Filozofická fakulta Univerzity Palackého v Olomouci
Katedra obecné lingvistiky

Aplikace strojového zpracování jazyka v automobilovém průmyslu

magisterská diplomová práce

Autor: Bc. František Špaček
Vedoucí práce: Mgr. Vladimír Matlach, PhD.

Olomouc
2023

Prohlášení

Prohlašuji, že jsem magisterskou diplomovou práci „Aplikace strojového zpracování jazyka v automobilovém průmyslu“ vypracoval samostatně a uvedl jsem veškerou použitou literaturu a veškeré použité zdroje.

V Olomouci dne 27.11.2023 ……

František Špaček

Poděkování

Chtěl bych poděkovat Mgr. Vladimíru Matlachovi, PhD. za rady a pomoc s řešením komplikací, a také všem, kteří mě v průběhu psaní práce podporovali.

Abstrakt

Název práce: Aplikace strojového zpracování jazyka v automobilovém průmyslu

Autor práce: Bc. František Špaček

Vedoucí práce: Mgr. Vladimír Matlach, PhD.

Abstrakt: Diplomová práce se zabývá aplikací strojového zpracování jazyka a analýzou sentimentu v oblasti automobilového průmyslu. Jsou nastíněny případy aplikací strojového zpracování jazyka a analýzy sentimentu specifické pro dané odvětví průmyslu s hlavním zaměřením na digitální asistenty v automobilech a jejich potenciálním využití při zlepšování pobytu ve vozidle. Jsou prozkoumávána možná rizika a také implikace v nejhorších případech. Krátce jsou také nastíněna specifika komerční sféry při trénování vlastních klasifikátorů pro analýzu sentimentu. V rámci této práce byl vytvořen malý doménově specifický dataset, jehož průběh vzniku je také popisován. Tento dataset později posloužil k finální evaluaci natrénovaných modelů. Postupně jsou vyzkoušeny a otestovány slovníkové metody, metody strojového učení jako SVM, Naive Bayes nebo k-NN, různé typy neuronových sítí a nakonec předtrénované modely. Je zkoumán vliv velikosti trénovacího vzniku i typu vektorizace na výsledné klasifikátory. Modely jsou hodnoceny na základě přesnosti klasifikací, ale i z hlediska vhodnosti pro automatickou anotaci. Přesnost modelů s nejlepšími výsledky byla nakonec otestována ještě na doménových datech.

Klíčová slova: analýza sentimentu, strojové učení, neuronové sítě, automobilový průmysl, strojové zpracování jazyka, NLP

Abstract

Title: Natural language processing application in automotive industry

Author: Bc. František Špaček

Supervisor: Mgr. Vladimír Matlach, PhD

Abstract: This master's thesis addresses the application of natural language processing and sentiment analysis in the automotive industry. It outlines case studies of natural language processing and sentiment analysis applications specific to the industry, with a primary focus on digital assistants in cars and their potential for enhancing the in-vehicle experience. Possible risks and implications of worst-case scenarios are explored. The specifics of the commercial sphere in training custom sentiment analysis classifiers are briefly outlined. Within this work, a small domain-specific dataset was created, and its development process is described. This dataset later serves for the final evaluation of trained models. Various methods are experimented with and tested, including lexicon-based methods, machine learning methods such as SVM, Naive Bayes, or k-NN, different types of neural networks, and pre-trained models. The impact of the training dataset size and vectorization type on the resulting classifiers is examined. Models are evaluated based on classification accuracy and suitability for automatic annotation. The accuracy of models with the best results is further tested on domain-specific data.

Keywords: sentiment analysis, machine learning, neural networks, automotive industry, natural language processing, NLP

Obsah

1. Úvod

Tato práce se zabývá analýzou sentimentu jakožto součásti textové analýzy a její aplikací v automobilním průmyslu. Většina poznatků v této práci vychází z mého působení v nejmenované automobilové firmě, kde oddělení, jehož součástí jsem, má na starosti především dokazování funkčnosti konceptů (proof of concept) a vytváření minimálních životaschopných produktů (minimum viable product neboli MVP). Naším cílem tedy není integrace hotového a optimalizovaného řešení do nějakého produktu, ale pouze dokázat, že něco je možné a upozornit na případné hardwarové či technologické požadavky potřebné pro funkční a efektivní implementaci.

Hlavním zkoumaným jazykem pro nás je angličtina, a to hned z několika důvodů – má poměrně jednoduchou morfologii a je pro ni dostupné asi největší množství materiálů, ať už se jedná o datasety nebo již hotové natrénované modely. Pro některé aplikace by stačila samotná angličtina, ale většinou byla pouze startovním bodem a byla snaha ověřit dostupnost materiálů a fungování našich řešení i na jiných jazycích, se kterými se v rámci firmy lze setkat.

Jelikož se však jedná o víceméně komerčně zaměřené aplikace, jsou naše možnosti dosti omezené v porovnání s výzkumníky působícími pouze v akademické sféře, protože některé materiály jsou zveřejňovány s licencí neumožňující komerční využití, například dataset Amazon recenzí (Ni et al., 2019)[1].

Prvním krokem tedy je vymezit si oblast analýzy sentimentu a definovat předmět jejího zkoumání, tedy co je to sentiment.

2. Co je to sentiment

Slovník spisovného jazyka českého definuje sentiment jako citlivost nebo senzitivitu (Sentiment, 2011), což je definice zajisté vhodná, ale pro případ této práce nedostačující. O něco rozsáhleji definuje sentiment třeba česká wikipedie (Sentiment, 2022), kde se píše, že sentiment může být mimo cit i náklonnost nebo odpor, a právě náklonnost nebo odpor je to, co se sentimentem myslí v kontextu textové analýzy. Jinak řečeno se v kontextu textové analýzy sentimentem myslí postoje mluvčího k nějakému danému tématu, osobě nebo produktu.

Další možnost je na sentiment nahlížet jako na součást jakéhosi soukromého vnitřního stavu, který není možné objektivně pozorovat a jehož součástí jsou mimo jiné emoce, názory a domněnky (Mejova, 2009, s. 3). Sentiment je tedy možné brát jako něco velmi subjektivního a v některých případech se i samotná subjektivita do analýzy sentimentu zahrnuje, viz například python knihovnu TextBlob nabízející s analýzou sentimentu i míru subjektivity (více o knihovně TextBlob v části 5.1.1).

[1] Autoři zmiňují, že data jim nepatří a nejsou tedy v pozici, kdy by mohli dataset nabízet pod licencí nebo nějakým způsobem kontrolovat přístup k datům, ale žádají o používání dat pouze pro výzkum viz https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/. [Přístup 2023-09-04]

Subjektivita však není předmětem zkoumání této práce, takže pojmem sentiment bude myšleno především zkoumání postojů a emocí mluvčích.

Spojením dvou předchozích výkladů lze sentiment definovat jako soukromý vnitřní stav, ve kterém se projevují emoce, postoje a názory mluvčího. A právě tato soukromost a niternost sentimentu velice komplikuje celou problematiku, protože sentiment je z těchto důvodů velice subjektivní, a tedy i poměrně složitý na zkoumání – dalo by se totiž říci, že se nezkoumá sentiment přímo, ale pouze jeho projevy skrze další prostředky jako například jazyk, což je oblast, kde může pomoct právě analýza sentimentu.

2.1. Analýza sentimentu

Analýza sentimentu jako výzkumná disciplína je propojena s komputační lingvistikou, strojovým zpracováním jazyka a vytěžováním textu odkud si vypůjčuje nástroje k hledání odpovědí na otázky zasahující i do psychologie (Mejova, 2009, s. 5). Pomocí metod z právě zmíněných disciplín lze poměrně efektivně analyzovat různé jazykové aspekty projevů a na základě toho sentiment určovat.

V úplně nejjednodušší podobě lze o analýze sentimentu přemýšlet jako o pouhém rozlišování na negativní/pozitivní, což může být pro některé aplikace efektivním řešením, jak celou problematiku trochu zjednodušit, avšak nese to s sebou i nějaká rizika (více v části 4.2). Je ale také možné provádět o něco podrobnější zkoumání a sentiment pojmout jako nějaké spektrum, kdy se bere v potaz i intenzita sentimentu (Mejova, 2009, s. 5).

Možností aplikací analýzy sentimentu v praxi je hned několik. Jednou z velice častých aplikací je analýza recenzí či komentářů spojených s produkty firmy nebo firmou samotnou. Implementací nějakého typu analýzy sentimentu může firma efektivněji na tyto komentáře reagovat a zlepšovat tak vztahy se zákazníky. Další možnou aplikací je průzkum mínění spojeného s produktem. Firma může nějakou dobu sbírat a analyzovat data, aby zjistila, na co si lidé stěžují a s čím mají nejčastěji problémy za účelem zlepšování svých produktů (Cambria et al., 2017, s. 2). Žádná z těchto aplikací však není specifická pro nějaké odvětví průmyslu nebo trhu a využít toho tedy může téměř kdokoliv. Část 3 tedy bude popisovat aplikace analýzy sentimentu, které jsou více specifické pro automobilový průmysl.

3. Analýza sentimentu v automobilovém průmyslu

Jednou z aplikací, která je tentokrát už pro automobilní průmysl specifickou, je zakomponování analýzy sentimentu přímo do auta. Moderní auta už často mívají k dispozici digitální asistenty, na které mohou být lidé již zvyklí z mobilních telefonů. Jejich účelem je především zjednodušovat ovládání vozidla za jízdy tím, že řidič nemusí odvádět pozornost od řízení a může některé věci jednoduše nastavit hlasem, aniž by musel hledat potřebná tlačítka na středovém panelu. V tom případě je ale důležité, aby digitální asistent mluvčímu dobře rozuměl a aby byl schopný zvládat příkazy. Je však jednoduché si představit situaci, ve které se řidič snaží hlasem například snížit teplotu, na kterou je auto vytápěno, ale digitální

asistent buďto nerozumí, nebo rozumí špatně. Něco takového může řidiče lehce naštvat, a nakonec je stejně nucen teplotu nastavit ručně, takže musí na chvíli odvrátit pozornost od řízení, čímž vzniká potenciálně nebezpečná situace.

Analýza sentimentu sice nepomůže vyřešit problém se špatným porozuměním, to je úkol spadající především do oblasti rozpoznávání řeči a jazykových modelů, může však pomoci zmírnit negativní dopady takto neúspěšné interakce. Mimo jiné lze podobným způsobem uklidnit řidiče rozrušeného z jakýchkoliv jiných důvodů. Původní představa byla taková, že by se pomocí analýzy sentimentu vylepšil digitální asistent v autě – na základě vyhodnocení stavu řidiče by digitální asistent mohl přizpůsobovat například svůj tón hlasu, vybírat vhodnější slova, nabízet různá doporučení nebo rovnou měnit různá nastavení auta. I když se naše zadání vztahovalo pouze na digitálního asistenta v autě (a tedy pouze texty ze speech-to-text přepisů), je snadné si představit zapojení dalších systémů do kontroly stavu řidiče vozidla, například měření síly stisku volantu (Sahar et al., 2021) nebo eye-tracking (Ahlström et al., 2021), což by dále usnadňovalo a zpřesňovalo získávání informací o řidiči. V této teoretické situaci by poté různé systémy spolupracovaly na monitorování řidiče a další systém by na základě měření z těchto systémů mohl aktivně reagovat na řidičův stav, čímž by bylo možné například regulovat stresové situace.

Při vyhnání této teoretické situace do extrému si lze představit, že by spolu všechna auta v určitém okruhu komunikovala a předávala by si informace o stavu jejich řidičů. Ostatní řidiči by tak mohli dostávat informace o tom, že je v jejich okolí vystresovaný nebo rozrušený řidič a měli by dbát zvýšené pozornosti, i když takováto poněkud futuristicko-dystopická aplikace by se nejspíše potýkala s potížemi minimálně kvůli GDPR.

Už i snaha o pouhé reagování digitálního asistenta na stav řidiče na základě analýzy sentimentu s sebou nese nějaká rizika. Každý projevuje své emoce trochu odlišným způsobem a každý také reaguje na různé věci jinak, takže řešení, které jednoho člověka uklidní může druhého ještě více naštvat, což je přesným opakem původního záměru. Takový systém by tedy nemohl být univerzální, musel by být schopný reagovat v souladu s povahou daného řidiče, takže systém by musel být schopný se přizpůsobovat podle toho, kdo zrovna sedí za volantem. A i když se může zdát extrémní mluvit o vzniku nebezpečných situací kvůli špatným reakcím digitálního asistenta v autě, je důležité si uvědomit, že všechny stresové situace, v izolaci jakkoliv banální, se mohou postupně kumulovat a nakonec vyústit v nějakou větší reakci, takže je v tomto ohledu bezpečnější přistupovat k této problematice s větší mírou rozvahy a obezřetnosti, aby výsledný produkt skutečně plnil svou roli jakéhosi uvolňovače stresu, než aby stres způsoboval.

4. Rozdíly v komunikaci člověka s člověkem a člověka s AI

Při komunikaci člověka s člověkem je k dispozici velké množství extralingvistického kontextu, které lze použít k vyhodnocení dané komunikační situace. Pokud se vezme v potaz komunikace mezi lidmi,

kteří nejsou například hluší nebo němí, tak tito lidé při komunikaci tváří v tvář slyší výšku hlasu mluvčího, vidí gesta, výrazy tváře a vnímají řeč těla, což vše může posloužit jako dodatečné informace k samotnému sdělení. A všechny tyto informace mohou napomoci ke zvýšení úspěšnosti komunikace, protože mohou pomoci k lepšímu porozumění významu mluvčího.

Na druhou stranu v mnoha komunikačních aplikacích umělé inteligence je umělé inteligenci k dispozici pouze text. Může to být samotná textová zpráva v případě chatbotů nebo speech-to-text přepis v případě digitálních asistentů jako například Alexa nebo Siri. Velká část extralingvistického kontextu je tímto ztracena při komunikaci s digitálními asistenty a jediný zbylý zdroj kontextu je doslovná zpráva.

Je snadné si představit zařízení schopné sledovat všechny zdroje extralingvistického kontextu, které jsou přítomné v mezilidské komunikaci, pomocí počítače vybaveného kamerami a různými senzory, které by sledovaly ty stejné věci, které v mezilidské komunikaci vnímá člověk. Určitě by také bylo možné zajít za hranice toho, co vnímá člověk, a měřit další biometrické údaje jako například tepovou frekvenci, krevní tlak, roztažení zorniček apod. Avšak některé z těchto věcí by mohly být značnou nevýhodou, pokud by cílem mělo být udělat komunikaci s umělou inteligencí co nejpřirozenější, protože změření některých z těchto údajů může vyžadovat fyzický kontakt s nějakým zařízením. Navíc by to představovalo nepotřebné množství komplikací a nežádoucí komplexnost celého systému, protože někteří z digitálních asistentů se nachází na malých zařízeních s limitovaným výkonem a komplexnější analýza by pravděpodobně měla za následek pomalejší reakce na promluvy mluvčího, což by akorát dále přispělo ke snížení nepřirozeného pocitu z komunikace.

4.1. Analýza sentimentu bez kontextu

Co je tedy řešením rozdílů mezi typy komunikace nastíněnými výše? Rozhodně není nemožné dělat analýzu sentimentu s velmi malým množstvím kontextu, jen je to o něco těžší kvůli větší nejednoznačnosti pouze textových dat.

Pro účely zachování jednoduchosti jak softwaru, tak hardwaru zařízení schopných provádět analýzu sentimentu by byl dobrý nápad nejprve zdokonalit analýzu sentimentu pouze z textu. Zatímco dva různí mluvčí mohou pronést stejná slova velice rozdílným způsobem, obsah sdělení zůstává stejný. Neverbální projevy sentimentu mluvčího se pravděpodobně budou různit více než ty verbální, a to je právě to, proč zaměřovat se pouze na text je dobrý nápad – je to ta nejméně variabilní proměnná.

Poté, co je inference sentimentu pouze z textu zdokonalena na uspokojivou úroveň lze implementovat navíc analýzu hlasu mluvčího k analýze z textu a tím dále výsledky zpřesnit.

4.2. Problém s daty

Když dojde na trénování nějakého vlastního modelu za nějakým konkrétním účelem, prvním krokem je získat kvalitní data v dostatečném množství. Zatímco pro různé ostatní úkoly týkající se textu existují

až tisíce datasetů, pro analýzu sentimentu jsou jich jen desítky, viz například výsledky vyhledávání na webu Papers With Code (Meta AI, b.r.)[2]. Navíc některé z těchto datasetů jsou poměrně malé, moc redukcionistické, používají potenciálně problematické metody anotace, nebo anotace neobsahují vůbec. V tomto ohledu je tedy analýza sentimentu trochu pozadu. Některé z těchto problémů datasetů jsou zřetelné při bližším prozkoumání dvou asi nejsnadněji dostupných datasetů – *IMDB movie review dataset* (Maas et al., 2011) a *Sentiment140* (Go et al., 2009).

IMDB dataset se skládá z 50 000 filmových recenzí s anotačními značkami pro pozitivní a negativní sentiment. I když tento dataset může být dobrým výchozím bodem, není dostatečně velký na to, aby se na něm natrénoval robustní model schopný dobře zpracovávat skutečná data, jak bude ukázáno v dalších částech.

Na druhou stranu dataset Sentiment140 obsahuje 1,6 milionu tweetů, což už je obstojná velikost. Anotace ale byla provedena automaticky pomocí emotikonů, což může vést ke komplikacím. Co když obsah tweetu není v souladu s použitým emotikonem? Takový tweet by poté nejspíše měl chybnou anotační značku, což by mohlo vést k natrénování nekvalitního modelu. Autoři tohoto datasetu si jsou toho však vědomi a k provedené anotaci referují jako „noisy labels“, tzn. anotace, která může nést nepřesnosti a nemusí zachycovat skutečný stav (Liang et al., 2022).

Oba zmíněné datasety navíc obsahují anotační značky pouze pro negativní a pozitivní sentiment, což je naprosto pochopitelný krok při formalizaci a zjednodušení zadání, ale je to velice redukcionistický přístup. Přidat alespoň třetí třídu pro neurčitý/neutrální sentiment by byl dobrý nápad. Takový krok by sice stále byl nedostatečným zlepšením při modelování lidských emocí, ale bylo by to alespoň nějaké zlepšení.

Kolem datasetu Sentiment140 panuje navíc nějaký zmatek, protože nejen weby jako Kaggle nebo Tensorflow, na kterých lze obvykle najít datasety, ale dokonce i oficiální web datasetu uvádí nesprávné informace. Všechny weby totiž uvádí, že dataset zahrnuje tři třídy – pozitivní, neutrální a negativní (Sentiment140 dataset with 1.6 million tweets, b.r.; Sentiment140, b.r.; For Academics, b.r.) – ale ve skutečnosti obsahuje jen dvě – pozitivní a negativní. Pouze dvě třídy jsou také zmiňovány v článku zveřejněném společně s tímto datasetem, kde se hned po úvodní kapitole píše: „V tomto výzkumu nebereme v úvahu neutrální tweety v našich trénovacích nebo testovacích datech. Používáme pouze pozitivní nebo negativní tweety.“ (Go et al., 2009), takže mi není zcela jasné, kde se tyto mylné informace vzaly.

K ověření podezření, že pouze binární klasifikace sentimentu v datasetu Sentiment140 je nedostatečná, byly podniknuty následující kroky. Prvním krokem bylo získání čtyř náhodných vzorků vytvořených z původního datasetu, z nichž každý obsahoval 100 tweetů. Poté byly skryty původní anotační značky a každý tweet byl znova manuálně anotován dvěma anotátory. Posledním krokem bylo vypočítat anotátorskou shodu pro jejíž výpočet byla použita Cohenova kappa (Cohen, 1960). Tato

[2] Stránky s výsledky vyhledávání dostupné na https://paperswithcode.com/datasets?mod=texts&page=1 a https://paperswithcode.com/datasets?mod=texts&task=sentiment-analysis. [Přístup 2023-09-28]

metrika se používá pro výpočet shody mezi dvěma anotátory, takže byla vypočítána shoda pro každou kombinaci mezi anotátory a původními značkami, viz následující tabulka.

	vzorek 1	vzorek 2	vzorek 3	vzorek 4	průměr
A + B, 3 třídy	0,802	0,819	0,79	0,875	0,8215
A + B	0,855	0,758	0,899	0,7	0,803
A + původní	0,776	0,758	0,74	0,613	0,72175
B + původní	0,711	0,638	0,641	0,56	0,6375

Tabulka 1: Cohenova kappa

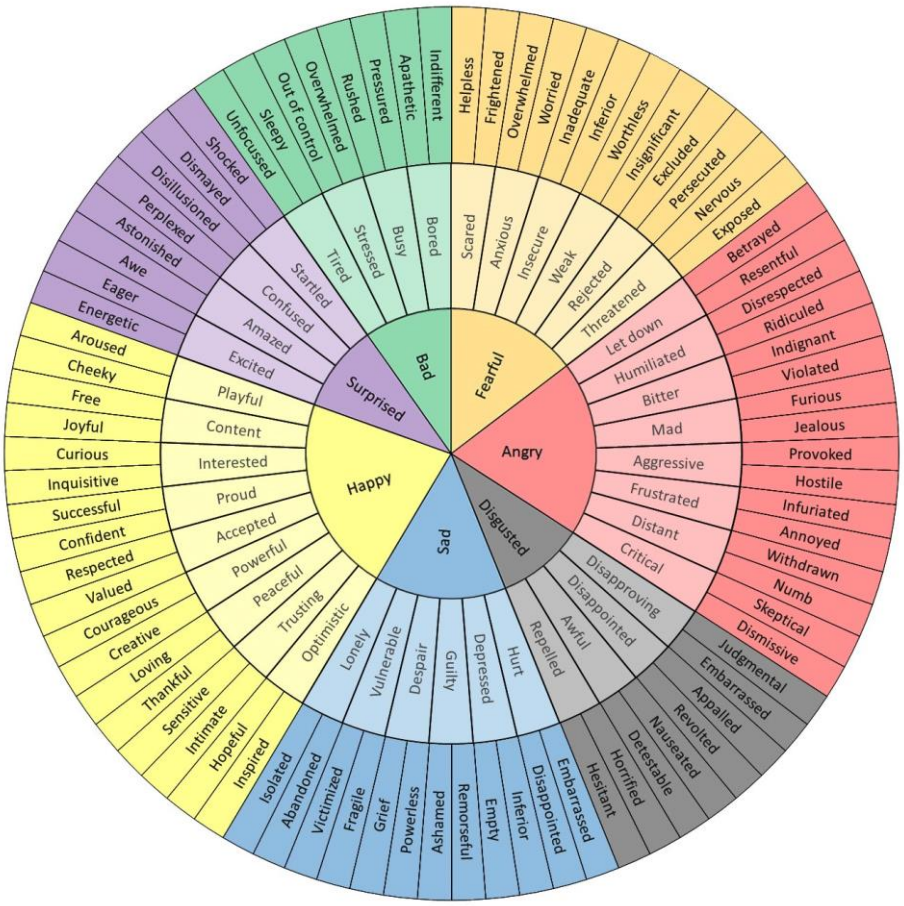
Jména řádků v tabulce zobrazují, jaká kombinace anotátorů byla použita pro výpočet shody. Řádek pojmenovaný „A + B, 3 třídy“ byl vypočítán z manuální anotace s přidanou třídou pro neutrální sentiment. Sloupce zobrazují, o jaký vzorek se jedná a poslední sloupec zobrazuje průměr všech předchozích hodnot na řádku. A a B jsou ID anotátorů pro zachování anonymity.

Ve všech případech měla manuální anotace vyšší nebo rovnou shodu mezi anotátory než původní anotace. To naznačuje, že v datasetu by se mohlo vyskytovat množství případů, ve kterých se sentiment obsahu tweetu a emotikonu neshodují, a anotátoři, kteří měli přístup pouze k textu, ze kterého byly odstraněny emotikony, udělali přesnější analýzu na základě jim dostupných informací.

I když největší míra anotátorské shody nastala u dvojice A + B v případě binární klasifikace, v průměru má lepší výsledek přidání třetí třídy sentimentu. Případy, kdy binární klasifikace dosahovala vyšší shody, jsou pravděpodobně dány subjektivností daného úkolu – při zavedení neutrální třídy se může stát, že anotátoři budou váhat, jestli je daný text ještě klasifikovatelný jako neutrální, nebo už by se měl spadat do pozitivního či negativního sentimentu. To, co jeden anotátor může vyhodnotit jako neutrální může být pro druhého už například pozitivní. K takovémuto váhání by u binární klasifikace nemělo téměř vůbec nastat, protože rozlišit pozitivní sentiment od negativního bude zpravidla mnohem jednodušší, pokud se pominou specifické případy jako například sarkasmus nebo ironie. Ovšem navzdory této větší variabilitě byla nejvyšší průměrná shoda naměřena u klasifikace na tři třídy, což nasvědčuje tomu, že pouhá binární klasifikace je nedostatečná

Mohou však existovat oblasti, kde je binární klasifikace dostatečná, ba žádoucí – například recenze zboží, u kterých se prodejce bude pravděpodobně zajímat především o ty negativní a pozitivní. Odpovídat na negativní recenze nebo komentáře bude nejspíše dobrou obchodní taktikou, protože tím prodejce dává najevo svůj zájem o produkt i své zákazníky (jestli to tak ve skutečnosti je, nebo je to pouze způsob, jak utišit nespokojené zákazníky bez opravdového zájmu o ně je věc jiná), navíc zpětná vazba může napomoci ke zlepšení daného produktu v budoucnu. Avšak recenze s neutrálním sentimentem budou obsahovat velice málo takovýchto informací, aby se vyplatilo je klasifikovat.

Když se pominou recenze, ve většině případů je nedostatečné rozlišovat pouze mezi negativním a pozitivním sentimentem. Lidé nejsou vždy pouze negativní nebo pouze pozitivní, lidé mají celé spektrum emocí, které se dají mnoha způsoby kombinovat, jak lze vidět na obrázku, který zobrazuje rozsah lidských emocí, níže.



Obrázek 1: Feelings wheel (Feelings Wheel, b.r.)

Z obrázku je jasné, že lidské pocity a emoce jsou komplexní záležitost a redukovat je na pouhé dvě kategorie může být mnohdy nedostačující, pokud je snaha vytvořit model schopný tyto emoce popisovat, jak bylo popsáno výše. Z tohoto důvodu by zavedení alespoň třetí kategorie pro neutrální sentiment mělo být minimem kategorií pro většinu případů, kde je potřeba o něco podobnější analýza.

V kontextu datasetu Sentiment140 je pochopitelné, proč byly použity pouze dvě kategorie, protože výzkumníci použili emotikony k automatické anotaci. Museli definovat dvě sady emotikonů a přiřadit jim sentiment. Je poměrně jasné, které emotikony mohou znamenat pozitivní a které negativní sentiment, ale u neutrálního sentimentu už to tak jasné není.

Jak to potom tedy je s méně specifickými případy, kdy je anotace prováděna manuálně, a ne pomocí emotikonů? Při ruční anotaci bude tendence celý proces co nejvíce zjednodušovat, protože to je

proces drahý a časově náročný, ale ani to by neměla být výmluva ke zjednodušování lidských emocí na pouhou pozitivní a negativní.

4.3. Doménová závislost dat

Ukázalo se, že data potřebná k natrénování kvalitního modelu, který by měl fungovat v automobilovém prostředí, jsou poměrně silně doménově závislá, to znamená, že nejlepší výsledky by měl dosahovat model natrénovaný na datech především z automobilního prostředí, nebo na dostatečném objemu obecných dat (i v takovém případě je ale potřeba, aby tato obecná data obsahovala alespoň z části data vztahující se k automobilnímu průmyslu). Model natrénovaný na filmových recenzích tedy sice dokáže provádět analýzu sentimentu s poměrně dobrými výsledky, tyto výsledky budou ale znatelně horší, jakmile se takto natrénovanému modelu předloží data, která nejsou recenze. Podobně to bude fungovat s každým dalším datasetem, jak bude ukázáno v dalších částech.

4.4. Tvorba vlastních datasetů

Kvůli problémům s doménovou závislostí byly pokusy o vytvoření vlastního datasetu, který by lépe sloužil potřebám konkrétního zadání. I když nějaké datasety, které by mohly posloužit pro trénování modelů pro automobilový průmysl, existují, mají licenci, která neumožňuje jejich komerční použití – například již zmíněný Amazon dataset složený z recenzí produktů.

Kvůli těmto omezením bylo přistoupeno k vytváření vlastních datasetů. Možností odkud čerpat data se naskytovalo hned několik, avšak každá se svými pro a proti.

Prvním nápadem bylo stahovat komentáře pod videi recenzí automobilů na YouTube, jenomže YouTube neposkytuje žádné API ke stahování komentářů, takže by se muselo přistoupit k web-scrapingu, což by celý proces zpomalovalo – YouTube totiž nenačítá komentáře, dokud se stránka neposune dolů na sekci komentářů, bylo by tedy potřeba vytvořit skript pomocí nějaké Python knihovny umožňující automatizaci webového prohlížeče (například Selenium). Navíc komentáře pod YouTube videi se nemusí vždy vyjadřovat k obsahu videa, ale i k jeho formě a zpracování, což by bylo mimo kýžené zaměření datasetu. Navíc by bylo potřeba ručně vybírat videa, z kterých komentáře stahovat. Kvůli těmto komplikacím bylo tedy od YouTube rychle upuštěno.

Další možností bylo využít nějakou ze sociálních sítí, z kterých nakonec byl vybrán Twitter jako vhodný kandidát. Twitter na rozdíl od YouTube sice poskytuje API ke stahování příspěvků, ale jsou zde různá omezení na počet příspěvků na časové období, případně omezení na komerční použití. První vyzkoušenou knihovnou na získávání tweetů byla knihovna Tweepy (Harmon, Roesslein), která je integrovaná do Twitter API, uživatel tedy potřebuje zažádat o vývojářský účet na Twitteru, získat ověřovací údaje a s jejich pomocí poté může přes Tweepy stahovat příspěvky. Je zde ale limit na počet stažených tweetů a Tweepy také nemá přístup k tweetům starším než týden, není to tedy vhodný nástroj k vytvoření dostatečně velkého datasetu.

Problémy, které má Tweepy, však řeší knihovna snscrape, která dovoluje stahovat příspěvky z různých sociálních sítí a zahrnuje i Twitter. Není zde žádný limit na časové období dostupných tweetů ani na jejich počet, protože snscrape nepoužívá API, ale emuluje webové rozhraní (v GitHub dokumentaci toto sice napsáno není[3], ale slovo „scrape“ v názvu prozrazuje něco o fungování knihovny), takže to je jako kdyby si uživatel otevřel Twitter v prohlížeči a příspěvky prohlížel ručně[4].

Díky metadatům obsažených v tweetech je také lze filtrovat podle jazyka, času apod., takže je poměrně snadné vytvořit jednojazyčný dataset. Bylo tedy vytvořeno několik datasetů, k jejichž vyhledání byl vždy použit název nějakého konkrétního modelu auta od různých automobilek. Tímto způsobem se získalo 10000 tweetů pro každý z vybraných modelů auta a celkem jich bylo staženo 50000. Následovala ruční anotace dvěma anotátory, avšak nejdříve bylo ještě potřeba texty pro anotaci připravit. Všechna označení profilů byla nahrazena za *@user* a všechny hypertextové odkazy byly odstraněny. Tweety byly dále rozděleny na jednotlivé věty pomocí teček, protože jednoznačné určení sentimentu na úrovni věty je jednodušší než na úrovni dokumentu (Lesch, 2017, s. 89), tedy celého tweetu v kontextu této práce. Brát tweet jako samostatný dokument se může zdát poněkud zvláštní, alespoň co se délky týče, ale poměrně často se vyskytovaly případy, kdy jeden tweet obsahoval více vět, z nichž každá měla jiný sentiment, což ve výsledku značně komplikuje klasifikaci. Rozdělení tweetů na věty s tímto problémem pomohlo, avšak ho to nevyřešilo úplně, protože vyjádření rozdílného sentimentu se může dít i na úrovni souvětí. Samozřejmě by bylo možné celé věty dále dělit na menší celky, avšak pro zachování jednoduchosti bylo dělení ponecháno pouze na úrovni vět.

Takto zpracované tweety již byly připraveny k anotaci. Tweety byly nejprve anotovány podle relevance na relevantní a nerelevantní, protože hlavním cílem bylo získat dataset silně zaměřený na automobilní doménu a rozdělením tweetů na jednotlivé věty nedávaly některé věty v izolaci v takovém datasetu smysl, jako těchto několik následujících příkladových vět:

1. and a wild animal
2. @user @user Just curious.
3. Maybe #PutinsWar will change this?

Po anotaci relevance následovala anotace sentimentu. Anotovalo se do čtyř tříd: pozitivní, neutrální, negativní a neurčitý sentiment. Třída pro neurčitý sentiment byla zavedena především ze dvou důvodů, prvním z nich byly již zmíněné věty se smíšeným sentimentem, kdy věty obsahovaly jak pozitivní, tak negativní sentiment a bylo by tedy nesprávné je klasifikovat jako kteroukoliv polaritu, druhým důvodem byly věty, u kterých v izolaci nebylo možné sentiment jednoznačně určit a sentiment by se mohl jednoduše měnit v závislosti na kontextu. Texty anotované v první fázi jako nerelevantní byly ignorovány a dohromady bylo anotováno 2000 textů. Kvůli časové intenzivnosti manuální anotace

[3] Dokumentace knihovny viz https://github.com/JustAnotherArchivist/snscrape. [Přístup 2023-09-28]
[4] Kvůli změnám na webu twitter.com z července 2023, kdy již nelze bez účtu prohlížet obsah, snscrape pro Twitter nefunguje viz https://github.com/JustAnotherArchivist/snscrape/issues/996. [Přístup 2023-09-28]

nebylo možné anotaci provést pro část větší než zmíněných 2000 textů. K analýze se poté použily pouze ty texty, kde došlo ke shodě obou anotátorů. Z původních 2000 textů tedy bylo 1497 relevantních, z nich 104 mělo neurčitý sentiment, 807 neutrální sentiment, 247 pozitivní sentiment, 89 negativní sentiment a u zbylých 250 textů se anotace lišily. Výsledky anotace lépe shrnuje následující tabulka.

Anotováno	2000
Relevantní	1497
Pozitivní	247
Negativní	89
Neutrální	807
Neurčitý	104
S určeným sentimentem	1143
S určenou polaritou	336

Tabulka 2: výsledky anotace

Tabulka také shrnuje u kolika textů byl určen sentiment (tedy součet počtů pozitivních, negativních a neutrálních textů) a u kolika textů polarita (pouze pozitivní a negativní sentiment). Počet textů s určeným sentimentem je důležitý pro případ, kdyby se trénoval kompletně vlastní model a nebylo by tedy potřeba omezovat se na problematické dvě třídy, zatímco počet textů s určenou polaritou je důležitý pro případ, kdyby se takto vytvořená doménová data použila při trénování modelu ve spojení s existujícími datasety, nebo pro ověření výkonu modelů natrénovaných na existujících datasetech. Z čísel vyplývá, že sentiment, avšak převážně neutrální, byl určen u 57 % anotovaných textů, a pouze u necelých 17 % byla určena polarita. Navíc jsou vzniklá data velice nevyvážená, což klade další požadavky na vytvoření většího datasetu a případně také na použití nějaké metody pro zmírnění vlivu nevyváženého datasetu, jako třeba oversampling nebo undersampling.

Vytváření vlastního datasetu uvnitř malého týmu lidí se tedy ukázalo jako nevhodný způsob pro vypořádání se s problémy spojenými s nedostatkem dat pro automobilní doménu. Největším problémem byla časová náročnost manuální anotace, což by se dalo v případě nutnosti řešit outsourcingem či crowdsourcingem. I tento malý dataset byl však využit pro analýzy v části 5.5.

4.5. State-of-the-art přístupy k analýze sentimentu

S neustále rostoucí výpočetní kapacitou moderních počítačů roste i poptávka po velkých AI modelech. Současné state-of-the-art přístupy k analýze sentimentu jsou neuronové sítě a transformery (Vaswani et al., 2017), což je také typ neuronové sítě. Tyto velké modely však potřebují obrovské množství dat, často stovky gigabytů textu, k natrénování modelu s vysokou přesností (Liu et al., 2019). Manuální anotace v takovémto měřítku je téměř nemožná, protože by zabrala extrémní množství času a pracovní síly.

Tradiční způsob trénování neuronové sítě je za pomoci učení s učitelem, kdy je k dispozici anotovaný dataset, takže jsou známy správné odpovědi pro pozdější kontrolu modelu. Takový dataset lze rozdělit na trénovací část, na které se model natrénuje, a na validační část, na které se model validuje, čímž se zkoumá jeho přesnost. Díky anotovanému datasetu je kontrola přesnosti modelu otázkou pouhého porovnání výstupního tagu se vstupním tagem. Učení s učitelem se ale bohužel rychle stává nepoužitelným, jakmile dojde řada na větší datasety. Z tohoto důvodu se pro velké modely používá učení bez učitele.

V případě učení bez učitele jsou data a anotace získány víceméně bez lidského zásahu. Jedním z běžných přístupů je shlukování. Shlukovací algoritmy seskupují data do shluků na základě vlastností vypozorovaných ze vzorů, které jsou přítomny v datech (What Is Unsupervised Learning?, b.r.). I když je tento přístup technologicky i výpočetně náročnější než učení s učitelem, je i přes to preferovaný pro velké objemy dat, protože nevyžaduje dohled ani zásah člověka, díky čemuž je učení bez učitele rychlejší než manuální anotace. Jednou velkou výhodou těchto modelů trénovaných bez učitele je to, že jsou méně ovlivněny redukcionistickými tendencemi přítomnými u manuální anotace, které byly zmíněny v části 4.2, protože je možné si vybrat do kolika shluků by data měla být seskupena. V případě analýzy sentimentu je to potom otázka změny jedné proměnné v kódu, aby byla klasifikace na tři třídy místo problematických dvou.

Je však důležité zmínit, že modely trénované bez učitele nedělají predikce samy o sobě, ale pouze shlukují data k sobě a interpretace těchto shluků je na člověku, který na daná data kouká. Takový člověk pak může prozkoumat výsledky a přidat anotace k jednotlivým shlukům dat, a s těmito novými informacemi lze pomocí daného modelu provádět predikce.

Hypoteticky vzato by mělo být možné mít tolik shluků dat jako je emocí v Obrázek 1, přeci jen existují modely schopné klasifikovat obrázky do desetitisíců kategorií (Wu et al., 2020). Potíž je v tom, že emoce jsou subjektivní a prožíváme je převážně vnitřně, což ztěžuje jejich identifikaci a přesnou klasifikaci.

5. Použité metody

Ze zadání bylo jasné, že neuronové sítě a umělá inteligence by měly být upřednostněny před ostatními metodami, takže primární cíl byl jasný. Přesto byly prozkoumány i ostatní přístupy k analýze sentimentu, aby bylo budováno vlastní povědomí a know-how v této problematice. Mimo to má každá metoda své pro a proti, takže i jednoduché metody mohou najít uplatnění ve speciálních případech, případně by se daly využít k automatické anotaci dat pro tvoření vlastních datasetů, které by následně sloužily k vytvoření komplexnějších řešení, například k natrénování již zmíněných neuronových sítí. Následující kapitoly budou popisovat použité metody a jejich výsledky.

5.1. Slovníkové metody

Slovníkové metody jsou, jak již název napovídá, založeny na slovnících, které ke každému obsaženému slovu obsahují ještě skóre daného slova vystihující intenzitu a polaritu sentimentu. Pro analýzu sentimentu na úrovni slov lze tedy jen jednoduše zaměnit slovo za jeho skóre, ale pro analýzu na vyšších úrovních už jsou potřeba i nějaká pravidla pro výpočet celkového sentimentu analyzovaného textového celku. Tato pravidla zpravidla zahrnují detekci intenzifikátorů významu (slova jako velmi, vážně, nejvíce) a zeslabovačů významu (slova jako málo, trochu, částečně), detekci negace apod. Tyto výrazy poté modifikují původní sentiment výrazů, vedle kterých se vyskytují, takže například výraz „velmi často“ bude mít při výpočtu sentimentu větší váhu než výraz „méně často“. Kvůli tomu, že slovníkové přístupy nespadají do metod strojového učení, nevyžadují žádná trénovací data, což je jejich nespornou výhodou – není tedy potřeba nejprve natrénovat klasifikátor a až tento klasifikátor používat k predikcím sentimentu.

5.1.1. TextBlob

Jednou z možných implementací této metody je python knihovna TextBlob (Loria, 2020), která má slovník o velikosti necelých 3000 slov. Výsledkem je hodnota v rozmezí od -1 do +1 znázorňující sentiment analyzovaného textu. Mimo sentiment jako takový také dokáže vyhodnotit míru subjektivity dané promluvy, ale jak již bylo zmíněno v úvodu, subjektivita zde není předmětem zkoumání.

5.1.2. VADER

Další implementací této metody je python knihovna VADER (Hutto, Gilbert, 2014). Zásadní rozdíl v porovnání s TextBlob knihovnou je, že VADER se zaměřuje hlavně na příspěvky ze sociálních sítí, takže umí rozpoznávat sentiment emotikonů, verzálek, opakování slov nebo opakování interpunkčních znamének. Také slovník, který VADER používá, má více než dvojnásobnou velikost v porovnání s TextBlob a obsahuje lehce přes 7500 tokenů (v tomto případě už ne pouze slov, protože slovník zahrnuje třeba již zmiňované emotikony). VADER také nabízí trochu podrobnější analýzu, protože výsledkem jsou čtyři hodnoty – positive, negative, neutral a compound. Positive, negative a neutral uvádějí poměr daných sentimentů v analyzovaném textu a vždy mají v součtu 1, zatímco compound hodnota uvádí celkový sentiment analyzovaného textu a je v rozmezí od -1 do +1.

5.1.3. Porovnání TextBlob a VADER

Pro ověření přesnosti analýzy sentimentu pomocí TextBlob a VADER knihoven byly použity IMDB a Sentiment140 datasety. Sentiment140 je dostupný i jako csv tabulka, takže načtení datasetu je poměrně snadné a obnáší pouze načtení souboru a zobrazení relevantních sloupců. V případě IMDB datasetu je

vyžadována trocha přípravy – data jsou po načtení vektorizovaná, takže je potřeba vektory zpět převést na text, o což se stará následující kód.

```
(x_train, y_train), (x_test, y_test) = imdb.load_data()
word_index = imdb.get_word_index()
reversed_index = dict([(value, key) for (key, value) in word_index.items()])
data = np.concatenate((x_train, x_test), axis=0)

def reverse(reversed_index, data):
    return ' '.join([reversed_index.get(i - 3, '#') for i in data])

texts = [reverse(reversed_index, review) for review in data]
```

Kód 1

Kód načte trénovací i testovací části datasetu a index slov – slovník ve formátu {slovo : číselný index}. Aby bylo možné tento index použít k převedení číselných reprezentací slov zpět na slova, je potřeba si vytvořit nový slovník, kde původní klíče budou hodnotami a naopak. Takto upravený index se poté používá ve funkci `reverse`, která každou vektorizovanou recenzi převede zpět na slova a vrátí ji jako jeden souvislý text, který je pro lidi čitelný a už je možné ho analyzovat.

Kvůli velikosti Sentiment140 datasetu bylo pro účely této analýzy použit pouze vzorek o velikosti 100000 tweetů, aby se zkrátil výpočetní čas. Výsledky obou knihoven byly měřeny pomocí přesnosti za použití funkce `accuracy_score` z knihovny scikit-learn (Pedregosa et al., 2011). V obou datasetech je sentiment anotovaný pomocí čísel, v případě IMDB datasetu 1 = pozitivní a 0 = negativní sentiment, u Sentiment140 datasetu 4 = pozitivní a 0 = negativní sentiment. Naměřené hodnoty sentimentu větší než nula jsou brány jako pozitivní sentiment a hodnoty menší než nula jsou brány jako negativní. U obou knihoven se v některých případech může stát, že naměřená hodnota je přesně nula, což se třeba u Sentiment140 datasetu dělo u velmi krátkých textů, které obsahovaly pouze pro obě knihovny neznámá slova. Z tohoto důvodu nebyly brány v potaz veškeré texty, u kterých kterákoliv z knihoven naměřila přesnou nulu. Obě knihovny tak analyzují ty stejné texty a výsledky jsou objektivnější a férovější. Výsledky shrnují následující tabulky.

IMDB	Pozitivní texty	Negativní texty	Celkem textů	TextBlob acc	VADER acc
Původní stav	25000	25000	50000		
Odebrány nulové hodnoty	24984	24963	49947	69,24 %	69,80 %

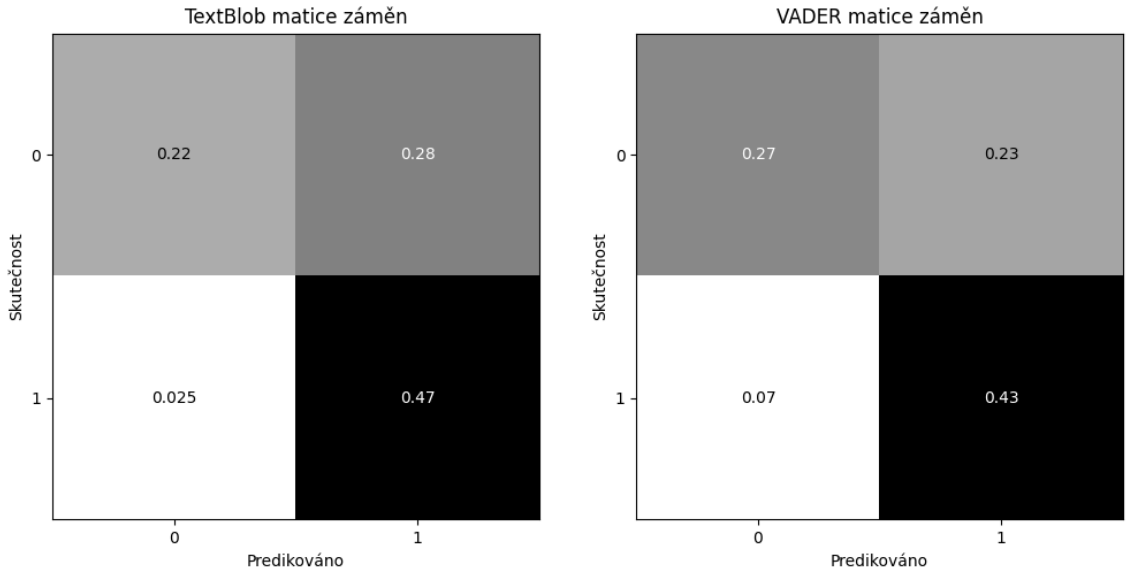
Tabulka 3: přesnost slovníkových metod na IMDB datasetu

Sentiment140	Pozitivní texty	Negativní texty	Celkem textů	TextBlob acc	VADER acc
Původní stav	800000	800000	1600000		
Zkoumaný vzorek	50000	50000	100000		
Odebrány nulové hodnoty	29082	27717	56799	68,85 %	71,70 %

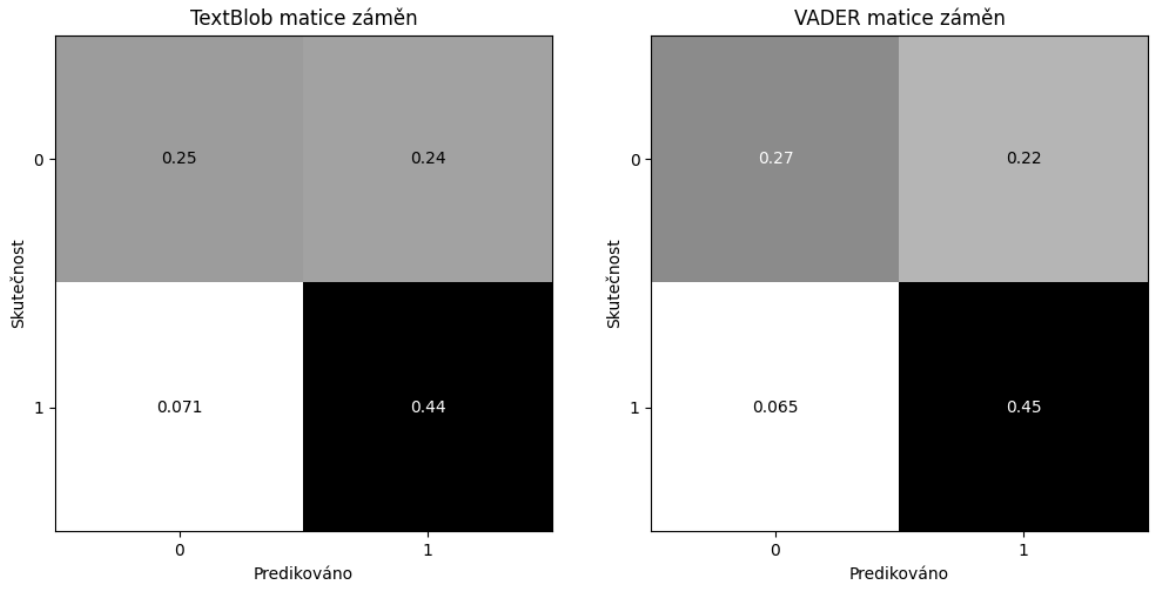
Tabulka 4: přesnost slovníkových metod na Sentiment140 datasetu

Tabulka 3 shrnuje analýzu provedenou na IMDB datasetu a lze z ní vyčíst, že v datasetu je obsaženo 53 textů, u kterých alespoň jedna knihovna naměřila přesnou nulu. Co se týče výsledků, tak je zřejmé, že obě knihovny dosahují poměrně podobných výsledků dosahujících téměř 70 %, avšak knihovna VADER dosáhla lepší přesnosti.

Tabulka 4 shrnuje analýzu provedenou na Sentiment140 datasetu. Jak již bylo řečeno, byl použit pouze vyvážený vzorek vytvořený z původního datasetu. I když původní velikost zkoumaného vzorku byla v porovnání s IMDB datasetem dvojnásobná, tak po odebrání nevhodných textů byla velikost vzorku téměř poloviční. To je nejspíše způsobeno tím, že tweety jsou zpravidla velice krátké texty v porovnání s recenzemi, a u kratších textů je větší šance, že budou obsahovat pouze nerelevantní tokeny. Ve srovnání s analýzou IMDB datasetu si TextBlob pohoršil a VADER polepšil. Potvrdilo se tím tedy tvrzení, že by VADER měl dosahovat lepších výsledků na datech pocházejících ze sociálních sítí. Lze předpokládat, že kdyby z tweetů nebyly odstraněny emotikony, tak by VADER dosáhl ještě o něco lepších výsledků. Další analýza, která nabízí o něco přesnější náhled na fungování obou knihoven, spočívá ve vytvoření matic záměn, ty byly vytvořeny pomocí knihoven scikit-learn a Matplotlib (Hunter, 2007).



Obrázek 2: matice záměn slovníkových metod na IMDB datasetu



Obrázek 3: matice záměn slovníkových metod na Sentiment140 datasetu

Obrázky zachycují matice záměn pro obě knihovny. Hodnoty zobrazené v maticích jsou normalizovány podle celkového počtu položek, takže čísla znázorňují, v jakém poměru jsou falešné/pravdivé negativy/pozitivy. Na obrázcích je vidět, že obě knihovny poměrně dobře odhalují pozitivní sentiment, což ale částečně bude dáno tím, že obě metody predikují pozitivní sentiment nevyváženě často. TextBlob predikuje pozitivní sentiment v 75 % případů u IMDB datasetu a v 68 % případů u Sentiment140 datasetu, zatímco u knihovny VADER to je 66 % a 67 %, tedy nižší v obou případech. VADER má také o něco více konzistentní výsledky napříč datasety a u obou z nich lépe predikuje negativní sentiment než TextBlob, zatímco TextBlob predikoval o něco lépe pozitivní sentiment u IMDB datasetu. Z těchto matic tedy vyplývá, že VADER má lepší výsledky při predikci negativního sentimentu, zatímco TextBlob odhaluje o něco lépe sentiment pozitivní.

U knihovny VADER se však počítá s tím, že hodnoty blízké nule – konkrétně hodnoty v rozmezí -0,05 až +0,05 – budou brány jako neutrální sentiment (Hutto, Gilbert, 2014, s. 224). TextBlob v tomto ohledu žádné rozmezí hodnot pro neutrální sentiment nedefinuje (Loria, 2020), takže následující analýza, u které nebyly brány v potaz neutrálně hodnocené texty, byla tedy provedena pro obě knihovny stejně. Výsledky opět shrnují následující tabulky.

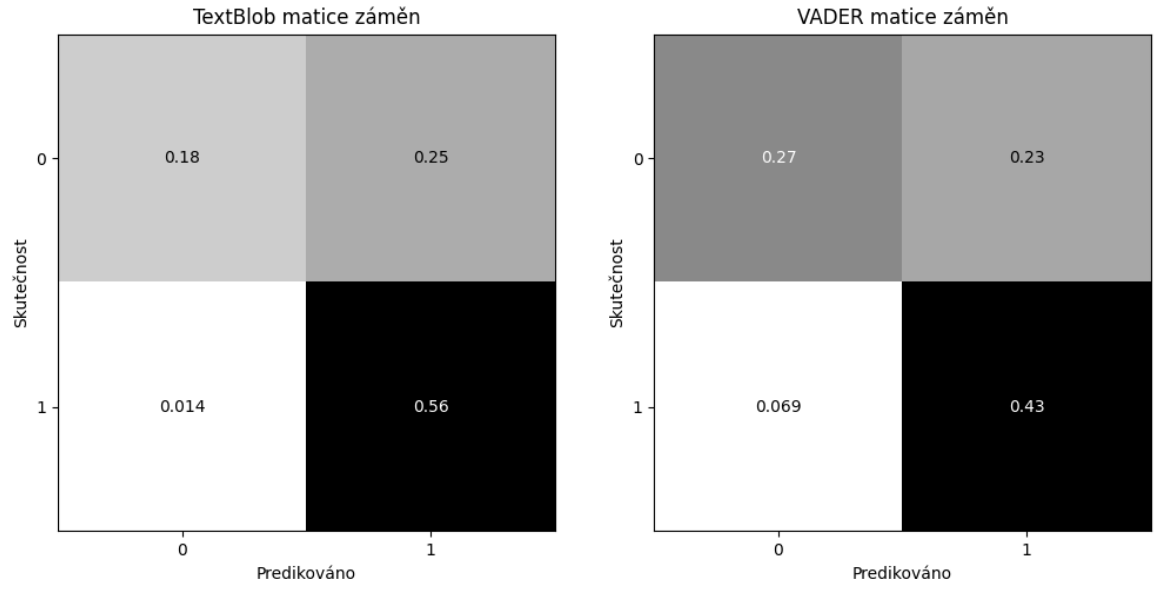
IMDB	Pozitivní texty	Negativní texty	Celkem textů	TextBlob acc	VADER acc
Původní stav	25000	25000	50000		
TextBlob bez neutrálního sent.	22738	16946	39684	73,87 %	
VADER bez neutrálního sent.	24844	24633	49477		70,01 %

Tabulka 5: přesnost slovníkových metod na IMDB datasetu s vynecháním neutrálních textů

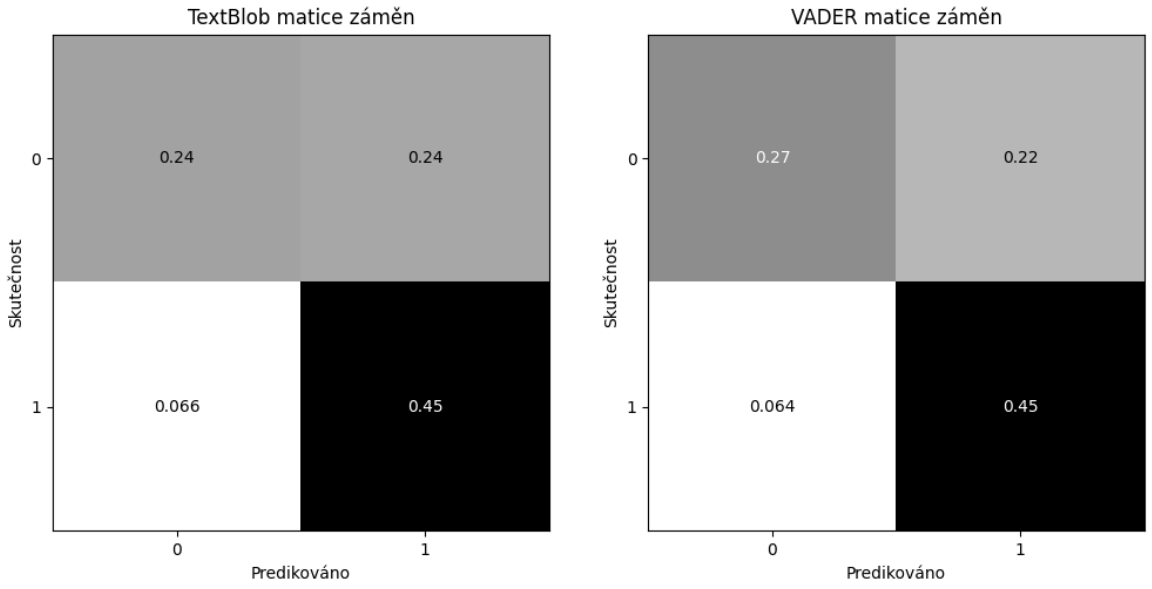
Sentiment140	Pozitivní texty	Negativní texty	Celkem textů	TextBlob acc	VADER acc
Původní stav	800000	800000	1600000		
Vzorek	50000	50000	100000		
TextBlob bez neutrálního sent.	28048	25983	54031	69,65 %	
VADER bez neutrálního sent.	28892	27240	56132		71,96 %

Tabulka 6: přesnost slovníkových metod na Sentiment140 datasetu s vynecháním neutrálních textů

Tabulky opět zachycují, kolik celkem zbylo textů po odstranění těch neutrálně hodnocených a v případě TextBlob knihovny u IMDB datasetu je velká nevyváženost mezi zbylými pozitivními a negativními texty, což má nejspíš také za následek vysokou naměřenou přesnost. 57 % z tohoto vzorku totiž tvořily pozitivní texty, a jak již bylo zmíněno u předchozí analýzy, TextBlob se jeví jako o trochu lepší nástroj na predikci pozitivního sentimentu v porovnání s knihovnou VADER, takže tato převaha pozitivních textů ve vzorku pravděpodobně zkresluje naměřenou přesnost ve prospěch TextBlob knihovny. Oproti tomu VADER jako neutrální hodnotil velice málo textů a má velmi podobný výsledek jako v předchozí analýze. U Sentiment140 datasetu jsou vzorky vyváženější a výsledky jsou opět dost podobné a konzistentní s předešlým měřením z Tabulka 4, i zde se tedy potvrdilo, že VADER dosahuje lepších výsledků na datech ze sociálních sítí. Podrobnější vhled opět nabízí matice záměn.



Obrázek 4: matice záměn slovníkových metod na IMDB datasetu bez neutrálních textů



Obrázek 5: matice záměn slovníkových metod na Sentiment140 datasetu bez neutrálních textů

U IMDB datasetu jsou v maticích vidět poměrně velké rozdíly, TextBlob predikoval pozitivní sentiment v 81 % případů, zatímco VADER pouze v 66 %, avšak tento rozdíl bude nejspíš opět způsoben nevyvážeností vzorku pro TextBlob. I zde se ale potvrzuje, že TextBlob predikuje lépe pozitivní sentiment a VADER lépe negativní. U Sentiment140 datasetu nejsou rozdíly mezi výsledky knihoven tak velké, obě správně predikovaly pozitivní sentiment stejně často, avšak VADER opět dosahuje lepších výsledků v případě negativního sentimentu a má výsledky velice podobné s předchozími měřeními.

Z provedených analýz tedy vyplývá, že obě metody predikují pozitivní sentiment ve většině případů – pozitivní sentiment byl predikován v 66-81 % případů. Dále se ukázalo, že VADER má velice konzistentní výsledky a predikuje negativní sentiment lépe než TextBlob – negativní sentiment predikoval správně vždy ve 27 % případů a pozitivní mezi 43-45 %, zatímco TextBlob správně predikoval negativní sentiment u 18-25 % textů a pozitivní u 44-56 % textů, avšak krajní hodnoty 18 a 56 % jsou způsobeny nevyvážeností vzorku po odstranění neutrálně hodnocených textů, takže nejsou úplně vypovídající o fungování knihovny TextBlob. Pokud se tyto zkreslené hodnoty neberou v potaz, tak TextBlob správně predikoval negativní sentiment ve 22-25 % případů a pozitivní ve 44-47 % případů, což stále podporuje tvrzení, že TextBlob lépe vyhodnocuje pozitivní sentiment a VADER negativní sentiment. Kromě toho jednoho problematického případu také VADER dosáhl vždy vyšší přesnosti než TextBlob, takže podle této analýzy se jeví být lepším nástrojem pro analýzu sentimentu. Pro posílení těchto tvrzení by však bylo vhodné provést podobnou analýzu na více vzorcích, aby se zredukoval možný vliv náhody. I přes obstojnou přesnost mají tyto metody své slabiny a hlavní z nich je omezená velikost slovníku, a tudíž velká šance narazit na neznámá slova, která nebudou do analýzy zahrnuta. Metody se tedy jeví jako velice nevhodné pro případ automatické anotace textů pro vytvoření

vlastního datasetu, protože lze očekávat, že zkreslení, která byla přítomna v popsaných analýzách by byla přítomna i při automatické anotaci nových dat. Slovníkové metody navíc neposkytují žádnou míru pravděpodobnosti o správnosti dané klasifikace, která by byla potřeba k efektivní automatické anotaci.

5.2. Statistické a prostorové metody

Dalším souborem přístupů k analýze sentimentu jsou metody založené na statistice či rozmístění dat v prostoru, kam spadá například SVM (Support Vector Machines, česky metoda podpůrných vektorů), Naive Bayes (česky naivní bayesiánský klasifikátor) nebo k-NN (k-nearest neighbors, česky algoritmus k-nejbližších sousedů). Tyto metody se dají považovat za takový mezistupeň mezi slovníkovými metodami a neuronovými sítěmi co se komplexnosti fungování týče. Všechny tři z uvedených metod spadají pod učení s učitelem, potřebují tedy nějaká anotovaná trénovací data, na kterých se nejprve natrénuje klasifikátor a ten se poté použije ke klasifikaci sentimentu na nových, neznámých datech. Jedním ze zásadních rozdílů proti slovníkovým metodám je, že tyto metody neumí pracovat přímo s texty, ale jen s jejich číselnými reprezentacemi. Pro každou z následujících metod tedy bylo potřeba text nějakým způsobem vektorizovat neboli reprezentovat pomocí čísel. Taková potřeba vektorizace textů přidává další úroveň komplexnosti těchto řešení, protože na výběr je mnoho způsobů vektorizace a každá z nich má své pro a proti. Zmíním však alespoň některé z nich.

Velice jednoduchou metodou pro vektorizace je vytvoření frekvenčního slovníku z trénovacích dat a nahrazení slov jejich rankem, takže nejčastější slovo = 1, druhé nejčastější slovo = 2 atd. V takové formě je právě IMDB dataset. Jednou nevýhodou tohoto postupu je však to, že vzniklé vektory nebudou zarovnané na stejnou délku (pokud tedy všechny texty v datasetu neobsahují stejný počet slov, což je velmi nepravděpodobné). Stejná délka vstupních vektorů je u statistických metod nutná, takže je potřeba všechny vektory nějakým způsobem zarovnat na stejnou délku, čehož se dá opět docílit různými způsoby – zkrátit všechny texty (zpravidla na délku nejkratšího textu), doplnit do vektorů prázdné hodnoty (tradičně až do délky nejdelšího textu), případně kombinace těchto dvou.

Další možností je použít bag of words model (Harris, 2015), který zachytává frekvenci výskytů slov v textech. Pro získání bag of words vektorů je nutné si nejdříve vytvořit jakýsi globální slovník ze všech unikátních slov neboli typů (nikoliv tokenů) v trénovacích textech. S tímto slovníkem se poté porovnávají všechny texty, pro každé slovo ze slovníku nevyskytující se v daném textu se do vektoru zapíše 0, pro každé slovo vyskytující se jak ve slovníku, tak v textu se do vektoru zapíše počet výskytů daného slova. Používá se i binární bag of words model, kdy se do vektorů zaznamenává pouze výskyt a absence slov, nikoliv jejich frekvence. Pomocí bag of words metody se tedy vytvoří vektory o stejné velikosti, takže je možné bez dalších úprav použít k trénování klasifikátoru. I přes to má však bag of words jeden nedostatek, a tím je ztráta informace o slovosledu. Pro češtinu to při ponechání původních tvarů slov není zase takový problém díky skloňování, například věty „lovec zabil medvěda“ a „medvěd zabil lovce“ nejsou tvořeny stejnými slovy, což však neplatí pro anglické ekvivalenty „the hunter killed

the bear“ a „the bear killed the hunter“, u kterých jsou věty tvořeny ze stejných slov, jejich význam je opačný, ale budou mít stejný bag of words vektor. I v rámci češtiny to ale může být problém, protože při přípravě dat se často používají lemmatizátory a další nástroje, které převedou vyskloňovaná slova do jejich základního tvaru, z původních vět by poté takovým zpracováním vznikly věty „lovec zabít medvěd“ a „medvěd zabít lovec“, které by také měly stejné bag of words vektory, protože obsahují ta samá slova. Informace o slovosledu by se daly zachytit například použitím bigramů namísto unigramů, čímž však poroste velikost slovníku a tím pádem i výsledných vektorů, což bude klást ještě větší požadavky na RAM paměť, takže ne vždy bude použití bigramů vhodným řešením.

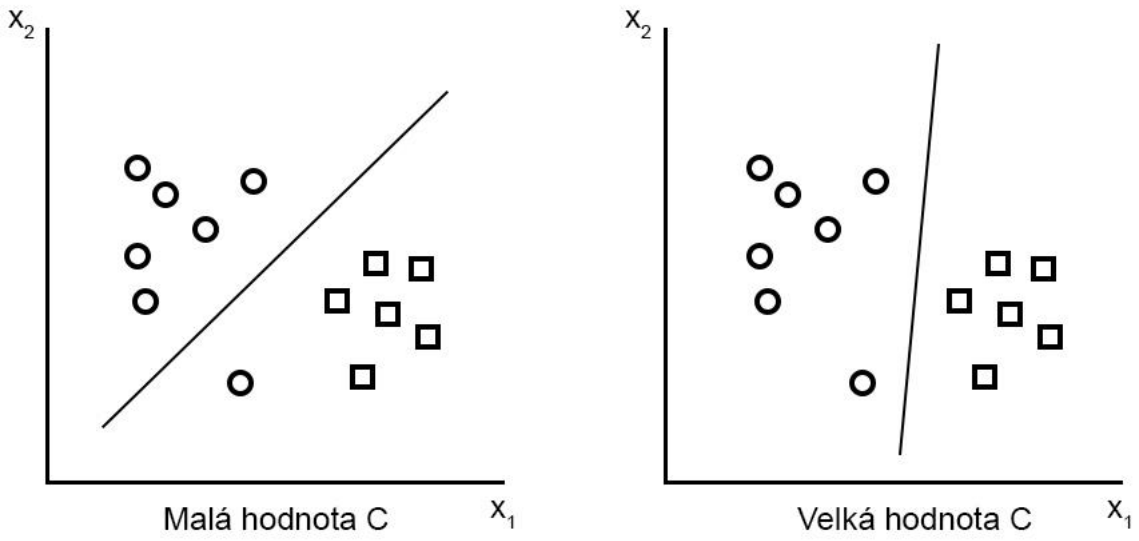
Poslední metodou, která bude zmíněna, je použití slovních embeddingů. Embedding v tomto kontextu znamená reprezentace významu slova ve vektorovém prostoru takovým způsobem, aby podobná slova byla umístěna blízko sebe (Jurafsky, Martin, 2023a). Myšlenka, že podobná slova mají i podobný kontext však není vůbec nová, protože tento koncept se objevuje již v 50. letech 20. století (Firth, 1957), pořádné uplatnění v NLP však našel až v posledních letech s představením technik pro vytváření a používání slovních embeddingů jako Word2Vec (Mikolov et al., 2013a; Mikolov et al., 2013b)[5] a fastText (Bojanowski et al., 2016)[6]. I když se obě techniky svou implementací liší, dělají alespoň v obecné rovině to stejné, tedy se snaží naučit, jak smysluplně umístit slova do vektorového prostoru a jak tyto vztahy číselně vyjádřit. Knihovna fastText má proti Word2Vec jednu zásadní výhodu, kterou je rozpoznávání neznámých slov, která nebyla v trénovacím datasetu. Word2Vec umí pracovat pouze se slovy, které se naučil při trénování, a neporadí si s žádnými novými slovy, která v trénovacím datasetu nebyla obsažena. FastText tohle řeší velice efektivně tím, že nepracuje na úrovni celých slov, ale používá dvojice znaků, ze kterých lze poté celá slova skládat.

5.2.1. SVM – Support Vector Machines

První z použitých statistických metod je SVM (Cortes, Vapnik, 1995), které se běžně používá ke klasifikaci a regresní analýze (Kecman, 2005, s. 10-11). SVM může využívat více různých jádrových funkcí, které dovolují transformovat data do nového prostoru a tím separovat i jinak lineárně neseparovatelná data, v této práci je však použito pouze lineární SVM. To se mezi skupinami dat snaží nalézt novou, optimální osu, která kolem sebe bude mít co největší hraniční pásmo (Kecman, 2005, s. 11-12; Cortes, Vapnik, 1995, s. 278), které je definováno pomocí bodů neboli podpůrných vektorů (anglicky support vectors, odtud název) ležících nejblíže okrajům tohoto pásma (Kecman, 2005, s. 15-16). Z toho vyplývá, že SVM není ovlivněno odlehlými hodnotami daleko od osy, protože záleží jen na bodech ležících na okrajích hraničního pásma. To vše platí však pouze na předpokladu, že data jsou lineárně oddělitelná a není žádný překryv mezi body z datových skupin, ale to se v praxi zřídkakdy

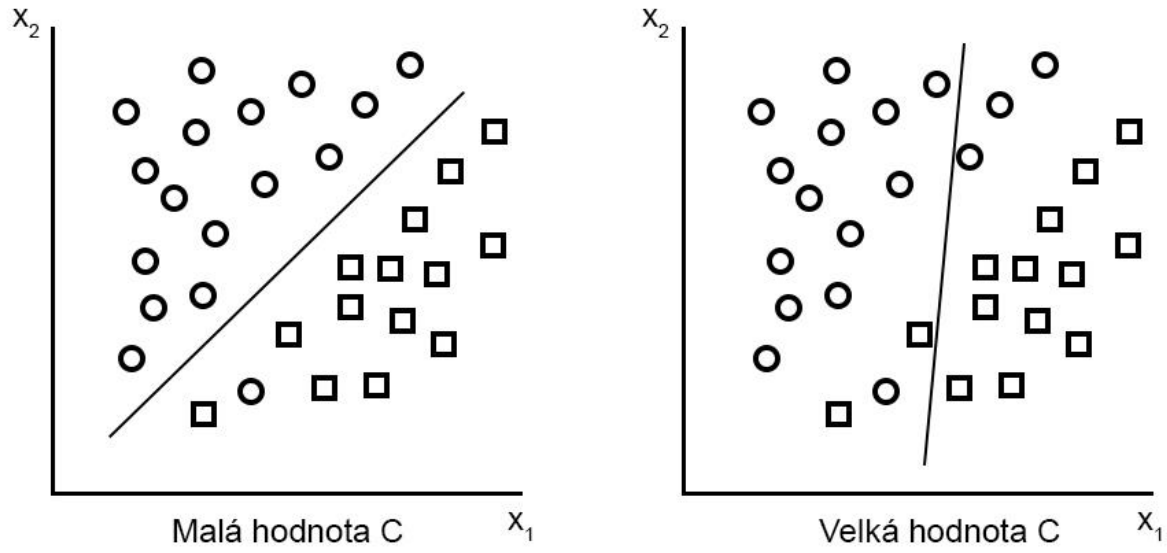
[5] Word2Vec implementace dostupná například zde: https://www.tensorflow.org/text/tutorials/word2vec. [Přístup 2023-09-06]
[6] Domovská stránka projektu: https://fasttext.cc/. [Přístup 2023-09-07]

stává, což tvrdí i Kecman (2005, s. 19). SVM má však řešení i pro případy, kdy se data překrývají a není tedy možné je jednoznačně a bezchybně lineárně oddělit. V takovém případě by se tedy mělo usilovat o to, aby co nejmenší množství dat bylo klasifikováno špatně (Cortes, Vapnik, 1995, s. 280). Řešení spočívá v zavedení parametru *C* do SVM výpočtů. Tento parametr určuje kompromis mezi chybovostí a hraničním pásmem, tedy jestli má model upřednostnit široké hraniční pásmo na úkor většího množství špatně klasifikovaných bodů, nebo se snažit mít co nejméně špatně klasifikovaných bodů za cenu užšího hraničního pásma. Roli parametru *C* ilustruje následující obrázek.



Obrázek 6: Vliv parametru C na SVM klasifikátor (vlastní tvorba)

Jak vyplývá z obrázku, malá hodnota *C* parametru upřednostňuje širší hraniční pásmo i za cenu špatně klasifikovaného bodu, zatímco velká hodnota *C* má za následek užší hraniční pásmo, avšak nedošlo k žádné chybě při klasifikaci. Takové nastavení klasifikátoru, které je v pravé části obrázku, se tedy jeví jako lepší nastavení pro daná data, což může být například situace při učení SVM klasifikátoru. Při aplikaci na nová data už klasifikátor s parametrem *C* nastaveným na vysokou hodnotu lepší výsledky mít nemusí, jak lze vidět na následujícím obrázku.



Obrázek 7: Vliv parametru C na SVM klasifikátor při vystavení novým datům (vlastní tvorba)

Jak je vidět, když se do grafu zanesou nové body, klasifikátor, který měl původně jednu chybu, rozděluje data lépe než druhý klasifikátor, který na trénovacích datech chybu neudělal. Jde však jen o ilustrační případ k nastínění vlivu parametru *C* na klasifikátor a mohlo by se tedy stát, že distribuce dat by byla jiná, než je v tomto příkladu a klasifikátor s velkou hodnotou *C* by data rozdělovat lépe v obou případech. Nejlepší nastavení *C* je tedy potřeba vyzkoušet experimentálně například pomocí křížové validace (Kecman, 2005, s. 23).

Pro učení SVM klasifikátoru byly použity tři velikosti rozdělení datasetů na trénovací a testovací sady v poměrech 70:30, 50:50 a protože by SVM mělo dosahovat dobrých výsledků i s malým množstvím trénovacích dat (Althnian et al., 2021, s. 11), bylo použito i rozdělení 5:95. Jako metody vektorizace byly použity bag of words, binární bag of words a předtrénované Word2Vec embeddingy (přístup přes knihovnu gensim (Řehůřek, Sojka, 2010)[7]). Před samotnou vektorizací textů ještě proběhlo jejich předzpracování. Pro bag of words byla odstraněna gramatická slova (tzv. stopslova) a ve snaze snížit počet různých slovních forem v důsledku inflexe byla zbylá slova zpracována Snowball Stemmerem, obojí z knihovny NLTK (Bird et al., 2009), a také byla z textů odstraněna interpunkční znaménka. V případě Sentiment140 datasetu byly z textů ještě navíc odstraněna označení profilů, hashtagy a internetové odkazy. Pro embeddingy byly podniknuty stejné kroky kromě použití stemmeru. Použití embeddingů však vyžadovalo ještě jeden další krok – trénování SVM klasifikátoru vyžaduje vstupy se stejnou velikostí, takže pouhé převedení slov na vektory nestačí, protože tím akorát vznikne dvourozměrná matice, která má počet sloupců stejný jako počet slov v textu a počet řádků jako velikost

[7] Domovská stránka projektu: https://radimrehurek.com/gensim/index.html. [Přístup 2023-09-07]

embeddingů (v tomto případě 300). Pro získání vstupů se stejnými rozměry byly tedy vytvořené embeddingy ještě zprůměrovány pomocí následující funkce.

```
def make_embeddings(texts, model):
    mean_embeddings = []
    for text in texts:
        text_embedding = []
        for word in text:
            if word in model.index_to_key:
                text_embedding.append(model[word])
        if text_embedding == []:
            pass
        else:
            mean_embeddings.append(np.mean(text_embedding, axis=0))
    return np.array(mean_embeddings)
```

Kód 2

Funkce postupně projde všechny texty a všechny slova v nich, pokud model obsahuje dané slovo, tak se jeho embedding přidá do seznamu `text_embedding`. Funkce poté zkontroluje, jestli byl vytvořen alespoň nějaký embedding pro některé ze slov v textu, pokud ne, přejde se na další text, pokud ano, vytvoří se z embeddingů pro všechna slova v daném textu průměr takovým způsobem, že se získá 300-rozměrný vektor zachycující průměr všech vektorů, takže v tomto vektoru je v podstatě zachycena průměrná sémantika daného textu. Tyto průměry již lze použít k trénování SVM klasifikátoru. V případě IMDB datasetu byl opět použit celý dataset, u Sentiment140 datasetu byl použit stejný vzorek jako v části 5.1.3, tedy celkem 100000 textů. V případě Word2Vec embeddingů u Sentiment140 datasetu také došlo k tomu, že model neobsahoval všechna slova ze vstupních textů, takže texty, které neměly žádný výsledný vektor ve funkci `make_embeddings` byly ignorovány. Tentokrát to však byly jen desítky až stovky textů. Naměřené přesnosti SVM klasfikátoru shrnuje Tabulka 7.

Dataset	Train:test split	BoW	BoW binární	W2V
IMDB	70:30	84,3 %	**84,64 %**	84,35 %
	50:50	82,91 %	83,78 %	**85,59 %**
	5:95	82,20 %	82,18 %	**85,67 %**
Sentiment140	70:30	74,74 %	**74,77 %**	72,82 %
	50:50	73,25 %	**74,01 %**	72,48 %
	5:95	68,49 %	68,64 %	**69,32 %**

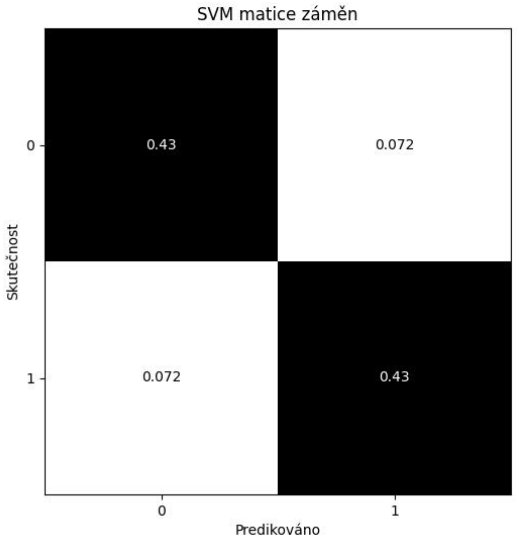
Tabulka 7: Přesnost lineárního SVM klasfikátoru

Tabulka 7 zachycuje vliv velikosti trénovacího datasetu a metody vektorizace na přesnost lineárního SVM klasifikátoru. Nejvyšší dosažená přesnost pro danou velikost trénovacího datasetu a metodu

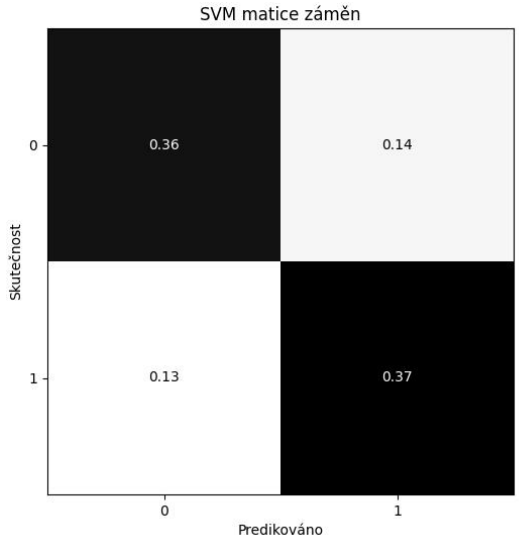
vektorizace je vždy vyznačena tučně. Ukázalo se, že až na jeden případ (IMDB dataset, train:test split 5:95), dosahuje binární bag of words lepších výsledků než bag of words zachycující frekvence, což je poněkud překvapivý výsledek, jelikož binární bag of words ze své binární podstaty zachycuje o něco méně informací než normální bag of words, avšak největší naměřený rozdíl je jen 0,87 %, což by mohl být i vliv náhody. Pro ověření tohoto tvrzení by bylo vhodné provést více měření na několika náhodných vzorcích a vypočítat, jestli jsou naměřené rozdíly statisticky signifikantní. Nejvyšší naměřené hodnoty byly tedy získány buďto pomocí binárního bag of words, nebo pomocí Word2Vec embeddingů. Dále z tabulky vyplývá, že ve všech případech kromě jednoho SVM zlepšovalo své výsledky s větším trénovacím datasetem, avšak rozdíly v naměřených přesnostech s rostoucím trénovacím datasetem jsou mnohem menší u IMDB datasetu než u Sentiment140 datasetu, což by mohlo být opět dáno tím, že filmové recenze jsou obecně delší než tweety, takže 5 % IMDB datasetu je větší objem textu než 5 % z použitého vzorku Sentiment140 datasetu (počet tokenů pro tento IMDB vzorek je 300857, zatímco pro Sentiment140 pouze 37073). Onou zmiňovanou výjimkou je použití Word2Vec embeddingů u IMDB datasetu, zde paradoxně naměřená přesnost klesá s rostoucím trénovacím datasetem, což je přinejmenším pozoruhodné. Z tabulky lze také vyčíst, že na největším trénovacím datasetu byla nejvyšší přesnost dosažena s binárním bag of words, zatímco na nejmenším datasetu s Word2Vec embeddingy. To by se dalo vysvětlit tím, že s rostoucím počtem textů roste i velikost slovníku pro bag of words, takže se jeho schopnost efektivně vektorizovat text bude zlepšovat, zatímco Word2Vec embeddingy jsou již předtrénované a neovlivňuje je velikost trénovacího datasetu (existuje ale i možnost natrénovat si embeddingy z vlastních dat), avšak bag of words klade mnohem větší nároky na RAM a není tedy možné ho efektivně používat s velkými datasety[8]. Posledním zjištěním je to, že u každého měření byla u IMDB datasetu naměřena lepší přesnost klasfikátoru, což může být dáno více faktory – například obecně rozdílnou délkou a typologií textů v datasetech, ale třeba i použití již zmiňovaných noisy labels pro anotaci Sentiment140 datasetu.

Další vhled do fungování klasifikátoru nabízí matice záměn. Matice byly vytvořeny pro každou velikost trénovacího datasetu a metodu vektorizace, ukázány však budou jen některé matice, protože se nijak zásadně neliší a pro ilustraci obecných trendů ve výkonu klasifikátoru to postačí.

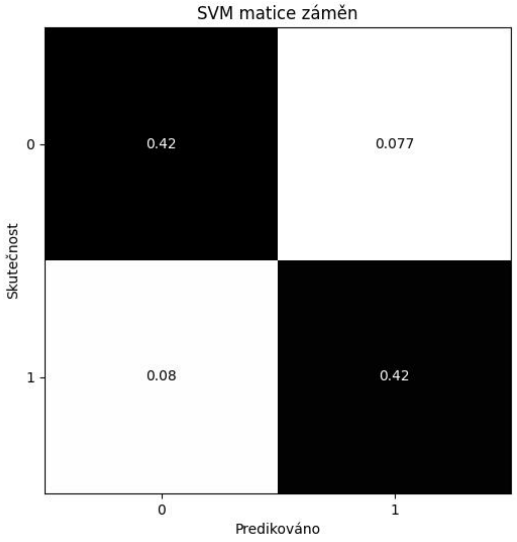
[8] Kdyby celý Sentiment140 dataset měl globální slovník o velikosti 35000 tokenů, velikost bag of words vektorů pro celý dataset by byla přibližně 200 GB.



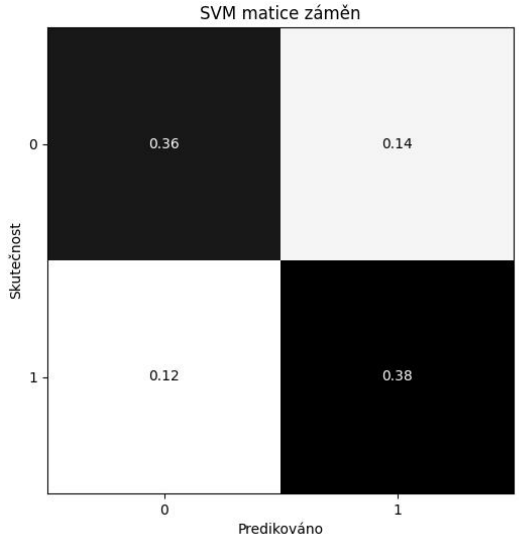
Obrázek 11: matice záměn SVM s embeddingy, IMDB dataset, split 70:30



Obrázek 10: matice záměn SVM s embeddingy, Sentiment140 dataset, split 70:30



Obrázek 9: matice záměn SVM s BoW, IMDB dataset, split 70:30



Obrázek 8: matice záměn SVM s BoW, Sentiment140 dataset, split 70:30

Z matic záměn je zřejmé, že natrénované SVM klasifikátory jsou poměrně robustní a mají velice malou míru biasu, z uvedených matic je největší míra biasu u Obrázek 8, kde pozitivní predikce převládají v poměru 52:48. Poměry biasů pro všechny kombinace vektorizací a velikostí trénovacích datasetů shrnuje následující tabulka.

		Bias negativní:pozitivní		
Dataset	Train:test split	BoW	BoW binární	W2V
IMDB	70:30	50:50	51:49	50:50
	50:50	52,5:47,5	52:48	50:50
	5:95	48:52	48:52	50:50
Sentiment140	70:30	48:52	47:53	49:51
	50:50	49:51	48:52	48:52
	5:95	47:53	47:53	49:51

Tabulka 8: poměry biasů SVM klasifikátorů

Jak lze v tabulce vidět, z použitých metod vektorizace má nejvyrovnanější výsledky použití embeddingů, zejména u IMDB datasetu, kdy byl klasifikátor vyrovnaný. V tomto ohledu mají obě varianty bag of words velice podobné výsledky a ve většině případů byl natrénovaný klasifikátor s jejich použitím lehce nevyrovnaný. Dalším zjištěním je, že u Sentiment140 datasetu byl klasifikátor vždy zkreslený a všech případech častěji predikoval pozitivní sentiment, což by ale opět mohlo být dáno daty. U IMDB datasetu s použitím bag of words metod je u nejmenšího trénovacího datasetu predikován častěji pozitivní sentiment, zatímco u větších trénovacích datasetů převažují predikce negativní.

Co se týče automatické anotace dat, tak je SVM lepším nástrojem než slovníkové metody, zejména kvůli velice konzistentním výsledkům, avšak má zásadní nedostatky, které je potřeba mít na paměti. SVM klasifikátory totiž přímo neposkytují pravděpodobnosti o správnosti klasifikace a jejich výpočet se provádí pomocí poměrně náročné pětinásobné křížové validace a výsledky navíc mohou být v některých případech nekonzistentní (Support Vector Machines, c2007-2023). Tyto nedostatky by mohly značně ovlivnit kvalitu nově vytvořeného datasetu a SVM pro automatickou anotaci se jeví jako nevhodné řešení.

V této části byla provedeno měření přesnosti lineárního SVM klasifikátoru, kde se ukázalo, že až na jeden případ dosahoval binární bag of words lepších výsledků než normální bag of words. Dále se ukázalo, že na větších trénovacích vzorcích dosahoval binární bag of words lepších výsledků než použití předtrénovaných Word2Vec embeddingů pro vektorizaci, zatímco u menších trénovacích vzorků měly lepší výsledky embeddingy. Také se potvrdilo, že SVM dosahuje dobrých výsledků i s poměrně málo daty, zejména pokud se jedná o delší texty a celkový počet tokenů dosahuje nižších statisíců. Z analýzy také vychází, že použití Word2Vec embeddingů vedlo k natrénování klasifikátoru s menší mírou biasu a klasifikátory trénované na Sentiment140 datasetu predikovaly ve všech nastaveních častěji pozitivní sentiment.

5.2.2. Naive Bayes

Další metodou je naivní bayesiánský klasifikátor, který se používá v různých odvětvích a v NLP se často používá například ve filtrech spamu (Jurafsky, Martin, 2023b, s. 9; Raschka, 2014, s. 3). Klasifikátor se

označuje jako naivní kvůli předpokladu, že mezi vlastnostmi v datasetu neexistuje žádná vzájemná závislost, což je však poměrně nereálný a často porušovaný předpoklad (Webb, 2011, s. 713). Tento předpoklad se v případě klasifikace textu projevuje například ignorováním pořadí slov. Jedná se o lineární klasifikátor, který staví na bayesově teorému. Je tedy založen na pravděpodobnosti a funguje tak, že na základě pravděpodobností výskytů slov v daném textu a pravděpodobnosti, že jakýkoliv text patří do nějaké z tříd se vypočítá pravděpodobnost, s jakou daný text patří do konkrétní třídy. Klasifikátor tedy v podstatě odpovídá na otázku „jaká je pravděpodobnost, že text T patří do skupiny Y, když má vlastnosti X?“, a odpoví na ni tolikrát, kolik je tříd pro klasifikaci a pro každou třídu zvlášť, z čehož se poté vybere jedna nejpravděpodobnější třída. Vlastnostmi se v tomto případě myslí frekvence výskytů slov, typicky ve formě bag of words (Jurafsky, Martin, 2023b, s. 3), což přináší jednu zásadní komplikaci. Protože Naive Bayes počítá s tím, že mezi vlastnostmi (slovy) neexistuje žádný vztah, tak postupně násobí pravděpodobnosti výskytů konkrétních slov v daném dokumentu (Jurafsky, Martin, 2023b, s. 4; Raschka, 2014, s. 5), avšak bag of words vektorizace vytváří řídké vektory, tedy vektory obsahující velké množství nul, a výsledek jakéhokoliv násobení s nulou je vždy nula. To tedy znamená, že jakmile by se měla vypočítat pravděpodobnost náležitosti k třídě pro dokument, který neobsahuje byť jedno slovo z globálního slovníku, bude výsledek nula. To se řeší zavedením parametru α, což je konstanta, která se přičte ke všem prvkům ve vektoru, aby se zabránilo výskytu nulových hodnot; v kontextu klasifikace textů se typicky $\alpha = 1$ a v takovém případě se této metodě říká Laplace smoothing (Jurafsky, Martin, 2023b, s. 5; Raschka, 2014, s. 12).

Je důležité také zmínit, že Naive Bayes není jen jedna metoda, ale existuje několik variant a každá je vhodná na něco trochu jiného. Ke klasifikaci textu se však primárně používají dvě varianty, konkrétně Multinomial Naive Bayes a Multi-variate Bernoulli Naive Bayes (McCallum, Nigam, 1998; Jurafsky, Martin, 2023b; Raschka, 2014; Webb, 2011, s. 714). Základní rozdíl mezi těmito dvěma typy bayesiánského klasifikátoru je, že multinomická verze pracuje s frekvencemi slov, zatímco Bernoulliho verze používá binární vlastnosti, tedy jestli se slovo v textu vyskytlo nebo ne. Dalším rozdílem je, že zatímco multinomický naivní bayes počítá pravděpodobnost náležitosti daného textu k nějaké třídě na základě pravděpodobnosti výskytu daných slov dohromady, Bernoulliho naivní bayes zkoumá výskyt daných slov izolovaně[9]. Jurafsky také píše, že v kontextu analýzy sentimentu je informace o výskytu slova důležitější než informace o frekvenci, takže binární vektorizace často zlepší výsledky; tuto verzi nazývá Binary Multinomial Naive Bayes (2023b, s. 7-8), z čehož vyplývá, že binární vlastnosti lze kombinovat i s multinomickou verzí bayesiánského klasifikátoru. Existují však výsledky, které toto tvrzení o lepších výsledcích při použití binárních vlastností vyvrací a na několika textových korpusech konzistentně dokazují přesný opak (McCallum, Nigam, 1998). Je tedy dobré si vždy empiricky ověřit, který klasifikátor dosahuje nejlepších výsledků s konkrétními daty a s konkrétním zadáním. V následujících analýzách naivních bayesiánských klasifikátorů tedy byly použity tři varianty:

[9] Pro podrobnější popis rozdílů viz například (Raschka, 2014), kde jsou popsány i samotné výpočty pravděpodobností.

multinomická, binární multinomická a Bernoulliho. Výsledky těchto klasifikátorů shrnuje následující tabulka.

Dataset	Train:test split	Multinomický NB	Binární multinomický NB	Bernoulliho NB
IMDB	70:30	84,83 %	82,11 %	**85,24 %**
	50:50	82,60 %	80,95 %	**84,15 %**
	5:95	82,39 %	78,76 %	**83,27 %**
Sentiment140	70:30	75,90 %	75,90 %	**76,14 %**
	50:50	74,88 %	74,89 %	**75,26 %**
	5:95	69,94 %	70,01 %	**70,53 %**

Tabulka 9: výsledky naivních bayesiánských klasifikátorů

Z tabulky jednoznačně vyplývá, že nejvyšší přesnosti dosáhl Bernoulliho naivní bayesiánský klasifikátor, což potvrzuje tvrzení, že binární vektorizace dat podává lepší výsledky, neplatí to však pro binární multinomický model, který popisoval Jurafsky (2023b, s. 7-8). Výsledky této analýzy také nejsou v souladu s výsledky, ke kterým došel McCallum, kdy se s rostoucí velikostí slovníku Bernoulliho klasifikátor začínal zhoršovat, zatímco výkon multinomického klasifikátoru se dále zlepšoval a v průměru měl o 27 % menší chybovost (1998). Dále měl případě IMDB datasetu binární multinomický klasifikátor ve všech situacích horší výsledky než běžný multinomický klasifikátor, zatímco u Sentiment140 datasetu měl binární multinomický klasifikátor přesnost vyšší nebo stejnou jako klasický multinomický klasifikátor, avšak rozdíly se pohybují jen v rámci setin procent. Ukázalo se také, že podobně jako SVM i naivní bayesiánský klasifikátor dosahuje dobré přesnosti i navzdory malému množství trénovacích dat. Tabulka 10 opět shrnuje poměry biasů klasifikátorů.

		Bias negativní:pozitivní		
Dataset	Train:test split	Multinomický NB	Binární multinomický NB	Bernoulliho NB
IMDB	70:30	51:49	52:48	51:49
	50:50	53:47	52:48	52:48
	5:95	52:48	53:47	52:48
Sentiment140	70:30	52:48	52:48	50:50
	50:50	53:47	53:47	51:49
	5:95	54:46	54:46	47:53

Tabulka 10: poměry biasů naivních bayesiánských klasifikátorů

Z těchto výsledků je jasné, že všechny natrénované bayesiánské klasifikátory jsou lehce zkreslené a predikují negativní sentiment častěji než pozitivní až na dvě výjimky, kdy v jednom případě je pozitivní sentiment predikován častěji a v druhém případě je klasifikátor vyrovnaný. Obecně ale platí, že na

největším trénovacím datasetu je míra biasu nejmenší u obou datasetů, což není nijak překvapivý výsledek.

Naive Bayes na rozdíl od SVM sice pravděpodobnosti poskytuje, avšak Naive Bayes je mnohem lepší pro klasifikaci než pro odhadování pravděpodobnosti, takže i když klasifikace může být správná, odhad pravděpodobnosti správnosti této klasifikace bude často dost zkreslený (Zhang, 2004) a těmto odhadům by se tudíž neměla přikládat příliš velká váha (Naive Bayes, c2007-2023). I přes to by se Naive Bayes dal použít k automatické anotaci dat poměrně efektivně – pokud je klasifikace správná, jen odhad pravděpodobnosti, že je tato klasifikace správná, je zkreslený, mělo by stačit jen snížit hranici, od které se anotované texty budou brát jako klasifikovány s velkou mírou jistoty. Tedy pokud by se u jiných metod, které poskytují lepší míry pravděpodobnosti, nastavila hranice pravděpodobnosti například na 0,9, tedy 90% jistota, že je klasifikace správná, u naivního bayesiánského klasifikátoru by mělo stačit tuto hranici akorát snížit například na 70 %. Přesné nastavení této hranice by však bylo potřeba experimentálně vyzkoušet a lze očekávat, že pro různé aplikace by nejlépe fungovala i různá nastavení. Naive Bayes by se tedy dal použít k automatické anotaci, ale je k tomu potřeba přistupovat s určitou mírou obezřetnosti.

V této části bylo krátce popsáno fungování různých variant naivních bayesiánských klasifikátorů běžně používaných pro klasifikaci textu. Z výsledků vyplývá, že na obou datasetech dosahoval nejlepších výsledků Bernoulliho klasifikátor, tedy klasifikátor pracující s binárními vlastnostmi textu, čímž se potvrdilo tvrzení, že binární vektorizace textu může podávat lepší výsledky a zároveň se tím nepotvrdilo tvrzení, že multinomické verze klasifikátorů by měly v průměru dosahovat mnohem lepších výsledků.

5.2.3. k-NN – k-Nearest Neighors

Poslední použitou statistickou metodou je k-Nearest Neighbors (Fix, Hodges, 1989; Cover, Hart, 1967), který se používá jak k regresi, tak ke klasifikaci. Metoda je koncepčně velice jednoduchá a vychází z předpokladu, že data podobného typu (například negativní a pozitivní texty) budou blízko sebe, podobně jako tomu je u slovních embeddingů a umisťování podobných slov blízko sebe. Trénovací data se na základě jejich vlastností rozmístí do prostoru a klasifikátor si toto rozmístění uloží. Klasifikace poté spočívá v nalezení nejbližších sousedů pomocí nějaké z metrik pro výpočet vzdálenosti mezi body, scikit-learn implementace k-NN používá Minkowského vzdálenost, která je však ve výchozím nastavení nastavena tak, že prakticky funguje jako Euklidovská vzdálenost (Sklearn.neighbors.KNeighborsClassifier, c2007-2023). Pro každou novou instanci dat se tedy najde k nejbližších sousedů a na základě náležitosti těchto sousedů do tříd se provede klasifikace. Pokud k=5 a alespoň tři z nejbližších sousedů patří do skupiny A, tak bude text zařazen do skupiny A. Je však možné použít i k=1, což je užitečné zejména v případech, kdy vzdálenosti mezi body náležícími k daným třídám jsou větší než vzdálenosti mezi třídami samotnými (Cover, Hart, 1967, s. 23), takže se v takovém případě

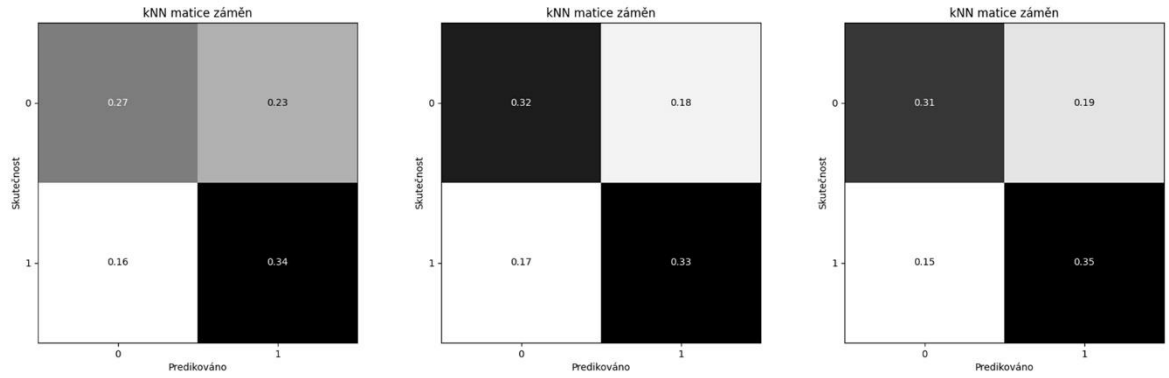
bude rozhodovat jednoduše na základě jediného nejbližšího souseda. Metoda je velice citlivá na hodnotu *k* a její správné nastavení je tedy klíčové k dobrému fungování klasifikátoru. Nejjednodušším způsobem, jak zjistit vhodnou hodnotu *k* pro konkrétní aplikaci je spustit algoritmus několikrát, vždy s jinou hodnotou *k* a zaznamenat si, s jakou hodnotou bylo dosaženo nejlepších výsledků; jsou však navrhovány i sofistikovanější metody výběru hodnoty *k*, kdy je hodnota variabilní v závislosti na datech (Guo et al., 2003, s. 986).

Výhodou k-Nearest Neighbors je, že žádným způsobem nezáleží na distribuci dat – nehledá se zde žádná nová osa pro lineární rozdělení jako třeba u SVM, pouze se přihlíží k tomu, jaké má nový bod nejbližší sousedy. Další výhodou je už zmíněný způsob trénování klasifikátoru, kdy si klasifikátor pouze uloží rozmístění dat, což v podstatě není trénování ve stejném smyslu jako opět například u SVM a výrazně to zkracuje dobu nutnou k naučení k-NN klasifikátoru, zároveň to však je i nevýhodou, protože se velká část výpočtů přesouvá až na chvíli, kdy dochází ke klasifikaci (Guo et al., 2003, s. 987). Zdlouhavost výpočtů k-NN by měla být poznat především na velkých datasetech, protože ke zjištění, který bod je nejbližší zkoumanému bodu, je v podstatě potřeba vypočítat vzdálenost zkoumaného bodu ke všem ostatním bodům, avšak existují i metody, které tyto výpočty významně zefektivňují (Zhu et al., 2014, s. 3737). Ve výsledku se tedy čas ušetřený při učení klasifikátoru pouze přesune na čas klasifikace, což zpravidla bude chvíle, kdy je preferována krátká výpočetní doba. Při provádění analýz se však neprokázalo, že by toto byl zásadní problém, protože vytváření predikcí s natrénovaným klasifikátorem trvalo v řádech jednotek sekund. Jako metody pro vektorizaci textů byla opět použita normální i binární varianta bag of words a také předtrénovaný Word2Vec model. Výsledky zachycuje Tabulka 11 níže.

Dataset	Train:test split	BoW	BoW binární	W2V
IMDB	70:30	62,25%	60,97%	**77,63%**
	50:50	60,44%	58,67%	**75,96%**
	5:95	58,85%	59,08%	**73,31%**
Sentiment140	70:30	66,15 %	65,92 %	**67,16 %**
	50:50	65 %	65,47 %	**66,53 %**
	5:95	60,37 %	60,89 %	**61,83 %**

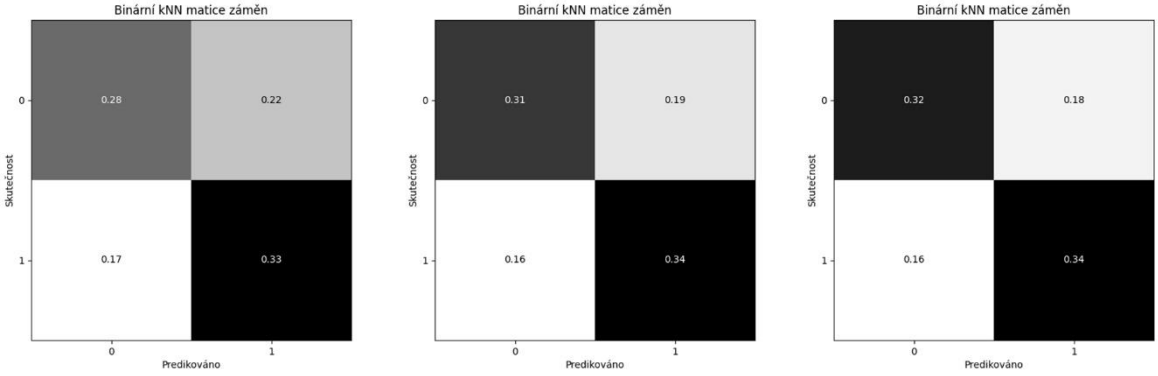
Tabulka 11: výsledky algoritmu k-nejbližších sousedů

Tabulka ukazuje, že obě varianty bag of words dosahovaly velice podobných výsledků, avšak na největším trénovacím datasetu dosáhl lepších výsledků normální bag of words, zatímco binární varianta byla lepší na nejmenším datasetu. Ve všech případech však k-NN dosahovalo nejlepších výsledků ve spojení s Word2Vec vektorizací. Je zajímavé, že při použití embeddingů se u IMDB datasetu výsledky zlepšily přibližně o 15 %, zatímco u Sentiment140 datasetu došlo ke zlepšení jen v rámci jednotek procent. Obecně lze také pozorovat zvyšování přesnosti s rostoucí velikostí trénovacího datasetu. Další informace poskytují matice záměn.



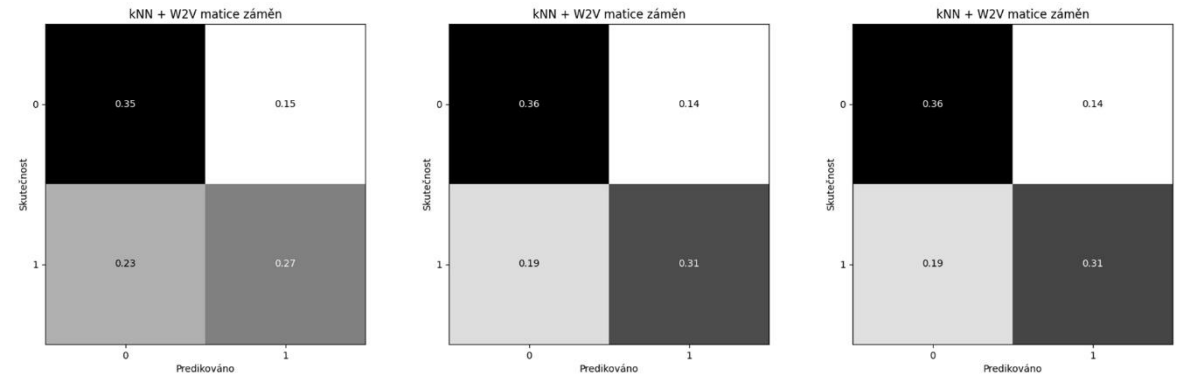
Obrázek 12: matice záměn k-NN s BoW, Sentiment140 dataset, split (zleva) 5:95, 50:50, 70:30

Tyto matice zobrazují výsledky k-NN klasifikátoru na Sentiment140 datasetu. Je vidět, že klasifikátor predikuje pozitivní sentiment s velmi podobnou přesností u všech splitů, avšak u splitu 5:95 je vidět poměrně výrazné zkreslení klasifikátoru, který predikuje častěji pozitivní sentiment. Je však zvláštní, že u splitu 50:50 model lépe predikoval negativní sentiment než u splitu 70:30.



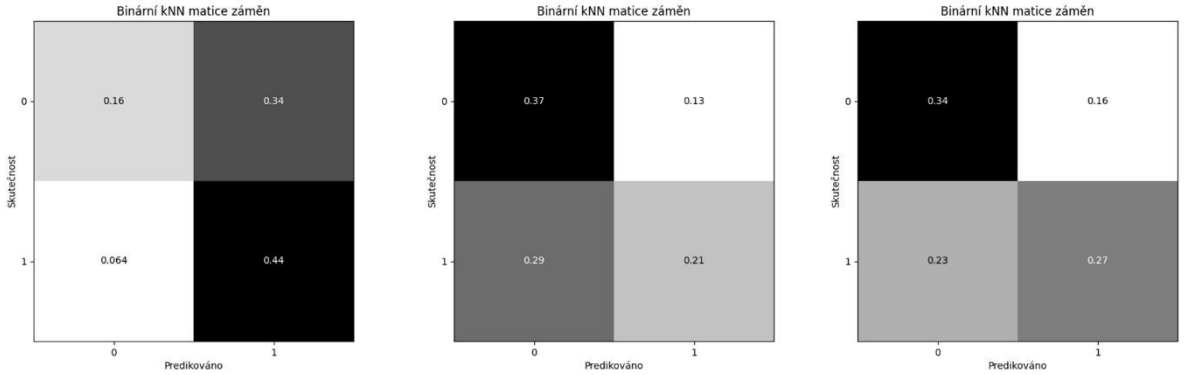
Obrázek 13: matice záměn k-NN s binárním BoW, Sentiment140 dataset, split (zleva) 5:95, 50:50, 70:30

Matice pro k-NN s binárním bag of words na Sentiment140 datasetu ukazují velice podobné výsledky jako při použití normálního bag of words – velice podobná přesnost u pozitivního sentimentu a podobné zkreslení u splitu 5:95. V tomto případě lze také pozorovat postupný nárůst přesnosti predikcí negativního sentimentu s rostoucím trénovacím datasetem.



Obrázek 14: matice záměn k-NN s Word2Vec, Sentiment140 dataset, split (zleva) 5:95, 50:50, 70:30

Při použití Word2Vec embeddingů jsou výsledky v podstatě opačné v porovnání s výsledky při použití bag of words – klasifikátor s použitím embeddingů lépe predikuje negativní sentiment, ale negativní sentiment také obecně predikuje častěji. Je zde také vidět zlepšení přesnosti mezi splity 5:95 a 50:50, ale mezi splity 50:50 a 70:30 jsou rozdíly tak malé, že se neprojeví při zaokrouhlování na dvě desetinná místa v zobrazení matic.



Obrázek 15: matice záměn k-NN s binárním BoW, IMDB dataset, split (zleva) 5:95, 50:50, 70:30

V případě k-NN s binárním bag of words na IMDB datasetu lze z matic vypozorovat poněkud nečekanou věc – klasifikátor natrénovaný na datech se splitem 5:95 sice predikuje velice dobře pozitivní sentiment, ale to bude nejspíše dáno tím, že dochází k velké míře zkreslení a pozitivní sentiment je předikován v 78 % případů; u ostatních splitů je to však naopak a dochází tedy k přesnějším predikcím negativního sentimentu, ale i ke zkreslení pro negativní sentiment. Všechna ostatní nastavení pro IMDB dataset se drží stejného vzoru – lepší přesnost predikcí negativního sentimentu, ale i zkreslení směrem k negativnímu sentimentu. Všechny míry biasu zachycuje následující tabulka.

		Bias negativní:pozitivní		
Dataset	Train:test split	BoW	BoW binární	W2V
IMDB	70:30	56:44	57:43	54:46
	50:50	60:40	66:34	56:44
	5:95	57:43	22:78	57:43
Sentiment140	70:30	46:54	48:52	55:45
	50:50	49:51	47:53	55:45
	5:95	43:57	45:55	58:42

Tabulka 12: poměry biasů k-NN klasifikátorů

Nejvyváženějšího klasifikátoru bylo dosaženo u Sentiment140 datasetu se splitem 50:50 za použití normálního bag of words, zatímco nejzkreslenější klasifikátor byl natrénován na IMDB datasetu se splitem 5:95 a binárním bag of words. Poměrně velká míra zkreslení je přítomna i klasifikátorů trénovaných pomocí Word2Vec embeddingů, což byly klasifikátory s nejvyšší přesností pro k-NN.

V kontextu automatické anotace dat nebude k-NN spolehlivou metodou, protože vypočtené pravděpodobnosti pouze zachycují, jaká část z nejbližších sousedů náleží ke konkrétní skupině dat, tedy například pokud jsou dvě třídy A a B, k=4 a tři sousedé patří do skupiny A a jeden do skupiny B, pak bude pravděpodobnost, že zkoumaný bod patří do skupiny A 75 % a 25 % že do skupiny B. Výsledné pravděpodobnosti budou tedy velice závislé na hodnotě parametru k i na celkové distribuci dat.

V této části bylo ukázáno fungování k-Nearest Neighbors klasifikátoru. Nejvyšší přesnost byla u obou datasetů dosažena s použitím Word2Vec embeddingů. U IMDB datasetu bylo dosaženo přesnosti přes 77 %, avšak u Sentiment140 byla nejlepší přesnost pouze lehce přes 67 %. Ukázalo se, že přesnost klasifikátoru roste s velikostí trénovacího datasetu, avšak se zdá, že od určitého objemu dat už bude nárůst přesnosti zanedbatelný, protože k největšímu skoku v přesnosti došlo mezi splity 5:95 a 50:50, zatímco mezi splity 50:50 a 70:30 už byl tento rozdíl menší a lze očekávat, že tento trend by pokračoval i s dodáním dalších dat.

5.2.4. Srovnání statistických a prostorových metod

Pro porovnání statistických a prostorových metod byly pro každou velikost trénovacího datasetu vybrány metody s nejvyšší dosaženou přesností, shrnuty jsou v následující tabulce.

Dataset	Train:test split	Metoda	Nejvyšší přesnost
IMDB	70:30	Bernoulli NB	85,24%
	50:50	SVM + W2V	85,59%
	5:95	SVM + W2V	85,67%
Sentiment140	70:30	Bernoulli NB	76,14%
	50:50	Bernoulli NB	75,26%
	5:95	Bernoulli NB	70,53%

Tabulka 13: shrnutí nejlepších klasifikátorů

Je vidět, že nejlepších výsledků dosahoval buďto Bernoulliho NB, nebo kombinace SVM s Word2Vec embeddingy. Zdá se, že v případech, kdy je k dispozici méně trénovacích dat, může SVM dosahovat velice dobrých výsledků, ale jak vyplývá z tabulky, nemusí to být vždy pravidlem, protože u všech rozdělení Sentiment140 datasetu měl nejlepší výsledky Bernoulliho NB. Obecně byl ale výkon SVM i Bernoulliho NB velmi podobný a Tabulka 13 tedy neposkytuje definitivní výsledky a neměla by být brána jako odpověď na otázku, který z klasifikátorů je lepší. Výsledky budou vždy velice závislé na konkrétních datech a je tedy doporučeno vyzkoušet více různých klasifikátorů pro zjištění, které nastavení je pro danou aplikaci nejlepší.

5.3. Neuronové sítě

Po vyzkoušení statistických a prostorových metod přišly na řadu neuronové sítě, které jsou velice silným a všestranným nástrojem a své uplatnění si našly v různých aplikacích v mnoha odvětvích (Abiodun et al., 2018). Doba jejich vzniku sahá až do roku 1943, kdy byl představen první model umělého neuronu, který měl svým fungováním napodobovat neurony biologické (McCulloch, Pitts, 1943) a i když moderní implementace umělých neuronů již nejsou založeny na této brzké inspiraci biologií, jsou biologické a umělé neurony spojeny alespoň jménem (Jurafsky, Martin, 2023c, s. 1). Navzdory slibným experimentům s jednoduchými neuronovými sítěmi se ke konci 60. let zjistilo, že technologie v tehdejší době dokázala řešit pouze lineárně oddělitelné úlohy čímž na nějakou dobu zájem o neuronové sítě zásadně ochabl, avšak od 80. let s vynalezením nových algoritmů zdokonalujících neuronové sítě zájem o ně opět stoupal (Jurafsky, Martin, 2023c, s. 25-26).

Základní stavební jednotkou neuronové sítě je tedy neuron, který dostává číselné vstupy, provede s nimi nějaké výpočty a vytvoří výstup (Jurafsky, Martin, 2023c, s. 2). I když jsou neuronové sítě schopné řešit různé komplexní problémy, jejich základní fungování je založené na jednoduchém

sčítání a násobení a schopnost řešit i složité problémy je dána použitím mnoha neuronů najednou, které společně pracují na nalezení vhodného řešení.

5.3.1. Tvorba vlastních modelů

Navzdory veliké popularitě a snadné dostupnosti velkých předtrénovaných modelů byla snaha vytvořit si vlastní model, a to z několika hlavních důvodů:

1. Nezávislost na třetích stranách. Vytvořením vlastního modelu by odpadla jakákoliv možná závislost na tvůrcích/poskytovatelích modelů. I když je zde primární zaměření na analýzu sentimentu, k čemuž existuje velké množství open-source modelů, je dobré si udržovat určitou míru prozřetelnosti pro případ, kdyby se podmínky změnily.
2. Kontrola nad trénovacími daty. Při tvorbě vlastních modelů je absolutní kontrola nad tím, jaká vstupní data se pro trénování použijí.
3. Více doménově zaměřený model. S použitím vlastních trénovacích dat v kombinaci s vybranými vhodnými open-source datasety lze hypoteticky předpokládat, že výsledný model bude na doménově závislých datech fungovat o něco lépe, protože by trénovací data byla tematicky koncentrovanější.
4. Získávání zkušeností. Nespornou výhodou trénování vlastních modelů je získávání cenných zkušeností v oblasti umělé inteligence, které se mohou hodit i v jiných aplikacích v budoucnu.

Při trénování vlastních modelů byla využita knihovna Keras (Cholet, 2015) a opět použity IMDB a Sentiment140 datasety, avšak již nebyl testován vliv velikosti trénovacího datasetu na výkon klasifikátoru, protože obecně platí, že neuronové sítě nemají moc dobré výsledky s malým množstvím trénovacích dat, a tedy čím více trénovacích dat, tím lepší klasifikační schopnosti. Pro IMDB dataset bylo tedy ve všech případech použito rozdělení 80:20. V případě Sentiment140 datasetu opět nastaly v některých případech komplikace kvůli jeho velikosti, zejména v kombinaci s bag of words. Kvůli tomu, že bag of words vytváří velké vektory, je od nějaké velikosti datasetu téměř nemožné natrénovat model na běžných počítačích bez použití generátoru dat, který bude data připravovat až ve chvíli, kdy jsou potřeba, protože mít celý vektorizovaný dataset načtený v RAM paměti může bez problému vyžadovat i stovky GB. S použitím generátoru však začne trénování brzdit procesor, protože musí neustále připravovat nová data a trénování modelů s bag of words vektorizací tedy trvalo hodiny, i když bylo celkem použito jen 15 % z celkové velikosti datasetu – 10 % pro trénování a 5 % pro testování. S jinými typy vektorizace však mohlo být použito více dat, a tak bylo použito stejné rozdělení jako u IMDB datasetu, tedy 80:20. U všech modelů byl nastaven ModelCheckpoint callback, který zaručil, že se uloží verze modelu s nejvyšší dosaženou přesností. U modelů, kde trénování trvalo dlouhou dobu byl ještě nastaven EarlyStopping callback, který proces trénování zastavil, pokud se přesnost modelu nezlepšila ve 2 epochách od epochy s nejlepší přesností.

Pro každý typ sítě bylo testováno několik konfigurací lišících se ve způsobu vektorizace textu. Byly použity stejné metody vektorizace jako v části 5.2, tedy bag of words, binární bag of words, Word2Vec embeddingy, ale přibyly i další dva typy vektorizace. První typ vektorizace je analogický k výchozí vektorizaci IMDB datasetu, kdy jsou tedy slova reprezentována svým rankem z frekvenčního slovníku. Druhý typ vektorizace je byte-pair encoding (BPE), který je schopen pomocí slučovacích pravidel pracovat s tokeny na úrovni slov i znaků tak, že časté kombinace znaků tvoří delší tokeny a méně časté kombinace zase kratší (Gage, 1994).

Při načítání IMDB datasetu je možnost určit, jak velký slovník se má použít a ve snaze alespoň trochu zkrátit výpočetní časy bylo tedy použito 10 000 nejčastějších slov. Pro Sentiment140 dataset byl vytvořen tokenizátor stejného typu, který používal stejně velký slovník. Dále byl vytvořen jeden BPE tokenizátor pro každý dataset, s velikostí slovníku tokenizátoru nastavenou na 25 000 tokenů. Pro zachování konzistence byl BPE tokenizátor pro IMDB dataset vytvořen z 10 000 nejčastějších slov v datasetu, u Sentiment140 byl BPE tokenizátor vytvořen z textů zbavených hypertextových odkazů, označení profilů, hashtagů apod. Kromě konfigurací s Word2Vec embeddingy a BPE tokenizací, kdy byly texty pouze zbaveny již zmíněných odkazů a dalších nepotřebných kusů textu, byly z textů odstraněna i stopslova a v případě bag of words vektorizace byl ještě navíc použit stemmer. Velikost vstupních dat byla nastavena na 200, v případě BPE na 400 jako kompenzace za používání tokenů menších než slovo, a v případě bag of words byla velikost vstupních dat dána počtem slov v globálním slovníku. V případech kromě použití bag of words a Word2Vec embeddingů byla do modelů přidána ještě embeddingová vrstva, která vytvářela 32 rozměrné embeddingy. V kombinaci Dense vrstev a Word2Vec embeddingů byly použity průměry vektorů jako vstupní data, což je dáno požadavky Dense vrstev na rozměry vstupních dat, avšak u ostatních typů sítí byly embeddingy zachovány v původních rozměrech, tedy každý token reprezentovaný 300 rozměrným vektorem.

Po dotrénování byly na několika příkladových větách u všech modelů otestovány výstupní hodnoty, aby se pro potenciální automatickou anotaci dat zjistilo, který model je nejvhodnější. Jednalo se o šest jednoduchých vět, z čehož polovina byla pozitivní a polovina negativní.

1. This is really great.
2. This is the best thing ever.
3. I love this so much.
4. That was extremelly horrible.
5. This is the worst thing ever.
6. I hate this so much.

Tím, že u každého modelu měla výstupní vrstva jeden neuron s aktivační funkcí sigmoid, výstupní hodnoty se vždy pohybovaly v rozmezí 0-1. Hodnoty pod 0,5 jsou klasifikovány jako negativní, nad 0,5 jako pozitivní, a rozdíl mezi predikovanou hodnotou a jednou z krajních hodnot lze interpretovat jako

míru jistoty, že je predikce správná. Pokud je tedy výstupní hodnota například 0,9, model si je nejspíše velice jistý tím, že vstupní text patří do kategorie pozitivního sentimentu.

Následující kapitoly shrnují základní konfigurace sítí a výsledky natrénovaných modelů. Kvůli rozdílným velikostem vstupních vrstev a (ne)používání embeddingových vrstev v různých scénářích bude ukázáno vždy jen jádro daných modelů, protože modely u každého typu sítě až na rozdílnou vstupní a embeddingovou vrstvu sdílely totožnou konfiguraci.

5.3.1.1. Plně propojená neuronová síť

Základním a zároveň asi i nejjednodušším typem neuronové sítě je plně propojená neuronová síť (anglicky dense neural network), jejíž název je odvozen z propojení neuronů uvnitř modelu – každý neuron v jedné vrstvě je napojen na všechny neurony ve vrstvě následující (Sarker, 2021, s. 6). Jádro modelů používajících plně propojené vrstvy bylo definováno následovně.

```
x = Dense(50, activation="relu")(x)
x = Dropout(0.5)(x)
x = Dense(25, activation="relu")(x)
x = Dropout(0.5)(x)
x = Dense(10, activation="relu")(x)
output = Dense(1, activation="sigmoid")(x)
```

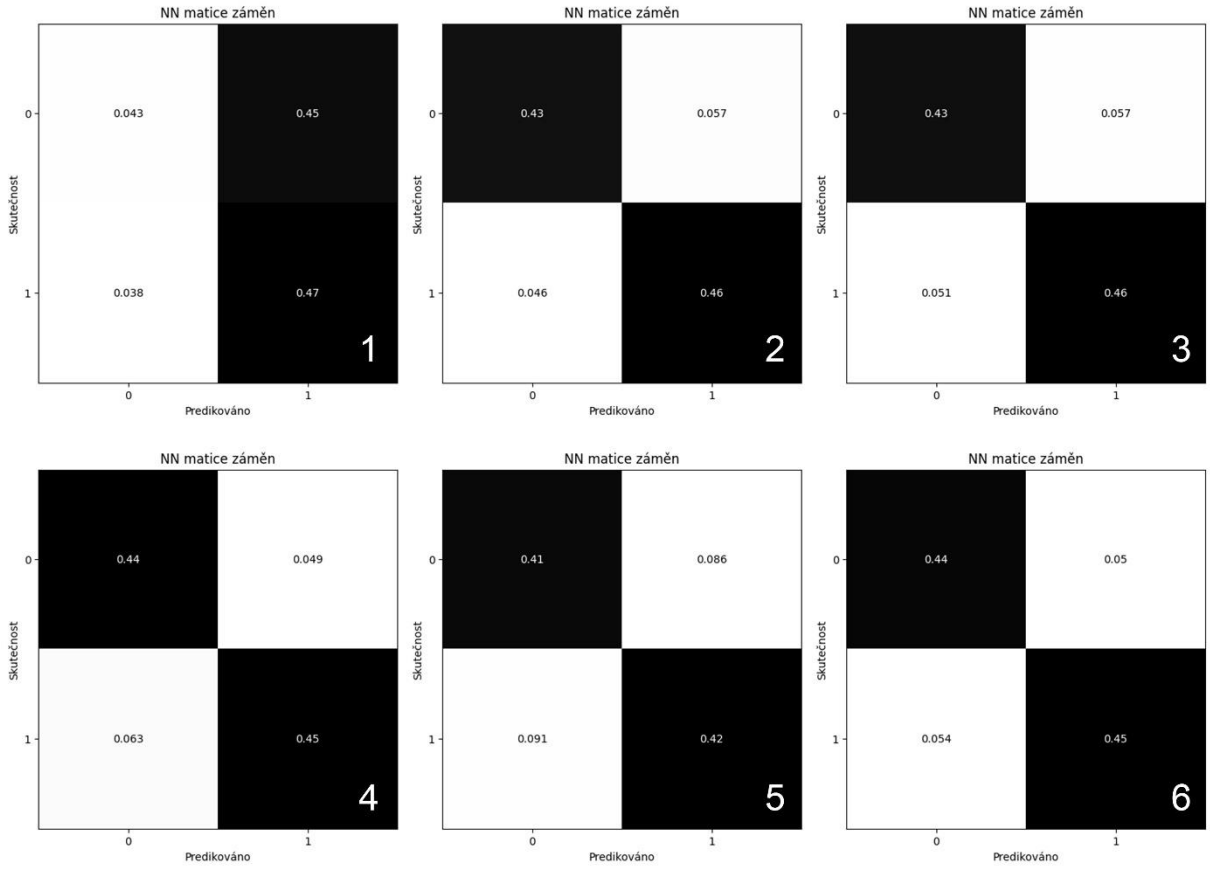
Kód 3: definice plně propojené sítě

Navíc kvůli požadavkům Dense vrstev na rozměry vstupních dat musela být ještě použita Flatten vrstva, která vícerozměrné vstupy (například embeddingy) přetransformuje tak, aby veškerá vstupní data byla v jediném vektoru. Kvůli jednoduchosti těchto vrstev se modely s touto konfigurací trénovaly poměrně krátkou dobu. Tabulka 14 shrnuje naměřené přesnosti těchto modelů.

Dataset	Typ vektorizace	Přesnost
IMDB	Tokenizátor	51,44 %
	Tokenizátor + embedding	89,64 %
	Bag of Words	89,19 %
	Binární Bag of Words	88,86 %
	Word2Vec	82,25 %
	BPE + embedding	89,60 %
Sentiment140	Tokenizátor + embedding	77,03 %
	Bag of Words	76,30 %
	Binární Bag of Words	76,39 %
	Word2Vec	74,82 %
	BPE + embedding	81,82 %

Tabulka 14: přesnosti modelů s Dense vrstvami

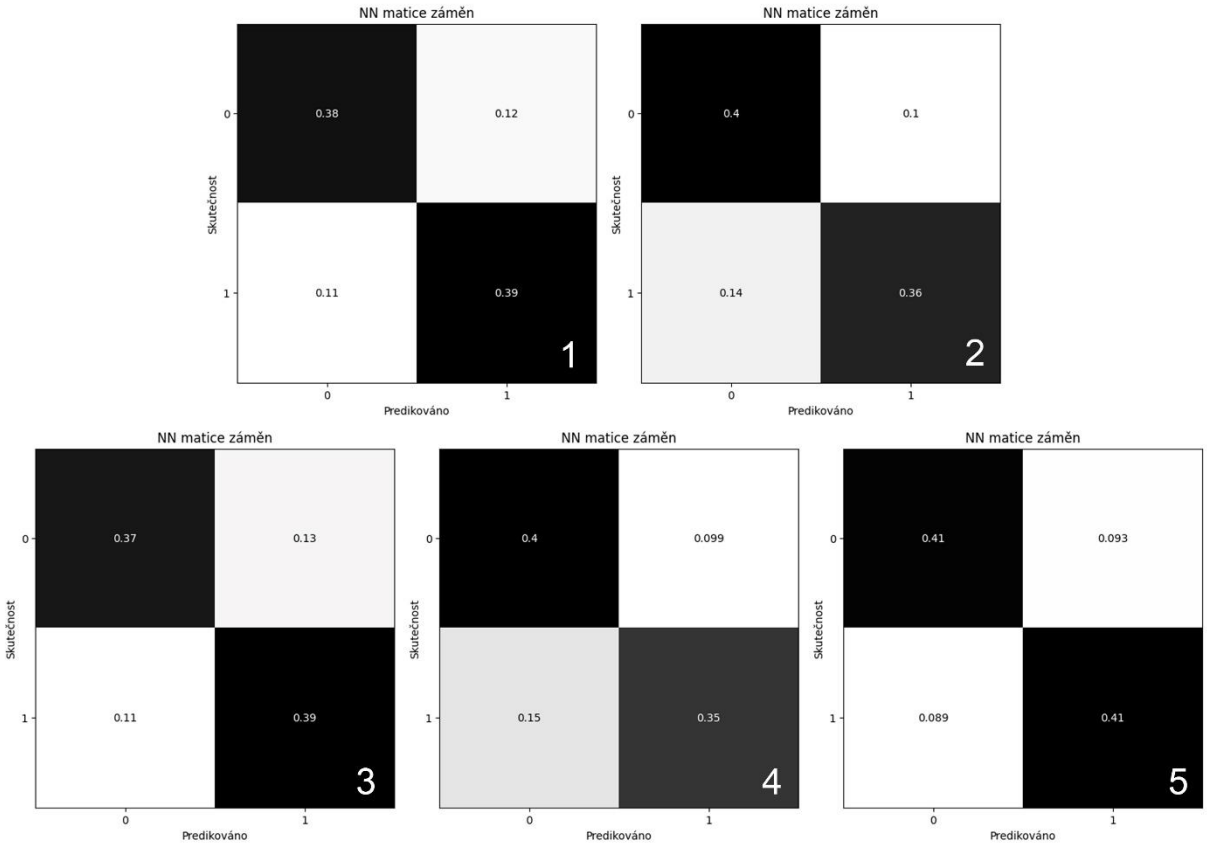
Tokenizátorem v typu vektorizace je myšlen tokenizátor vytvořený z frekvenčního slovníku a z tabulky jasně vyplývá, že vektorizace pomocí tohoto samotného tokenizátoru nestačí, protože model trénovaný na IMDB datasetu dosáhl přesnosti jen 51,44 %, což je jen o trochu lepší než náhoda. Z tohoto důvodu nebyl u žádných dalších modelů použit tokenizátor samotný, ale vždy se kombinoval s použitím embeddingové vrstvy, což extrémním způsobem zlepšilo přesnost. U IMDB datasetu model s kombinací tokenizátoru a embeddingové vrstvy dosáhl nejvyšší přesnosti, konkrétně 89,64 %, avšak u Sentiment140 datasetu bylo nejvyšší přesnosti dosaženo při použití BPE tokenizátoru, konkrétně 81,82 %. Bag of words vektorizace se také jeví jako dobrý nástroj a obě varianty dosahovaly podobně dobrých výsledků. Použití Word2Vec embeddingů mělo ze všech metod (kromě samotného tokenizátoru) nejhorší výsledky. Pro tento typ modelů, jak již bylo zmíněno, byly embeddingy opět zprůměrovány, čímž se dostal jeden stejně velký vektor pro každý vstupní text a je možné, že toto průměrování nezachycuje význam slov dostatečně detailně pro neuronové sítě. Další informace poskytnou matice záměn.



Obrázek 16: matice záměn Dense modelů na IMDB datasetu, 1 –tokenizátor, 2 – tokenizátor+emb, 3 – BoW, 4 – binární BoW, 5 – W2V, 6 – BPE

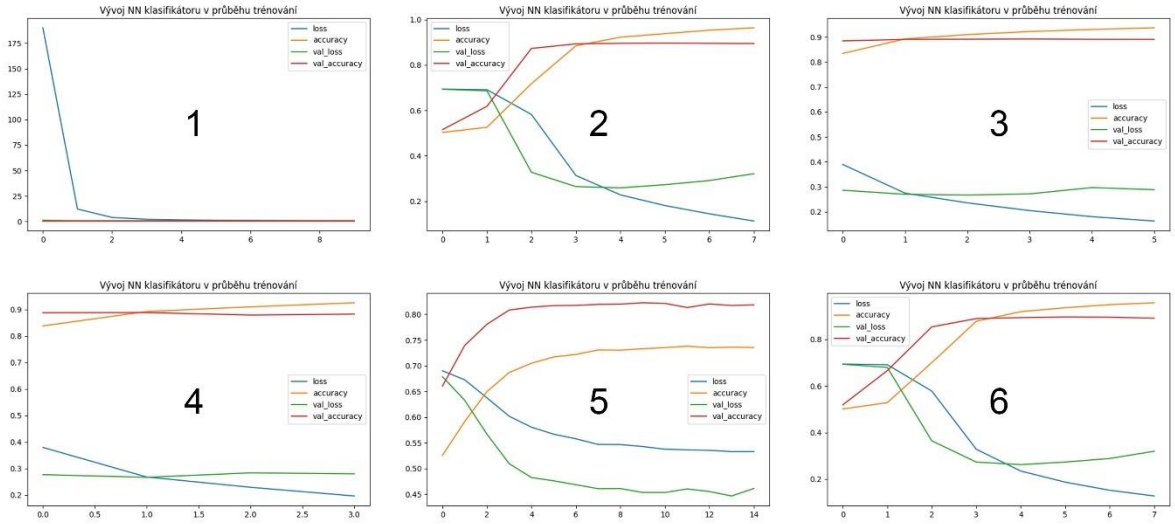
Z matic je zřejmé, že model, který používal jen samotný tokenizátor (1), se téměř nenaučil klasifikovat a v naprosté většině případů predikoval pozitivní sentiment. Na všech ostatních maticích lze vidět, že

modely sice klasifikují pozitivní sentiment o trochu častěji a lépe než ten negativní, ale jsou to jen malé rozdíly a jinak jsou modely velice vyvážené.



Obrázek 17: matice záměn Dense modelů na Sentiment140 datasetu, 1 – tokenizátor+emb, 2 – BoW, 3 – binární BoW, 4 – W2V, 5 – BPE

Obrázek 17 zobrazuje matice záměn modelů trénovaných na Sentiment140 datasetu. V tomto případě použití binárního bag of words (3) vedlo k natrénování vyváženějšího modelu v porovnání s normálním bag of words (2). Dále je zřejmé, že Word2Vec embeddingy (4) vedly k nejméně vyváženému modelu, který častěji predikoval negativní sentiment. Zbytek modelů byl opět poměrně vyvážený.



Obrázek 18: vývoj Dense modelů v průběhu trénování na IMDB datasetu, 1 –tokenizátor, 2 – tokenizátor+emb, 3 – BoW, 4 – binární BoW, 5 – W2V, 6 – BPE

Obrázek 18 zachycuje grafy zobrazující vývoj modelů v průběhu trénování, a i zde lze opět vidět, že v případě použití samotného tokenizátoru (1) se model nic neučil. U zbytku modelů lze pozorovat, že modely poměrně rychle dosáhly své maximální přesnosti – v prvních několika málo epochách přesnost prudce stoupala nebo byla vysoká již od začátku, jako při použití bag of words (3, 4), a poté docházelo jen k minimálním zlepšením.

		Pozitivní			Negativní		
Dataset	Typ vekzorizace	1	2	3	1	2	3
IMDB	Tokenizátor + embedding	0,89	0,83	0,81	0,16	0,11	**0,72**
	Bag of Words	0,90	0,86	0,80	0,15	0,09	0,35
	Binární Bag of Words	0,83	0,81	0,71	0,26	0,13	0,44
	Word2Vec	0,998	0,97	0,97	0,004	0,20	0,31
	BPE + embedding	0,89	0,59	0,64	0,10	0,05	0,27
Sentiment140	Tokenizátor + embedding	0,81	**0,03**	0,62	**0,70**	0,02	**0,71**
	Bag of Words	0,66	0,61	0,65	0,39	0,23	0,22
	Binární Bag of Words	0,75	0,73	0,67	0,47	0,34	0,27
	Word2Vec	0,84	0,72	0,67	0,06	0,23	0,17
	BPE + embedding	0,78	0,63	0,67	0,09	0,13	0,12

Tabulka 15: hodnoty predikcí Dense modelů pro automatickou anotaci

Tabulka výše zachycuje hodnoty predikcí na již zmíněných testovacích větách. V tabulce jsou tučně vyznačeny případy, kdy model špatně klasifikoval. U IMDB datasetu velice dobře fungují Word2Ve embeddingy, které se zejména u pozitivního sentimentu velice blíží 1, avšak pro negativní sentiment predikuje lepší hodnoty model s BPE tokenizátorem. U Sentiment140 datasetu jsou obecně hodnoty o

něco více konzistentní, avšak platí to samé jako u IMDB datasetu, tedy že model s Word2Vec embeddingy vrací nejlepší hodnoty pro pozitivní sentiment, zatímco pro negativní sentiment je to opět model s BPE tokenizátorem.

Jak matice záměn, tak grafy s vývojovými křivkami jsou ve všech situacích velice podobné těm, které byly ukázány v této části, tudíž v následujících částech popisujících ostatní typy neuronových sítí budou matice záměn a vývojové grafy ukázány jen v případě, že jsou nějak výrazně odlišné.

5.3.1.2. Konvoluční síť

Konvoluční sítě jsou specificky určené pro zpracování 2D dat, tedy typicky obrázků (Sarker, 2021, s. 7), ale existuje i typ 1D konvoluční sítě, který se více hodí pro sekvenční data, tedy data jako text nebo audionahrávky. Modely používající Conv1D vrstvy byly definovány následovně.

```
x = Conv1D(128, 5, activation="relu")(x)
x = MaxPooling1D()(x)
x = Conv1D(64, 5, activation="relu")(x)
x = MaxPooling1D()(x)
x = Conv1D(32, 5, activation="relu")(x)
x = MaxPooling1D()(x)
x = Flatten()(x)
x = Dropout(0.5)(x)
output = Dense(1, activation="sigmoid")(x)
```

Kód 4: definice konvoluční sítě

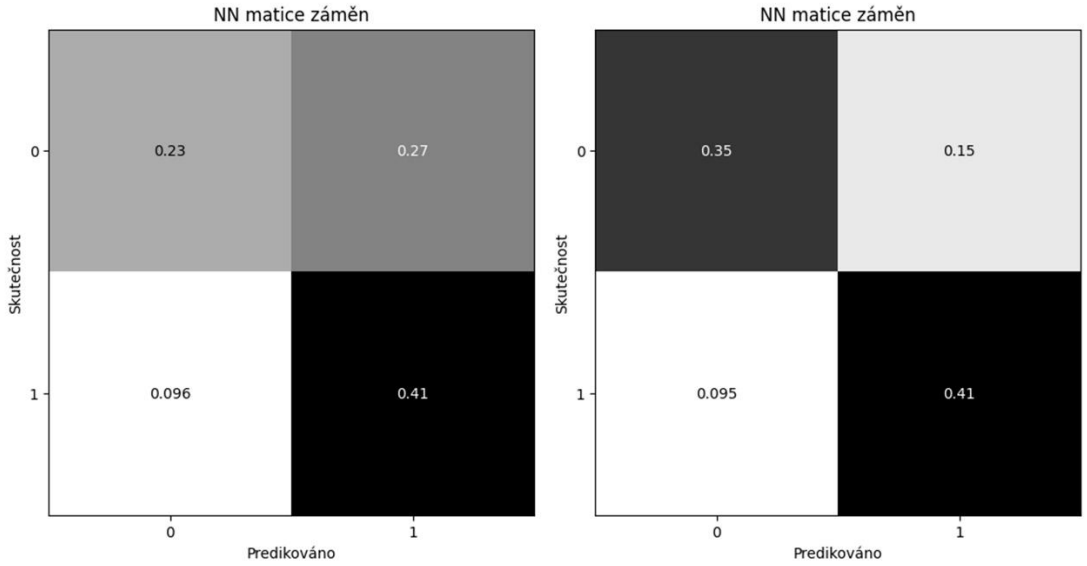
Následující tabulka shrnuje výsledky modelů s konvolučními sítěmi.

Dataset	Typ vektorizace	Přesnost
IMDB	Tokenizátor + embedding	87,51 %
	Bag of Words	88,48 %
	Binární Bag of Words	88,33 %
	Word2Vec	89,60 %
	BPE + embedding	89,13 %
Sentiment140	Tokenizátor + embedding	63,35 %
	Bag of Words	76,28 %
	Binární Bag of Words	76,16 %
	Word2Vec	81,39 %
	BPE + embedding	75,89 %

Tabulka 16: přesnosti modelů s Conv1D vrstvami

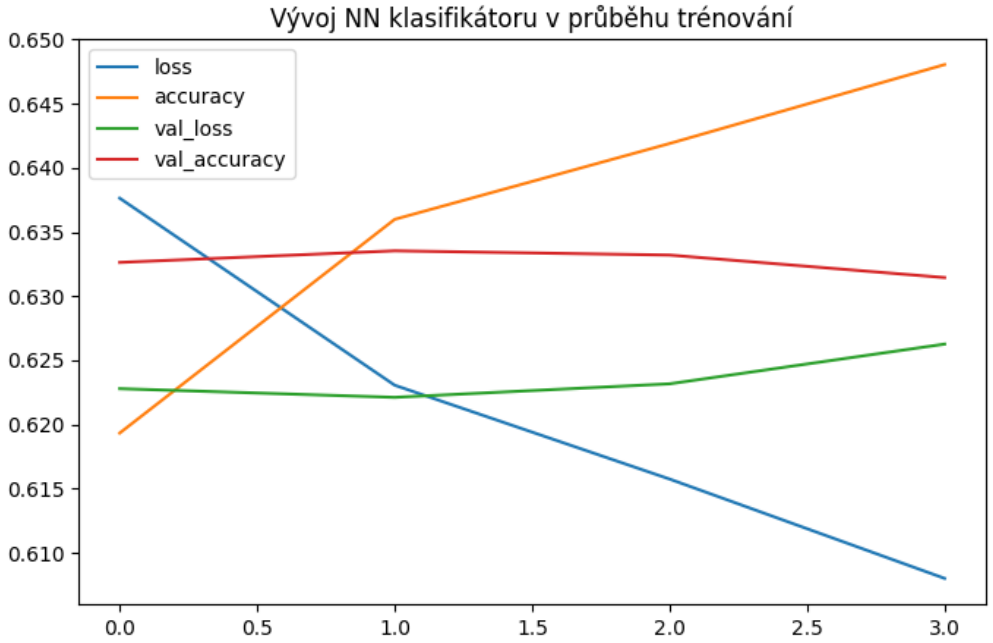
V tomto případě pro oba datasety bylo dosaženo nejvyšší přesnosti s použitím Word2Vec embeddingů, konkrétně 89,6 % pro IMDB dataset a 81,39 % pro Sentiment140 dataset. Všechny metody u IMDB

datasetu dosahovaly velice vysoké přesnosti, avšak u Sentiment140 datasetu byly rozdíly v přesnostech o něco větší, a především kombinace tokenizátoru s embeddingovou dosáhla velice špatných výsledků s přesností pouhých 63,35 %.



Obrázek 19: matice záměn konvolučních sítí na Sentiment140 datasetu, tokenizátor+emb (vlevo) a BPE (vpravo)

Matice záměn dvou modelů s nejhorší přesností na Sentiment140 datasetu odhalují, že klasifikátory jsou poměrně nevyvážené a predikují častěji pozitivní sentiment, především v případě, kdy byl použit tokenizátor s embeddingovou vrstvou. Zbytek modelů pro oba datasety je vyvážený a matice záměn tedy neodhalují nic zajímavého.



Obrázek 20: vývoj Conv1D modelu na Sentiment140 datasetu s použitím tokenizátoru a embeddingové vrstvy

Jediný graf zachycující vývoj klasifikátoru, který je odlišný od ostatních, je graf pro model s použitím tokenizátoru a embeddingové vrstvy trénovaný na Sentiment140 datasetu. Z grafu lze vyčíst, že přesnost na testovacím datasetu téměř vůbec nestoupala v průběhu trénování a v druhé polovině dokonce začala klesat, zatímco přesnost na trénovacím datasetu stále stoupala. Je však důležité si všimnout stupnice na svislé ose – i když se rozdíl mezi křivkami zobrazujícími přesnost může zdát velký, jsou to ve skutečnosti méně než 2 %. Trénování však bylo ukončeno po třetí epoše, protože se model nezlepšoval a mohlo by se tedy stát, že kdyby byl model trénovaný na více epoch, rozdíly budou ještě větší, alespoň soudě podle dosavadního trendu. U tohoto modelu tedy velice rychle docházelo k overfittingu, což by mohlo být zlepšeno zavedením například nějakého typu regularizace do modelu.

		Pozitivní			Negativní		
Dataset	Typ vekzorizace	1	2	3	1	2	3
IMDB	Tokenizátor + embedding	0,58	0,58	0,58	**0,58**	**0,58**	**0,58**
	Bag of Words	0,67	0,66	0,60	**0,51**	0,15	**0,504**
	Binární Bag of Words	0,73	0,72	0,61	**0,52**	0,11	0,45
	Word2Vec	0,79	0,72	0,55	0,31	0,20	0,47
	BPE + embedding	0,56	0,56	0,56	**0,56**	**0,56**	**0,56**
Sentiment140	Tokenizátor + embedding	0,53	0,53	0,53	**0,53**	**0,53**	**0,53**
	Bag of Words	0,73	0,75	0,75	0,14	0,19	0,16
	Binární Bag of Words	0,75	0,70	0,71	0,16	0,23	0,20
	Word2Vec	0,97	0,98	0,95	0,04	0,04	0,03
	BPE + embedding	0,56	**0,49**	**0,35**	**0,55**	0,49	0,35

Tabulka 17: hodnoty predikcí Conv1D modelů pro automatickou anotaci

Z tabulky výše je zřejmé, že u obou datasetů dosahoval nejlepších hodnot model s použitím Word2Vec embeddingů, a to jak pro pozitivní, tak negativní sentiment. U Sentiment140 datasetu jsou však hodnoty mnohem lepší než u IMDB datasetu, což může být dáno tím, že Sentiment140 obsahuje kratší texty a testovací věty, které jsou také poměrně krátké, se tedy více podobají původním trénovacím datům. U modelů tokenizátor + embedding a BPE + embedding na IMDB datasetu model predikoval vždy stejnou hodnotu, což bude asi dáno právě rozdílem v délce testovacích vět a trénovacích textů – z textů z IMDB datasetu vznikaly především vektory plné nějakých hodnot, zatímco z těchto testovacích vět po vektorizaci vznikly řídké vektory obsahující převážně nulové hodnoty, se kterými se modely nedokázaly efektivně vypořádat. Nejlepším modelem pro automatickou anotaci z této části by tedy byl model trénovaný na Sentiment140 datasetu s Word2Vec embeddingy.

5.3.1.3. Long Short-Term Memory (LSTM) síť

LSTM síť je primárně určena ke zpracování sekvenčních dat, takže je vhodná ke zpracování textu. Je to typ rekurentní neuronové sítě, což znamená, že předchozí krok z učení se použije k učení kroku

následujícího, což v důsledku funguje jako forma paměti. Obecně mají rekurentní sítě problém s dlouhými sekvencemi dat, což se právě LSTM síť snaží řešit, ale i tak má tento typ sítě svá omezení. Jedním z problémů LSTM sítě je výpočetní náročnost, která značně prodlužuje dobu trénování – ze všech trénovaných modelů v rámci této práce trvalo trénování LSTM modelů nejdéle. V případě bag of words vektorizace u Sentiment140 datasetu by trénování trvalo desítky hodin na jednu epochu, takže tyto modely jako jediné byly úplně vynechány. Pro LSTM modely byla zvolena architektura s obousměrnými LSTM vrstvami, což znamená, že vstupní data jsou zpracována v původním pořadí, ale i pozpátku, což síti poskytuje více informací.

```
x = Bidirectional(LSTM(64, return_sequences=True))(x)
x = Dropout(0.5)(x)
x = Bidirectional(LSTM(64, return_sequences=True))(x)
x = Dropout(0.5)(x)
x = Bidirectional(LSTM(64))(x)
output = Dense(1, activation="sigmoid")(x)
```

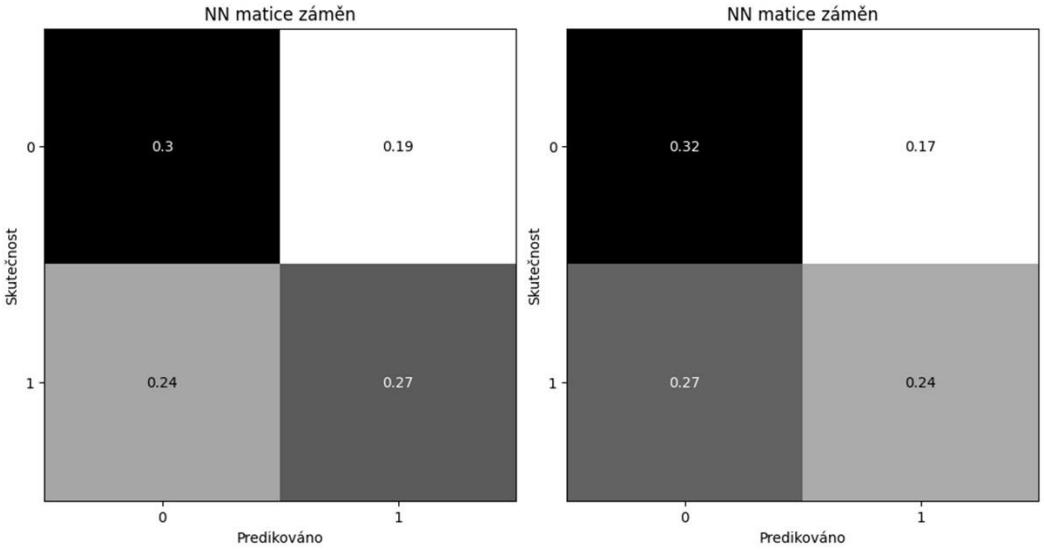
Kód 5: definice obousměrné LSTM sítě

Tabulka 18 zachycuje naměřené přesnosti LSTM modelů.

Dataset	Typ vektorizace	Přesnost
IMDB	Tokenizátor + embedding	89,43 %
	Bag of Words	57,08 %
	Binární Bag of Words	55,61 %
	Word2Vec	86,3 %
	BPE + embedding	88,96 %
Sentiment140	Tokenizátor + embedding	79,32 %
	Word2Vec	82,59 %
	BPE + embedding	84,1 %

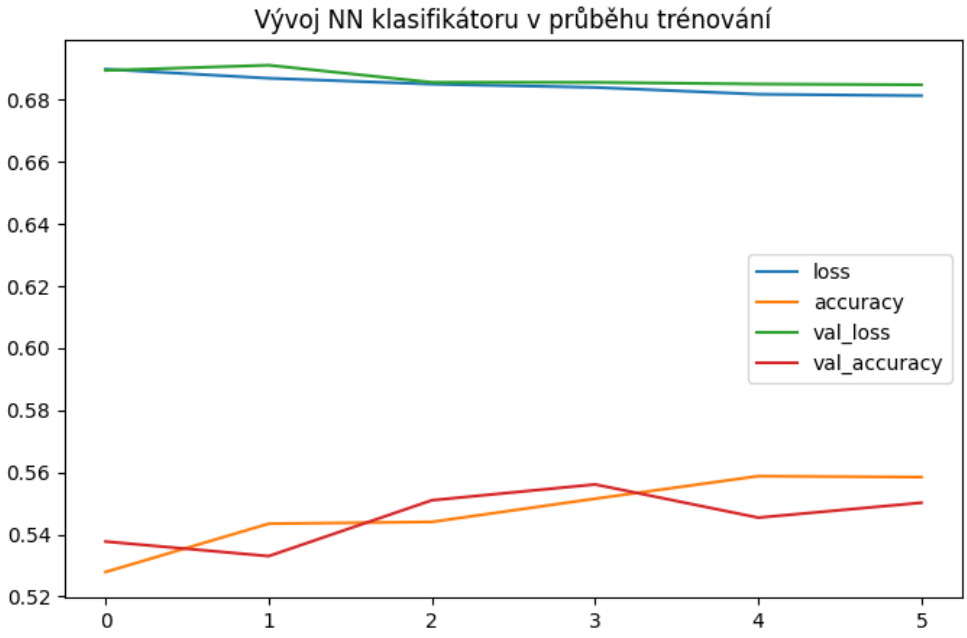
Tabulka 18: přesnosti modelů s LSTM vrstvami

Již na první pohled je zřejmé, že LSTM sítě v kombinaci s bag of words vektorizací nebyly efektivní a dosáhly přesnosti jen přes 55 %. U všech ostatních typů vektorizace však modely dosáhly velice dobrých výsledků, nejvyšší přesnost byla u IMDB datasetu 89,43 % a u Sentiment140 datasetu 84,1 %.



Obrázek 21: matice záměn LSTM sítí na IMDB datasetu, BoW (vlevo) a binární BoW (vpravo)

Obrázek 21 zobrazuje matice záměn LSTM sítí na IMDB datasetu s bag of words vektorizací a jak lze vidět, oba modely predikují častěji negativní sentiment a pozitivní sentiment predikují o něco hůře než ten pozitivní. Zbytek modelů je velice vyvážený a matice zde tedy nestojí za to ukazovat.



Obrázek 22: vývoj LSTM modelu na IMDB datasetu s použitím BoW

Graf vývoje klasifikátoru odhaluje, že při trénování LSTM modelu s bag of words vektorizací nedocházelo k overfittingu, protože trénovací i testovací přesnost jsou si v grafu velice blízko a mají velmi podobný trend. Je možné, že pokud by se EarlyStopping callback nastavil s větší tolerancí, možná

by to vedlo k dosažení větší přesnosti, avšak podle dosavadního trendu by zlepšení bylo velice pozvolné, a tudíž i zdlouhavé, což se vzhledem k vysoké přesnosti ostatních LSTM klasifikátorů zkrátka nevyplatí.

		Pozitivní			Negativní		
Dataset	Typ vekzorizace	1	2	3	1	2	3
IMDB	Tokenizátor + embedding	0,75	0,64	0,62	0,27	0,17	0,48
	Bag of Words	**0,49**	**0,49**	**0,49**	0,49	0,49	0,49
	Binární Bag of Words	**0,49**	**0,48**	**0,49**	0,49	0,49	0,49
	Word2Vec	0,98	0,97	0,88	0,19	0,06	**0,67**
	BPE + embedding	0,89	0,84	0,78	0,17	0,15	**0,64**
Sentiment140	Tokenizátor + embedding	0,93	0,93	0,85	0,03	0,05	0,05
	Word2Vec	0,97	0,96	0,93	0,07	0,01	0,02
	BPE + embedding	0,91	0,69	0,82	0,03	0,05	0,02

Tabulka 19: hodnoty predikcí LSTM modelů pro automatickou anotaci

U hodnot predikcí pro automatickou anotaci mají všechny modely trénované na IMDB datasetu kromě modelu s tokenizátorem a embeddingovou vrstvou alespoň jednu chybu ve svých predikcích. Modely s bag of words vektorizací opět predikovaly téměř identické hodnoty ve všech případech, a tudíž se pro automatickou anotaci vůbec nehodí. Modely s použitím Word2Vec a BPE vrací pěkné hodnoty, pokud se pomine případ s chybnou predikcí. U všech modelů trénovaných na Sentiment140 datasetu jsou hodnoty predikcí velice blízko krajním hodnotám, avšak model s Word2Vec embeddingy má opět nejlepší výsledky.

5.3.1.4. Transformer

Transformer (Vaswani et al., 2017) je poměrně nová architektura neuronové sítě, která využívá ještě dalšího mechanismu k řešení problému s dlouhými sekvencemi dat, takzvaný attention mechanismus, který dovoluje odvozovat závislosti v datech na velké vzdálenosti. Architektura transformeru navíc dovoluje velkou míru paralelizace, což významně zkracuje potřebný výpočetní čas pro trénování (Vaswani et al., 2017, s. 2). Díky výhodám paralelizace se transformer modely trénovaly ze všech modelů nejkratší dobu. Pro transformer byla použita implementace dostupná na Keras webu (Nandan, b.r.) bez žádných dalších změn kromě změny výstupní vrstvy, která má v původním kódu dva neurony a aktivační funkci softmax, zatímco v této práci byla použita vrstva s jedním neuronem a sigmoid aktivační funkcí pro zachování konzistence s ostatními modely. Byl použit jen jeden transformer blok, který si vytváří embeddingy o velikosti 32, používá dvě attention hlavy a uvnitř bloku je plně propojená vrstva s 32 neurony. Jelikož transformer využívá embeddingových vrstev, byly tyto modely kombinovány pouze s frekvenčním a BPE tokenizátorem – kombinace bag of words a embeddingů je přinejmenším neobvyklá (a při krátkém testování nevedla k dobrým výsledkům) a vytvářet embeddingy

z Word2Vec embeddingů pomocí další embeddingové není možné, protože embeddingová vrstva vyžaduje na vstupu celá čísla. Celkem tedy byly natrénovány čtyři modely a jejich výsledky jsou shrnuty v následující tabulce.

Dataset	Typ vektorizace	Přesnost
IMDB	Tokenizátor + embedding	89,61 %
	BPE + embedding	89,23 %
Sentiment140	Tokenizátor + embedding	78,39 %
	BPE + embedding	82,19 %

Tabulka 20: přesnosti Transformer modelů

Z tabulky vyplývá, že u IMDB modelu byly výsledky velice podobné, avšak verze s použitím tokenizátoru a embeddingové vrstvy měla přesnost o něco vyšší, konkrétně 89,61 %. U Sentiment140 datasetu jsou rozdíly v přesnosti větší a model s BPE tokenizátorem dosáhl 82,19% přesnosti. Na maticích záměn ani vývojových grafech pro tyto modely není nic zvláštního, natrénované modely jsou poměrně vyvážené a nejvyšší přesnosti dosáhly v malém počtu epoch.

		Pozitivní			Negativní		
Dataset	Typ vekzorizace	1	2	3	1	2	3
IMDB	Tokenizátor + embedding	0,87	0,81	0,75	0,16	0,05	**0,69**
	BPE + embedding	0,89	0,86	0,77	0,20	0,06	**0,68**
Sentiment140	Tokenizátor + embedding	0,94	0,85	0,75	0,02	0,07	0,05
	BPE + embedding	0,95	**0,35**	0,70	0,02	0,07	0,05

Tabulka 21: hodnoty predikcí Transformer modelů pro automatickou anotaci

Tabulka 21 opět ukazuje hodnoty predikcí natrénovaných modelů a jak lze vidět, až na jednu chybu v klasifikaci u Sentiment140 modelu s BPE tokenizátorem mají modely trénované na stejných datech velice podobné výsledky. Modely trénované na IMDB datasetu opět chybně klasifikovaly třetí negativní větu, u které často docházelo ke špatné klasifikaci i u předchozích modelů. Obecně se z těchto modelů opět jeví model trénovaný na Sentiment140 datasetu jako vhodnější pro automatickou anotaci.

5.3.1.5. Ensemble model

Ensemble model není samostatným typem neuronové sítě, ale pouhým spojením různých typů sítí či klasifikátorů s cílem dosáhnout co nejlepších výsledků. Základní myšlenkou je, že spojením více klasifikátorů by mělo být možné dosáhnout lepších výsledků, než kterých je schopen kterýkoliv z dílčích klasifikátorů (Sagi, Rokach, 2018, s. 1). Toho se typicky docílí hlasováním – jednotlivé klasifikátory vytvoří nějaký výstup a ten, který je většinově zastoupený, je zvolen jako konečný výstup (Sagi,

Rokach, 2018, s. 6). Pro ensemble model byly vybrány tři typy neuronových sítí – plně propojená síť, LSTM síť a Transformer, vše s použitím BPE tokenizátoru pro jednotný proces zpracování dat. Navzdory tomu, že u IMDB datasetu verze modelů s použitím BPE tokenizátoru většinou nedosáhly těch nejvyšších výsledků, dosáhly alespoň na druhou příčku v přesnosti. Co se Sentiment140 datasetu týče, tam BPE tokenizátor ve většině případů vedl k nejvyšším naměřeným přesnostem pro danou skupinu modelů, a pro předejití zbytečného komplikování tokenizace dat byl tedy vybrán jediný způsob tokenizace, kterým je již zmíný BPE tokenizátor. Každý z použitých typů sítě používal stejné nastavení jako v předchozích částech, k tomu měla každá síť svou vlastní embeddingovou vrstvu a výstupy z poslední vrstvy každé sítě byly konkatenovány, aby byly zpracovány posledním výstupním neuronem.

```
input_layer = Input(shape=(400,))

embedding_layer = TokenAndPositionEmbedding(400, 25000, 32)
x1 = embedding_layer(input_layer)
transformer_block = TransformerBlock(32, 2, 32)
x1 = transformer_block(x1)
x1 = GlobalAveragePooling1D()(x1)
x1 = Dropout(0.5)(x1)
x1 = Dense(20, activation="relu")(x1)

x2 = Embedding(25000, 32)(input_layer)
x2 = Bidirectional(LSTM(64, return_sequences=True))(x2)
x2 = Dropout(0.5)(x2)
x2 = Bidirectional(LSTM(64, return_sequences=True))(x2)
x2 = Dropout(0.5)(x2)
x2 = Bidirectional(LSTM(64, return_sequences=True))(x2)
x2 = Flatten()(x2)

x3 = Embedding(25000, 32)(input_layer)
x3 = Flatten()(x3)
x3 = Dense(50, activation="relu")(x3)
x3 = Dropout(0.5)(x3)
x3 = Dense(25, activation="relu")(x3)
x3 = Dropout(0.5)(x3)
x3 = Dense(10, activation="relu")(x3)

concat = Concatenate()([x1, x2, x3])
concat = Dropout(0.5)(concat)
output = Dense(1, activation="sigmoid")(concat)
```

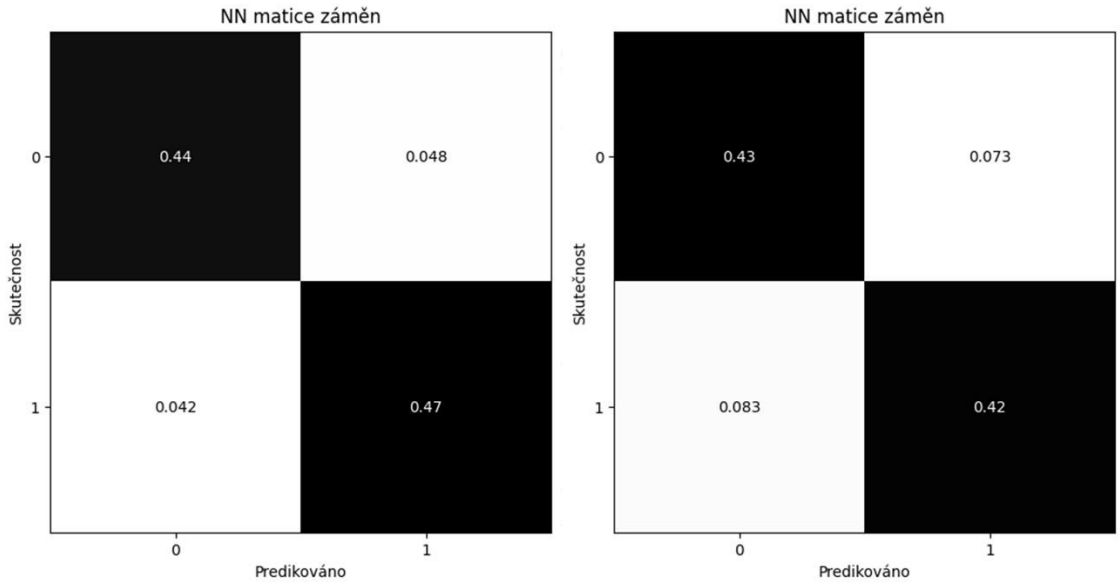
Kód 6: definice ensemble modelu

Následující tabulka zachycuje výsledky ensemble modelů.

Dataset	Typ vektorizace	Přesnost
IMDB	BPE + embedding	91,04 %
Sentiment140	BPE + embedding	84,41 %

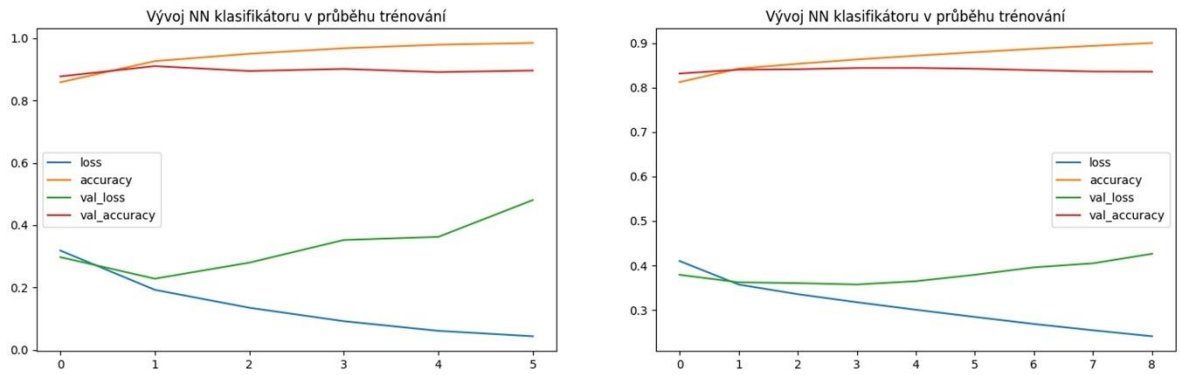
Tabulka 22: přesnosti ensemble modelů

Výsledky ukazují, že ensemble modely pro oba datasety dosáhly ze všech modelů nejvyšší přesnosti, u IMDB datasetu se přesnost poprvé vyšplhala přes hranici 90 %, konkrétně na 91,04 % a u Senitment140 datasetu je nejvyšší přesnost 84,41 %. Další informace poskytnou matice záměn.



Obrázek 23: matice záměn ensemble modelů trénovaných na IMDB datasetu (vlevo) a Sentiment140 datasetu (vpravo)

Na maticích záměn je vidět, že IMDB model predikuje pozitivní sentiment o trochu častěji než ten negativní, zatímco Sentiment140 model to má naopak, i přes to jsou však natrénované modely velice vyvážené.



Obrázek 24: vývoj ensemble modelů trénovaných na IMDB datasetu (vlevo) a Sentiment140 datasetu (vpravo)

Obrázek 24 zachycuje grafy vývoje ensemble modelů při trénování a lze z nich vyčíst, že modely dosáhly své maximální přesnosti v poměrně malém počtu epoch a také že u nich docházelo k overfittingu, což je zřejmé ze stoupající přesnosti na trénovacích datech a zvyšující se chybě (loss hodnotě) na testovacích datech. Je možné, že by se modely by mohly dosáhnout ještě o pár procent lepší přesnosti, kdyby se použil nějaký typ regularizace. Avšak i navzdory overfittingu modely dosáhly nejvyšší přesnosti ze všech natrénovaných modelů.

		Pozitivní			Negativní		
Dataset	Typ vekzorizace	1	2	3	1	2	3
IMDB	BPE + embedding	0,89	0,85	0,55	0,08	0,01	0,36
Sentiment140	BPE + embedding	0,85	0,64	0,80	0,01	0,03	0,03

Tabulka 23: hodnoty predikcí ensemble modelů pro automatickou anotaci

Na hodnotách predikcí lze vidět, že ani jeden z modelů neudělal chybu v klasifikaci testovacích vět a až na pár výjimek se čísla blíží krajním hodnotám, což znamená, že modely by mohly být vhodné pro automatickou anotaci textů.

5.3.2. Shrnutí vlastních natrénovaných modelů

V předchozích částech byly popsány jednotlivé natrénované modely a tato část má za cíl všechny výsledky shrnout. Z použitých vektorizačních metod dosahovala u Sentiment140 datasetu nejlepších výsledků metoda BPE, zatímco u IMDB datasetu to byla častěji metoda frekvenčního tokenizátoru, BPE však bylo s výsledky velice blízko a v některých scénářích také dosáhlo nejlepší přesnosti pro daný scénář. V průběhu analýzy se BPE tokenizátor jevil jako nejvšestrannější způsob vektorizace – kvůli jeho operování s tokeny menšími než slovo je možné pomocí poměrně malého slovníku tokenů

vektorizovat v podstatě jakýkoliv text, což se o žádné jiné z vyzkoušených metod vektorizace tvrdit nedá. Frekvenční tokenizátor je omezen počtem slov a pokud nějaké slovo nezná, nemůže dané slovo reprezentovat jako číslo, tudíž je tento typ vektorizace velice závislý na objemu dat, ze kterého se tokenizátor tvoří. Bag of words sice v některých případech vedlo k velice dobrým výsledkům, avšak vektory, které metoda vytváří pro každý vstupní text jsou obrovské, což extrémně prodlužuje dobu trénování, navíc si bag of words také neporadí s novými slovy a stejně jako frekvenční tokenizátor je efektivita vektorizace závislá na velikosti dat při vytváření globálního slovníku používaného při vektorizaci. Word2Vec embeddingy budou s neznámými slovy potkávat stejné obtíže, avšak tyto předtrénované embeddingy se osvědčily jako velice efektivní způsob reprezentace tokenů, a i když modely, které je používaly, byly v přesnosti často poraženy jinými metodami, jejich vektorové reprezentace slov se i tak zdají být velice kvalitní, o čemž mohou svědčit velice dobré výsledky klasifikátorů při testu hodnot predikcí pro automatickou anotaci.

Co se týče nejvyšší dosažené přesnosti, pro oba datasety byla největší přesnosti dosažena s ensemble modelem skládajícím se z Dense vrstev, LSTM vrstev a Transformeru. Pro IMDB dataset byla nejvyšší přesnost 91,04 % a pro Sentiment140 84,41 %, což jsou celkově nejlepší dosažené výsledky ze všech dosud vyzkoušených metod. V kontextu automatické anotace dat dosahovaly nejlepších výsledků modely trénované na Sentiment140 datasetu s použitím Word2Vec embeddingů, kdy se pokaždé hodnoty predikcí blížily velice blízko jedničce nebo nule, dále také LSTM modely trénované na stejných datech, ale i ensemble modely pro oba datasety. Modely trénované na IMDB datasetu však u testovacích vět pro automatickou anotaci často dělaly chyby v klasifikaci, což by mohlo být způsobeno rozdílnou délkou textů v datasetech – testovací věty se totiž délkou více podobaly trénovacím datům ze Sentiment140 datasetu, což může být důvodem větší úspěšnosti těchto modelů v této analýze.

Pro testování toho, jak moc je model obecný, a tedy aplikovatelný na nová data, která jsou odlišná od těch trénovacích, byl nejlepší model trénovaný na IMDB datasetu otestován na testovací části ze Sentiment140 datasetu a naopak. V obou případech tedy byla testována přesnost ensemble modelu. Výsledky shrnuje tabulka níže.

Trénovací dataset	Testovací dataset	Model	Přesnost
IMDB	Sentiment140	Ensemble	59,45 %
Sentiment140	IMDB	Ensemble	69,02 %

Tabulka 24: přesnost ensemble modelů na nových datech

Z tabulky je hned jasné, že ani jeden model nedosahuje převratných výsledků, avšak model trénovaný na IMDB datech dosáhl přesnosti jen 59,45 %, zatímco model trénovaný na Sentiment140 datech měl přesnost téměř o 10 % vyšší, konkrétně 69,02 %. Zdá se tedy, že model trénovaný na Sentiment140 datasetu je více obecný a je schopnější v klasifikaci neznámých dat než model trénovaný na IMDB

datech. V tomto případě by to však mohlo být dáno tím, že Sentiment140 dataset obsahuje přibližně třicetkrát více dat, což by mohlo vysvětlovat lepší výsledky na nějakých datech výrazně odlišných od těch trénovacích.

5.4. Předtrénované modely

Poslední vyzkoušenou metodou jsou předtrénované modely, které zaslouženě získávají stále větší popularitu. Základem jejich úspěchu je to, že výpočetní výkon potřebný pro inferenci je pouhým zlomkem výkonu potřebného k natrénování modelů, což platí asi pro všechny typy klasifikátorů ze strojového učení. Situace je tedy často taková, že někdo, kdo má přístup k velkému výpočetnímu výkonu, zpravidla tedy velké korporace, natrénuje velký model na obrovském množství dat, který se následně zveřejní s open-source licencí a instrukcemi, jak lze model dále ladit pro nějaký specifický úkol, takzvaný finetuning. Těmto velkým modelům se někdy říká foundation modely, protože samy o sobě v ničem nevynikají, avšak díky velkému množství trénovacích dat mají dobré obecné znalosti, které se poté pomocí finetuningu dají dobře využít při nějakém specializovaném úkolu. Hlavní výhodou těchto modelů tedy je, že někdo jiný vynaložil čas a energii na sběr dat a trénování modelu a další uživatelé se mohou soustředit na více specifické úkoly. Z oblasti předtrénovaných modelů byly vyzkoušeny dva přístupy, konkrétně finetuning obecného modelu a použití předtrénovaného modelu doladěného pro analýzu sentimentu.

5.4.1. Finetuning BERT modelu

BERT (Bidirectional Encoder Representations from Transformers) je model, který se učí obousměrné reprezentace jazyka, tedy kontext napravo i nalevo od aktuálně zkoumaného slova, z neanotovaného textu, a výsledkem je model, který lze pomocí finetuningu snadno upravit v model pro nějaký konkrétní downstream úkol (Devlin et al., 2018). V tomto případě byl model doladěn zvlášť na 50 % IMDB datasetu (10 % z celého datasetu bylo použito pro testování) a na 5 % Sentiment140 datasetu (s 1 % pro testování). Oproti dosud trénovaným modelům je BERT mnohem větší, tudíž i trénování trvá déle a z časových důvodů nebylo možné použít větší část dat. Kvůli požadavkům modelu na rozměr vstupních dat musely být v případě IMDB datasetu vektorizované texty zkráceny na 512 tokenů. Nastavení modelu pro finetuning byl následovný.

```
tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
model= TFAutoModelForSequenceClassification.from_pretrained("bert-base-cased")
model.compile(optimizer=Adam(3e-5))
model.fit(train_generator, epochs=5)
```

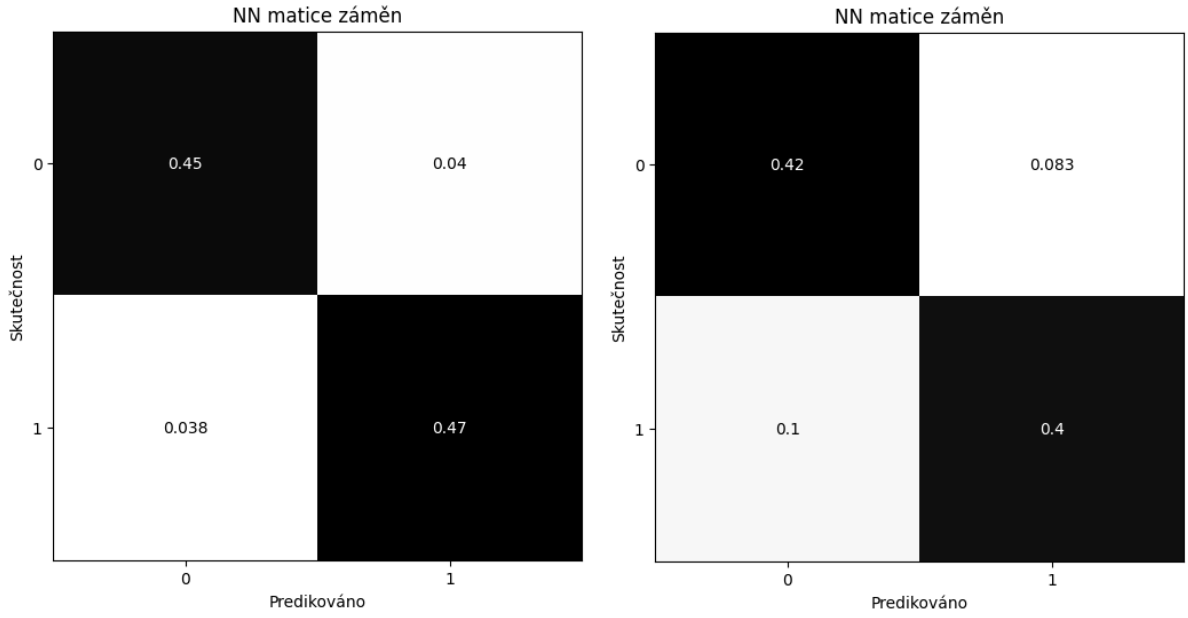
Kód 7: nastavení BERT modelu pro finetuning

Následující tabulka shrnuje výsledky BERT modelu po finetuningu.

Dataset	Přesnost
IMDB	92,22 %
Sentiment140	81,43 %

Tabulka 25: přesnost BERT modelu po finetuningu

Tabulka ukazuje, že u IMDB datasetu se doladěný BERT model s přesností vyšplhal až na 92,22 %, což je dosud nejvyšší naměřená hodnota pro tento dataset. Avšak u Sentiment140 datasetu to je jen 81,43 %, což sice stále velice dobrý výsledek, ale předchozí natrénované modely dosahovaly přesnosti přibližně o 3 % vyšší. Další informace poskytují matice záměn.



Obrázek 25: matice záměn BERT modelu po finetuningu na IMDB (vlevo) a Sentiment140 (vpravo) datasetech

Jak lze na obrázku vidět, model dotrénovaný na IMDB datasetu sice lehce upřednostňuje pozitivní sentiment a model dotrénovaný na Sentiment140 datasetu zase ten negativní, i přes to jsou však oba modely dostatečně vyvážené a dosahují dobrých výsledků. V obou případech je možné, že by se výsledky ještě o něco zlepšily, pokud by model byl dotrénovaný na větším objemu dat, případně i na více epoch.

5.4.2. Twitter RoBERTa

Pro tuto analýzu byl použit model typu RoBERTa, který byl na datech z Twitteru doladěn pro analýzu sentimentu (Loureiro et al., 2022)[10]. Analýzu však trochu komplikovala skutečnost, že tento model zahrnuje i neutrální sentiment a klasifikuje tak sentiment do tří tříd, zatímco oba použité datasety obsahují jen binární klasifikace. Z tohoto důvodu tedy byly všechny neutrální predikce modelu vynechány a pro výpočet přesnosti se použily jen ty pozitivní a negativní. Pro otestování výkonu tohoto předtrénovaného modelu bylo vybráno 2000 textů z každého datasetu. Výsledky shrnuje následující tabulka.

Testovací dataset	Původní počet textů	Počet textů bez neutrálního sentimentu	Přesnost
IMDB	2000	1618	86,89 %
Sentiment140	2000	1436	82,94 %

Tabulka 26: přesnost RoBERTa modelu

Po odstranění neutrálních predikcí zůstalo v IMDB vzorku 1618 textů a v Sentiment140 vzorku 1436 textů. Co se týče přesnosti, tak ensemble modely sice dosáhly vyšší přesnosti, ale jen na testovacím vzorku dat vytvořeného ze stejných dat, na kterých byly modely trénovány. A i když nejsou naměřené přesnosti pro tento model ty nejvyšší v rámci této práce, jsou i přes to výsledky velice dobré a z přesností naměřených u obou datasetů lze tedy usoudit, že RoBERTa model je mnohem obecnější model, který bude pravděpodobně dosahovat velice dobrých výsledků nezávisle na datech. V situacích, kdy by se dalo zajistit, že do modelu budou vždy vstupovat data stejného typu, by tedy ensemble modely mohly být lepším řešením, pokud se dá však očekávat nějaká míra variability v datech, předtrénovaný model bude nejspíš lepší volbou.

5.5. Evaluace modelů na vlastních datech

Posledním krokem je zjistit, jak si modely povedou na vlastním vytvořeném datasetu z části 4.4. Jelikož byl dataset anotován na tři třídy, pro model RoBERTa mohl být použit téměř ve výchozím stavu – stačilo odstranit sporné případy a nerelevantní texty. Pro vlastní natrénované modely však musely být všechny neutrálně anotované texty odstraněny. Doteď byla u všech analýz použita přesnost jako jediná metrika pro hodnocení výkonu modelů, protože datasety byly vyvážené a každá třída tedy byla zastoupena stejnou mírou. V případě tohoto vlastního datasetu jsou však třídy silně nevyrovnané, takže kromě přesnosti byly spočítány i metriky precision, recall a F1-score. Z vlastních modelů byly opět vybrány

[10] Konkrétní použitá verze modelu dostupná z: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest. [Přístup 2023-11-27]

pouze ty nejlepší dva pro každý dataset, tedy ensemble modely. Následující tabulka shrnuje naměřené výsledky.

	Sentiment	Precision	Recall	F1-score	Support
RoBERTa	neutrální	0,93	0,83	0,88	807
	pozitivní	0,64	0,87	0,74	247
	negativní	0,73	0,78	0,75	89
	přesnost			83,2 %	1143
IMDB BERT	negativní	0,54	0,92	0,68	89
	pozitivní	0,96	0,71	0,82	247
	přesnost			76,79 %	336
Sentiment140 BERT	negativní	0,75	0,80	0,77	89
	pozitivní	0,93	0,90	0,91	247
	přesnost			87,5 %	336
IMDB ensemble	negativní	0,41	0,69	0,51	89
	pozitivní	0,85	0,64	0,73	247
	přesnost			65,17 %	336
Sentiment140 ensemble	negativní	0,41	0,88	0,56	89
	pozitivní	0,92	0,55	0,69	247
	přesnost			63,39 %	336

Tabulka 27: výsledky vlastních a předtrénovaných modelů na vlastním datasetu

Precision měří, jaká část z označených dat byla označena správně, recall měří, jaká část z dat náležící k nějaké skupině byla označena správně a F1-score je harmonický průměr vypočítaný z precision a recall. Sloupec s názvem Support zachycuje, kolik položek bylo v dané kategorii. Konkrétně tedy model RoBERTa ze všech textů, které klasifikoval jako neutrální, správně klasifikoval 93 %, ale celkem odhalil jen 83 % neutrálních textů. Nižší precision hodnota u pozivního sentimentu nasvědčuje tomu, že model by mohl texty častěji klasifikovat jako pozitivní, protože ze všech pozitivně predikovaných textů jich bylo správně klasifikováno jen 64 %, avšak model těchto pozitivních textů také klasifikoval nejvíce správně. Negativní sentiment má sice vyšší precision než pozitivní sentiment, má však nejnižší recall, takže model nejhůře odhaluje negativní sentiment. Celkem se přesnost modelu RoBERTa vyšplhala na 83,2 %, což je poměrně vysoká hodnota.

IMDB BERT model má poměrně nízkou hodnotu precision u negativního sentimentu, což může znamenat, že model dává v predikcích přednost negativnímu sentimentu na úkor pozitivních predikcí, na druhou stranu je však recall na velmi vysoké hodnotě. Pro pozitivní sentiment je precision na velice vysoké hodnotě, avšak recall na tom už tak dobře není. Celková přesnost tohoto modelu je 76,79 %, což je sice více než ensemble modely, ale nejméně ze všech předtrénovaných modelů.

Pro Sentiment140 BERT jsou jak precision tak recall na dost podobných hodnotách, což znamená, že když už model dělá nějakou predikci, má tendenci predikovat správně. U pozitivního

modelu jsou však naměřené hodnoty výrazně lepší než u negativního. Tento model dosáhl přesnosti 87,5 %, což je na tomto vlastním datasetu ze všech modelů nejvíce.

U IMDB ensemble modelu lze vidět, že je zde dost nízká precision hodnota pro negativní sentiment, což svědčí o tom, že model častěji predikuje negativní sentiment i v případech, kdy je text pozitivní. U pozitivního sentimentu je precision hodnota mnohem lepší, avšak recall není ani v jednom případě moc vysoký a celková přesnost dosáhla 65,17 %.

Pro Sentiment140 model je to trochu podobné, precision hodnota pro negativní sentiment je stejná, avšak recall pro negativní sentiment je zde ze všech modelů nejvyšší, což ale opět akorát svědčí o tom, že model predikuje negativní sentiment mnohem častěji, čímž se nutně musí schopnost ho odhalovat zlepšit. Tento model má i nejvyšší precision pro pozitivní sentiment, takže pokud tento model predikuje pozitivní sentiment, ve většině případů bude nejspíš klasifikace správná, avšak odhalil celkem jen 55 % všech poozitivních textů. Celková přesnost je ze všech zkoušených modelů nejnižší, a to 63,39 %.

6. Závěr

V této práci bylo nastíněno, jaké specifické aplikace strojového zpracování jazyka a analýzy sentimentu se mohou objevit v automobilovém průmyslu. V úvodu byla vysvětlena motivace celé práce, dále byly v teoretické rovině rozvedeny příkladové situace a možná rizika, kde byla ilustrována důležitost dbání na bezpečnost a etiku. Byly otestovány slovníkové metody, které se při klasifikaci sentimentu příliš neosvědčily. Další na řadu přišly metody strojového učení. Z vyzkoušené trojice metod měla metoda k-nejbližších sousedů (k-NN) nejhorší výsledky, zatímco SVM a Naive Bayes měly poměrně srovnatelné výsledky, avšak Naive Bayes má značnou výhodu v kontextu automatické anotace dat. Také se potvrdila schopnost těchto klasifikátorů dosahovat dobrých výsledků i s malým trénovacím datasetem. Poté byly vyzkoušeny různé typy neuronových sítí, konkrétně Dense, Conv1D, Bidirectional LSTM, Transformer a ensemble modely skládající se z Dense, Bi-LSTM a Transformer vrstev. Ukázalo se, že ensemble modely na datasetech použitých k trénování dosahovaly velice dobrých, avšak ne vždy těch nejlepších nejlepších výsledků – u Sentiment140 datasetu byla nejvyšší naměřená přesnost 84,41 % naměřena u ensemble modelu, ale pro IMDB dataset byla nejvyšší přesnost dosažena pomocí finetuningu předtrénovaného BERT modelu, kdy se přesnost vyšplhala na 92,22 %. Mimo jiné byly také analyzovány způsoby vektorizace, ze kterých byl byte pair encoding vyhodnocen jako nejlepší volba, avšak v kontextu automatické anotace se kombinace neuronových sítí s Word2Vec embeddingy jevila jako nejlepší postup. Závěrem byly také otestovány předtrénované modely, konkrétně model BERT a model RoBERTa doladěný na Twitter datech. U těchto modelů se ukázalo, že jejich všestrannost a klasifikační schopnosti napříč různými typy dat jsou mnohem lepší než u ensemble modelů, které dosahovaly vysoké přesnosti jen na datech stejného typu jako byla jejich trénovací data. U vlastního doménového datasetu byla nejvyšší přesnost 87,5 % dosažena pomocí BERT modelu s finetuningem na

části dat ze Sentiment140 datasetu. Takže i když malé specializované modely určitě mají své místo i využití, v kontextu této práce se ukázalo, že velké předtrénované modely jsou robustnější a podávají konzistentnější výsledky, což je užitečné zejména v případě malých datasetů, na kterých se kvůli jejich velikosti nedá natrénovat obstojný model. Ukázala se také důležitost podoby dat pro finetuning, protože jak vyšlo najevo, finetuning předtrénovaného modelu pomocí dat, která jsou dostatečně podobná datům, na které se má hotový model následně aplikovat, může dosahovat lepší přesnosti než model, který je sice určen a doladěn ke stejnému zadání, avšak neměl při finetuningu data dostatečně podobná těm ostrým. Dalšími kroky ke zlepšení přesnosti některých z těchto modelů by mohl být finetuning na větším objemu dat či na více epoch. Dalo by se tedy říct, že v komerční sféře je pro účely rychlého vytváření prototypů vhodnější použít předtrénovaný model a finetuning, protože díky tomu odpadá nutnost experimentace s hyperparametry vlastnoručně trénovaných modelů, ale jak již bylo zmíněno, i přes to však mohou existovat pádné důvody, proč si trénovat modely vlastní.

7. Bibliografie

ABIODUN, Oludare; JANTAN, Aman; OMOLARA, Abiodun; DADA, Kemi; MOHAMED, Nachaat et al., 2018. State-of-the-art in artificial neural network applications: A survey. online. *Heliyon*. roč. 4, č. 11, s. 1-41. Dostupné z: https://doi.org/https://doi.org/10.1016/j.heliyon.2018.e00938. [cit. 2023-10-09].

AHLSTRÖM, Christer; KIRCHER, Katja; NYSTRÖM, Marcus a WOLFE, Benjamin, 2021. Eye Tracking in Driver Attention Research—How Gaze Data Interpretations Influence What We Learn. online. *Frontiers in Neuroergonomics*. roč. 2. ISSN 2673-6195. Dostupné z: https://doi.org/10.3389/fnrgo.2021.778043. [cit. 2023-09-04].

ALTHNIAN, Alhanoof; ALSAEED, Duaa; AL-BAITY, Heyam; SAMHA, Amani; DRIS, Alanoud et al., 2021. Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain. online. *Applied Sciences*. roč. 11, č. 2. ISSN 2076-3417. Dostupné z: https://doi.org/doi.org/10.3390/app11020796. [cit. 2023-09-15].

BIRD, Steven; KLEIN, Ewan a LOPER, Edward, 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

BOJANOWSKI, Piotr; GRAVE, Edouard; JOULIN, Armand a MIKOLOV, Tomáš, 2016. Enriching Word Vectors with Subword Information. online. *ArXiv preprint*. Dostupné z: https://arxiv.org/pdf/1607.04606.pdf. [cit. 2023-09-07].

CAMBRIA, Erik; DAS, Dipankar; BANDYOPADHYAY, Sivaji a FERACO, Antonio (ed.), 2017. *A Practical Guide to Sentiment Analysis*. online. Socio-Affective Computing. Cham: Springer International Publishing. ISBN 978-3-319-55392-4. Dostupné z: https://doi.org/10.1007/978-3-319-55394-8. [cit. 2023-11-02].

COHEN, Jacob, 1960. A Coefficient of Agreement for Nominal Scales. online. *Educational and Psychological Measurement*. roč. 20, č. 1, s. 37-46. ISSN 0013-1644. Dostupné z: https://doi.org/10.1177/001316446002000104. [cit. 2023-09-28].

CORTES, Corinna a VAPNIK, Vladimir, 1995. Support-vector networks. *Machine Learning*. roč. 20, č. 3, s. 273-297.

COVER, T. a HART, P., 1967. Nearest neighbor pattern classification. online. *IEEE Transactions on Information Theory*. roč. 13, č. 1, s. 21-27. ISSN 0018-9448. Dostupné z: https://doi.org/10.1109/TIT.1967.1053964. [cit. 2023-09-15].

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton a TOUTANOVA, Kristina, 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Online. ArXiv preprint. Dostupné z: https://doi.org/https://doi.org/10.48550/arXiv.1810.04805. [cit. 2023-12-09].

Feelings Wheel, b.r. online. Feelings Wheel. Dostupné z: https://feelingswheel.com/. [cit. 2023-08-15].

FIRTH, John Rupert, 1957. A Synopsis of Linguistic Theory, 1930-1955. In: *Studies in Linguistic Analysis*. Oxford: Blackwell.

FIX, Evelyn a HODGES, J., 1989. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. online. *International Statistical Review / Revue Internationale de Statistique*. roč. 57, č. 3, s. 238-247. Dostupné z: https://doi.org/https://doi.org/10.2307/1403797. [cit. 2023-09-22].

For Academics, b.r. online. In: Sentiment140. Dostupné z: http://help.sentiment140.com/for-students. [cit. 2023-08-09].

GAGE, Philip, 1994. A New Algorithm for Data Compression. online. roč. , č. 12, s. 1-14. Dostupné z: https://docplayer.net/184964300-A-new-algorithm-for-data-compression.html. [cit. 2023-11-27].

GO, Alec; BHAYANI, Richa a HUANG, Lei, 2009. Twitter Sentiment Classification using Distant Supervision. online. Dostupné z: https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf. [cit. 2023-08-09].

GUO, Gongde; WANG, Hui; BELL, David; BI, Yaxin a GREER, Kieran, 2003. KNN Model-Based Approach in Classification. online. In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*. Berlin, Heidelberg: Springer Berlin Heidelberg, s. 986-996. ISBN 978-3-540-39964-3. Dostupné z: https://doi.org/https://doi.org/10.1007/978-3-540-39964-3_62. [cit. 2023-09-25].

HARMON, a ROESSLEIN, Joshua. Tweepy (software). online. Dostupné z: https://doi.org/10.5281/zenodo.7259945. [cit. 2023-09-17].

HARRIS, Zellig S., 2015. Distributional Structure. online. *IWORD/i*. roč. 10, č. 2-3, s. 146-162. ISSN 0043-7956. Dostupné z: https://doi.org/10.1080/00437956.1954.11659520. [cit. 2023-09-06].

HUNTER, J. D., 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. roč. 9, č. 3, s. 90-95.

HUTTO, C. a GILBERT, E., 2014. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In: *Proceedings of the 8th International Conference on Weblogs and Social Media*. ISBN ISBN 978-1-57735-659-2. Dostupné z: https://doi.org/https://doi.org/10.1609/icwsm.v8i1.14550.

CHOLET, Francois, 2015. *Keras*. online. Dostupné z: keras.io. [cit. 2023-11-27].

JURAFSKY, Daniel a MARTIN, James, 2023a. Vector Semantics and Embeddings. online. In: *Speech and Language Processing*. 3rd ed. draft. ISBN 978-0-13-095069-7. Dostupné z: https://web.stanford.edu/~jurafsky/slp3/6.pdf. [cit. 2023-11-27].

JURAFSKY, Daniel a MARTIN, James, 2023b. Naive Bayes and Sentiment Classification. online. In: *Speech and Language Processing*. 3rd ed. draft. ISBN 978-0-13-095069-7. Dostupné z: https://web.stanford.edu/~jurafsky/slp3/4.pdf. [cit. 2023-11-27].

JURAFSKY, Daniel a MARTIN, James, 2023c. Neural Networks and Neural Language Models. online. In: *Speech and Language Processing*. 3rd ed. draft. s. 1-27. ISBN 978-0-13-095069-7. Dostupné z: https://web.stanford.edu/~jurafsky/slp3/7.pdf. [cit. 2023-10-09].

KECMAN, V., 2005. Support Vector Machines – An Introduction. In: *Support Vector Machines: Theory and Applications*. Berlin: Springer, s. 1-47.

LESCH, Kateřina, 2017. *Sentiment analysis in Czech*. Studies in computational and theoretical linguistics. Praha: Ústav formální a aplikované lingvistiky. ISBN 978-80-88132-03-5.

LIANG, Xuefeng; LIU, Xingyu a YAO, Longshan, 2022. Review–A Survey of Learning from Noisy Labels. *ECS Sensors Plus*. roč. 1, č. 2. ISSN 2754-2726. Dostupné z: https://doi.org/10.1149/2754-2726/ac75f5.

LIU, Yinhan; OTT, Myle; GOYAL, Naman; DU, Jingfei; JOSHI, Mandar et al., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. online. Dostupné z: https://doi.org/doi.org/10.48550/arxiv.1907.11692. [cit. 2023-11-27].

LORIA, Steve, 2020. *TextBlob*. online. In: Read the Docs. Dostupné z: https://textblob.readthedocs.io/en/dev/. [cit. 2023-08-09].

LOUREIRO, Daniel; BARBIERI, Francesco; NEVES, Leonardo; ESPINOSA ANKE, Luis a CAMACHO-COLLADOS, Jose, 2022. TimeLMs: Diachronic Language Models from Twitter. online. *ArXiv preprint*. Dostupné z: https://doi.org/https://doi.org/10.48550/arXiv.2202.03829. [cit. 2023-11-27].

MAAS, Andrew; DALY, Raymond; PHAM, Peter; HUANG, Dan; NG, Andrew et al., 2011. Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland: Association for Computational Linguistics, s. 142-150. ISBN 978-1-932432-88-6.

MCCALLUM, Andrew a NIGAM, Kamal, 1998. A comparison of event models for naive bayes text classification. online. In: *AAAI Conference on Artificial Intelligence*. Dostupné z: https://api.semanticscholar.org/CorpusID:7311285. [cit. 2023-09-21].

MCCULLOCH, Warren a PITTS, Walter, 1943. A logical calculus of the ideas immanent in nervous activity. online. *The bulletin of mathematical biophysics*. roč. 5, s. 115-133. Dostupné z: https://doi.org/10.1007/BF02478259. [cit. 2023-10-09].

MEJOVA, Yelena, 2009. *Sentiment Analysis: An Overview*. online, Comprehensive Exam Paper. University of Iowa. Dostupné z: https://www.researchgate.net/profile/Yelena-Mejova/publication/264840229_Sentiment_Analysis_An_Overview/links/590ad68e0f7e9b1d0823eff2/Sentiment-Analysis-An-Overview.pdf. [cit. 2023-08-09].

META AI, b.r. online. In: META AI. Papers With Code. Dostupné z: https://paperswithcode.com/datasets. [cit. 2023-09-28].

MIKOLOV, Tomáš; CHEN, Kai; CORRADO, Greg a DEAN, Jeffrey, 2013a. Efficient Estimation of Word Representations in Vector Space. online. *ArXiv preprint*. Dostupné z: https://arxiv.org/pdf/1301.3781.pdf. [cit. 2023-09-06].

MIKOLOV, Tomáš; SUTSKEVER, Ilya; CHEN, Kai; CORRADO, Greg a DEAN, Jeffrey, 2013b. Distributed Representations of Words and Phrases and their Compositionality. online. *ArXiv preprint*. Dostupné z: https://arxiv.org/pdf/1310.4546.pdf. [cit. 2023-09-06].

Naive Bayes, c2007-2023. online. In: Scikit-learn. Dostupné z: https://scikit-learn.org/stable/modules/naive_bayes.html#naive-bayes. [cit. 2023-10-05].

NANDAN, Apoorv, b.r. *Text classification with Transformer*. online. In: Keras. Dostupné z: https://keras.io/examples/nlp/text_classification_with_transformer/. [cit. 2023-11-26].

NI, Jianmo; LI, Jiacheng a MCAULEY, Julian, 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. online. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, s. 188-197. Dostupné z: https://doi.org/10.18653/v1/D19-1018. [cit. 2023-09-04].

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B. et al., 2011. Scikit-learn: Machine Learning in Python. online. *Journal of Machine Learning Research*. roč. 12, č. 85, s. 2825-2830. Dostupné z: http://jmlr.org/papers/v12/pedregosa11a.html. [cit. 2023-08-15].

RASCHKA, Sebastian, 2014. Naive Bayes and Text Classification – Introduction and Theory. online. *ArXiv preprint*. s. 1-20. Dostupné z: https://arxiv.org/abs/1410.5329. [cit. 2023-09-19].

ŘEHŮŘEK, Radim a SOJKA, Petr, 2010. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, s. 45-50.

SAGI, Omer a ROKACH, Lior, 2018. Ensemble learning: A survey. online. *WIREs Data Mining and Knowledge Discovery*. roč. 8, č. 4, s. 1-18. ISSN 1942-4787. Dostupné z: https://doi.org/10.1002/widm.1249. [cit. 2023-10-17].

SAHAR, Yotam; ELBAUM, Tomer; WAGNER, Michael; MUSICANT, Oren; HIRSH, Tehila et al., 2021. Grip Force on Steering Wheel as a Measure of Stress. online. *Frontiers in Psychology*. roč. 12. ISSN 1664-1078. Dostupné z: https://doi.org/10.3389/fpsyg.2021.617889. [cit. 2023-09-04].

SARKER, Iqbal H., 2021. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. online. *SN Computer Science*. roč. 2, č. 6, s. 1-20. ISSN 2662-995X. Dostupné z: https://doi.org/10.1007/s42979-021-00815-1. [cit. 2023-10-15].

Sentiment, c2011. online. In: *Slovník spisovného jazyka českého*. Ústav pro jazyk český. Dostupné z: https://ssjc.ujc.cas.cz/search.php?hledej=Hledat&heslo=sentiment&sti=EMPTY&where=hesla&hsubstr=no. [cit. 2023-08-15].

Sentiment, c2022. online. In: *Wikipedia: the free encyclopedia*. San Francisco (CA): Wikimedia Foundation. Dostupné z: https://cs.wikipedia.org/wiki/Sentiment. [cit. 2023-08-09].

Sentiment140 dataset with 1.6 million tweets, b.r. online. In: Kaggle. Dostupné z: https://www.kaggle.com/datasets/kazanova/sentiment140. [cit. 2023-08-09].

Sentiment140, b.r. online. In: Tensorflow. Dostupné z: https://www.tensorflow.org/datasets/catalog/sentiment140. [cit. 2023-08-09].

Sklearn.neighbors.KNeighborsClassifier, c2007-2023. online. In: Scikit-learn. Dostupné z: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier. [cit. 2023-10-05].

Support Vector Machines, c2007-2023. In: Scikit-learn. Dostupné z: https://scikit-learn.org/stable/modules/svm.html#scores-probabilities. [cit. 2023-10-05].

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion et al., 2017. Attention Is All You Need. In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. s. 5998-6008. ISBN 9781510860964.

WEBB, G.I., 2011. Naïve Bayes. online. In: *Encyclopedia of Machine Learning*. Boston, MA: Springer, s. 713-714. ISBN 978-0-387-30164-8. [cit. 2023-09-19].

What Is Unsupervised Learning?, b.r. online. In: IBM. Dostupné z: https://www.ibm.com/topics/unsupervised-learning. [cit. 2023-08-14].

WU, Bichen; XU, Chenfeng; DAI, Xiaoliang; WAN, Alvin; ZHANG, Peizhao et al., 2020. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. online. *ArXiv preprint*. Dostupné z: https://arxiv.org/abs/2006.03677. [cit. 2023-11-27].

ZHANG, Harry, 2004. The Optimality of Naive Bayes. online. In: *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*. Dostupné z: https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf. [cit. 2023-10-05].

ZHU, Xiaofeng; ZHANG, Lei a HUANG, Zi, 2014. A Sparse Embedding and Least Variance Encoding Approach to Hashing. online. *IEEE Transactions on Image Processing*. roč. 23, č. 9, s. 3737-3750. ISSN 1057-7149. Dostupné z: https://doi.org/10.1109/TIP.2014.2332764. [cit. 2023-09-25].