve klasifikovat jednotlivé věty a následně provést syntézu těchto výsledků. Na výstupu potom dostaneme sentiment celého dokumentu.

Další metodou rozdělení granulity je na úrovni vět. Tento způsob probíhá analogicky stejně jako je klasifikace na úrovni celého dokumentu jenom místo celých vět klasifikujeme jednotlivá slova ve větách a následně znovu provádíme již popsanou syntézu sentimentu.

Předposledním přístupem, jak lze rozmělnit a klasifikovat data je na úrovni slov. Tato metoda je vlastně základním vyhodnocovacím kamenem analýzy sentimentu. Pomocí klasifikace sentimentu na úrovni slov jsme totiž schopni aproximovat další sentiment vyšších úrovní které jsou popsány výše. Ke klasifikaci samostatných slov lze přistupovat dvěma způsoby. Prvním způsobem je klasifikace pomocí slovníku a druhá pomocí klasifikovaného datového korpusu (Kharde, a další, 2016).

Proto aby bylo možné zmíněnou techniku použít je nutná ještě poslední úroveň granulity dat, která se v analýze sentimentu používá. Tato technika umožňující co možná nejpreciznější klasifikaci slov používá metody, které umí z textu vyčíst směr zprávy a ostatní její vlastnosti (Kharde, a další, 2016).

2.6 Metody zpracovávání dat pro NLP

Jak už bylo popsáno výše analýzu sentimentu lze rozdělit do několika kategorií. Jedná se o analýzy z pohledu dokumentu, odstavců, vět a slov. Pro každou z těchto metod se hodí jiná metoda pro zpracování vstupních dat (Mohey, 2016). Tyto metody jsou vhodné zároveň i pro jiné NL problémy, a nejen pro analýzy sentimentu. Tato práce se zabývá vyhodnocením analýzy sentimentu pomocí umělé inteligence, tudíž zde jsou popsány hlavně modely, které lze v tomto přístupu využít a které mají pro zpracování práce potenciální smysl.

Analýza sentimentu musí zpracovat obrovské množství lexikologických informací, proto aby byla schopna určit polaritu výsledných dat (kladná, záporná). Je proto nutné znát metody úprav textových dat, které dokážou rozeznat, jak už bylo stručně nastíněno výše,

různé atributy analyzovaných textů. Tato vlastnost je velmi důležitá, pokud se jedná o analýzu dat, které jsou psány ve složitějších jazycích (Mohey, 2016).

Složitějším jazykem se rozumí takový, které obsahuje velké množství mnohoznačných slov, do této kategorie jazyků spadá například právě i angličtina ve které jsou psány analyzované zpravodajské texty v praktické části práce.

2.6.1 Bag of Words

Pokud použijeme naivní popis metody Bag of Words, aby bylo jasné, co na první pohled metoda dělá, řekneme že metoda převádí podle předem definovaného matematického aparátu části analyzovaných textových dat na vektory. Tyto vektory jsou různé podle kontextu, ve kterém se nachází data, které jsou analyzována. V té nejjednodušší formě BOW se jedná o shromáždění dat do jednoho pytle, který je bez duplikátů a následně ohodnocení jednotlivých dat pomocí vektoru, jak je vidět v tabulce níže (IBM Corporation, 2022).

Jako příkladové věty byli zvoleny: „Poznání je smysl života“ (věta 1) a „Tato věta nedává smysl“ (věta 2). Potom výsledné vektory budou vypadat jak je znázorněno v tabulce 1.

Tabulka 1 Ukázka naivní metody BOW, zdroj: autor

	poznání	je	smysl	života	tato	věta	nedává
Věta 1	1	1	1	1	0	0	0
Věta 2	0	0	1	0	1	1	1

Tato definice metody BOW není jediná, existují další variace, které přiřazují jednotlivým datům vektory podle jinak zvolených matematických aparátů. Například jednou z pokročilejších variant metody BOW je vážené BOW. Tato metoda nebere v potaz pouze na slova v množině, ale zároveň pracuje i s jeho váhou. Tato váha je stanovena jako počet výskytů slova v množině. Tímto dostaneme přesnější reprezentaci dat, z kterých tvoříme vektor. Přesnější je proto že dáváme slovům, která jsou zastoupena v celkové množině vícekrát vyšší váhu (George K, a další, 2014).

Představme si dva různé dokumenty D1 a D2. D1 obsahuje následující data: „Les je skvělé místo. Už se těším až uvidím les.“ A D2 obsahuje: „Hlavní je klid a nohy v teple.“ Potom tedy množina, ze které se budou tvořit vektory vypadá následovně:

{„les“, „je“, „skvělé“, „místo“, „už“, „se“, „těším“, „až“, „uvidím“, „hlavní“, „klid“, „a“, „nohy“, „v“, „teple“}

Z takto sestrojené množiny slov víme že dostaneme pro každý dokument vektor o 15 prvcích. Výsledný vektor z množiny výše tedy bude pro D1 a D2 následující:

$$D1 = (2,1,1,1,1,1,1,1,1,0,0,0,0,0,0)$$

$$D2 = (0,0,0,0,0,0,0,0,0,0,1,1,1,1,1)$$

Druhá zvolená metoda má výhodu, že dokáže lépe rozlišovat mezi dokumenty díky vahám, které jsou součástí konstrukce vektorů (George K, a další, 2014).

2.6.2 Metoda latentního sémantického indexování

Metoda latentní sémantického indexování se používá pro popsání významu textu. Ostatní metody jako je například Bag of Words nebo TF-IDF jsou v tomto směru poněkud nepřesné. Důvodem nepřesnosti výše zmíněných metod je jejich způsob interpretace slov v dokumentech. Jednoduché vektory, kterými jsou slova reprezentována v těchto metodách nedokážou zachytit význam slov v takové míře jako metoda LSI.

Tyto nepřesnosti jsou způsobené tím, jak je nastavená lidská řeč. Lidská řeč obsahuje vysoké množství synonym, ale i citově zabarvených slov, které mohou měnit význam zprávy, a právě tyto nevyzpytatelné znaky nelze klasickými metodami předvídat přesně.

LSI předpokládá že v dokumentu se nachází ještě významová podvrstva, která tyto metadata dokáže popsat. Tato vrstva se zpracovává pomocí matematické metody singulárního rozkladu (Rosario, 2000).

Metoda singulárního rozkladu (SVD) je rozklad reálné matice na součin dvou unitárních matic které mají rozměry *m x m* a *n x n*. Podle Rosario (Rosario, 2000) data z testů výkonnosti naznačují že takto upravené dokumenty vykazují lepší tendenci popsat směr a význam sentimentu zkoumaného dokumentu.

Latentní vrstva dokumentu vzniklá pomocí SVD má z pravidla méně dimenzí nežli prostor zkoumaného dokumentu. Lze tedy říct že metoda LSI nejenže dokáže rozeznat směr

sentimentu dokumentu, ale zároveň ji zle použít pro zjednodušení prostoru, z toho tedy vyplývá že LSI je zároveň metodou redukce prostoru (Rosario, 2000).

Metoda má však i svoje nevýhody. Jako jednu z největších lze zmínit paradoxně paměťovou neefektivitu. Mohlo by se zdát, že s počtem snižujících se dimenzí prostoru se sníží i nutné množství paměti pro uložení vektorů vzniklých z SVD metody. Opak této hypotézy dokázal David Hull (Hull, 1995) ve svém pokusu. Pokusný dokument se skládal z 90 441 nenulových vstupů dokumentu. Vytvoření pouhých 100 LSI vektorů z 1 399 možných vyžadovalo 139 900 nenulových vstupů. Důvodem, (Blei, a další, 2003)proč je k reprezentaci tak malého počtu LSI vektorů nutná taková spousta vstupů je ten že pro vytvoření LSI vektoru jsou nutné vstupy složené z reálných čísel kdežto pro jednoduchou vektorovou reprezentaci dokumentu nám postačují celá čísla.

2.6.3 Metoda TF-IDF

Metoda TF-IDF se řadí do skupiny statistických metod pro určování váhy testovaného souboru (slov). Metoda zkoumá váhy slov v textu a na tomto základě se jim pokouší přiřadit určitou váhu. Na základě této váhy je množné říct, jak je zkoumaný prvek hodnotný ve smyslu celého dokumentu. Metoda vytváří vektory v prostu, kde každý vektor reprezentuje jedinečné slovo (ZHANG, 2005).

Stanovení váhy se počítá pomocí dvou hlavních kritérií, a to kolikrát se slovo vyskytuje ve zkoumaném souboru a jaká je jeho inverzní frekvence vzhledem k dokumentu. Pro výpočet první hodnoty existuje více metod, ale všechny jsou ve své podstatě velmi jednoduché. Počet slov v dokumentu se dá zjistit jednoduchým podílem počtu výskytů daného slova ku celkovému počtu slov v dokumentu. Hodnoty inverzní frekvence se již ale dosahuje více sofistikovanými metodami.

Jedná se vlastně o logaritmus podílu všech dokumentů a dokumentů obsahujících určité slovo. Tyto hodnoty se mohou pohybovat od 0 do 1. Čím blíže se hodnota nachází 0 tím je slovo důležitější, protože se nachází ve více dokumentech, pokud se hodnota přibližuje 1, jedná o opak a slovo nemá takovou důležitost vzhledem k ostatním dokumentům. Formální zápis výpočtů jednotlivých hodnot:

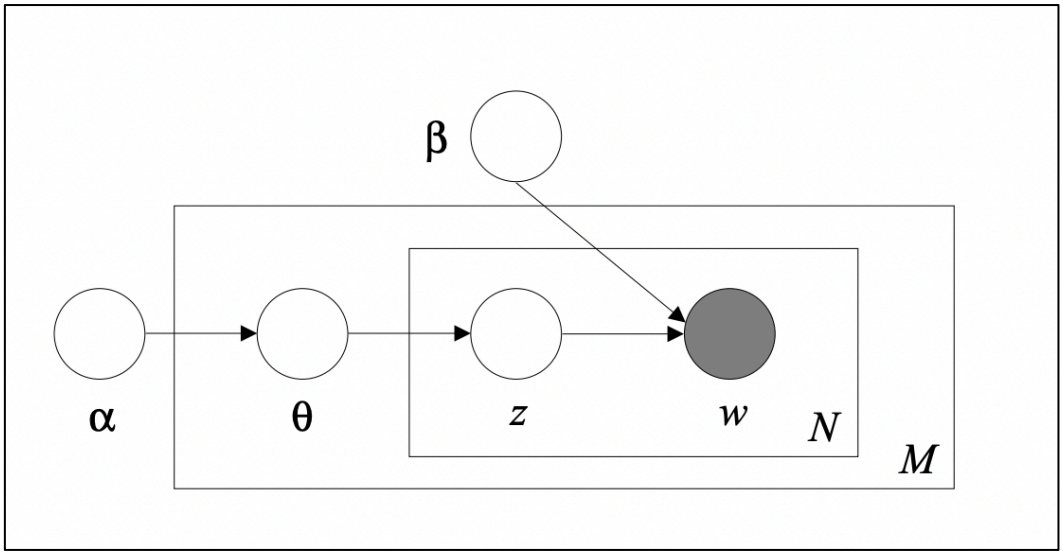
$$TF_t = \frac{n_w}{n_{wa}}, \qquad IDF_t = \log_e\left(\frac{n_{da}}{n_d}\right)$$

Rovnice 1 - Stanovení váhy kritérií pro TF-IDF, zdroj: (ZHANG, 2005)

Kde n_w reprezentuje výskyt hledaných slov, n_{wa} celkový počet slov, n_{da} reprezentuje celkový počet dokumentů a n_d počet dokumentů, kde se vyskytuje zkoumané slovo (TFIDF).

2.6.4 Metoda latentního Diritchletova přiřazení

Metoda latentního Diritchletova přiřazení je jedna z dalších redukčních metod, které lze použít pro zjednodušení korpusu, tedy sady dokumentů. LDA je třívrstvá metoda využívající Bayesiánských statistických metod pro redukci zkoumaného korpusu. Každá vrstva je větší „abstrakcí“ nadřazené vrstvy. Metoda pracuje se slovy, ze kterých generuje konečné shluky, které jsou nakonec transformovány na pravděpodobnostní množiny dokumentů. Tyto pravděpodobnostní množiny mohou relativně věrně reprezentovat samotné originální dokumenty (Blei, a další, 2003). Obrázek 3 graficky znázorňuje LDA rozdělení.



Obrázek 3 - Grafické zobrazení vrstev LDA, zdroj: (Blei, a další, 2003)

Parametry α a β reprezentují korpus. Tyto parametry jsou inicializované pouze jednou při vytváření korpusu. Parametr θ se váže k dokumentům obsaženým v korpusu, tento parametr je inicializován pro každý dokument v korpusu. Posledními parametry jsou z a w, tyto parametry jsou součástí jednotlivých slov, která jsou uvnitř dokumentů. Každý parametr z,w je inicializován pro každé slovo v dokumentu (Blei, a další, 2003).

LDA je oproti jiným metodám jako je například LSI nebo TF-IDF mnohem efektivnější v redukcích prostorů se zachováním užitečných statistických pojítek mezi daty, které lze následně využít pro další analýzy.

2.6.5 Word2Vec

Jedná se o metodu, která jsou součástí skupiny výpočetně rychlejších a jednodušších metod na zpracování dat. Metoda se zakládá, například, stejně jako metoda Bag of Words na tvoření vektorů z textu obsaženého v dokumentech, který je vložen na vstup této metody. Metoda Word2Vec zpracovává tokenizovaná, ale jinak neupravená data a převádí je na vektory (Sienčnik, 2015). Výhodou této metody je její jednoduchost a dovoluje tedy velmi rychlé učení neuronové sítě na rozdíl o metod které fungují na bázi vytváření shluků dat, které jsou si navzájem podobné nebo mezi sebou mají určité užitečné statistické vazby jako je například LDA, LSI TF-IDF apod.

Další důvod, proč je tato metoda tak oblíbená například společnostmi jako je například Google je že nabízí schopnost velmi rychle vyhledávat a doplňovat texy. Velmi dobře to lze demonstrovat na analogii kterou popsal ve své práci Kenneth W. Church (Church, 2017) a to sice: *man is to woman as king is to x'*. Síť, která využívá Word2Vec je potom velmi rychle schopna dohledat slovo queen tím způsobem, že prohledává slovník slov *V*, který má metoda k dispozici a hledá takové *x'* maximalizuje funkci definovanou níže (Church, 2017).

$$\hat{x} = ARGMAX_{x' \in V}\, sim(x', king + woman - man)$$

Rovnice 2 - Maximalizační funkce pro vyhledávání textu pro metodu Word2Vec, zdroj: (Church, 2017)

Každé slovo v této rovnici (king, woman, man) je v samotném výpočtu reprezentováno jako vektor. Tyto vektory jsou reprezentovány jako sekvence K desetinných čísel, kde K zároveň reprezentuje počet dimenzí vektoru. V praxi se často používá K = 300. Výsledná shodnost *sim* je potom tedy funkcí *sim(x,y)* a je definovaná je cosinus (Church, 2017). Potom platí:

$$sim(x, y) \equiv \cos(\vec{x}, \vec{y}) = \frac{\vec{x}.\vec{y}}{|\vec{x}|.|\vec{y}|}$$

Rovnice 3 – Zápis shodnosti pro sim parametr v metodě Word2Vec, zdroj: (Church, 2017)

Pro tyto rovnice bylo navrhnuto několik přepisů, ale ve výsledku nezáleží, jaký je použit. Další možností optimalizace metody Word2Vec může být ve formě úpravy velikosti prohledávaného dokumentu pro hledání maxima funkce *sim(x,y)*. Tato optimalizace spočívá ve statistické redukci prohledávaného souboru, tudíž metoda nemusí prohledávat celý dokument, ale jen jeho malou část.

2.7 Způsoby vyhodnocování sentimentu pomocí umělé inteligence

Analýza sentimentu se tradičně vyhodnocuje pomocí algoritmů strojového učení. Tyto algoritmy mohou být různých typů a jejich použití se může lišit. Typ použitého algoritmu závisí na typu dat, které vyhodnocujeme (na jednodušší data nám postačují velmi jednoduché algoritmy), na druhu prvotního zpracování dat, pro vstup do učeného modelu a na výpočetní úrovni infrastruktury do které patří jednak velikost úložiště a za druhé výpočetní výkon, který jsme schopni poskytnout.

Pokud subjekt (jednotlivec, firma, expert), který bude vypracovávat a vyhodnocovat analýzu sentimentu nemá k dispozici vhodnou infrastrukturu je možné si výpočetní výkon pronajmout. Jako příklad může sloužit Google Cloud, Microsoft Azure, Alibaba Cloud, Amazon Web Services, IBM apod.

2.7.1 Učení algoritmů

Pro správné vyhodnocování vstupních dat je nutné dávat předpřipravená data na vstup takovým modelům strojového učení, které mají maximalizují úspěšnost klasifikace (Veselý, 2012). Tento proces maximalizace se nazývá učení sítě a může probíhat několika způsoby. Samotný princip učení algoritmů strojového učení stojí na principu trénování modelu v generování takových prediktivních modelů, které dosahují, nebo se přibližuje požadované přesnosti.

Pro to abychom mohli algoritmus natrénovat jsou potřeba **trénovací soubory**. Trénovací soubory jsou takové, které obsahují stejná (nebo podobná) data jako klasický dokument s tím rozdílem, že jednotlivá data jsou v něm ohodnocena správným výsledkem (Veselý, 2012).

Na základě tohoto ohodnocení je algoritmus, který hlídá správnost vyhodnocování schopen poznat, jak sít pracuje a popřípadě upravit pro další data váhy sítě tak, aby přiblížil přesnost vyhodnocování správným směrem.

Způsoby učení se dělí na dvě hlavní skupiny, a to učení pod dohledem (supervised learning) a samostatné učení (unsupervised learning). Každá z těchto metod má svoje postupy, výhody a nevýhody. Tyto atributy učení jsou popsány níže.

Učení pod dohledem na rozdíl od samostatného učení je vedené kontrolou, která ověřuje správnost předpovědí modelu oproti trénovacímu souboru, algoritmus se pokouší

v každé další iteraci programu přiblížit očekávanému výsledku a upravovat prediktivní model tak, aby se tomu tak dělo. Existuje několik algoritmů a matematických, informatických postupů, jak tohoto efektu docílit (Osisanwo, a další, 2017).

Do skupiny algoritmů, které se řadí do učení pod dohledem spadají algoritmy pro klasifikační a regresní úlohy (Veselý, 2012). Každý z algoritmů má jiné atributy a je tedy vhodnější pro řešení specifických problémů. Dalším kritériem, na které je třeba myslet, když se navrhuje řešení a vybírají se algoritmy učení je rychlost a přesnost učení. Podle Osisanwa a spol. (Osisanwo, a další, 2017) platí pravidlo, že čím vyšší je rychlost učení algoritmu tím nižší je potom jeho výsledná přesnost. Dalším atributem, který nás může zajímat je náchylnost algoritmů na šum v datech která algoritmem procházejí. Tolerance na šum se u většiny jmenovaných algoritmů příliš neodlišují, tento krok lze tedy zanedbat (Osisanwo, a další, 2017).

Pokud vyžadujeme od klasifikátoru rychlost nabízí se například lineární klasifikátor, který funguje na principu shlukování dat (vektorů) do tříd, které jsou rozděleny pomocí lineárních hranic (Osisanwo, a další, 2017). Pokud chceme použít tyto klasifikátory je nutné znát vstupní data a určit, jestli jsou třídy (množiny) lineárně separabilní (Veselý, 2012), tedy jestli mezi nimi existují jasně definované hranice, podle kterých lze data přiřadit do nějaké ze skupin. Například pokud budeme používat algoritmus SVM stačí nám dvě lineárně separabilní množiny, protože algoritmus dokáže pomocí podpůrných vektorů najít takovou přímku prostorem která rozděluje soubor na dvě separabilní množiny.

Samostatné učení (unsupervised learning) je metodou učení predikátových modelů, kde se model pokouší zdokonalovat v přesnosti výstupních dat podle předem daných pravidel a pokouší se co nejvíce se přiblížit ukončující podmínce která označuje konec učení (Dike, a další, 2018). Toto snažení algoritmu je povětšinou realizováno pomocí shluků, které algoritmy zpracovává na základě dat, která rozpoznává při průchodu vstupních dat algoritmem.

Příkladem samostatného učení (někdy označovaného jako samoorganizace) je Kohonenova síť. Tento typ neuronové sítě je uspořádán do mřížky o velikosti *rxs* potom je pro učební soubor vytvořeno m shluků, které jsou definovány jako m=rs. Následně se síť pokouší upravovat váhy tak aby minimalizovala rozptyl mezi shluky a zároveň aby byla zachována topologie vstupního vektoru. Ta se zachová úpravou vah i pro neurony v určité

vzdálenosti (počet hran nutný pro cestu z výstupního neuro do dalšího) od výstupního neuronu (Veselý, 2012).

Samotné upravování vah potom může fungovat například následovně. Na začátku jsou váhy nastavené na hodnotu, která je náhodně zvolena. V první iteraci algoritmu se náhodně vybere jeden ze vstupů, který se nechá projít sítí a na základě výsledků se upraví váhy takovým způsobem kdy se posunou váhy směrem ke vstupnímu vektoru o poměrnou část (Veselý, 2012).

Dalším způsobem, jak lze dosáhnout učení klasifikačních algoritmů bez učitele je učení sítě pomocí genetických algoritmů. Tento typ algoritmů spočívá ve vytváření postupem času čím dál tím lepší entity (neurony, resp. váhy), které jsou schopny vyhodnocovat vstupní data s vyšší přesností než generace předchozí. Vznik nové generace je podmíněn ukončující podmínce, která je daná například výslednou přesností po jedné epoše (vyhodnocení celého [testovacího] souboru).

Nové klasifikátory (v některých případech se do nových generací přesouvají i ty nejlepší z generace minulé) se vytvářejí několika způsoby. Mezi způsoby, kterými lze vytvořit nové entity je křížení neuronů a mutace neuronů (jev podmíněný náhodou). Výběr klasifikátoru pro křížení je řízen pomocí několika metod selekce entit (Veselý, 2012).

Může se jednat o volbu pomocí ruletového kola, nebo například turnajová selekce kde proti sobě „zápasí“ dvě vybrané entity. Ohodnocení entit je zajištěno pomocí takzvané „fitness“ funkce která udává kvalitu klasifikátoru. Fitness funkce může být definovaná podle nároků jaké si žádá řešení problémů (Veselý, 2012).

2.7.2 Typy algoritmů strojového učení pro aplikaci klasifikátorů

Jako první je potřeba se zamyslet který typ problému (závisí na podobě zkoumaných dat) chceme pomocí algoritmů strojového učení řešit a na základě toho vybrat správný typ algoritmu. Existují dva základní problémy, které lze pomocí algoritmů strojového učení řešit. Jedná se o problémy regresní, kde se snažíme poznat odchylku nebo se k ní alespoň přiblížit pomocí aproximačních funkcí, které jsou obsaženy v učících souborech.

Druhým problémem jsou klasifikační úlohy. Jedná se o rozdělování dat do předem definovaných kategorií. V případě analýzy sentimentu se může jednat o množiny které reprezentují texty které mají negativní a pozitivní zabarvení.

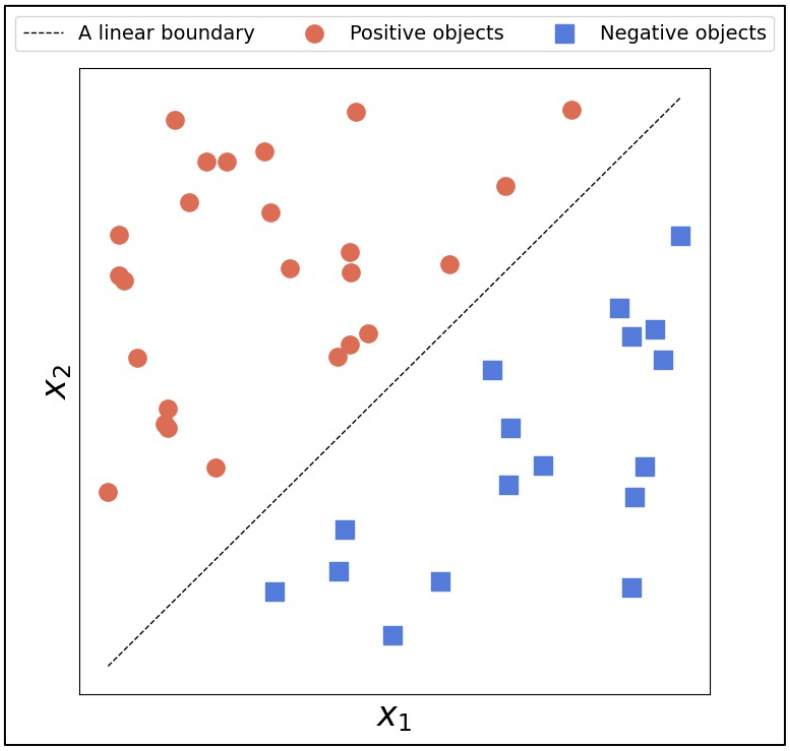
Do těchto dvou kategorií spadá právě i v tomto textu zkoumaná analýza sentimentu dokumentů. Třetím případem jsou shlukovací algoritmy, které vytváření ze vstupních dat shluku, které obsahují data na základě odhadnuté podobnosti.

2.8 Klasifikátory

Jak už bylo popsáno v kapitole, která se věnovala učení pod dohledem existuje několik druhů klasifikátorů, které lze využít. V následujících podkapitolách jsou popsány vybrané druhy klasifikačních algoritmů, které lze využít v klasifikačních modelech. Klasifikátory se odlišují způsobem implementace, výpočetní náročností a způsobem jakým jsou klasifikátory učeny pro klasifikaci dat, která jim nebyla nikdy předtím předložena.

2.8.1 Lineární klasifikátory

Lineární klasifikátory jsou druhem algoritmu, který je součástí skupiny klasifikátorů, které spadají, co se přístupu k učení týče, do skupiny algoritmů učených pod dohledem. Klasifikuje vstupní data (nebo se o to pokouší podle, jak je algoritmus natrénovaný) do předem definovaných od sebe jasně oddělených množin výsledků. To že jsou množiny jasně oddělené, jsou lineárně separabilní, je důležité pro samotné použití algoritmu. Pokud by množiny nebyli lineárně separabilní nelze použít jako klasifikátor lineární algoritmus. Toto omezení je jednoduše způsobeno tím, že pokud by neexistoval jasně rozdělený prostor řešení mohl by se algoritmus dostat do nedefinovaného stavu. Zobrazení separabilních množin je vidět na obrázku 4.



Obrázek 4 - Lineárně separabilní množiny, zdroj: Milos Simic (https://www.baeldung.com/cs/nn-linearly-separable-data)

Další výhodou lineární klasifikátorů je již zmíněná rychlost. Rychlost algoritmu je zapříčiněná nízkou algoritmickou složitostí v porovnání s ostatními typy klasifikátorů (Chen, a další, 2004), které jsou zmíněny níže. Níže je popsaná funkcionalita implementace lineárního klasifikátoru v jednoduché neuronové síti.

Klasifikace v tomto algoritmu funguje na principu porovnávání kombinací, které přijdou na vstup algoritmu. Tyto kombinace mají povětšinou tvar vektorů a jsou definované jako:

$$v = [v_1, v_2, \dots, v_n]$$

Potom co na vstup algoritmus dostane vektor *v* je tento vektor vynásoben váhami *w*, které jsou na hranách neuronů, pomocí kterých je implementovaný algoritmus. Váhy w jsou definované jako:

$$w = [w_1, w_2, \dots, w_n\ ]$$

Pomocí takto zadaných vektorů v a w lze spočítat postsynaptický potenciál neuronu, tedy hodnotu, kterou lze prověřit pomocí klasifikační funkce. Postsynaptický potenciál lze

spočítat jako skalární součin vektorů v a w a přičtením prahu neuronu, kde práh je definovaný jako w_0. Následný zápis vypadá takto:

$$h = \sum_{i=1}^{n} v_i w_i + w_0$$

Rovnice 4 - Postsynaptický potenciál neuronu, zdroj: (Veselý, 2012)

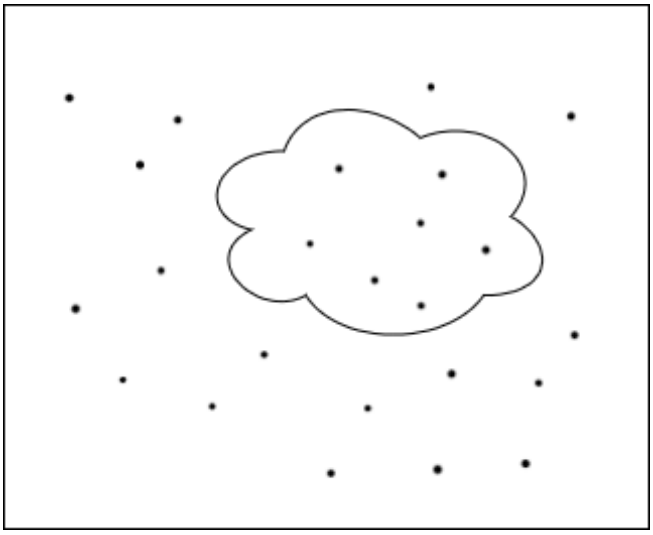
Výsledné zařazení vstupních dat do jedné z lineárně separabilních množin potom závisí na hodnotě proměnné *h* (Veselý, 2012). V případě že vyhodnocujeme pomocí skokové přenosové funkce, která se používá pro dělení do dvou množin záleží pouze jestli hodnota *h* překročila prahovou hodnotu, která je určena ve skokové přenosové funkci.

Všechny hodnoty, které mají vyšší hodnotu, než práh jsou zařazeny do skupiny, která může reprezentovat například kladná ohodnocení textu a všechny hodnoty *h* které tento práh nepřesáhne, se nacházejí ve skupině druhé, která reprezentuje negativní hodnoty. Skoková přenosová funkce je nespojitá v nule, a tedy všechny hodnoty které se budou nacházet v této nespojité části funkce můžeme označit například jako neutrální.

Dalším používaným a velmi oblíbeným algoritmem pro řešení klasifikačních úloh je **SVM** (Support vector machines) algoritmus. Tento algoritmus s jinou, než lineární funkcí pro rozdělení souboru do množin se používá v případech, kdy množiny, které chceme klasifikovat nejsou mezi sebou jasně lineárně separabilní a nelze tedy použít výše zmíněný přístup bez úprav datového souboru.

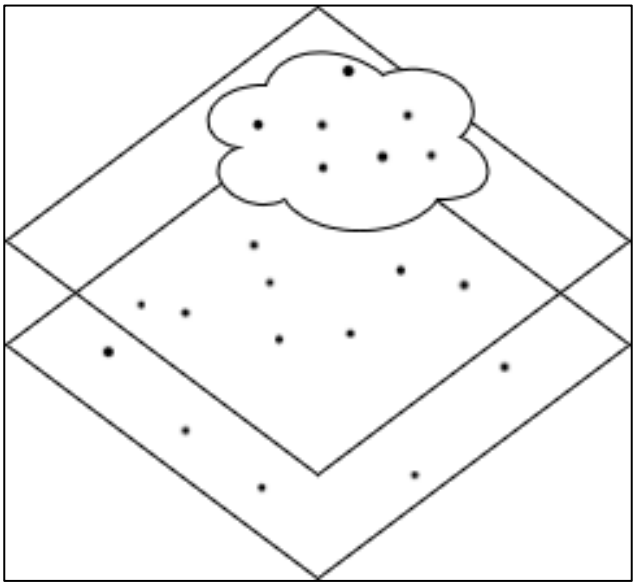
Pokud množiny nejsou jasně lineárně separabilní lze použít jiné funkce pro rozdělení souboru do množin. Mezi ostatní funkce použitelné v tom případě se řadí polynomiální funkce, sigmoidální funkce a RBF funkce.

Způsob, jakým dokáže SVM separovat množiny které nejsou lineárně separabilní stojí na principu převodu dimenze ve které jsou původní cvičná vstupní data převést do jiné, kde už lze rozdělit množiny například pomocí nadrovin nebo rovin. Na obrázku 5 jsou vidět dvě množiny které nejsou lineárně separabilní, na tento problém je klasický lineární klasifikátor nepoužitelný (Veselý, 2012).



Obrázek 5 - Lineárně neseparabilní množina, zdroj: autor

Pokud ovšem využijeme možnost algoritmu SVM změnit dimenzi tohoto prostoru která je nyní dvě na vyšší dimenzi lze již velice snadno množiny oddělit, jak je znázorněno na obrázku 6 níže.



Obrázek 6 - Příklad vytvoření dvou oddělitelných množin pomocí přidané dimenze, zdroj: autor

Pokud se do obrázku přidá třetí dimenze lze množiny oddělit pomocí roviny na vertikální úrovni.

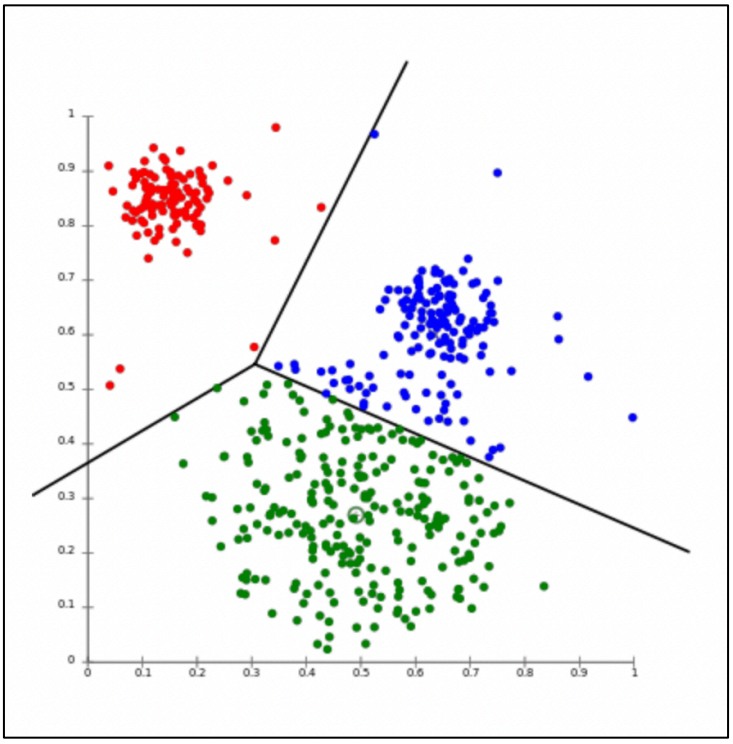
2.8.2 Shlukové klasifikátory

Shlukové algoritmy jsou součástí skupiny algoritmů, které se snaží učit samostatně, bez dohledu. Je používaný k rozdělování datových souborů, které jsou algoritmu předložené

na vstupu, do *n* shluků. Tyto shluky jsou množinou dat, které mezi sebou sdílejí určité podobnosti. Jako další definice může posloužit výklad který říká, že shluk je podmnožinou dat, která mezi svými prvky znatelně nižší vzdálenosti oproti zbytku universa (Likas, a další, 2003).

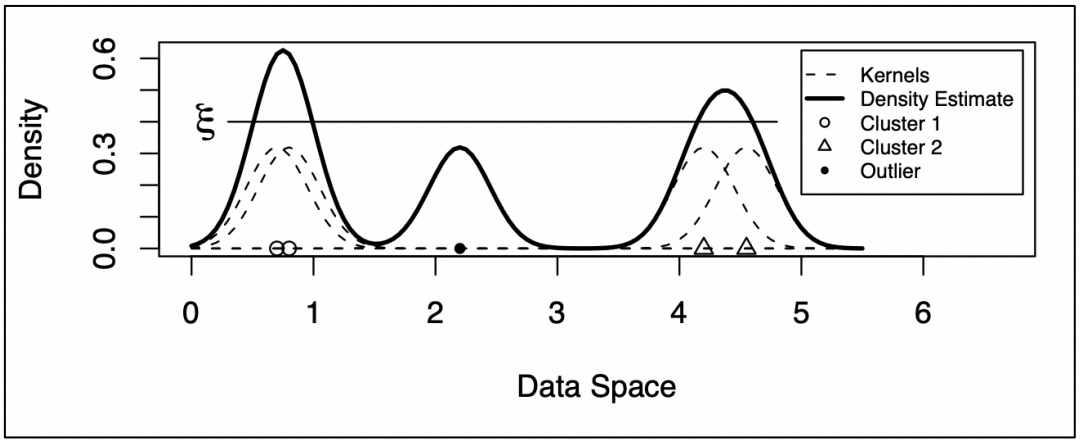
Nevýhodou shlukových algoritmů je jich vysoká výpočetní náročnost a bylo zpracováno několik metod které se pokouší tento problém minimalizovat a algoritmus optimalizovat. Jednou z touto metod jsou například výše zmíněné genetické algoritmy, ale tento přístup optimalizace je používaný spíše zřídka (Likas, a další, 2003).

Velmi rozšířeným algoritmem na řešení úloh, které vyžadují shlukování je „k-means“ algoritmus. Předpokladem pro použití tohoto algoritmu je splnění podmínky že data mohou být reprezentována jako body v euklidovském prostoru a zároveň předem známe konečný počet shluků. Princip, na kterém potom funguje shlukování jednotlivých bodů do shluků je řízem pomocí takzvané shlukovací podmínky. Tato podmínka stanovuje, kdy bod do shluku náleží a kdy se nachází mimo. Jednou z často používaných shlukovačích podmínek je euklidovská vzdálenost od středu shluku (Veselý, 2012) (Likas, a další, 2003). Příklad shlukování je znázorněn na obrázku 7 níže.



Obrázek 7 - Příklad shlukování s třemi shluky, autor: Gitansh Chadha, Piali Das, and Zohar Karnin (https://aws.amazon.com/blogs/machine-learning/k-means-clustering-with-amazon-sagemaker/), 2018

Dalším možným algoritmem pro vytvoření shluků ze vstupního objemu dat je algoritmus DENCLUE. Tento algoritmus staví na metodě KDE. Tato metoda, v češtině nazývaná jádrovým odhadem hustoty, je jedním ze statistických neparametrických odhadů. Tato metoda vlastně konstruuje klouzavý vážený průměr (obrázek 8) a přesnost odhadu závisí na zvolených zpřesňováních parametrech. (Orava, 2008).



Obrázek 8 - DENCLUE model, autor: (Hinneburg, a další, 2007)

V DENCLUE algoritmu je shluk definován jako lokální maximum z funkce odhadu jádrové hustoty. Data jsou do shluků přiřazována pomocí metody algoritmu DENCLUE, která nese označení „hill climbing“. Metoda funguje na principu procházení dat a hledání jejich lokálních maximum se nachází ve stejných místech jako lokální maximum shluku. Pokud se lokální maximum právě zpracovávaných dat nachází blízko k lokálnímu maximu shluku jsou data do toho shluku přiřazena.

Krok hill climbing byl v první verzi algoritmu jeho největší nevýhodou, protože velmi zpomalovat výpočet. V druhé verze algoritmu byl tento krok optimalizován. (Hinneburg, a další, 2007). Existují i další verze algoritmu, které se všechny více méně zaobírají optimalizací kroku hill climbing. Jednou z takových metod určených pro obrovské datové soubory je varianta DENCLUE-IM vyvinutá Rehioui a spol. (Rehioui, a další, 2016).

2.8.3 Regresní modely

Další používanou skupinou klasifikačních modelů jsou regresní modely. Tyto modely jsou vhodné pro data, pro která se snažíme předpovídat pravděpodobnost výskytu sledované proměnné. V tomto případě se jedná o sentiment jednotlivých zpravodajských textů ze zemědělství.

Existují různé druhy regresních modelů, jedním z nich je lineární regrese, která nám udává které z nezávislých proměnných se v jaké míře podílejí na sledované proměnné.

Často používaným regresním modelem pro tento případ odhadování pravděpodobnosti výskytu je logická regrese. Metoda funguje na principu odhadu pravděpodobnosti závislé (sledované) proměnné na nezávislých proměnných (Sperandei, 2013).

Logistická regrese je metodou použitou v této analýze. Logistická regrese je velmi podobná lineární regresy v tom rozdílu že lineární regrese nemá svůj výstup v podobě binomického rozdělení, tedy pouze dvou hodnotách. Vzhledem k tomu že analyzujeme zpravodajské texty ze zemědělství, na kterých hledáme buď kladný (1) nebo záporný (0) sentiment je logistická regrese ideálním kandidátem na klasifikační model.

2.9 Určení přesnosti klasifikátoru sentimentu

Určení přesnosti klasifikátoru je klíčové pro zvolení správné metody klasifikace analýzy sentimentu a zároveň je to velmi důležitým aspektem pro reprezentaci výsledků modelu. Podle Vinodhiniho a Chandrasekarama (Vinodhini, a další, 2016) lze na určení přesnosti použít pět poměrových ukazatelů. Mezi tyto ukazatele patří míra chybné klasifikace, správnost, kompletnost, účinnost a efektivita. Jednotlivé poměrové ukazatele jsou po jednom popsány a vysvětleny níže.

Míra chybné klasifikace je poměrovým ukazatelem definovaným jako součet chyb prvního a druhého druhu děleno celkovým počtem klasifikovaných dat. Chybu prvního druhu v tomto případě lze vypočítat následovně. Nechť C1 je počet negativně klasifikovaných dat, která měli být klasifikovány jako pozitivní a nechť CP je celkový počet pozitivně klasifikovaných dat. Potom chyba prvního druhu (E1) má vztah:

$$E_1 = \frac{C_1}{C_p}$$

Rovnice 5 - Chyba prvního druhu pro míru chybné klasifikace, zdroj: (Kanmani, a další, 2007)

Chyba druhé druhu je definována takto. Nechť C2 je počet pozitivně klasifikovaných dat, která měli být klasifikována jako záporná a CN je celkový počet negativně klasifikovaných dat (Vinodhini, a další, 2016) (Kanmani, a další, 2007). Potom chyba druhého druhu (E2) má vztah:

$$E_2 = \frac{C_2}{C_n}$$

Rovnice 6 - Chyba druhého druhu pro míru chybné klasifikace, zdroj: (Kanmani, a další, 2007)

Z takto definovaných vztahů lze následně vypočíst celkovou míru chybné klasifikace která má následující vztah.

$$E = \frac{C_1 + C_2}{C_p + C_n}$$

Rovnice 7 - Celková míra chybné klasifikace, zdroj: (Kanmani, a další, 2007)

Druhým možným poměrovým indexem, které lze použít pro vyjádření efektivnosti klasifikátoru je **správnost**. Tento ukazatel je definován velmi jednoduše, a to jako podíl počtu správně klasifikovaných dat jako pozitivních oproti celkovému počtu pozitivně klasifikovaných dat (Kanmani, a další, 2007).

Kompletnost je dalším z možných ukazatelů přesnosti klasifikátorů. Tento ukazatel byl definován dvojicí Briand a Wust v roce 2002. V jejich publikaci byla kompletnost definovaná jako poměr správně kladně ohodnocených dat oproti všem kladně ohodnoceným datům nehledě na výslednou správnost. Tímto poměrem získáme číslo, které říká, jak moc klasifikátor chybuje v označování dat jako pozitivních (Kanmani, a další, 2007) (Vinodhini, a další, 2016).

Předposlední metodou popsanou Vinodhinim a kol. (Vinodhini, a další, 2016) jako užitečnou pro měření **efektivity** klasifikace a definovanou Kanmanim a kol (Kanmani, a další, 2007) je tzv. účinnost. Účinnost je definovaná jako poměrná část, které přiřadíme vyšší riziko chybovosti (fp) ze všech ostatních dat (c). Potom můžeme definovat efektivitu vztahy:

$$Efektivita = \left(\frac{fp}{c}\right) = 1 - \left(\frac{nfp}{c}\right)$$

Rovnice 8 - Efektivita klasifikace, zdroj: (Kanmani, a další, 2007)

Poslední metodou měření podle Vinodhiniho (Vinodhini, a další, 2016; ZHANG, 2005) je účinnost. Účinnost je popsána jako poměrová část klasifikátorem pozitivně ohodnocených dat ze všech ohodnocených dat (Kanmani, a další, 2007). Výpočet účinnosti je definován jako:

$$\text{Účinnost} = \frac{fp\pi_{fp}}{nfp}$$

Rovnice 9 - Účinnost klasifikátoru, zdroj: (Kanmani, a další, 2007)

Dalším možným způsobem, jak definovat přesnost klasifikátoru je kombinací dalších čtyř indexů které popsali Kharde a Sonawane (Kharde, a další, 2016). Jedná se o indexy složené z jednoduché matice (tabulka 2), která je zobrazena níže.

Tabulka 2 Matice pro pro výpočet přesnoti klasifikátoru, zdroj: (Kharde, a další, 2016)

	Pravděpodobně negativní	Pravděpodobně pozitivní
Negativní	TN	FP
Pozitivní	FN	TP

Hodnoty v matici reprezentují správných vyhodnocení (TP), falešně pozitivních ohodnocení (FP), falešně negativních ohodnocení (FN) a pozitivně negativních ohodnocení (TN). Pomocí těch hodnot je možné definovat indexy: přesnosti, preciznosti a odvolání výsledků. Index jsou podle Khardeho a spol. (Kharde, a další, 2016) definovány jako:

$$Přesnost = \frac{TP}{TP + FP}$$

Rovnice 10 - Přesnost vyhodnocování modelu, zdroj: (Kharde, a další, 2016)

$$Senzitivita = \frac{TP}{TP + FN}$$

Rovnice 11 - Senzitivita modelu, zdroj: (Kharde, a další, 2016)

$$Specifičnost = \frac{TN}{TN + FP}$$

Rovnice 12 - Specifičnost modelu, zdroj: (Kharde, a další, 2016)

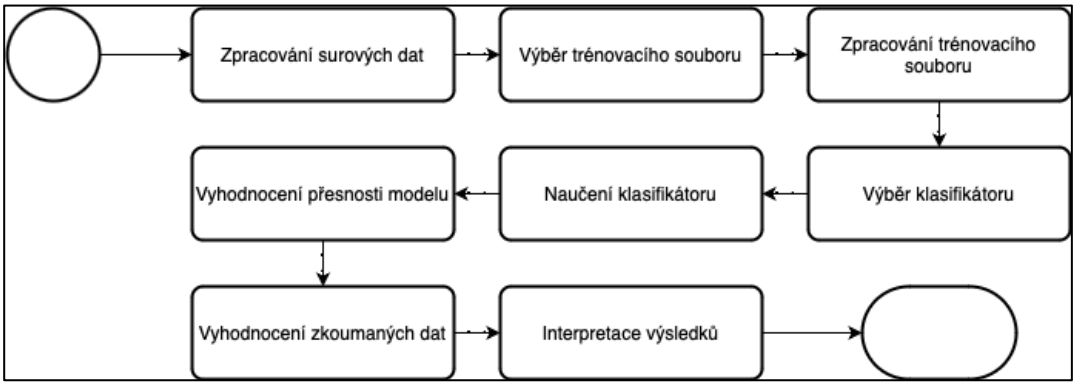
Jako další způsoby ověření kvality modelu se používají grafy typu Lift, CGC, Gini. Tyto grafy porovnávají zkoumané predikované třídy a vynášejí jejich křivky do grafů, podle kterých lze posléze určit přesnost/kvalitu modelu. (Drezner, a další, 2010) (Piatetsky-Shapiro, a další, 1999) (Brandenburger, a další, 2009)

3 Praktická část práce

Zpracování sentimentu je analytický proces, který se skládá z několika kroků, které jsou v kompozitním vztahu. Je tedy nutné provést každý jednotlivý krok pro dosažení celkového výsledku. Další kapitoly práce se zaobírají zpracováním a vyhodnocováním těchto jednotlivých kroků, které vedou k dosažení cíle práce.

3.1 Fáze realizace experimentu

Fáze realizace analýzy sentimentu zpravodajských dat ze zemědělství se skládá z osmi po sobě jdoucích analytických fází. Jednotlivé kroky jsou zmíněny na obrázku 9 níže.



Obrázek 9 - Schéma pracovního postupu analýzy sentimentu, zdroj: autor

Ve fázi jedna, která se zabývá zpracováním surových dat, které jsou předmětem zkoumání probíhá očištění dat od nepotřebných atributů, které nejsou pro samotnou analýzu sentimentu důležité. Dále se v ní zkoumá kvalita souboru. Kvalitou souboru se rozumí základní metriky určující kvalitu dat. Řadí se mezi ně míra chybějících hodnot, počet duplicitních hodnot apod.

Druhá fáze se zaobírá výběrem vhodného trénovacího souboru pro analýzu sentimentu. Důležitými parametry při výběru trénovacího souboru je jeho datová struktura. Volí se takový soubor, který je připraven pro podobné účely podobou jeho dat, abychom nemuseli tato data, kromě následného dělení ve fázi tři, nijak upravovat. Nutností pro správné použití v analýze sentimentu je přítomnost již ohodnoceného sentimentu na trénovacích datech, ze kterých se bude učit vybraný klasifikátor ve fázi čtyři.

Ve třetí fázi probíhá zpracování testovacího souboru. Jedná se o průzkumovou analýzu, která nám poodhalí vlastnosti, které by mohli ovlivňovat přesnost souboru. Jedná

se například o počet měření, které soubor obsahuje (nebezpečí přeučení algoritmu) nebo vyváženost souboru.

Vyváženost souboru říká, jaký je poměr mezi hodnotami sentimentu. Soubor, který obsahuje příliš mnoho hodnot jedné nebo druhé složky může způsobovat nižší citlivost, nebo specifičnost výsledného modelu (tyto míry jsou popsány v kapitole a přesnosti klasifikátorů 2.9). Dále se soubor náhodně rozdělí v předem definovaném poměru na dvě podskupiny. První podskupinou je samotná trénovací množina a druhou je množina testovací. Poměr stanovený pro toto rozdělený bývá různý a záleží na testovacím souboru a klasifikátoru (Muraina, 2022).

Ve čtvrté fázi experimentu dochází k výběru klasifikátoru pro analýzu sentimentu. Na výběru správného klasifikátoru závisí přesnost výsledků, jak je již popsáno v teoretické části (kapitoly 2.8, 2.9), které se zaobírají typy klasifikátorů. Analýza sentimentu je většinou chápána jako klasifikační nebo regresní úloha. Pro účely této práce byla zvolena regrese. Důvod pro výběr regresního modelu je popsán v kapitole 3.6.

Pátá fáze se zaobírá učením zvoleného klasifikačního algoritmu na trénovacím algoritmu. Tato fáze probíhá automaticky po navržení a implementaci zvoleného algoritmu. Do této fáze se také řadí tokenizace dat pro učení algoritmu (toto následně ve fázi sedm platí i pro samotná zkoumaná data).

V šesté fázi se vyhodnocuje přesnost algoritmu dle obecných metrik jako je senzitiva, specifičnost, přesnost a úplnost. Jako další ukazatele přesnosti algoritmu může posloužit Kolmogorov-Smirnovovův graf, graf kumulativního přínosu a graf Gini koeficientu. Pokud výsledky přesnosti modelu nejsou uspokojivé je nutné znovu prozkoumat trénovací data, nebo zvolit jiný klasifikační algoritmus.

V sedmé, předposlední, fázi probíhá samotné vyhodnocení zkoumaných dat na již dostatečně přesném klasifikačním modelu. Výstupní data jsou znovu ohodnocena podobně jako ve fázi šest s tím rozdílem, že již není nutné sledovat grafy přesnosti apod.

V poslední fázi analýzy sentimentu se interpretují výsledky sentimentu zpravodajských dat ze zemědělství, kterých bylo dosaženo analýzou.

3.2 Výběr použitých technologií

Pro zpracování analýzy sentimentu byl sestaven a použit balík nástrojů běžně používané v praxi, který umožňuje efektivní práci s daty, klasifikaci a vyhodnocení. Jako další byli vybrány podpůrné nástroje pro statistické analýzy datových souborů.

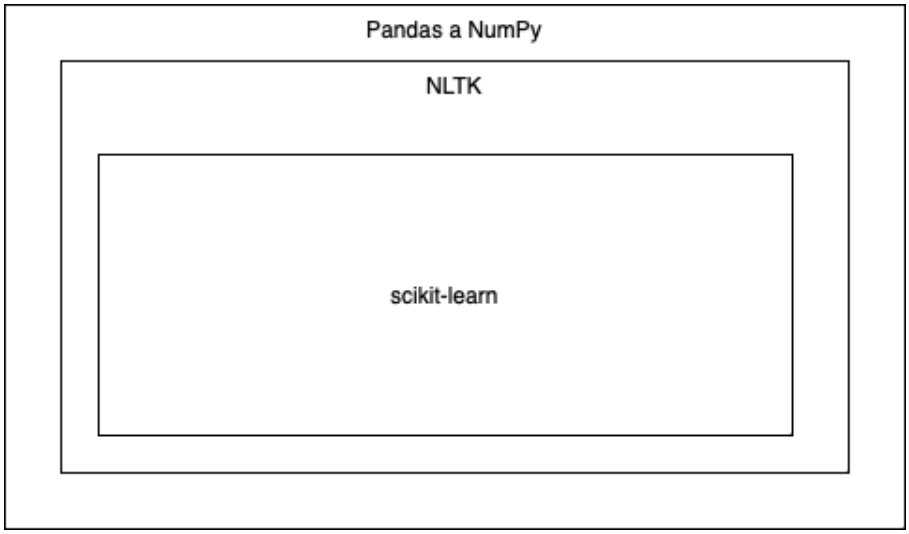
Pro celkovou implementaci algoritmů pro zpracování dat byl vybrán programovací jazyk Python. Tento jazyk je vhodný pro tento typ úloh svým typem. Jazyk Python se řadí mezi vysokoúrovňové, multi-paradigmové programovací jazyky. Tyto vlastnosti říkají, jak lze jazyk Python používat a popisují jeho vlastnosti. To že je jazyk multi-paradigmový je obrovskou výhodou pro psaní kódu, protože nenutí programátora k jednomu stylu návrhu algoritmů, ale programátor si může zvolit který styl je mu bližší, nebo je to může kombinovat (Python Software Foundation, 2023).

Skripty v jazyce Python jsou přehledné a jednoduché na implementaci. Další výhodou jazyku je to, že je vyvíjen jako open-source projekt a je tedy přístupný zdarma pro všechny. Verze jazyku Python použitá pro implementaci analýzy sentimentu zpravodajských textů ze zemědělství v této práci je 3.9.6.

Protože implementace některých potřebných algoritmů pro analýzu sentimentu by byla časově neefektivní byli vybrány a použity již hotové a Python komunitou prověřené knihovny (Stack overflow, 2022), které mají tyto funkce již v sobě zabudované a jsou k dispozici přes API. Funkce, které jsou nutné pro analýzu sentimentu jsou například práce s datovými rámci, pomocí kterých jsou v jazyce Python reprezentovány jak zkoumaná data, tak data trénovacího datového souboru.

Dále jsou použity knihovny pro zpracovávání textu. Konkrétně jsou algoritmy pro výpočet tokenizace a vektorizace textu, které je nutné pro učení klasifikačních algoritmů a také pro následné vyhodnocení samotných zpravodajských textů ze zemědělství.

Zvolené knihovny jsou Pandas pro zpracování a manipulaci s datovými rámci, NumPy pro vylepšenou práci s vektory v jazyce Python, MatPlotLib pro vykreslování grafů, NLTK pro zpracování textů a SciKit-Learn (Pedregosa, a další, 2011) pro samotné modelování a trénování modelu pro klasifikaci sentimentu zpravodajských textů ze zemědělství. Návaznost jednotlivých knihoven na analýzu sentimentu a na další využití výstupů dalšími knihovnami je znázorněn na obrázku 10 níže.



Obrázek 10 - Hierarchie použitých knihoven pro analýzu sentimentu, zdroj: autor

Knihovny byli vybrány na základě podporovaných funkcionalit a také dle oblíbenosti/používanosti jak v profesionální komunitě programátorů zabývající se umělou inteligencí, nebo problémy typu data mining tak v komunitě laické veřejnosti (Stack overflow, 2022). Verze knihoven jsou obsaženy v souboru *requirements.txt*, který je společně se zdrojovým kódem součástí přílohy. Jako podpůrné programy jsou používány Microsoft Excel a IBM IPSS Modeler 18. Podpůrné programy jsou využívány pro tvorbu a ověřování grafů.

3.3 Popis zkoumaného souboru

Soubor, který je předmětem analýzy sentimentu obsahuje data z webového portálu agris.cz (ČZU a MZČR, 2022), který se zabývá zpravodajstvím z oblasti zemědělství České republiky (jedná se o popis vývoje trendů zemědělské výroby reprezentovanými statistickými údaji z českého statistického úřadu), vztahy českých zemědělců vzhledem k evropské unii (zde se jedná o zpravodajské texty týkající se importu a exportu produktů zemědělské výroby).

Pro samotnou analýzu sentimentu byla vybrána sekce týdenních shrnutí v anglickém jazyce. Jedná se o soubor, který obsahuje v surovém nezpracovaném stavu 4 468 záznamů. Jedná se o záznamy od 1.4.2004 do 11.10.2022, které obsahují shrnutí v anglickém jazyce za poslední týden, perex, datum vzniku a další metadata.

Všechny údaje obsažené v jednotlivých záznamech s vysvětlivkami jsou uvedeny v tabulce 3 níže.

Tabulka 3 - Seznam atributů datového souboru pro analýzu, zdroj: autor

Název sloupce	Vysvětlivka
id_text	Primární klíč záznamu
stamp	Otisk označující hodnotu v šestnáctkové soustavě ve formátu 0x0000000000000000
id_zdroj	Cizí klíč odkazující na zdroj
id_obrazek	Cizí klíč odkazující na obrázek
id_jazyk	Cizí klíč odkazující na jazyk textu
id_soubor	Cizí klíč odkazující na soubor
zobrazit	Ditochomická proměnná starající se o viditelnost
porizeno	Datum pořízení článku
datum	Datum zveřejnění článku
nazev	Název souhrnu
url	Webová adresa na odkaz v textu
cesta	Číselná hodnota ukazující na pozici na stránce
perex	Úvodní název článku
fulltext	Samotný souhrn za poslední týden v anglickém jazyce
old	Ditochomická proměnná označující zastaralost článku

3.4 Zpracování surových dat

Datový soubor z portálu Agris.cz (ČZU a MZČR, 2022) pro analýzu byl poskytnut v surovém formátu a bylo nutné ho dále upravit pro vyhotovení analýzy sentimentu. Prvotním problémem se zpracováváním souboru byl jeho formát. Soubor byl poskytnut ve formátu TSV, problémem bylo že při použití standartních nástrojů (IBM SPSS Modeler 18, Microsoft Excel apod.) pro zpracování tohoto typu souboru nebylo možné soubor správně syntakticky zanalyzovat.

Jako první bylo nutné vyřešit problémy s formátem souboru. Problém se stával hlavně z chyb ve sloupci „fulltext“, který samotný obsahoval v některých řádcích znak tabulátoru, a tedy odsazoval nesprávně zbytek textu na další řádek a analyzátor ho

zaměňoval za další hodnotu. Tento problém odsazení v datovém souboru byl vyřešen odebráním těchto nesprávně umístěných hodnot ze sloupce fulltext.

Datový soubor se tím pádem stal konzistentním a byl již zpracovatelný. Pokud by byla bývala data v jiném formátu než TSV (například CSV) (Shafranovich, 2005) tomuto problému by se dalo zamezit již z podstaty zvoleného separátoru.

Celá úprava datového souboru probíhá v programovacím jazyce Python s podporou datové knihovny Pandas. Průzkumová analýza byla zhotovena v software IBM SPSS Modeler 18.

Dalším krokem v přípravě dat bylo očištění souboru od nepotřebných sloupců. Data, která byla ze souboru odstraněna neměla pro analýzu žádnou informační hodnotu, není tedy nutné mít takto složitý datový soubor.

Z datového souboru byli odstraněny sloupce, které byli primárními, nebo cizími klíči, dále byli odstraněny sloupce obsahující nepotřebná data o textu. Jako nepotřebná data o textu lze označit například sloupce: *stamp, zobrazit, porizeno, url, cesta, perex*. Pro přehlednost byl ponechám odkaz na primární klíč souhrnu textů. Výsledný soubor pro další analýzu tedy obsahuje pouze data znázorněná v tabulce 4 níže.

Tabulka 4 - Datová struktura upraveného zdrojového souboru, zdroj: autor

	Popis dat			
	id_text	datum	nazev	fulltext
Data	Primární klíč odkazující na článek	Datum pořízení článku	Název článku	Celý článek

Tato forma dat není přizpůsobená pro samotnou analýzu sentimentu, ale pro spíše pro průzkumovou analýzu data. Data byla poté očištěna o nulové hodnoty ve sloupci fulltext. Z průzkumové analýzy byli zjištěny parametry jako časový rozsah zkoumaných článků a počet pozorování a průměrná délka textů.

Časový rozsah a počet pozorování byli už zmíněné výše v popisu datového souboru. Pro zjištění průměrné délky textu ve sloupci fulltext byla použita následní logika zobrazená na obrázku 11.

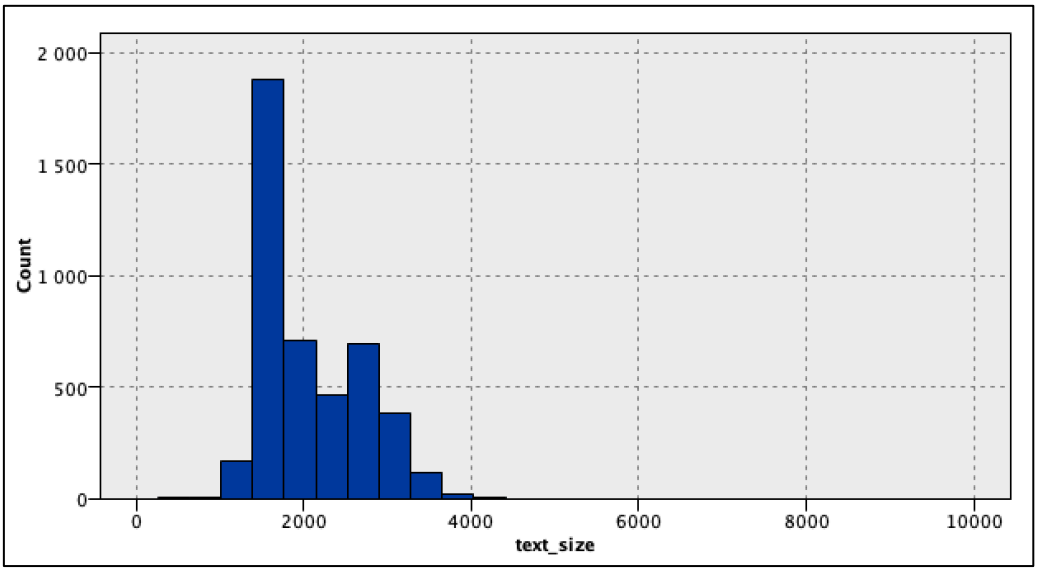
```
pandas.options.display.max_rows = 999999

mean_df = pandas.read_csv("forcut.csv", delimiter="\t")

mean_df.drop(['stamp',
              'id_zdroj',
              'id_obrazek',
              'id_jazyk',
              'id_soubor',
              'zobrazit',
              'perex',
              "nazev",
              "url",
              "cesta",
              "porizeno",
              "datum" ,
              'old',
              "id_text"], axis=1, inplace=True)

mean_df.dropna(inplace=True)

fulltext_size = []

mean_df["fulltext"].apply(lambda x: fulltext_size.append(len(x)))

mean_df["text_size"] = fulltext_size
```

Obrázek 11 - Algoritmus pro zjištění délky textů souboru, zdroj: autor

Tento algoritmus nejdříve stanoví maximální možnou délku datového rámce v Pandas a dalším kroku načte definovaný CSV soubor s definovaným oddělovačem jako dalším parametrem. Následně proběhne očištění datového rámce od sloupců, které nejsou pro výpočet algoritmu potřeba a zachová jen ty, které jsou předmětem výpočtu. V tomto případě se jedná se samotný text.

Jako poslední krok se (Kába, a další, 2012) spočtená délka uloží do nového sloupce v datovém rámci a tento nově vzniklý datový rámec se vyexportuje v CSV pro další analýzu probíhající v IBM IPSS Modeleru 18. Tímto vznikl soubor obsahující délku týdenních souhrnů textů a bylo možné provést další krok průzkumové analýzy.

Z vypočítaných délek bylo zjištěno že průměrná délka textů v tomto souboru dosahuje 2076,737 znaku. Rozložení délek v souboru na histogramu je vidět na obrázku 12 níže.



Obrázek 12 - Histogram zobrazující rozložení délek textů, zdroj: autor

Z histogramu je vidět ze více odlehlejších hodnot od průměru se vyskytuje na pravé straně grafu. To potvrzuje i kladná šikmost která nabývá hodnoty 1.344. Šikmost charakterizuje symetrii rozložení četností (Kába, a další, 2012). Kladná šikmost v tomto případě 1.344 poukazuje na to, že se většina hodnot nachází pod průměrem souboru.

Takto upravený soubor není vhodný pro použití pro analýzu sentimentu a musí být podroben dalším úpravám. Jedním z hlavních problémů souboru jsou relativně málo pozorování (4468) a příliš vysoká délka textů, která by snižovala přesnost klasifikátoru. Proto byl soubor znovu rozdělen, ale nyní na základě jednotlivých odstavců, ze kterých se skládal souhrn textů ve sloupci fulltext.

Rozdělení odstavců bylo realizováno pomocí metody nahrazení zástupných znaků pro nový řádek, kterým byli v textu odstavce rozdělené. Jednotlivé odstavce reprezentují jednotlivé články v českém jazyce.

Souhrn se tedy skládá z několika odstavců v angličtině. Tento souhrn reprezentuje sadu článků v českém jazyce. Zástupný znak byl nahrazen zvoleným substitutem (###) pro lepší parsovatelnost a čitelnost a eliminaci chyb. Pro rozdělení jednotlivých souhrnů textů na jednotlivé odstavce byla použita následující logika.

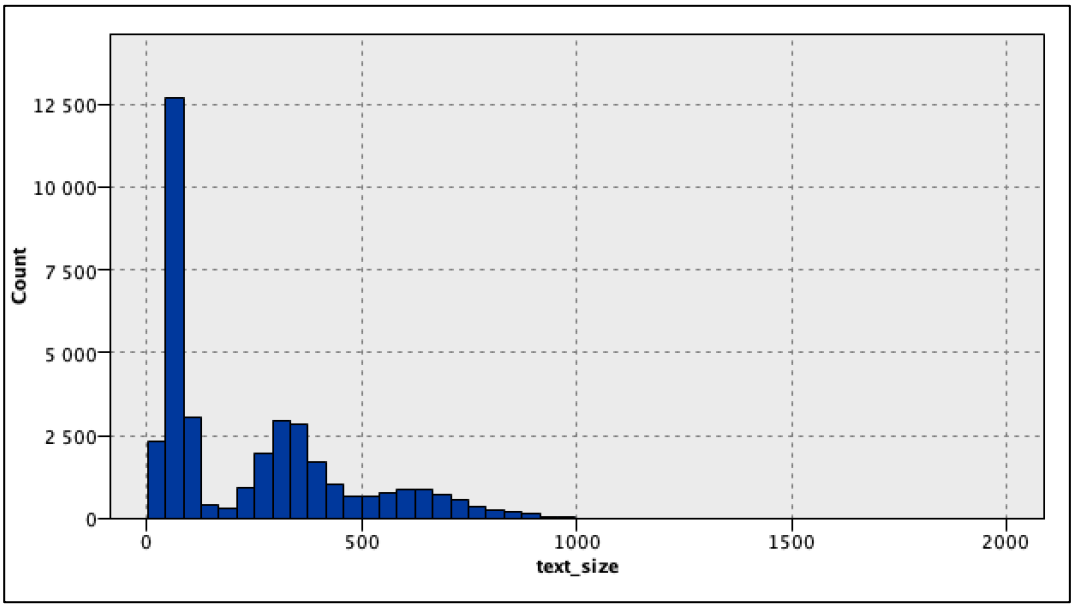
Algoritmus postupuje řádek po řádku a vždy rozdělí text podle odstavců s tím, že hodnoty z rozdělovaného řádku duplikuje na řádky nově vzniklé. Tyto duplikace dat pro analýzu sentimentu nevadí z důvodu jejich eliminace pro samotnou analýzu sentimentu.

Po této úpravě má nově vzniklý soubor 36 439 záznamů, toto číslo už je pro analýzu sentimentu mnohem více použitelné. Upravený algoritmus je zobrazen na obrázku 13.

```
pandas.options.display.max_rows = 99999

base_csv = pandas.read_csv("forcut.csv", delimiter="\t")

base_csv.drop(['stamp',
               'id_zdroj',
               'id_obrazek',
               'id_jazyk',
               'id_soubor',
               'zobrazit',
               'porizeno',
               'url',
               'cesta',
               'perex',
               'old'], axis=1, inplace=True)

base_csv.dropna(inplace=True)

def create_dataframe_by_para(data_frame):
    holder = []
    frame = data_frame

    for index, row in frame.iterrows():
        selected_row = row
        text_value = selected_row["fulltext"]
        split = text_value.split("###")

        for para in split:
            if re.search("[a-z]", para):
                row_copy = selected_row
                row_copy["fulltext"] = para.strip()
                holder.append(collections.OrderedDict(row_copy))

    return pandas.DataFrame(holder, columns=frame.columns)


clean = create_dataframe_by_para(base_csv)
```

Obrázek 13 - Algoritmus pro rozdělení jednotlivých textů na odstavce, zdroj: autor

Průměrná délka textů byla zpracována stejně jako v případě souboru po prvotní úpravě a činí 251,635 znaku se šikmostí 1,001. Zde by se dalo mluvit o téměř souměrném rozložení, soudě podle vypočítané šikmosti četností v souboru. Rozložení délek je vidět na obrázku 14 níže.



Obrázek 14 - Rozložení dat po druhé úpravě, zdroj: autor

Takto upravená data již lze použít pro analýzu sentimentu. Jednak počet pozorování je uspokojivý pro reprezentativnost výsledku a zároveň pro průměrnou délku 251,635 klasifikovaného textu se nebude snižovat přesnost klasifikátoru.

3.5 Trénovací soubor

Pro účely učení algoritmu klasifikátoru pro analýzu sentimentu byl zvolen dataset obsahující zprávy ohledně akciových trhů v Spojených státech amerických (Chaudhary, 2020). Zprávy jsou získané ze sociální sítě Twitter. Dataset obsahuje 5 791 pozorování, která jsou ohodnocena hodnotami -1 pro negativní sentiment a 1 pro pozitivní sentiment.

3.5.1 Rozdělení souboru na testovací a trénovací množiny

Při trénování algoritmů je pravděpodobnost výskytu jevu, jež nazýváme „přetrénování algoritmu“. Tento jev má za následek zkreslení přesnosti klasifikátoru, protože je zde šance, že se algoritmus naučil předpovídat výsledek na základě vstupních dat, nikoli atributů obsažených ve stupních datech.

Tomuto jevu lze zabránit pomocí metody data splitting. Tento postup funguje na principu rozdělování datasetu na trénovací a testovací množinu. Někdy se zde používá také třetí, validační, množina. V tomto případě použita není. Trénování algoritmů je již popsáno v předchozích kapitolách.

Pro účely analýzy sentimentu zpravodajských textů ze zemědělství vzhledem k vybranému testovacímu souboru a nejlepším kombinacím podle Muraina (Muraina, 2022) rozdělení do tří testovacích skupin. Jedná se o poměry 80:20, 50:50 a 65:35. Soubor lze takto rozdělovat bez dalších úprav, protože se jedná o vyvážený dataset. Vyvážený je takový dataset, který ani v jedné kategorii nemá o patrnou část více ohodnocení než v dalších.

Rozdělení souborů může probíhat pomocí implementace v knihovně Pandas v jazyce Python. Níže je použitá logika pro rozdělení testovacího datasetu v poměru 80:20, který zajišťuje parametr metody sample *frac*. Je zde použit parametr *random_state=200,* který zajišťuje přepoužitelnost generování pro další modely. Příklad rozdělení souboru je zobrazen na obrázku 15 níže.

```
import pandas

pandas.options.display.max_rows = 99999

stock_df = pandas.read_csv("../dp_data/stock_data.csv", delimiter=",")

stock_df.to_csv("stock_data_semicol.csv", index=False, sep=";")

train_df = stock_df.sample(frac=0.8, random_state=200)

test_df = stock_df.drop(train_df.index)
```

Obrázek 15 - Implementace rozdělení souborů na testovací a učicí množiny, zdroj: autor

Samotné rozdělení datových souborů (testovací data, zkoumané zpravodajské texty ze zemědělství) jsou rozděleny pomocí implementace v knihovně SciKit-Learn. Výše uvedený postup pro Pandas je také validní, protože SciKit-Learn je schopna na vstupních argumentech akceptovat Pandas datové rámce.

K tomuto kroku se autor přiklonil pouze z hlediska udržitelnosti a přehlednosti kódu v jazyce Python. Níže na obrázku je vidět implementace rozdělení testovacího souboru na testovací a verifikační množinu pomocí SciKit-Learn knihovny.

Algoritmus je na první pohled o velkou část jednodušší, než tomu tak je v případě implementace v samotném Pandas. V první části jsou naimportovány všechny potřebné metody ze SciKit-Learn a Pandas knihovna.

V dalším kroku je načten trénovací, již tokenizované a metodou BagOfWords optimalizovaný dokument obsahující vektory a k nim příslušné sentimenty. V posledním

kroku algoritmu je volána metoda *train_test_split* z knihovny SciKit-Learn. Tato metoda má na výstupu čtyři pole, které jsou uloženy do proměnných *sent_train, sent_test, bow_train a bow_test* (Buitinck, a další, 2013). První dvě proměnné uchovávají hodnoty sentimentu testovacího datasetu, druhé dvě potom nesou vektory vzniklé pomocí metody BagOfWords. Poměr pro trénovací množinu zůstává stejně jako v případě implementace v Pandas stejný, tedy 80 % pro trénovací množinu a 20 % pro testovací. Stejně tak byl použit i stejný náhodný stav (200) jako v případě Pandas implementace, jak je znázorněno na obrázku 16.

```
import pandas as pd
from sklearn.model_selection import train_test_split

training_data = pd.read_csv("./bog_df_training.csv", delimiter=",")

sentiment = training_data["Sentiment-XX"]

training_data.drop(columns=["Sentiment--XX"])

bow_train, bow_test, sent_train, sent_test = train_test_split(training_data,
sentiment, test_size=0.2, random_state=200)
```

Obrázek 16 - Implementace rozdělení souborů na trénovací a testovací množinu v knihovně SciKit-Learn, zdroj: autor

3.6 Výběr klasifikátorů

Analýza sentimentu probíhá pomocí implementace knihovny NLTK pro jazyk Python a knihovny SciKit-Learn. Knihovna NLTK disponuje velkým množstvím funkcí, které jsou použity hlavně pro tokenizační zpracování textu. Tato metoda popsána v teoretické části nám pomůže lépe připravit klasifikátor na samotnou klasifikaci textu. Jako další metoda je na data aplikována BagOfWords, která generuje z textových tokenů vektory. Fungování metody je vysvětleno v kapitole 2.6.1.

Text v klasické podobě je pro klasifikátory totiž hůře zpracovatelný a je tedy lepší text převést do podoby, která je lepší pro výpočet. Knihovna SciKit-Learn je konkurentem velmi známé knihovny Tensorflow, která nabízí podobné implementace jako SciKit-Learn, důvod pro použití SciKit-Learn namísto Tensorflow je přehlednější a rychlejší implementace potřebných algoritmů.

Jako klasifikátor pro analýzu sentimentu zpravodajských textů ze zemědělství byla zvolena logistická regrese, která se hodí pro klasifikační úlohy tohoto typu. Logistická regrese má schopnost predikovat sledovanou veličinu s určitou pravděpodobností a lze tedy

i říct s jakou mírou jistoty klasifikátor určil sentiment pro daný text. Tento algoritmus a jeho implementace je výhodná v rychlosti s jakou lze takový klasifikační model vyvinout a otestovat.

Pro analýzu sentimentu byli ještě zvažované další modely, konkrétně Bayesiánská regrese a náhodné rozhodovací stromy. Tyto modely nedosahovali takové přesnosti, aby je bylo vůbec možné zařadit jako další kandidáty na klasifikátor sentimentu. Model Bayesiánské regrese dosahoval skóre s = 0,38 a náhodné rozhodovací stromy s = 0,68 oproti 0,79 (zaokrouhleno na dvě desetinná místa). U Bayesiánské regrese bychom kvůli s <0,5 mohli říci, že model dosahuje přesnosti 1-s tedy 0,62, ale to je stále nízká hodnota na to, aby byl model použit.

Modely s takto nízkým metrikou skóre nebyli ani dále analyzovány na další ukazatele přesnosti jednoduše proto, že to v porovnání s modelem logistické regrese nemělo význam.

3.7 Úprava datových souborů pro použití v klasifikátoru

Po rozdělení trénovacího souboru na trénovací a verifikační množinu následuje tokenizace obou části. Tento krok lze udělat i před samotným rozdělením souborů, ale na pořadí těchto dvou kroků nezáleží.

Tokenizace je implementována pomocí knihovny NLTK, která obsahuje již hotový algoritmus pro vytvoření tokenů ze zkoumaných/trénovacích textů. Důležitou věcí, kterou je potřeba dodržet je řazení originálních textů v poli ku řazení vzniklých tokenů, pro pozdější zpětné rozklíčování a interpretaci výsledků. Na obrázku 17 je zobrazen algoritmus pro tokenizace textů testovacího souboru.

```
import pandas
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

pandas.options.display.max_rows = 6000

stopwrds = set(stopwords.words("english"))

whole_df = pandas.read_csv("./stock_data_semicol.csv", delimiter=";")

tokens = []

whole_df["Text"].apply(lambda x: tokens.append(word_tokenize(x)))

post_stopwords = []

for token in tokens:
    post_stopwords.append(list(filter(lambda x: x.lower() not in stopwrds, token)))
```

Obrázek 17 - Algoritmus pro tokenizaci textu, zdroj: autor

Algoritmus prvně naimportuje všechny potřebné knihovny, které potřebuje pro následující operace. V tomto případě se jedná o Pandas pro práci s datovým rámcem, do kterého je načten testovací dataset ve formě souboru CSV. Následně je nastavena maximální délka zobrazovaných řádků, tato opera je čistě pro přehlednost při vyváření algoritmu, ale není nutná pro samotnou funkčnost algoritmu.

Jako další jsou načteny *stopwords* z knihovny NLTK. Metoda je uvedena s parametrem *english* ten říká, že jako výstupní hodnota metody jsou slova v anglickém jazyce. Stopwords jsou následně uložena do datové struktury typu Set, pro lepší použitelnost dat. Set obsahuje anglická slova, která nejsou pro samotné zpracování přirozeného jazyka důležitá. Jedná se například o spojky, předložky apod. Na obrázku 18 níže je výčet použitých *stopwords*.

```
{"she's", 'having', 'can', "hasn't", 'what', 'whom', 'you', 'am', 'she', 'how', 'very', 'needn', 'too', 'there',
's', 'aren', 'yourself', 'your', 'its', 'herself', 'below', "you'd", 'their', 'by', 'few', 'only', 'he', 'were',
'the', 'our', "mightn't", 'should', 'itself', 'which', 'because', "you'll", 'in', 'some', 'll', "didn't", 'ma',
'wasn', 'any', 'it', 'again', 'ain', 'couldn', 'doesn', "doesn't", "aren't", "couldn't", 'm', 'out', "shouldn't",
'why', "don't", 'a', 'then', 'each', 'after', 'didn', 'his', 've', 'under', 'weren', 'between', 'that', 'nor',
'this', 'own', 'yourselves', "that'll", 'about', 'was', 'mightn', 'more', 're', 'or', 'y', 'yours', "you've",
'they', 'up', 'both', 'doing', 'same', 'be', 'at', 'o', "wouldn't", "haven't", 'most', 'did', "weren't", 'him',
'an', 'these', 'do', 'above', 'won', 'during', 'so', "isn't", 'will', 'themselves', 'who', 'of', 'is', 'into',
"mustn't", 'through', 'been', 't', 'myself', 'down', 'where', 'haven', 'here', 'when', 'shan', 'shouldn', 'don',
'as', "hadn't", 'me', 'her', 'are', 'being', "wasn't", 'isn', "shan't", 'but', 'once', 'them', "you're", 'and',
'now', "won't", 'further', 'other', 'no', 'against', 'than', 'has', 'himself', 'while', 'my', 'd', 'hadn', 'from',
"it's", 'i', 'until', 'not', 'those', 'such', 'had', 'theirs', 'wouldn', 'all', 'over', 'for', 'hers', 'does', 'on',
'if', 'we', 'to', 'just', 'mustn', 'off', 'with', 'ourselves', "should've", 'before', "needn't", 'have', 'ours',
'hasn'}
```

Obrázek 18 - Výstup metody stopwords, zdroj: autor

Takovéto očištění dat je výhodné, protože jednak zmenšuje celkovou velikost zkoumaných/testovacích dat a zároveň pomáhá optimalizovat data pro následující krok, který data vektorizuje například metodou BagOfWords.

V dalším kroku algoritmu jsou textové hodnoty, na něž je pomocí lambda funkce volána funkce, která zajišťuje tokenizace. Tokenizace je rozdělení jednotlivých slov na samostatné hodnoty, jak je již popsáno v teoretické části. Tyto hodnoty jsou překopírovány do nového pole, které je následně profiltrováno pomocí lambda funkce, která hledá shodu mezi prvkem v poli a hodnotami v setu stopwords.

Pokud shodu najde hodnota je vynechána, pokud platí opak potom je položka uložena do nového pole, které obsahuje již pouze očištěné tokenizované textové hodnoty. Na obrázku 19 jsou zobrazeny k porovnání hodnoty před (první řádek) a po optimalizaci (druhý řádek).

```
['Kickers', 'on', 'my', 'watchlist', 'XIDE', 'TIT', 'SOQ', 'PNK', 'CPW', 'BPZ', 'AJ', 'trade', 'method', '1', 'or',
'method', '2', ',', 'see', 'prev', 'posts']

['Kickers', 'watchlist', 'XIDE', 'TIT', 'SOQ', 'PNK', 'CPW', 'BPZ', 'AJ', 'trade', 'method', '1', 'method', '2',
',', 'see', 'prev', 'posts']
```

Obrázek 19 - Ukázka optimalizace pomocí stopwords, zdroj: autor

3.7.1 Vektorizace dat

Pro vektorizaci dat byla vybrána metoda BagOfWords, pro svoji aritmetickou nenáročnost a jednoduchost implementace. Implementace metody BagOfWords byla použita z knihovny SciKit-Learn, která má pro tuto metodu již předpřipravené rozhraní a není ji tedy složité naimplementovat. Výstupem metody je matice, která má, jak již bylo popsáno v teoretické části (tabulka 1) nulové hodnoty tam kde není výskyt určitého slova a hodnotu jedna na pozici kde byl výskyt úspěšně nalezen. S tím že vyhledávaná slova se neopakují. Implementace vektorizace testovacího souboru je zobrazena na obrázku 20.

```
import pandas
import sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(stop_words="english")

bog = vectorizer.fit_transform(test_df)

bog_df = pandas.DataFrame(bog.toarray(), columns=vectorizer.get_feature_names_out())
```

Obrázek 20 - Vektorizace dat pomocí SciKit-Learn, zdroj: autor

Jako první zavoláme konstruktor třídy CountVectorizer s parametrem upřesňující *stopwords* popsané dříve, která je naimportována z knihovny SciKit-Learn (Buitinck, a další, 2013) a inicializujeme její instanci do proměnné vectorizer. Následně na instanci třídy CountVectorizer voláme metodu *fit_transform*, která má na vstupních parametrech dokumenty (v tomto případě Pandas datový rámec), který vektorizuje. Výstup je znovu uložen jako Pandas datový rámec pro lepší použitelnost dále.

Datový rámec je posléze vyexportován do CSV souboru pro jednodušší manipulaci vzhledem k implantačním nárokům SciKit-Learn modelu.

3.8 Trénování modelu pro vyhodnocení sentimentu

Po rozdělení dat na trénovací a testovací množiny se může přistoupit k samotnému trénovaní modelu. Všechny modely použity v této analýze jsou typu učení pod dozorem. Je tedy nutné předat algoritmu ve cvičící fázi dva datové soubory, a to právě trénovací množinu na které se algoritmus učí předpovídat nezávislou proměnou v případě regresního modelu, nebo přidělovat hodnoty do předem známých kategorií v případě klasifikačního modelu.

Algoritmus pro implementaci učení algoritmu (v tomto případě logistické regrese) je zobrazen na obrázku 21.

```
# create Model
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
# display model score
model_score = logreg.score(x_test, y_test)
# Confusion matrix
test_prediction = logreg.predict(x_test)
test_prediction_proba = logreg.predict_proba(x_test)
```

Obrázek 21 - Algoritmus pro trénování modelu Logistické regrese, zdroj: autor

Část algoritmus na obrázku výše funguje následovně: Jako první se do proměnné *logreg* inicializuje instance třídy *LogisticRegression*, která je součástí knihovny SciKit-Learn. Všechny modely v této knihovně jsou implementované jako třídy, manipulace s nimi

je tedy velmi intuitivní. Všechny potřebné hodnoty jsou dosažitelné právě skrze vytvořenou instanci třídy modelu logistické regrese (Buitinck, a další, 2013).

V dalším kroku algoritmu se na vytvořené instanci *logreg* volá metoda *fit*. Tato metoda je určená pro trénování algoritmu. Metoda *fit* přijímá celkem tři parametry. Prvním argumentem je trénovací soubor, který obsahuje vektorizované textové tokeny z trénovacího datového souboru. Druhý argument obsahuje trénovací část ohodnoceného sentimentu, která je součástí trénovacího datového souboru (sentiment byl od textu oddělen právě kvůli následnému učení algoritmu).

Při učení algoritmu se objevil problém, který byl způsoben nedostatečnou homogenitou trénovacího souboru a datového souboru, který obsahuje vektorizované textové toky zpravodajských dat ze zemědělství. Problém spočíval v délce vektorů, které má na výstupu funkce BagOfWords. Přesněji šlo o to, že trénovací soubor dosahoval maximálních délek vektorů okolo 9 300 hodnot, kdežto datový soubor se zpravodajskými texty měl maximální délky na hranici 36 000.

Tato nesouměrnost způsobila problém, který znemožňoval vyhodnocení zpravodajských textů ze zemědělství, protože model jednoduše takové kombinace, které přicházeli na vstup modelu neznal a nemohl na jejich základně nijak predikovat nezávislou proměnou, tedy sentiment textu. Řešení problému naštěstí poskytuje vektorizační metoda BagOfWords. V metodě lze způsob, jakým skládá výsledné vektory, resp. omezit jejich délku na konstantní délku pro několik vektorizovaných na sobě nezávislých souborů. Konstrukce takových vektorů s omezenou maximální délkou funguje na výběru pouze určitého počtu výskytů slov, kde kritériem pro výběr je četnost výskytu těchto slov v datovém souboru, který podléhá vektorizaci. Toto řešení může způsobit ztrátu informace, pokud by se vektor určil moc krátký. Z tohoto důvodů byl vektor omeze na 9 000 hodnot. Tímto jsme dosáhli relativně vysokého zachování informace, kterou text nese a zároveň umožní použít trénovací data a data zpravodajských textů ze zemědělství. Na obrázku 22 níže je zobrazená metoda, která tento problém řeší pro nezávislé datové soubory.

```
def vectorize(df, label, col):
    print("Starting vectorize data frame" + " : " + label)

    vectorizer = CountVectorizer(stop_words="english", max_features=9000)

    vec = vectorizer.fit_transform(df[col])

    print("Creating vectorized data frame from df" + " : " + label)

    return pd.DataFrame(vec.toarray())
```

Obrázek 22 - Metoda pro fixní délku vektorů z metody BagOfWords, zdroj: autor

Po natrénování modelu pomocí metody *fit* získáme model, který by měl na základně získané přesnosti mít schopnost predikovat nezávislé proměnné na základně jemu neznámých dat.

V posledním kroku algoritmu je na nacvičeném algoritmu spuštěná metoda *predict*, která jak již její název naznačuje spustí na naučeném modelu predikci dosud neznámých dat. V tom to případě se jedná o testovací množinu trénovacího datového souboru. Výsledkem metody je datový typ *numpy_array*, který obsahuje predikované nezávislé proměnné. Pro získání pravděpodobnostní, s jakou model predikoval nezávislé proměnné je dále spuštěna metoda *predict_proba*, která má výstupu pravděpodobnostní odhad, se kterým se model rozhodl ohodnotit vektor buďto kladným nebo záporným sentimentem. Tento výsledek je dále nutný pro výpočet přesnosti a celkové evaluace modelu logistické regrese pomocí metod popsaných v následující kapitole.

3.9 Kvalita modelu logistické regrese

Rozhodnutí, který model bude použitý pro následnou analýzu sentimentu zpravodajských dat ze zemědělství závisí především na přesnosti vytrénovaných modelů, které jsou implementovány v rámci této práce. Kritéria, dle kterých lze měřit přesnost modelu jsou popsány v teoretické části v kapitole 2.9.

Tato kapitola se zaměřuje na analýzu přesnosti modelu logistické regrese, která byla natrénovaná na trénovacím datovém souboru. Prvním naivním ukazatelem přesnosti je takzvané skóre modelu. Tento parametr je vypočítán jako průměrná přesnost pro model, a proto je považována pouze za orientační metriku, která by nikdy neměla sama o sobě vypovídat o kvalitě modelu (Buitinck, a další, 2013).

Tabulka 5 - Matice záměn modelu logistické regrese, zdroj: autor

Skutečné hodnoty	Predikované hodnoty	
	0	1
0	262	166
1	85	646

Jednou z používaných způsobů určení přesnost/kvality modelu je matice záměn (obrázek 23). Tato matice obsahuje ve sloupcích predikované hodnoty a v řádcích skutečné. V případě analýzy sentimentu je matice záměn o velikosti 2x2, protože máme pouze ditochomickou proměnou sentimentu a to kladný (1) a záporný (-1). Na obrázku výše je zobrazena matice záměn vygenerovaná z modelu logistické regrese, která predikovala testovací množinu z testovacího datového souboru a byla ověřena proti testovací množině sentimentu z testovacího datového souboru.

Z těchto hodnot lze vyčíst několik metrik, které popisují jednotlivé silné a slabé stránky naučeného modelu. Řadí se mezi ně specifita, senzitivita a přesnost. Další metrika může být například F1 míra, která je harmonickým průměrem součtu senzitivity a specificity. Níže jsou ohodnocené míry kvality modelu popsané výše.

$$Specificita = \frac{FP}{TN + FP} = \frac{166}{428} = 0{,}387$$

Rovnice 13 - Výpočet specificity modelu logistické regrese, zdroj: autor

$$Senzitivita = \frac{TP}{FN + TP} = \frac{646}{731} = 0{,}884$$

Rovnice 14 - Výpočet senzitivity modelu logistické regrese, zdroj: autor

$$Přesnost = \frac{TP}{FP + TP} = \frac{646}{812} = 0{,}795$$

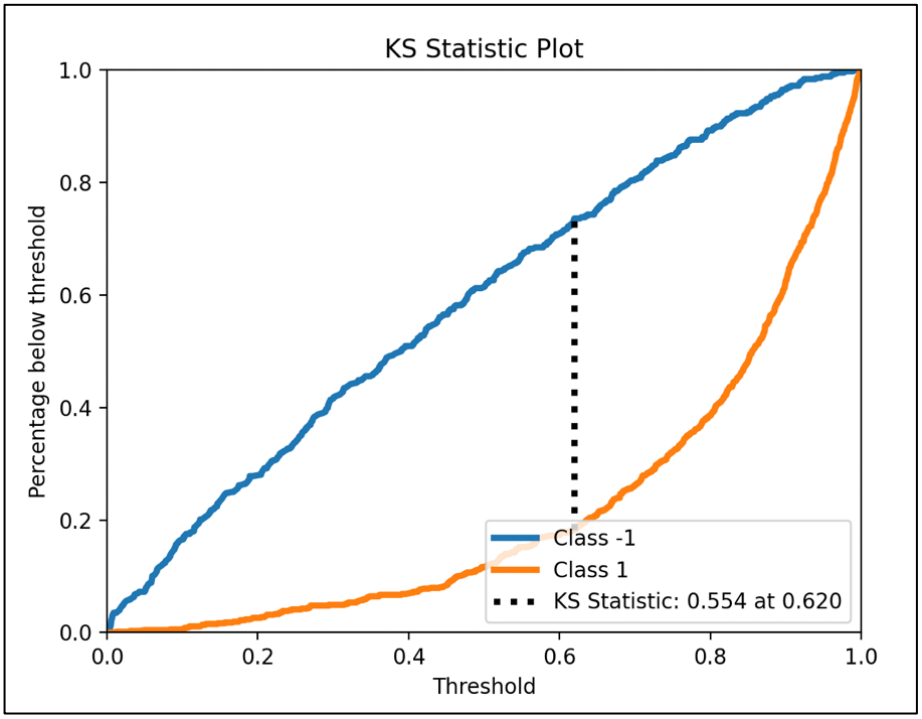
Rovnice 15 - Výpočet přesnosti modelu logistické regrese, zdroj: autor

Z naměřených hodnot lze říct následující. Specificita s hodnotou 0,387 ukazuje na relativně nízkou úspěšnost klasifikace negativních případů. Oproti tomu Senzitivita, tedy schopnost modelu klasifikovat pozitivní případy je 0,884. To značí že model náchylný na chyby typu FP (false positive) tedy že predikce modelu označil sentiment textu za kladný, ale reálná hodnota sentimentu byla záporná.

Přesnost 0,795 která se odlišuje od naivní přesnosti, která je popisována v předchozí kapitole, kterou lze získat jako výstup metody *score* je k analýze sentimentu dostatečná. Model je podle této hodnoty schopen ohodnotit 80 % případů validně. Přesnost vypočítaná z hodnot matice záměn se odlišuje od naivní o 0,012.

Jako další pomocné metody pro vyhodnocení přesnosti modelu jsou vybrány některé typy grafů, ze kterých lze vyčíst určité vlastnosti modelu. Jako grafy jsou vybrány Kolmogorov-Smirnov graf, ROC graf, Lift graf, a graf kumulativního přínosu.

Na obrázku 23 níže je zobrazen Kolmogorov-Smirnov graf na kterém je významných ukazatelem Kolmogorov-Smirnov koeficient.



Obrázek 23 - Kolmogorov-Smirnov graf modelu logistické regrese, zdroj: autor

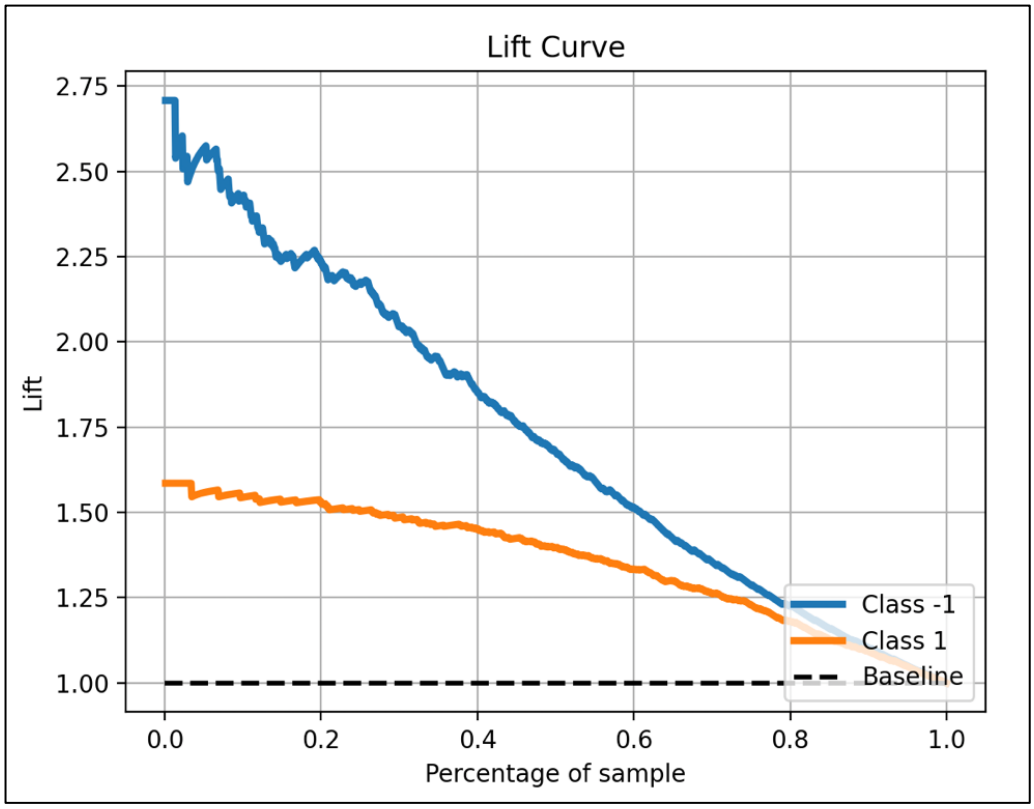
Kolmogorov-Smirnov graf porovnává dvě distribuce (pro každou třídu). Test se pohybuje na intervalu (0,1) na ose X i Y. Hlavním parametrem, který vypovídá o kvalitě modelu je KS koeficient, který určitě schopnost modelu diverzifikovat data mezi třídami. Hodnota KS koeficientu se může pohybovat na intervalu (0,1) kde hodnota 0 značí naprostou neschopnost modelu rozlišit hodnoty mezi třídami a na druhé straně hodnota 1 říká že model nabývá maximální schopnosti diverzifikace mezi třídami (Drezner, a další, 2010).

Hodnota naměřená na modelu logistické regrese je 0,554 a nachází se na ose X v hodně 0,620. Hlavním vodítkem pro určení přesnosti je samotná hodnota KS koeficientu,

protože to, kde se na ose X koeficient nachází naznačuje pouze průběh distribuce. Hodnota koeficientu 0,554 je dostačující pro potvrzení jednoho argumentu hypotézy že model je dostatečně kvalitní. Běžně se u modelů toho typu kvalitní KS koeficient pohybuje mezi hodnotami 0,5 až 0,7. Takové hodnoty lze považovat za dostatečné. (Drezner, a další, 2010)

Dalším grafem, který má ověřit kvalitu modelu je graf typu Lift. Graf typu Lift pracuje s pravděpodobností, s jakou se model logistické regrese rozhoduje přiřadit vektorizovanému textovému tokenu hodnotu -1 nebo 1.

Na obrázku 24 níže je zobrazen graf typu Lift pro model logistické regrese.



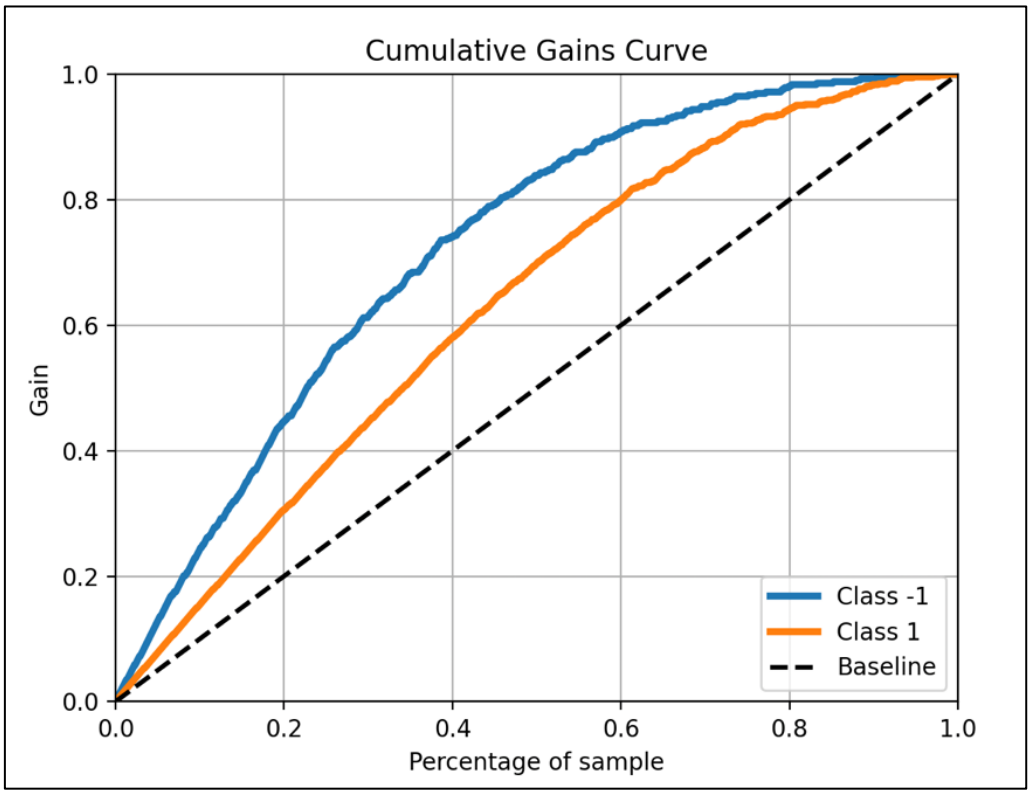
Obrázek 24 - Graf typu Lift medelu logistické regrese, zdroj: autor

Z grafu je patrné (třída pro -1) že s přibývajícími daty ztrácíme schopnost správně klasifikovat negativní případy, avšak křivka není tak strmá, aby tento průběh křivky znamenal problém pro samotný model. Tento argument je zároveň možné podpořit nízkou specificitou modelu vypočítanou z matice záměn. Avšak je nepopiratelné že model tímto deficitem trpí (Piatetsky-Shapiro, a další, 1999).

Pokud se podíváme na druhou křivku (pro třídu 1) vidíme že nemá zdaleka tak strmý průběh jako první již popsaná křivka. Z toho lze usuzovat, že schopnost modelu zvládat vyhodnocovat pozitivní případy je stabilnější v celém průběhu vyhodnocování.

Toto tvrzení lze podložit stejně jako v případě první třídy vypočítanou měrou z matice záměn. Hodnota senzitivity modelu dosahuje z vypočítaných hodnot 0,844. To naznačuje vysokou schopnost ohodnocovat správně pozitivní případy. To je dále potvrzeno i hodnotami z grafu Lift (Piatetsky-Shapiro, a další, 1999).

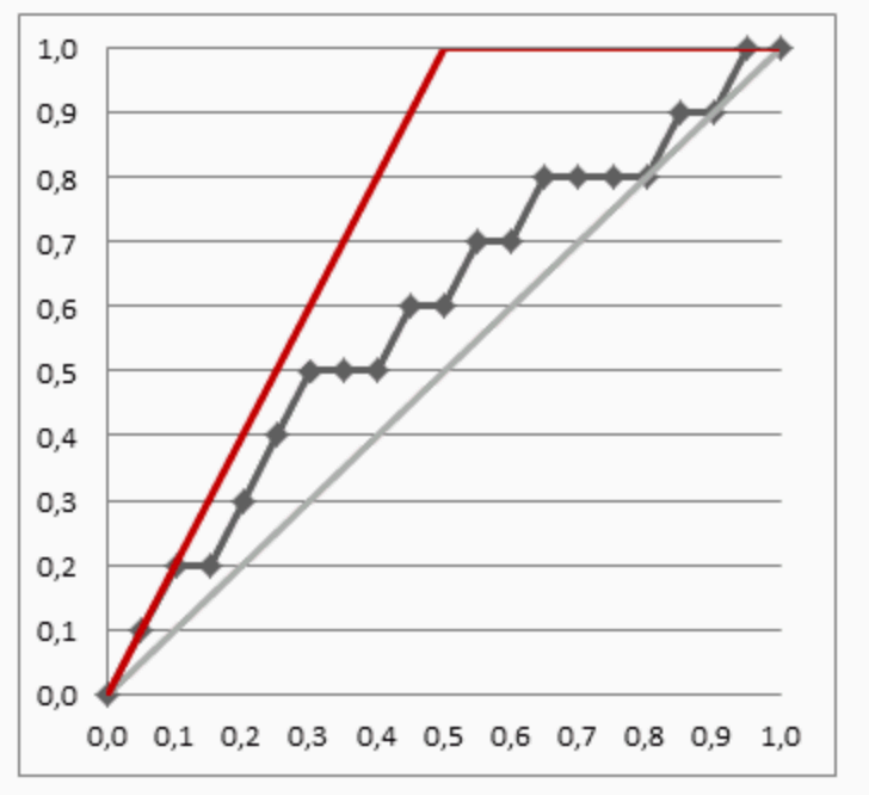
Další metrikou, díky které můžeme vyvodit přesnost/kvalitu modelu je CGC (Cumulative Gain Curve) graf. Tento graf sleduje znovu dvě třídy modelu a zobrazuje jejich rozložení ve dvourozměrném prostoru. Tento graf se často právě využívá pro modely, které predikují ditochomickou proměnou. Graf pro natrénovaný model logisticé regrese je zobrazen na obrázku 25.



Obrázek 25 - Graf pro zobrazení CGC modelu logistické regrese, zdroj: autor

Na grafu sledujeme narůstající přínos v poměru k procentuálnímu nárůstu dat, které jsou na vstupu grafu. Ideální křivka roste lineárně k 1 ose Y (Gain) a po dosažení 1 zůstává konstantní po celou dobu zbývajícího růstu procent dat. (Brandenburger, a další, 2009)

Ideální křivka je zobrazena na ukázkovém grafu na obrázku 27.



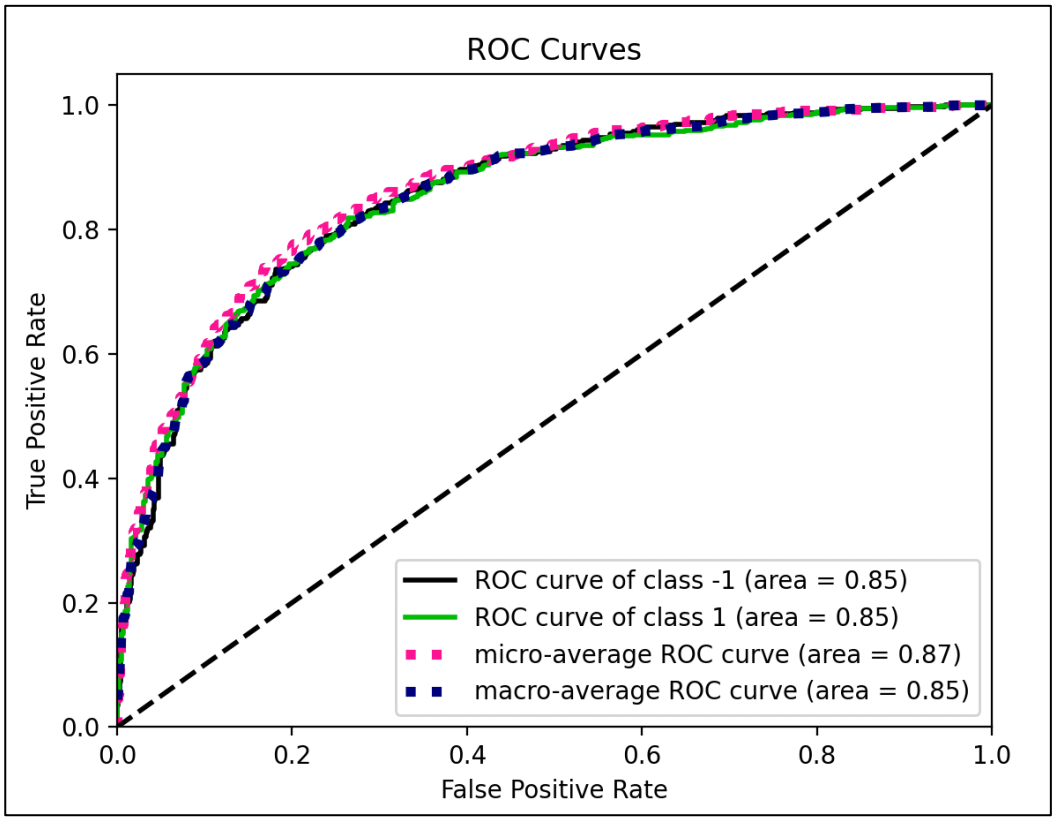
Obrázek 26 - Ideální křivka pro CGC graf, zdroj: (ML Wiki, 2014)

Pokud porovnáme ideální křivku z obrázku 26 s výslednými křivkami z grafu na obrázku 25 můžeme zjistit že křivky z obrázku 25 se přibližují ideálnímu průběhu grafu na obrázku 26.

Z tohoto zjištění lze tedy vyvodit závěr že natrénovaný model má dostatečný přínos (gain) dle procentuálního množství dat.

Posledním ukazatelem kvality modelu logistické regrese je ROC graf (obrázek 27). Tento graf popisuje závislost na senzitivitě grafu oproti 1-specifičnosti.

Každý bod na grafu vyjadřuje scóre modelu, ale to se do grafu nevynáší.



Obrázek 27 - ROC graf modelu logistické regrese, zdroj: autor

Kvalitu modelu logistické regrese na tomto typu grafu odhadujeme podle Gini koefientu. Pokud bychom křivky grafu zrcadlili podle diagonály a následně spočítali plochu mezi těmito křivkami dostali bychom ve výsledku tzv. Gini koefient, který lze také spočíst jako dvojnásobek plochy mezi křivkou a diagonálou.

Tento koeficient popisuje míru diverzifikace dat. V případě analyzovaného modelu logistické regrese dosahuje velikost plochy pro první třídu (-1) 0,85. Pro druhou třídu modelu (1) je tato plocha shodně 0,85. Tyto data nám říkají že model je schopen na dostačující úrovni diverzifikovat data. Je zde ale třeba počítat s nižší specifitou.

3.10 Klasifikace sentimentu zpravodajských dat ze zemědělství

Data získaná z portálu Agris.cz (ČZU a MZČR, 2022) byla podle dříve popsaných postupů upravena do podoby, která je vhodná pro vyhodnocení sentimentu. Podoba dat, která je použita pro kvalifikaci sentimentu zpravodajských dat ze zemědělství je tedy souborem vektorizovaných textových tokenů, která jsou vložena na vstup modelu logistické regrese.

Celý algoritmus výpočtu sentimentu zahrnuje také učení algoritmu, a tedy také zpracování testovacích dat. Toto je teoreticky zbytný krok, protože algoritmus by mohl být natrénován nezávisle na logice vyhodnocení zpravodajských dat ze zemědělství, ale vzhledem k nenáročnosti algoritmu to není nutné.

Výsledky predikce jsou vyexportované do CSV souboru pro další analýzu a spojení například dat ohledně časového bodu publikování článku na webových stránkách Agris.cz (ČZU a MZČR, 2022).

3.10.1 Časová náročnost klasifikačního algoritmu

Pokud prověříme nutnou dobu, kterou počítač potřebuje k celkovému zpracování algoritmu, které zahrnuje tokenizaci obou souborů (trénovacího, zpravodajských dat ze zemědělství), natrénování algoritmu, ověření přesnosti algoritmu pomocí spočítání matice záměn a následné vytvoření predikce s exportem vyhodnoceného sentimentu pro 36 438 záznamů dostaneme se podle programu *time* (Linux manual page, 2019) hodnotu okolo 19 vteřin. Program byl spočítán na procesoru Apple M1, které disponuje maximální frekvencí 3,2 GHz na jádro.

3.11 Vyhodnocení sentimentu

Vyhodnocení sentimentu zpravodajských dat ze zemědělství bylo spočítáno pomocí natrénovaného modelu logistické regrese. Model je po předešlé kvalitativní analýze shledán dostatečně kvalitním pro použití na analyzovaných zpravodajských textech ze zemědělství.

Výsledky predikují sentiment 36 438 záznamů. Z toho bylo ohodnoceno 31 451 jako kladný sentiment a 4988 jako záporný sentiment. Zde se v potaz musí brát nízká specifita modelu, reálné číslo tedy může o něco vyšší u hodnot ohodnocených záporným sentimentem, řešením je tzv. „boosting“ trénovacího souboru abychom dosáhli vyšší specifity modelu, ale pro tento model, resp. pro jeho přesnost to není nutné.

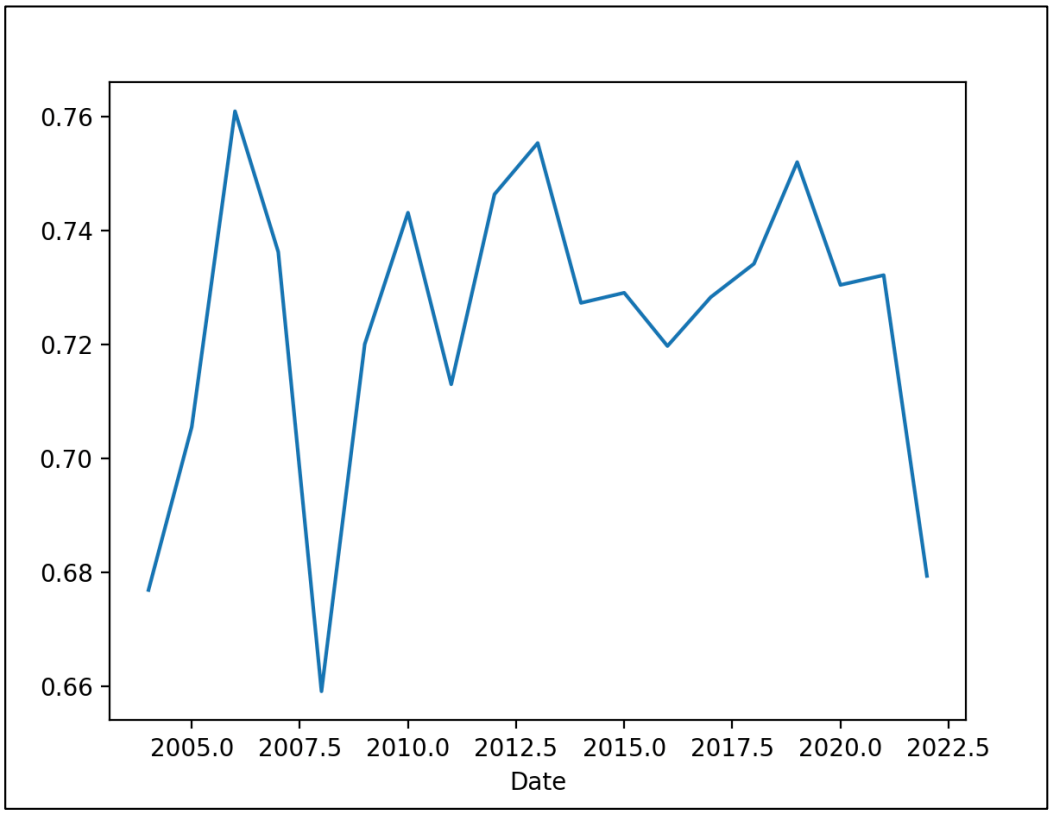
Tento výsledek modelu říká že většina zkoumaných zpravodajských dat ze zemědělství, které byli podrobené analýze jsou zabarveny kladně. Tedy jejich vyznění, jak je popsáno v kapitole 2.1 o analýze sentimentu, je od autora textu směrem ke čtenáři kladné.

Vzhledem k ditochomické povaze dat, která stanovuje pouze kladný a záporný sentiment (neuvažujeme neutrální sentiment) je to dobrá zpráva, vzhledem k tomu že texty

jsou podle několika studií uvedené v teoretické části práce schopny ovlivňovat emoce čtenáře textů. V našem případě je tedy lepší vyšší počet kladně ohodnocených dat, než kdyby tomu bylo obráceně.

Jako další vyhodnocovací metrika byla zvolena metrika vývoje sentimentu v čase, resp. v čase publikací analyzovaných dat. K tomuto kroku je nutné zkombinovat prvotní data s vyhodnoceným sentimentem zpravodajských dat ze zemědělství. Jelikož byli jednotlivé části dat v průběhu procesu úpravy dat exportována jako CSV souboru, nebo jako Pandas datového rámce jedná se o jednoduché spojení dvou souborů a náčrtu jeho hodnot do grafu.

Pro zobrazení v čase byla provedena agregace dat z důvodu vysokého počtu hodnot. Počet hodnot přesahující 36 000 záznamů by znamenal nepřehlednost grafu a jeho nepoužitelnost. Jako jednotka byl tedy použit průměrný sentiment za jeden rok. Graf průměrného sentimentu za rok je zobrazen na obrázku níže.



Obrázek 28 - Zobrazení vývoje sentimentu zpravodajských textů ze zemědělství v rocích, zdroj: autor, Pozn: osa y = průměrný sentiment, osa x = čas.

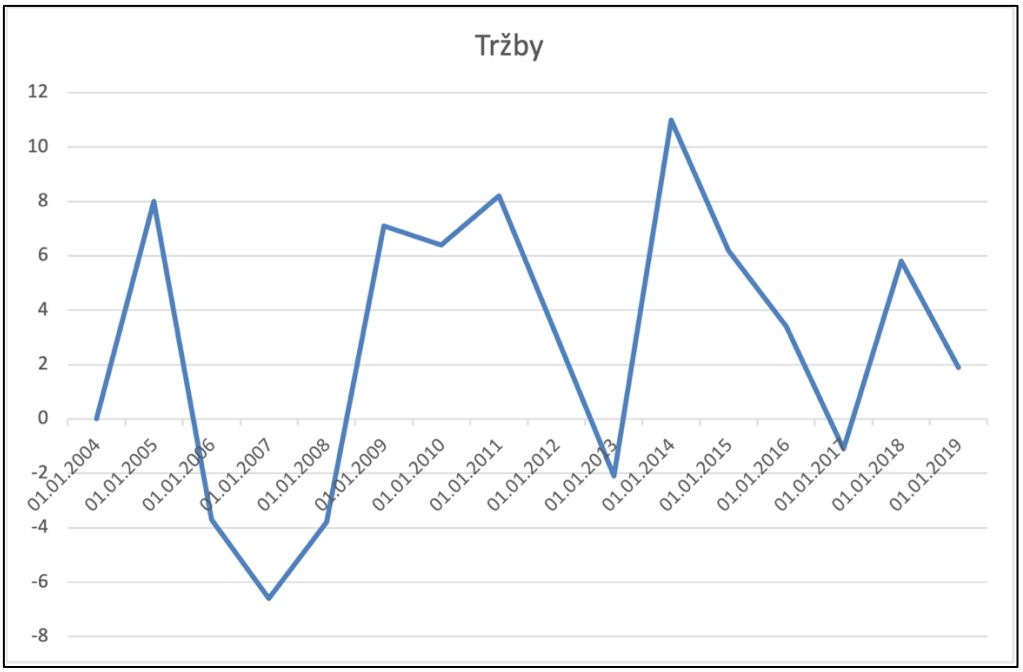
Z grafu na obrázku 28 jsou vidět propady okolo roku 2008 a na konci roku 2021. Tyto propady mohou být způsobeny například ekonomickými krizemi (2007-2008) anebo začátkem válečného napětí ze strany Ruské federace k Ukrajině, které eskalovalo koncem

roku 2021 a na začátku roku 2022 přerostlo v otevřenou válečný agresi ze strany Ruské federace. Tyto události mohou mít vliv na sentiment zpravodajských textů, protože se na obecné úrovni zhoršila nálada ve společnosti. Tyto dvě události zmíněné výše však nemusí plně souviset s propadem sentimentu.

4 Diskuse

4.1 Korelace s ekonomickými daty

Na obrázku 29 je zobrazen graf tržeb ze zemědělství (CZ-NACE 01) a rybářství (CZ-NACE 03) za vlastní výrobky bez DPH v procentech oproti minulému roku z Českého statistického ústavu (Český statistický úřad, 2023).



Obrázek 29- Tržby v zemědělství (CZ-NACE 01 a 03) v procentech, zdroj: (Český statistický úřad, 2023). Pozn: osa y = %, osa x = čas.

Při pohledu na přehled tržeb ze zemědělství a rybářství můžeme pozorovat podobnou křivku jako je na obrázku 30. Toto zjištění může naznačovat spojitost zabarvení sentimentu nejen s globálními daty, které ovlivňují dění na celém světě, jak bylo popsáno výše, ale také s vnitrostátní ekonomickou situací v souvisejících odvětvích (Balounová, 2012). Na grafu tržeb můžeme pozorovat propad okolo konce roku 2007 a začátku roku 2008. Ostatní výkyvy křivky již tak evidentně sentiment zpravodajských textů ze zemědělství neopisují.

Z těchto dat lze usuzovat že může existuje spojitost mezi zabarvením textů a procentuální změnou v tržbách ze zemědělství a rybářství (CZ-NACE 01 a 03).

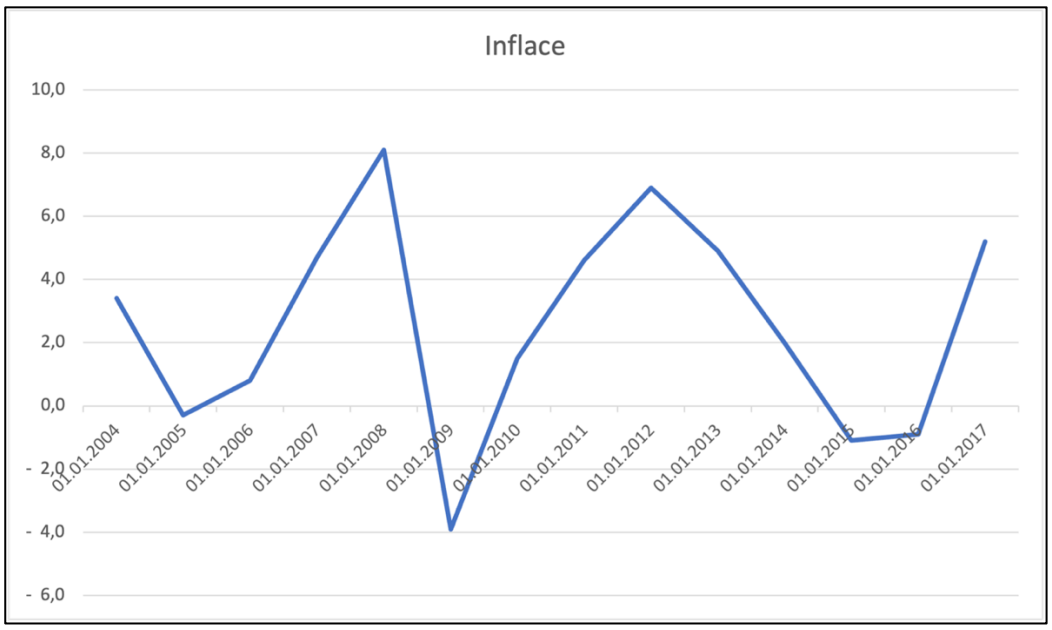
Toto tvrzení lze však vyvrátit, pokud nad daty provedeme korelační analýzu průměrného sentimentu za rok a dat, které se týkají procentuálního nárůstu/poklesu tržeb v potravinářství a rybářství.

Po provedení Pearsonovy korelační analýzy zjistíme že pearsonův korelační koeficient vykazuje velmi nízkou korelaci mezi dvěma zmíněnými soubory. Hodnota korelačního koeficientu je uvedena níže.

$$r = -0.006651106$$

Z těchto výsledků lze soudit že dva zmíněné soubory nemají mezi sebou žádnou vazbu, která by ovlivňovala hodnoty v jednom nebo druhém souboru.

Dalšími porovnávanými hodnotami je inflace týkající se potravin a nealkoholických výrobků z dat Českého statistického ústavu, které spadají pod potravinářský průmysl (Český statistický úřad, 2023). Na grafu níže je vynesena změna inflace v procentech oproti minulému roku.



Obrázek 30 - Inflace potravin a nealkoholických výrobků, zdroj: (Český statistický úřad, 2023). Pozn: osa y = %, osa x = čas.

Na obrázku 30 je vidět znatelný nárůst inflace v rokách 2007/2008, toto zvyšování je zapříčiněné zhoršující se ekonomické situace, která se dotkla i České republiky (Balounová, 2012). Mezi roky 2011 až 2015 kde inflace začala klesat. V roce 2017 se inflace znovu začala zvedat. Pokud tato statistická data porovnáme s průměrným sentiment v rocích 2008,2007 a 2011 až 2015 můžeme si všimnout že křivka sentimentu se v rocích 2011 až 2015 spíše zvedná, ale u předchozích let se podobá křivce inflace, ale v roce 2017 kdy se začala inflace znovu zvedat průměrný sentiment lehce poklesl.

Nesrovnalosti, které vidíme v rocích 2011 až 2015 mohou být způsobené nízkou specificitou modelu logistické regrese, a tedy chyby v predikci algoritmu vzhledem k sentimentu textu.

I vzhledem k možné chybě predikce způsobené nízkou specificitou můžeme podle porovnání grafů výše že sentiment zpravodajských textů ze zemědělství může do jisté míry odrážet ekonomickou situaci v zemědělském sektoru a potravinářství.

Pokud znovu podrobíme průměrný sentiment Pearsonově regresní analýze zjistíme, že výsledný regresní koeficient dosahuje hodnoty:

$$r = -0.1807581$$

Tato hodnota je sice o něco vyšší než při prvním měření v případě tržeb v potravinářství a rybářství, ale pořád dosahuje hladiny korelace, kterou lze označit za „velmi slabou“. Tudíž ani v tomto případě nelze potvrdit domněnky zmíněné výše že by míra meziroční změny inflace ovlivňovala predikovaný sentiment u zpravodajských textů ze zemědělství.

Lze tedy jistě konstatovat že ani inflace v potravinářství a ani průměrná změna tržeb v zemědělství a rybářství nemá prokazatelný vliv na naměřený sentiment zpravodajských textů ze zemědělství. Hodnoty, které byli naměřeny se tedy spíše týkají hlavně textů a zpravodajského webu Agris.cz (ČZU a MZČR, 2022) jako celku, než že by měli nějaký vyšší přesah.

4.2 Detailní shrnutí popisu zpracování analýzy sentimentu

Analýza sentimentu se skládá z několika po sobě jsoucích kroků. První fáze analýzy sentimentu se zaobírá prvotním zpracováním dat. Způsob zpracovávání dat vychází ze struktury zkoumaného souboru a cíle kterého se snažíme dosáhnout.

V našem případě budou data z portálu Agris (ČZU a MZČR, 2022) zpracována pomocí Python knihovny Pandas. V tomto kroku dojde k prvotnímu očištění dat od nepotřebných atributů, které nejsou pro samotnou analýzu sentimentu nutné. Po tomto kroku následuje zpracování jednotlivých článků, které datový soubor obsahuje. Články budou oddělené po jednotlivých odstavcích. Důvod pro provedení této operace je následující. Články, které datový soubor obsahuje jsou do angličtiny přeloženým týdenním souhrnem důležitých zpráv, které byli na portálu Agris (ČZU a MZČR, 2022) publikovány v češtině.

V zájmu přesnosti výsledků je tedy lepší jednotlivé týdenní souhrny rozdělit podle jednotlivých překladů zpráv z portálu.

Jako další následuje průzkumová analýza souboru s daty z portálu Agris (ČZU a MZČR, 2022). Tato analýza obsahuje statistické postupy, které následně pomohou pro interpretaci výsledků analýzy sentimentu. Průzkumem dat získáme různé statistické ukazatele, které se týkají analyzovaných textů.

Třetí krok analýzy sentimentu se vztahuje k výběru trénovacího souboru, na kterém bude následně vycvičen model pro vyhodnocování sentimentu zpravodajských textů ze zemědělství. Soubor bude vybrán pomocí kritérií, které budou nejlépe vyhovovat záměru analýzy. Bude se tedy jednat o soubor s daty, která se podobají datům, které jsou předmětem analýzy sentimentu a jsou již ohodnocená sentimentem, to je pro trénovací množinu klíčové. Dalším důležitým parametrem pro výběr testovacího datového souboru je počet pozorování, které soubor obsahuje. Pokud by soubor obsahoval nízký počet pozorování mohlo by to ovlivnit schopnost algoritmu naučit se klasifikovat data správně. Další důležitou části trénovacího souboru je vyváženost dat, na kterou musí být brán zřetel.

Jako další krok proběhne rozdělení trénovacího souboru na více částí dle doporučených poměrů (Muraina, 2022). První část reprezentuje trénovací data pro algoritmus a druhá reprezentuje testovací sadu dat, na které se ověřuje přesnost natrénovaného algoritmu.

Následně proběhne samotné trénování modelů na trénovací množině. Budou zde popsány jednotlivé metody, jak lze algoritmu trénovat pro maximalizaci přesnosti algoritmu. Trénování se týká všech autorem implementovaných algoritmů.

Po natrénování algoritmů pro zpracování a určení sentimentu na zkoumaných datech je přesnost algoritmu ověřena na testovací množině dat. Ze vzniklých pozorování chování algoritmů je následně vyhodnocena procentuální přesnost algoritmů a další znaky, které určují kvalitu algoritmu společně s jeho silnými a slabými stránkami.

Dalším krokem je samotná analýza sentimentu na zpravodajských datech ze zemědělství z portálu Agris (ČZU a MZČR, 2022). Tyto data budou postaveny na vstup všem vybraným algoritmům, které budou splňovat minimální hranici pro kvalitu výsledků, která bude popsána v kapitole o přesnosti algoritmů.

4.3 Využití projektu Semex pro analýzu sentimentu

Další možností, jak zpracovávat analýzu sentimentu je pomocí služby Semex z projektu PoliRULAR (CORDIS, 2023). Tato služba nabízí volatelný endpoint, pomocí kterého lze zpracovat analýzu sentimentu. Nevýhodou tohoto přístupu zpracování analýzy sentimentu je fakt že systém Semex neumí přijímat data v podobě souborů CSV. Možností jak, zpracovat sentiment pomocí systému Semex je nahrát soubor skrze API ve formátu DOCX, PDF nebo ve formě klasického textu (TXT) (KajoServices, 2022). Tyto datové formáty mohou být vhodné pro ohodnocení sentimentu na hotových dokumentech, ale pro případ, kdy jsou analyzovanými daty data z webu, a tedy pravděpodobně jsou extrahována z nějakého datového úložiště nedostaneme bez vysokého úsilí soubory ve formátu, který by Semex podporoval, a proto nebyl pro tuto práci použit.

5 Závěr

Hlavním cíle této práce bylo zanalyzovat sentiment zpravodajských textů ze zemědělství. Tento cíl byl následně rozdělen na dílčí cíle, které popisují proces pro dosažení analýzy sentimentu zpravodajských dat.

V teoretické části jsou shrnuty poznatky současného stavu problematiky a metody, které mohou být použity pro řešení zkoumaného problému. Konkrétně se jedná o proces analýzy sentimentu, ať už se jedná o klasický lingvistický přístup nebo přístup, který volí jako vyhodnocovací aparát algoritmy umělé inteligence. Dále jsou v teoretické části uvedeny postupy, které jsou nutné pro vypracování analýzy sentimentu. Jedná se o metody zpracování a přípravy textu, výčet používaných metod pro tokenizaci a vektorizaci textu, popis klasifikačních algoritmů a zhodnocení kvality trénovaných modelů.

V praktické části byla vypracována analýza sentimentu zpravodajských textů ze zemědělství. Analýza se skládá z několika po sobě jdoucích kroků. Jedná se o zpracování surových dat, výběr trénovacího souboru, zpracování trénovacího souboru, výběr klasifikátoru, učení klasifikačního modelu, vyhodnocení přesnosti klasifikačního modelu, vyhodnocení zkoumaných dat a interpretace výsledků.

Samotná příprava dat pro vyhodnocení sentimentu byla náročná a zabrala většinu času, který analýza vyžadovala. Jednalo se zde o zpracování surových dat z portálu Agris.cz (ČZU a MZČR, 2022) do podoby, která byla vhodná pro následnou tokenizace a vektorizaci. Surový soubor obsahuje 4 468 záznamů, které se skládají ze souhrnných zpravodajských textů v anglickém jazyce. Tento soubor byl následně po zpracování rozdělen na jednotlivé věty textů (36 439 záznamů). U rozdělených textů byla dodržena příslušná časová razítka, aby bylo možné záznamy později analyzovat v čase.

Body zabývající se zpracováním surových dat a trénovacího souboru jsou popsány podrobně v kapitole 3.4. Jedná se o proces tokenizace a vektorizace dat pro použití v klasifikátoru pro vyhodnocení analýzy sentimentu. V procesu vektorizace dat bylo nutné vyřešit překážky, které by jinak zabraňovali vyhodnocení sentimentu zpravodajských dat ze zemědělství. Tento problém a jeho řešení je popsán v kapitole 3.8 zabývající se trénováním modelu.

Vyhodnocení kvality natrénovaného modelu probíhá promocí kvalitativních metrik popsaných v teoretické části v kapitole 2.9. Po zvolení klasifikátoru bylo vypracováno vyhodnocení sentimentu zpravodajských dat ze zemědělství.

Data byla následně zpracována do podoby průměrného sentimentu v průběhu roků 2004 až 2021. Výsledky analýzy sentimentu byli následně dále zpracované pro lepší interpretaci a pro porovnání s externími daty, které jsou popsány v diskusi práce. Vyhodnocení sentimentu dopadlo následovně: 31 451 záznamů bylo ohodnoceno predikcí modelu logistické regrese jako kladný sentiment a 4 988 jako záporný sentiment.

Průměrný sentiment byl následně porovnán pomocí korelační analýzy s daty z Českého statistického úřadu, které se zabývají inflací a tržbami v zemědělství a rybářství. Korelace mezi těmito daty je velmi nízká a lze tedy soudit že sentiment není ovlivněn ani jednou z testovaných metrik.

6 Bibliografie

Chaudhary, Yash. 2020. Kaggle.com. *Stock-Market Sentiment Dataset.* [Online] 2020. [Citace: 21. 12 2022.] doi:10.34740/kaggle/dsv/1217821.

Chen, Xinjian, a další. 2004. Segmentation of Fingerprint Images Using Linear Classifier. *EURASIP Journal on Applied Signal Processing.* 2004, 4.

CORDIS. 2023. Projects & results. *Future Oriented Collaborative Policy Development for Rural Areas and People.* [Online] 26. 02 2023. [Citace: 01. 03 2023.] https://cordis.europa.eu/project/id/818496.

Church, Kenneth Ward. 2017. Word2Vec. *Natural Language Engineering.* 2017, Sv. 23, 1, stránky 155-162.

Český statistický úřad. 2023. Český statistický úřad. *Indexy spotřebitelských cen podle klasifikace COICOP - míra inflace.* [Online] 14. 2 2023. [Citace: 14. 2 2023.] https://vdb.czso.cz/vdbvo2/faces/cs/shortUrl?su=d437aeb4.

—. 2023. Hlavní makroekonomické ukazatele. *Český statistický úřad.* [Online] 1. 2 2023. [Citace: 13. 2 2023.] https://www.czso.cz/documents/10180/196622028/chmucr020123.xlsx/2d681272-5439-4929-945f-9ecf7193b39b?version=1.0.

ČZU a MZČR. 2022. Agris.cz. *Agris.cz.* [Online] 2022. [Citace: 01. 03 2023.] www.agris.cz.

Balounová, Marcela. 2012. Vývoj inflace v České republice a její dopady na domácnosti a podniky. [Online] 2012. [Citace: 06. 03 2023.] https://dspace.tul.cz/bitstream/handle/15240/45849/V_26112_E.pdf?sequence=-1.

Blei, David M., Ng, Andrew Y. a Jordan, Michael I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research 3.* 2003.

Brandenburger, Thomas a Furth, Alfred. 2009. Cumulative Gains Model Quality Metric. *Journal of Applied Mathematics and Decision Sciences.* 2009, Sv. 2009.

Buitinck, Lars, a další. 2013. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning.* 2013, stránky 108-122.

D'Andrea, Alessia, a další. 2015. Approaches, Tools and Applications for Sentiment Analysis Implementation. *International Journal of Computer Applications.* 2015, Sv. 125, 3.

Dike, Hapiness Ugochi, a další. 2018. Unsupervised Learning Based On Artificial Neural Network: A Review. *Proceedings of the 2018 IEEE International Conference on Cyborg and Bionic Systems.* 2018.

Dohare, Shibhansh, Karnick, Harish a Gupta, Vivek. 2017. Text Summarization using Abstract Meaning Representation. 2017.

Drezner, Zvi, Turel, Ofir a Zerom, Dawit. 2010. A Modified Kolmogorov–Smirnov Test for Normality. *Communications in Statistics - Simulation and Computation* . 2010, Sv. 39, 4.

Feldman, Ronen. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM.* 2013, Sv. 56, 4.

George K, Soumya a Joseph, Shibily. 2014. Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature. 2014, Sv. vol. 16, issue 1, stránky 34-38.

Hinneburg, Alexander a Gabriel, Hans Henning. 2007. DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation. 2007.

Hull, David. 1995. Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.* 1995, stránky 282-291.

IBM Corporation. 2022. Text classification algorithms. *IBM Documentation.* [Online] 2022. https://www.ibm.com/docs/en/rpa/21.0?topic=classification-text-algorithms.

IBM. 2020. Text mining. *IBM.* [Online] 2020. https://www.ibm.com/cloud/learn/text-mining.

KajoServices. 2022. Semantic Explorer API. *GitHub.* [Online] 20. 11 2022. [Citace: 21. 2 2023.] https://github.com/KajoServices/polirural-semex-doc/tree/master/API#semantic-analysis.

Kanmani, S., a další. 2007. Object-oriented software fault prediction using neural networks. 2007, Sv. vol. 49, issue 5, stránky 483-492.

Kharde, Vishal A. a Shonawane, Sheetal S. 2016. Sentiment Analysis of Twitter Data: A Survey of Techniques. 2016, Sv. 11, 139.

Kába, Bohumil a Svatošová, Libuše. 2012. *Statistické nástroje ekonomického výzkumu.* Plzeň : Vydavatelství a nakladatelství Aleš Čeněk, s.r.o., 2012. ISBN 978-80-7380-359-9.

Likas, Aristidis, Vlassis, Nikos a Verbeek, Jakob J. 2003. The global k-means clustering algorithm. *Pattern Recognition.* 2003, Sv. 36, 2.

Linux manual page. 2019. time(1) — Linux manual page. *Linux manual page.* [Online] 6. 3 2019. [Citace: 13. 2 2023.] https://man7.org/linux/man-pages/man1/time.1.html.

McShane, Marjorie a Nirenburg, Sergei. 2021. *Linguistics for the Age of AI.* místo neznámé : The MIT Press, 2021.

ML Wiki. 2014. Cumulative Gain Chart. *ML Wiki.* [Online] 8. 6 2014. [Citace: 13. 2 2023.] http://mlwiki.org/index.php/Cumulative_Gain_Chart.

Mohey, Doaa. 2016. Enhancement Bag-of-Words Model for Solving the Challenges of Sentiment Analysis. 2016, Sv. vol. 7, issue 1.

Muraina, Ismail Olaniyi. 2022. IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS. *7th INTERNATIONAL MARDIN ARTUKLU SCIENTIFIC RESEARCHES CONFERENCE.* 02 2022.

Nadkarni, Prakash M, Ohno-Machado, Lucila a Chapman, Wendy W. 2011. Natural language processing: an introduction. 2011, Sv. 16, 5, stránky 544–551.

Orava, Jan. 2008. Volba vyhlazovacího parametru při jádrových odhadech hustoty. 2008.

Osisanwo, F Y, a další. 2017. Supervised Machine Learning Algorithms: Classification and Comparison. *nternational Journal of Computer Trends and Technology (IJCTT.* 2017, Sv. 3, 48.

Pedregosa, F, a další. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research.* 2011, 12, stránky 2825-2830.

Piatetsky-Shapiro, Gregory a Masand, Brij. 1999. Estimating Campaign Benefits and Modeling Lift. 1999.

Python Software Foundation. 2023. Python documentation. *Python.* [Online] 2023. [Citace: 01. 03 2023.] https://www.python.org/doc/.

Qiu, Guang, a další. 2010. DASA: Dissatisfaction-oriented Advertising based on Sentiment Analysis. 2010.

Rehioui, Hajar, a další. 2016. DENCLUE-IM: A New Approach for Big Data Clustering. *Procedia Computer Science.* 2016, 83, stránky 560-567.

Rosario, Barbara. 2000. Latent Semantic Indexing: An overview. 2000.

Shafranovich, Y. 2005. Common Format and MIME Type for Comma-Separated Values (CSV) Files. [Online] 2005. https://www.rfc-editor.org/rfc/rfc4180#page-2.

Shumaker, Robert P, a další. 2012. Evaluating sentiment in financial news articles. 2012.

Sienčnik, Scharolta Katharina. 2015. Adapting word2vec to Named Entity Recognition. *Proceedings of the 20th Nordic Conference of Computational Linguistics.* 2015.

Socher, Richard, a další. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Conference on Empirical Methods in Natural Language Processing.* 2013.

Sperandei, Sandro. 2013. Understanding logistic regression analysis. [Online] 26. Listopad 2013. [Citace: 7. 2 2023.] https://hrcak.srce.hr/file/171128.

Stack overflow. 2022. Developer survey 2022. *Stackoverflow developer survey 2022.* [Online] 2022. [Citace: 01. 03 2023.] https://survey.stackoverflow.co/2022/#section-most-popular-technologies-other-frameworks-and-libraries.

Tandoc, Edson C. 2019. Techniques and applications for sentiment analysis. *Sociology Compass.* 2019, Sv. 13, 9.

TFIDF. TFIDF. [Online] [Citace: 15. 8 2022.] http://www.tfidf.com/.

Veselý, Arnošt. 2012. *Metody umělé inteligence.* Praha : Česká zemědělská univerzita v Praze, 2012. 978-80-213-2295-0.

Vinodhini, G. a Chandrasekaran, R.M. 2016. A comparative performance evaluation of neural network based approach for sentiment classification of online reviews. 2016, Sv. vol. 28, issue 1, stránky 2-12.

Wang, Chuan-Ju, a další. 2013. Financial Sentiment Analysis for Risk Prediction. *International Joint Conference on Natural Language Processing.* 2013.

ZHANG, Yun-tao, Ling GONG a Yong-cheng WANG. 2005. An improved TF-IDF approach for text classification. *Journal of Zhejiang University SCIENCE.* 2005, Sv. 6, 1.

7 Seznam tabulek

8 Seznam obrázků

9 Seznam rovnic

10 Seznam použitých zkratek

AI = artificial intelligence (umělá inteligence)

TMRs = Text Meaning Representations

ARM = Abstract Meaning Representation

NLP = Natural Language Processing

NLTK = Natural Language Toolkit

LSI = Latent semantic indexing

SVD = singular value decomposition

LDA = Latent Dirichlet allocation

FP = false positive

NFP = not false positive

BOW = Bag of Words

TF-IDF = term frequency-inverse document frequency

KDE = kernel density estimation

DENCLUE = density clustering

SVM = support vector machines

RBF = radial basis function

TSV = table separeted values

11 Přílohy

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import matplotlib.pyplot as pp
import scikitplot as skplt

# loads test dataframe

print("Reading test data")

test_df = pd.read_csv("./stock_data_semicol.csv", delimiter=";")

# loads sharp data

print("Reading sharp data")

agri_df = pd.read_csv("../dp_data/cutted-data.csv", delimiter=",")


def vectorize(df, label, col):
    print("Starting vectorize data frame" + " : " + label)

    vectorizer = CountVectorizer(stop_words="english", max_features=9000)

    vec = vectorizer.fit_transform(df[col])

    print("Creating vectorized data frame from df" + " : " + label)

    return pd.DataFrame(vec.toarray())


# vectorize test data frame

vec_test_df = vectorize(test_df, "test", "Text")


# vectorize sharp data

vec_sharp_df = vectorize(agri_df, "sharp", "fulltext")

# split testing data to training and testing sets

print("Splitting sets to train and test in ratio 80:20")

x_train, x_test, y_train, y_test = train_test_split(vec_test_df, test_df["Sentiment"], test_size=0.2,
random_state=200)


# create Model

logreg = LogisticRegression()

print("Training model on train sets")

logreg.fit(x_train, y_train)

# display model score

model_score = logreg.score(x_test, y_test)

print("Score is :" + str(model_score))

# Confusion matrix

test_prediction = logreg.predict(x_test)

test_prediction_proba = logreg.predict_proba(x_test)

cof_matrix = metrics.confusion_matrix(y_test,test_prediction)

print("Confusion matrix of test model")

print(cof_matrix)

# Evaluating agri data

print("Predicting sharp data")

prediction = logreg.predict(vec_sharp_df)

prediction_df = pd.DataFrame(prediction, columns=["Sentiment"])

#cof_matrix_display = metrics.ConfusionMatrixDisplay(cof_matrix)

#cof_matrix_display.plot()

#skplt.metrics.plot_ks_statistic(y_test, test_prediction_proba)

#skplt.metrics.plot_cumulative_gain(y_test, test_prediction_proba)

#skplt.metrics.plot_lift_curve(y_test, test_prediction_proba)

#skplt.metrics.plot_roc(y_test, test_prediction_proba)

#pp.show()
```

Obrázek 31 - Zdrojový kód pro analýzu sentimentu

11.1 Seznam knihoven v requirements.txt

- pandas
- nltk
- scikit-learn
- scikit-plot