Filozofická fakulta Univerzity Palackého v Olomouci

Katedra obecné lingvistiky

# Software na klasifikaci textů

*bakalářská diplomová práce*

Autor: Veronika Vlasáková

Vedoucí práce: Mgr. Vladimír Matlach, Ph.D.

**Olomouc**

2020

## Prohlášení

Prohlašuji, že jsem bakalářskou/magisterskou diplomovou práci „Software na klasifikaci textů“ vypracoval/a samostatně a uvedl/a jsem veškerou použitou literaturu a veškeré použité zdroje.

V Praze dne 17.08.2020 Podpis

## Poděkování

Tímto děkuji vedoucímu mé bakalářské práce, Mgr. Vladimíru Matlachovi, Ph.D., za jeho odborné vedení, rady a ochotu.

# Abstrakt

Název práce: Software na klasifikaci textů

Autor práce: Veronika Vlasáková

Vedoucí práce: Mgr. Vladimír Matlach, Ph.D.

Počet stran a znaků: 73 stran, 82945 znaků

Počet příloh: 2

Abstrakt (minimálně 900 znaků): Cílem této bakalářské práce je v programovacím jazyce Python vytvořit software, který umožní uživateli trénovat klasifikaci textů a evaluovat její výsledky. Teoretická část této práce představuje software a vysvětluje, jak ho má uživatel použít. Konkrétně je v ní popsáno, s jakými vlastnostmi software pracuje, jak se dají texty předzpracovat, jak vyhodnotit výsledky použitých metod a jak software nainstalovat a nastavit. V praktické části této práce je ukázáno, jak software pracuje s konkrétními problémy. Vyzkoušeno je několik různých druhů klasifikace textů. Jedná se o rozpoznání jazyka, a to u náhodně vybraných jazyků a jazyků ze stejné rodiny, určení autorství u profesionálních autorů a u neprofesionálních autorů, rozpoznání sentimentu a rozpoznání spamu. Výsledky jednotlivých klasifikací textů jsou poté evaluovány. Závěr celou práci shrnuje a představuje možnosti dalšího rozšíření a zlepšení softwaru.

Klíčová slova: Zpracování přirozeného jazyka, Python, Kvantitativní lingvistika, Klasifikace textů, Bag of words

## Abstract

Title: Software for text classification

Author: Veronika Vlasáková

Supervisor: Mgr. Vladimír Matlach, Ph.D.

Number of pages and characters: 73 pages, 82945 characters

Number of appendices: 2

Abstract (900 characters): The aim of this bachelor thesis is to create software in the programing language Python, which would allow the user to train text classification and evaluate its results. The theoretical part of this thesis introduces the software and explains how to use it. Specifically, it is described what feature the software works with, how the texts can be pre-processed, how to evaluate the results of the chosen methods and how to install and configure the software. In the practical part of this thesis it is shown how the software works with specific issues. Several different kinds of text classification are tested. It is language recognition, on randomly chosen languages and on languages from the same family, author identification with professional authors and with nonprofessional authors, sentiment detection and spam detection. The results of each text classification are then evaluated. The conclusion summarizes the entire thesis and introduces the possibilities of expansion and improvement of the software.



Keywords: Natural language processing, Python, Quantitative linguistics, Text classification, Bag of words

# Obsah

# Seznam tabulek

# Seznam grafů

# 1 Úvod

Primárním cílem této práce je napsat program v jazyce Python, který za pomocí zpracování přirozeného jazyka umožní uživateli trénovat klasifikace textů a následně evaluovat výsledky. Tento program přiblíží strojové učení a automatickou klasifikaci textů uživatelům bez další znalosti programování. Zdrojový kód tohoto programu je dostupný na přiloženém CD. Cílem této psané práce je tento program představit. Text této práce je rozdělen na teoretickou a praktickou část.

Teoretická část obsahuje čtyři kapitoly. První se zabývá vlastnostmi, na které se text převede, a se kterými následně pracují modely strojového učení. Druhá kapitola se zabývá možnostmi předzpracování textu za cílem lepších výsledků. V třetí kapitole jsou popsány typy použitých metody, v jakém formátu se ukládají výsledky a jak je interpretovat. Poslední kapitola popisuje, jak samotný program vypadá, jak ho nainstalovat, nastavit a spustit.

Následuje praktická část. V té budou vyzkoušeny různé typy klasifikace textu a následně vyhodnoceny jejich výsledky. Jedná se o rozpoznání jazyka, zkoušené na angličtině a češtině, pěti náhodně vybraných jazycích a čtyřech jazycích ze stejné rodiny. Ve všech případech se jedná o paralelní korpusy. Dále bude vyzkoušeno určení autorství, a to na textech profesionálních autorů a neprofesionálních autorů. Zároveň bude i vyzkoušeno, zda je těžší určit autory, kteří jsou si bližší. V případě profesionálních autorů se jedná o autory stejného žánru a v případě neprofesionálních autorů se jedná o autory ze stejné demografické skupiny. Poté bude vyzkoušeno rozpoznání sentimentu, a nakonec rozpoznání spamu. V závěru bude práce shrnuta a budou popsány možnosti rozšíření programu.

# 2 Teoretická část

## 2.1 Vlastnosti

Jelikož počítače zpracovávají pouze čísla, musí být texty reprezentovány jako číselné vlastnosti. V případě této práce půjde o jejich primární reprezentaci pomocí indexů, které obsahují informace o textu, zejména o bohatosti jeho slovníku, a o *bag-of-words*, řešící výskyt konkrétních slov v textu. Tyto jednotlivé vybrané reprezentace textu si nyní představíme.

### 2.1.1 Types to tokens ratio

*Types to tokens ratio* (poměr typů k tokenům, dále TTR) měří počet unikátních slov (typů) vůči celkovému počtu slov (tokenům). Nejvyšší TTR je jedna, a to za předpokladu, že jsou všechna slova použita je jednou. Nejnižší TTR ovšem bude vždy vyšší než nula, protože se v textu musí vyskytovat alespoň jedno unikátní slovo. Hodnota potom záleží na velikosti textu. Pokud má

text sto stejných slov, TTR bude 0,01. Pokud má text sto tisíc stejných slov, TTR bude 0,00001. Na TTR se tedy přímo podepisuje délka textu.

### 2.1.2 Repeat rate

$$RR = \sum_{r=1}^{V} p_r^2$$

*Repeat rate* (míra opakování, dále RR) ukazuje, jak moc se slova v textu opakují. Vypočítá se vydělením počtu opakujících se slov počtem všech slov. Ve zde uvedené rovnici *V* značí počet typů, *p* pravděpodobnost a *r* slovo. Výsledek RR se pohybuje mezi nulou a jedničkou. Nula znamená, že se žádná slova v textu neopakují. Jednička znamená, že se v textu opakují všechna slova. Tudíž čím větší je RR, tím více se slova v textu opakují. (Čech, Popescu a Altman 2014)

### 2.1.3 Giniho koeficient

$$G = \frac{1}{V}\left(V + 1 - 2m_1^{'}\right)$$

Giniho koeficient je oblast mezi pravidelnou distribucí a Lorenzovou křivkou. Pravidelná distribuce nastává ve chvíli, kdy se všechna slova v text vyskytují ve stejném počtu. Lorenzova křivka „graficky vyjadřuje kumulativní distribuční funkci slov v textu.“ (Čech, Popescu a Altman 2014) Větší Giniho koeficient značí menší slovník. Na výše uvedené zjednodušené rovnici *V* značí typy, *N* tokeny a *m* průměr frekvenční distribuce. (Čech, Popescu a Altman 2014)

### 2.1.4 Shannonova entropie

$$H = -\sum_{r=1}^{V} p_r \log_2 p_r$$

Shannonova entropie označuje míru nahodilosti. Větší entropie značí větší slovník. U každého unikátního slova (r) se vypočítá pravděpodobnost (p) jeho výskytu a ta se vynásobí binárním logaritmem pravděpodobnosti. Poté se výsledky všech slov sečtou. (Čech, Popescu a Altman 2014)

### 2.1.5 Průměrná délka slov v textu

Poslední z indexů jako jediný nepracuje s bohatostí slovníku. Je jím průměrná délka slov, která je v případě této práce počítána průměrem počtu písmen u jednotlivých grafických slov.

### 2.1.6 Bag-of-words

Poslední a největší vlastností je *bag-of-words*. Jedná se frekvenční tabulku. U každého textu je zaznamenáno kolikrát se v něm vyskytuje každé konkrétní slovo, které lze najít v slovníku všech použitých textů. *Bag-of-words* může být binární. Místo konkrétních frekvencí se pouze zaznamenává, zda se slovo v textu vůbec vyskytuje. Dále je možné hranici posunout a slovo považovat za přítomné, pouze pokud se vyskytuje alespoň n-krát. Tímto se může eliminovat vliv slov, které se v textu málo vyskytují. V této práci je používána frekvenční *bag-of-words* a binární *bag-of-words* s citlivostí 1 a s citlivostí 2.

Jako vlastnosti se potom použijí frekvence jednotlivých slov v každém textu. Na rozdíl od indexů, *bag-of-words* nepopisuje texty jednou vybranou vlastností. Počet vlastností se rovná počtu sjednocených typů všech textů.

### 2.1.7 Normalizace vlastností

Jednotlivé vlastnosti jsou reprezentovány různými rozsahy čísel. Například index RR se pohybuje na škále 0-1. Naproti tomu průměrná délka slov nikdy nebude nižší než 1. 1 je nejvyšší možná hodnota u indexu RR a zároveň nejnižší možná hodnota u průměrné délky slov. Aby se vlastnosti daly vzájemně porovnat musí být převedeny na stejnou škálu. Tomu se říká normalizace. (Han, Kamber a Pei 2011) V programu vytvořeným v rámci této práce jsou na výběr dva typy normalizace: *min-max* a *z-score*. *Min-max* pracuje s nejnižšími a nejvyššími hodnotami na základě kterých určí škálu pro celou vlastnost. *Z-score* používá průměr a směrodatnou odchylku ke standardizaci vlastnosti tak, aby měla průměr vždy v 0 a směrodatnou odchylku 1. (Han, Kamber a Pei 2011) V této práci je z ilustrativních důvodů používán *min-max*.

## 2.2 Úprava textu

Než je z textu možné získat představené vlastnosti, prostý text je nutné předzpracovat. Text se nejprve musí rozdělit na kratší segmenty. Poté se může dál upravovat, nicméně další úpravy jsou již závislé na jazyku textu. Tudíž všechny úpravy textů nejsou dostupné ve všech jazycích a nejde je použít, pokud jsou v jednom datasetu texty v různých jazycích. Jednotlivé možnosti úpravy textu budou nyní blíže vysvětleny.

### 2.2.1 Tokenizace

Tokenizace je z úprav textu jediná povinná, program by bez ní nefungoval. Jedná se o rozdělení textu na tokeny. Ty mohou mít různou podobu, která se specifikuje pomocí regulérního výrazu. V praktické části této práce je použit regulérní výraz \W+, který oddělí grafická slova pomocí nealfanumerických znaků (mezery, čárky, tečky a další interpunkce).

### 2.2.2 Odstranění stop slov

Další úpravy jsou závislé na jazyku textu. První z nich je odstranění stop slov. Stop slova jsou typicky nejběžnější slova v jazyce, která sama o sobě nenesou význam. V programu je k odstranění stop slov použita knihovna stop-words (Savand 2018), která obsahuje seznamy stop slov v různých jazycích. Jedná se o 21 převážně evropských jazyků. V této práci je použita pouze čeština a angličtina.

### 2.2.3 Stemming

Další možnost úpravy je *stemming*. Jedná se o zkrácení na kořen slova. Prakticky však jde o odstranění nejčastějších afixů. V programu je *stemming* možný pouze v angličtině, použit je *PorterStemmer*, který je součástí knihovny *Natural Language Toolkit* (NLTK). (Bird, Loper a Klein 2009)

### 2.2.4 Lemmatizace

Místo *stemmingu* mohou být slova převedena na lemma. Lemma jsou reprezentativní formy slov, například u podstatných jmen nominativ singuláru, u sloves infinitiv. Lemmatizace sjednotí například různé pády stejného slova, které jsou následně počítány pouze jako jeden typ. V programu je možné lemmatizovat texty pouze v češtině a angličtině. V češtině je na lemmatizaci použita Morphodita (Straka a Straková, MorphoDiTa: Morphological Dictionary and Tagger 2014) a v angličtině *WordNetLemmatizer*, který je součástí knihovny NLTK. (Bird, Loper a Klein 2009)

### 2.2.5 Tagy

Namísto používání tokenizovaných slov je možné použít tagy. Tagy obsahují morfologické a další informace o slově. V češtině mají tagy patnáct pozic. Každá z pozic nese informaci o slově, například první pozice indikuje slovní druh. Některé pozice se neurčují u všech slovních druhů (např. čas u adjektiv) a jsou tedy naznačeny pomlčkou. V programu je možné převést na tagy pouze texty v češtině nebo angličtině. V češtině je použita opět Morphodita, (Straka a Straková, MorphoDiTa: Morphological Dictionary and Tagger 2014) v angličtině funkce *pos-tag*, která je součástí knihovny NLTK. (Bird, Loper a Klein 2009)

## 2.3 Interpretace výsledků

Výsledky strojového učení a nastavení, které bylo k získání výsledků použito, se uloží do vybrané složky ve formě reportu ve formátu html či json. Metody strojového učení, které jsou použity v této práci, se dají rozdělit na metody učící se s učitelem a vizualizační metody. Zde

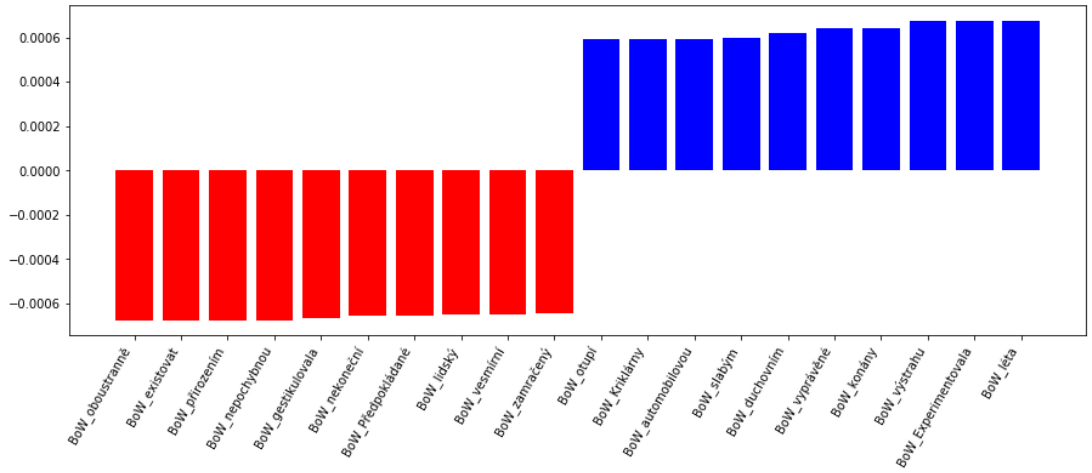
použité metody učící se s učitelem jsou *Support Vector Machine* (metoda podpůrných vektorů, dále SVM), *Linear Discriminant Analysis* (lineární diskriminační analýza, dále LDA), *K-Nearest Neighbors* (*K* nejbližších sousedů, dále KNN), *Naive Bayes* (Naivní Bayesovská metoda, dále NB) a *Decision Tree* (rozhodovací strom).

Výsledky metod učících se s učitelem jsou uložené v podobě tabulek s přesností. Tyto tabulky obsahují hodnoty testovacího a trénovacího datasetu. Čím vyšší přesnosti, a to hlavně u testovacích dat, tím lépe. Data jsou na škále 0-1, jedná se ovšem o procenta. 1.0 tedy znamená 100 %. Kromě přesnosti (*accuracy*) se ukládá i *recall*, *precision* a f1, a to u všech skupin zvlášť. *Recall* ukazuje, jak často byla určena daná skupina, *precision* jak často byla určená správně z těch, které označené byly. F1 je průměr *recallu* a *precision*. *Accuracy* ukazuje kolik procent dat bylo určeno správně. Příklad jednotlivých výsledků pro klasifikaci autorství můžete vidět v tabulce 1.

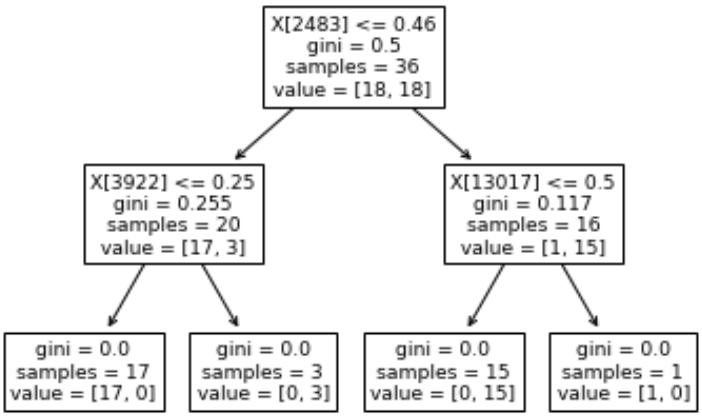
| | | |
|---|---|---|
| accuracy train | | 1.0 |
| accuracy test | | 0.9166666666666666 |
| recall train | Asimov | 1.0 |
| | Lem | 1.0 |
| recall test | Asimov | 1.0 |
| | Lem | 0.8333333333333334 |
| precision train | Asimov | 1.0 |
| | Lem | 1.0 |
| precision test | Asimov | 0.8571428571428571 |
| | Lem | 1.0 |
| f1 train | Asimov | 1.0 |
| | Lem | 1.0 |
| f1 test | Asimov | 0.923076923076923 |
| | Lem | 0.9090909090909091 |

**Tabulka 1**: Úspěšnost modelu Naive Bayes.

U lineárního SVM a NB se do výsledků ukládají také grafy s nejpodstatnějšími vlastnostmi. Jedná se o vlastnosti, které byly nejvíce užitečné při určení skupiny. Takto se děje jen za předpokladu, že se jedná o binární klasifikaci. V případě SVM se vlastnosti rozdělí a jedna vlastnost tedy nemůže být přítomna u obou skupin. U NB toto možné je. U *Decision Tree* se také ukládá graf s rozhodnutími o klasifikaci textů. Příklady těchto grafů můžete vidět na grafu 1 a grafu 2.

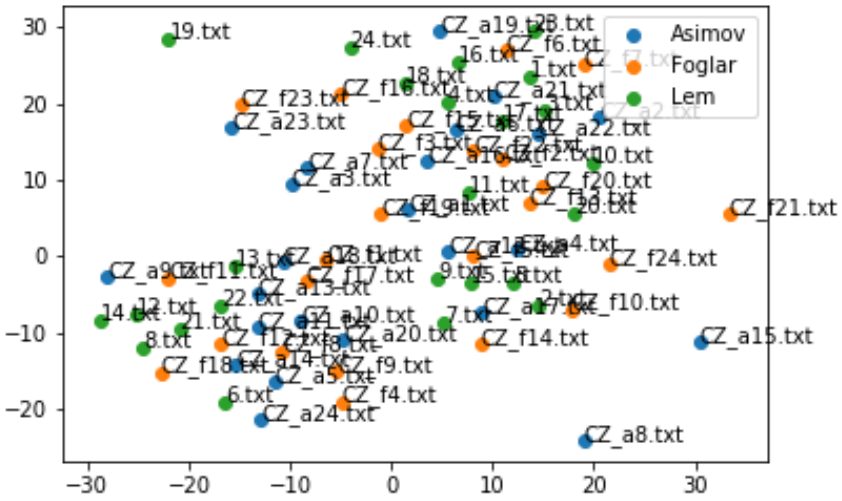



**Graf 1**: Nejpodstatnější vlastnosti u SVM.




**Graf 2**: Proces rozdělování do skupin u *Decision Tree*.

Vybrané metody vizualizace vícerozměrných dat jsou *Multidimensional Scaling* (multidimenzionální škálování, dále MDS), *Principal Component Analysis* (analýza hlavních komponent, dále PCA) a *t-distributed stochastic neighbor embedding* (*t*-distribuované stochastické vkládání sousedů, dále t-SNE). Ty jsou uloženy jako bodové grafy, kde jsou třídy zaznačeny rozlišnými barvami. Tyto grafy musí být interpretovány člověkem. V ideálním případě by se skupiny zcela oddělily. Na grafu 3 můžete vidět příklad výsledků klasifikace tří autorů.




**Graf 3**: Výsledky t-SNE.

## 2.4 Implementace v Pythonu

Program je napsán v programovacím jazyce Python, konkrétně ve verzi 3.7. (Van Rossum a Drake 2009) Python je pro zpracování přirozeného jazyka (dále NLP) vhodný, jelikož již obsahuje velké množství *open source* knihoven, které se NLP věnují. Do jisté míry je také pro tyto účely jednodušší na použití oproti jiným programovacím jazykům. (Thanaki 2017, Srinivasa-Desikan 2018) Následující části práce se zabývají instalací a nastavením programu.

### 2.4.1 Instalace programu

Pro spuštění programu je potřeba mít nainstalovaný Python. Pro instalaci Pythonu je vhodné použít prostředí Anaconda, (Anaconda inc. 2016) které je dostupné zdarma pro Windows, MacOs a Linux na https://www.anaconda.com/products/individual. Python se dá naistalovat i jinak, nicméně Anaconda je pro tento účel nejvhodnější, jelikož již obsahuje předinstalované knihovny na zpracování dat, včetně knihoven použitých v této práci. Instalace probíhá jednoduše, z již zmíněné stránky se stáhne potřebná verze instalatéru a projde se instalací.

Po instalaci pythonu je potřeba naistalovat knihovny, které nejsou součástí Anacondy. Jedná se o knihovnu stop-words, (Savand 2018) která obsahuje seznamy stop slov v nejpoužívanějších evropských jazycích, Morphoditu, (Straka a Straková, MorphoDiTa: Morphological Dictionary and Tagger 2014) která umožňuje lemmatizaci a tagování v češtině a json2html, (Malhotra 2019) na převádění formátu json na formát html. Pro instalaci knihoven je vhodné použít pip. K tomu je potřeba otevřít aplikaci Anaconda Prompt, která je součástí Anacondy, a zadat příkaz pip install a název knihovny, kterou chcete instalovat, a odsouhlasit klávesou *enter*. Názvy knihoven jsou v tomto případě stop-words, ufal.morphodita a json2html.

Aby fungovala Morphodita, je také za potřebí stáhnout jazykový model. V této práci je použita nejnovější verze českého modelu s číslem 161115, který je k dispozici na stránce ufal.mff.cuni.cz/morphodita. (Straka a Straková, Czech Models (MorfFlex CZ 161115 + PDT 3.0) for MorphoDiTa 161115 2016)

### 2.4.2 Nastavení a spuštění programu

Pro používání programu stačí, aby uživatel interagoval pouze se souborem main.py. Do něj se zadávají konkrétní parametry, které chce uživatel použít.

První informace, kterou musí v nastavení uživatel zadat, je adresa dat, které chce uživatel zkoumat. V programu jsou na výběr dva formáty, a to csv a textové soubory txt. Při použití formátu csv je nutné zadat celou adresu souboru. Použití textových souborů je o něco komplikovanější. Jednotlivé texty musí být v samotných souborech, roztříděných do složek podle skupin, které mají být rozděleny. Adresa je napsána až po složku, ve které se data nacházejí. Poté následuje první hvězdička, která značí libovolnou podsložku a druhá hvězdička, která značí všechny soubory s vybranou koncovkou, v tomto případě .txt. V další kolonce je nutné zapsat, který formát je použit.

Užití formátu CSV:

```
SETTING_PATH = "c:/Data/spam.csv"
SETTING_INPUT_FORMAT = "csv
```

Užití jednotlivých textových souborů ve vlastních adresářích:

```
SETTING_PATH = "c:/Data/*/*.txt"
SETTING_INPUT_FORMAT = "txt"
```

Další nastavení programu je předzpracování textu, které udává kompletní proces úpravy textu (tzv. *pipeline*) ve formě *dictionare*. Pod „n“ se píše název daného procesu, pod „p“ jeho jednotlivé parametry. Povinná je pouze tokenizace, bez které by se text nerozdělil na menší segmenty. Nastavení tokenizace se zadává v regulárních výrazech. Základní nastavení je „\W+“, které oddělí grafická slova, ale je možné použít i jiná rozdělení. Ostatní zpracování je možné vynechat.

U redukce se specifikuje počet tokenů, na které chce uživatel texty zkrátit, a zda chce, aby byly tokeny brány náhodně, či aby bylo vybráno prvních *n* tokenů. V této práci jsou je promíchání slov zapnuto. Vypnutí promíchání slov je nutné, pokud je potřeba ponechat kontext slov. Tento a několik následujících parametrů se ovládá napsáním *True* pro spuštění a *False* pro vypnutí. U ostatních zpracování je třeba zadat jazyk, a to v angličtině a malými písmeny.

U lemmatizace a tagování se také zadává umístění souboru jazykového modelu. Toto je potřeba pouze u češtiny, v jiném případě zde uživatel může nechat *None*.

```
  SETTING_NLP_PIPELINE = [{
  "n" : "tokenize", "p" : {"regex_split" : "\W+"}},
  {"n" : "reduce", "p" : {"to" : 1000, "randomize": True}},
  {"n" : "lemmatize", "p" : {"language" : "english", morpho-
  dita_path" : None}},
  {"n" : "remove_stopwords", "p" : {"language" : "english"}}
]
```

Následuje vektorizace. V tomto nastavení se volí, na jakou formu budou data převedena. Stejně jako u zpracování textu se do „n“ píše název a pod „p“ parametry. Parametry jsou zde potřeba pouze u *bag-of-words*, kde se zadává, zda se jedná o binární či frekvenční *Bag-of-words*, případně na jaké číslo má být binární *bag-of-words* nastavena. Uživatel si může vybrat, zda chce *bag-of-words* a konkrétní indexy použít. Program však potřebuje víc než jednu vlastnost, tudíž při vynechání *bag-of-words* musí zůstat alespoň dva indexy. *Bag-of-words* se může použít samostatně, jelikož na rozdíl od indexů vytváří více vlastností.

```
  SETTING_VECTORIZATION = [
  {"n" : "bow", "p" : {"binary" : True, "binary_number" : 1}},
  {"n" : "index_ttr"},
  {"n" : "index_gini_coeficient"},
  {"n" : "index_shannon_entropy"},
  {"n" : "index_rr"},
  {"n" : "index_average_word_lenght"}
]
```

Uživatel také volí, zda chce, aby se pořadí textů promíchalo. Promíchání textů se vypíná v případě, že je potřeba pokus zopakovat se stejným pořadím textů. V této práci je promíchání textů zapnuté.

```
  SETTING_SHUFFLE = True
```

Dále je potřeba vybrat normalizaci. Na výběr je *min max* nebo *z-score*.

```
  SETTING_NORMALIZATION = {"min_max" : True, "z_score" :
  False}
```

Uživatel si může vybrat, které metody chce použít. Ideálně lze použít všechny, nicméně některé metody můžou s větším množstvím dat trvat delší dobu. Při velkém množství dat se navíc vizualizační metody stávají nečitelné (viz graf 43) a mohou být vynechány.

```
SETTING_METHODS = {"svm" : True, "lda" : True, "knn" : True,
"nb" : True, "decision_tree" : True, "mds" : True, "pca" :
True, "tsne" : True}
```

KNN a t-SNE navíc potřebují specifické parametry. Jedná se o způsob měření vzdálenosti. Nejběžnější volby jsou euklidovská a kosinová vzdálenost. Vzdálenost se zadává názvem vzdálenosti pod „n“ a názvem, který je pro vzdálenost použit v knihovně *SciPy* pod „p“. První název může být napsaný jakkoliv, slouží jen k uložení do výsledného reportu. Všechny vzdálenosti, které se nacházejí v knihovně *SciPy* (Virtanen 2020) jsou dostupné na https://docs.scipy.org/doc/scipy/reference/spatial.distance.html, nicméně ne všechny jsou vhodné pro typy výpočtů, používaných v tomto programu, a je možné, že nebudou fungovat. KNN také potřebuje nastavit počet sousedů a t-SNE potřebuje nastavit *perplexity*. Počet sousedů je na uživateli, ale nesmí být větší, než počet textů. *Perplexity* se musí nacházet mezi 5 a 50. V této práci bude použita kosínová vzdálenost, *perplexity* 30 a 11 sousedů. Pouze v případě, že je počet textů menší, než 11, bude počet sousedů snížen.

```
SETTING_DISTANCE_METRIC = {"n" : "cosine", "p" : scipy.spa-
tial.distance.cosine}
SETTING_KNEIGHBORS = 11
SETTING_TSNE_PERPLEXITY = 30
```

Důležité je nastavění poměru trénovacích a testovacích dat. Trénovacích dat by mělo být více než testovacích. V této práci bude použito 75 % trénovacích dat ku 25 % testovacích.

```
SETTING_TRAIN_RATIO = 0.75
```

Pro zobrazování nejdůležitějších vlastností u SVM a NB musí uživatel zadat počet vlastností, které chce zobrazit. Pokud je vlastností méně, než je zadáno, například při vynechání *bag-of-words* a použití samotných indexů, budou zobrazeny všechny dostupné vlastnosti.

```
SETTING_TOP_N_FEATURES = 10
```

Uživatel si také může zvolit formát, v jakém chce výsledky uložit. Na výběr je html a json. Také je potřeba zvolit název složky, do jaké se výsledky uloží. Tato složka se vytvoří v místě uložení samotného programu. Pokud v tomto adresáři již složka s tímto názvem existuje, k názvu se přidá číslo.

```
SETTING_SAVE_AS = "html"
SETTING_OUTPUT_PATH = "results"
```

Poslední nastavení je smyčka. Uživatel může nastavit, aby se program spustil opakovaně. To je výhodné, pokud je použita *n-fold* křížová validace a výpočet trvá déle. Uživatel nastaví, jestli chce smyčku spustit, a pokud ano, kolikrát má být opakována. Výsledky jsou uloženy stejně, jako by se byly uloženy bez smyčky, tedy se každá iterace se uloží do vlastní složky. Pouze ve složce poslední iterace se navíc uloží soubor se shrnutím celé smyčky. V něm je uloženo, která z iterací měla nejlepší výsledky metod učící se s učitelem a poté průměr přesnosti testovacího datasetu u každé z metod.

```
SETTING_LOOP = {"loop" : True, "number" : 5}
```

Pokud je nastavení zadáno správně, program se může spustit. Uživatel musí pamatovat na zachování všech uvozovek, závorek a jiných znaků, bez kterých Python příkazy nemůže provést. Pokud se na konzoli objeví hláška *syntax error*, došlo někde k překlepu.

# 3 Praktická část

V praktické části bude program použit na několik datasetů s cílem ilustrovat použití a ukázat aplikovatelnost různých metod na různé klasifikační problémy. Jedná se o určení jazyka, určení autorství u profesionálních autorů a neprofesionálních autorů, rozpoznání sentimentu a rozpoznání spamu. U všech dat bude použita *5-fold* křížová validace výsledků, aby se omezil vliv náhody.

## 3.1 Určení jazyka

První z úloh je automatické rozpoznání jazyka. Na rozdíl od dalších úloh by mělo rozpoznání jazyka fungovat lépe, jelikož rozdílné jazyky mají odlišná slova. *Bag-of-words* by si tedy měla vést dobře. I informace, které se z textů získávají v podobě indexů, by měly být ovlivněny jazykem.

### 3.1.1 Oddělení češtiny a angličtiny

První úloha bude použita částečně na ukázání postupu a ověření fungování programu. Jedná se o binární klasifikaci, konkrétně o rozpoznání angličtiny od češtiny. Na rozdělení bylo použito prvních osm kapitol Asimovova románu Roboti úsvitu v angličtině a češtině. Stejně jako následující dva datasety se jedná o paralelní korpus. Některé části slovníku budou tedy společné i v odlišných jazycích. Minimálně se jedná o jména a slovo robot, které angličtina převzala z češtiny. Nicméně v českých textech budou tato slova skloňována, tudíž by nemuselo dojít k záměně. Dá se předpokládat, že za použití *bag-of-words* se texty oddělí. Kromě tokenizace

není texty možné dále zpracovat, ovšem i přesto by se lexikální složka angličtiny a češtiny měla lišit.

#### 3.1.1.1 Nastavení a průběh programu

Vše, co se v programu děje, se zobrazuje na konzoli, jak je vidět na ukázce níže. Program nejprve načte texty ze zvolené zdrojové cesty. Uloženy jsou jako objekt třídy text, do které se zapíše jejich jméno, třída, prostý text a zdrojová cesta. Později se k těmto informacím přidá také tokenizovaný text, typy, frekvence, ranky a pravděpodobnosti. Tím, že jsou texty zavedené jako objekty se s nimi lépe pracuje a všechny informace o jednom textu jsou na jednom místě.

Po načtení textů je zkontrolováno, zda mají všechny skupiny stejný počet textů. Skupiny s odlišnou velikostí by mohly zkreslit výsledky. Pokud je počet rozdílný, zkrátí se na počet textů v nejmenší skupině. Z ostatních skupin se náhodně vybere potřebný počet textů.

Poté se jednotlivé texty tokenizují, případně zkracují, odstraňují se z nich stop slova, stemují, lemmatizují a/nebo převádí na tagy. V tomto případě jsou texty pouze tokenizované a zkrácené na 50 slov. Jelikož jsou texty v této úloze ve dvou různých jazycích, ostatní úpravy se nemohou použít. Jsou totiž závislé na jazyku. Pro tuto úlohu je tedy nastavení předzpracování textu následující:

```
SETTING_NLP_PIPELINE = [
{"n" : "tokenize", "p" : {"regex_split" : "\W+"}},
{"n" : "reduce", "p" : {"to" : 50, "randomize": True}}]
```

V této úloze jsou zvoleny všechny indexy i *bag-of-words* a jejich nastavení je následující:

```
SETTING_VECTORIZATION = [
{"n" : "bow", "p" : {"binary" : False, "binary_number" :
None}},
{"n" : "index_ttr"},
{"n" : "index_gini_coeficient"},
{"n" : "index_shannon_entropy"},
{"n" : "index_rr"},
{"n" : "index_average_word_lenght"}]
```

Získaná data se následně mohou náhodně zamíchat a rozdělit na testovací a trénovací. Nastavení zamíchání dat je zadáno následovně:

```
SETTING_SHUFFLE = True
```

Na konzoli se poté zobrazí počty skupin a trénovacích a testovacích dat. Lze si tak ověřit, že nikde nedošlo k chybě. Pak se vytvoří se nová složka, do které se budou ukládat výsledky. Data

jsou následně použita v jednotlivých metodách. U této úlohy jsou zvoleny všechny metody, se zde zobrazeným nastavením:

```
SETTING_NORMALIZATION = {"min_max" : True, "z_score" :
False}
SETTING_METHODS = {"svm" : True, "lda" : True, "knn" : True,
"nb" : True, "decision_tree" : True, "mds" : True, "pca" :
True, "tsne" : True}
SETTING_DISTANCE_METRIC = {"n" : "cosine", "p" : scipy.spa-
tial.distance.cosine}
SETTING_KNEIGHBORS = 3
SETTING_TSNE_PERPLEXITY = 30
SETTING_TRAIN_RATIO = 0.75
```

Metody se nejprve natrénují a poté se vypočítá a uloží jejich úspěšnost. Po dokončení výpočtů se vytvoří grafy. Ty se uloží do nově vytvořené složky a zároveň se i zobrazí na konzoli, odkud se případně také mohou uložit či zkopírovat. Výsledky metod učících se s učitelem se na konzoli nezobrazí, pouze se ukládají do souboru v nové složce. Pokud by je uživatel chtěl zobrazit na konzoli, stačí po dokončení programu do konzole zadat *report["supervised"]*.

Ukázka kontrolního výstupu z konzole programu:

```
>Loading files...
>Testing if all classes have the same ammount of files...
>Applying nlp...
>c1.txt
>Tokenizing...
>Reducing...
>Randomizing...
>c2.txt
>Tokenizing...
...
>Vectorizing...
>Shuffling...
>Splitting...
>Training dataset:  {'english': 6, 'czech': 6}
>Testing dataset:  {'english': 2, 'czech': 2}
>{'Number of classes': 2, 'number of train data': 12, 'num-
ber of test data': 4}
>Creating a folder...
>Training...
>Normalizing...
>Training supervised methods...
>svm...
>evaluating...
...
>Training unsupervised methods...
>mds...
...
>Creating graphs...
```

#### 3.1.1.2 Výsledky

| | | Celé texty | 50 slov |
|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % |
| | Testovací | 100 % | 100 % |
| LDA | Trénovací | 83,3 % | 55 % |
| | Testovací | 100 % | 70 % |
| KNN | Trénovací | 100 % | 100 % |
| | Testovací | 100 % | 100 % |
| NB | Trénovací | 100 % | 100 % |
| | Testovací | 100 % | 100 % |
| Decision Tree | Trénovací | 100 % | 100 % |
| | Testovací | 90 % | 100 % |

**Tabulka 2**: Průměrná přesnost rozpoznání angličtiny a češtiny.

Přesnost při rozpoznání angličtiny a češtiny je vysoké. SVM, KNN a NB dosáhly v obou případech 100% úspěšnosti jak u trénovacího, tak i testovacího datasetu. *Decision Tree* dosáhlo 100 % úspěšnosti pouze při zkrácení textů na 50 slov, LDA naopak pouze při použití celých textů. LDA navíc dosáhla lepších výsledků u testovacího datasetu, než u trénovacího.

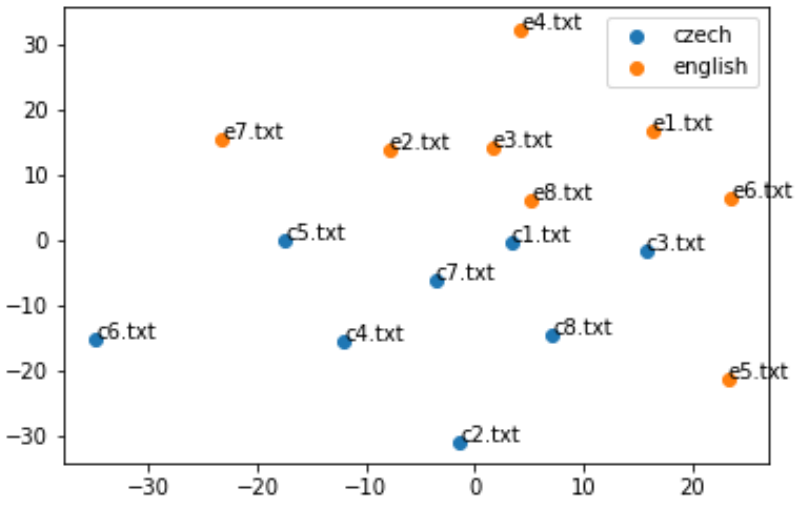
| | | |
|---|---|---|
| accuracy train | 1.0 | |
| accuracy test | 0.5 | |
| recall train | czech | 1.0 |
| | english | 1.0 |
| recall test | czech | 1.0 |
| | english | 0.0 |
| precision train | czech | 1.0 |
| | english | 1.0 |
| precision test | czech | 0.5 |
| | english | 0.0 |
| f1 train | czech | 1.0 |
| | english | 1.0 |
| f1 test | czech | 0.6666666666666666 |
| | english | 0.0 |

**Tabulka 3**: Výsledky *Decision Tree* při vynechání zkrácení textů.

| | | |
|---|---|---|
| accuracy train | 0.5833333333333334 | |
| accuracy test | 0.25 | |
| recall train | czech | 0.6666666666666666 |
| | english | 0.5 |
| recall test | czech | 0.0 |
| | english | 0.5 |
| precision train | czech | 0.5714285714285714 |
| | english | 0.6 |
| precision test | czech | 0.0 |
| | english | 0.3333333333333333 |
| f1 train | czech | 0.6153846153846153 |
| | english | 0.5454545454545454 |
| f1 test | czech | 0.0 |
| | english | 0.4 |

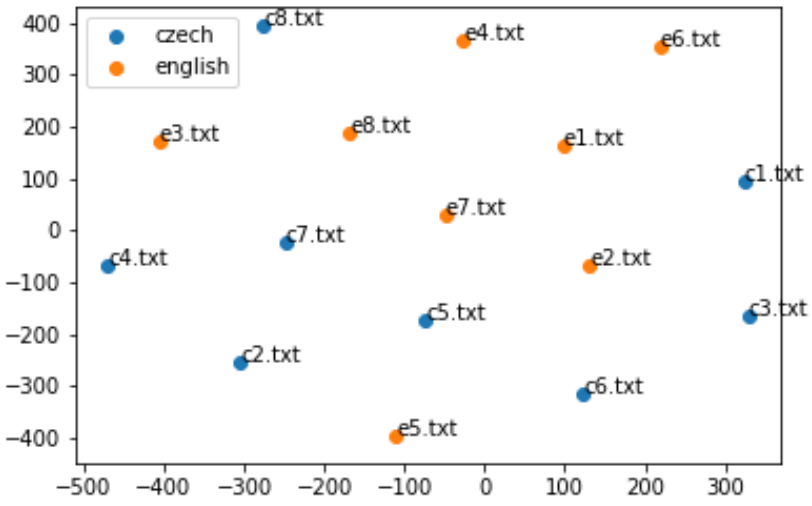
**Tabulka 4**: Výsledky LDA při zkrácení na 50 slov.

Na výsledcích *Decision Tree* a LDA je vidět, jaký vliv na výsledky může mít náhoda. Zda bude text součástí trénovacího či testovacího datasetu se vybírá náhodně. V případě zkrácení textu se i náhodně vybírají slova, která budou ponechána. Kvůli tomu se data, a výsledky z nich získané, mohou lišit. Ve čtyřech případech z pěti byla přesnost testovacího datasetu *Decision Tree* 100 %, nicméně v jednom případě byla přesnost pouhých 50 %. Na tabulce 3 můžeme vidět, že *Decision Tree* označilo všechny texty jako české a žádné jako anglické. Průměr těchto pěti přesností dává 90 %. Podobný problém nastal u LDA při zkrácení textů na 50 slov. Průměrná přesnost testovacího datasetu LDA byla 70 %, nicméně jednotlivé výsledky se pohybovaly od 25 % do 100 %. Je vhodné tedy výpočty zopakovat, aby se vliv náhody omezil. Nicméně je zároveň třeba nedbat pouze na průměr, jelikož se jednotlivé výsledky mohou značně lišit.



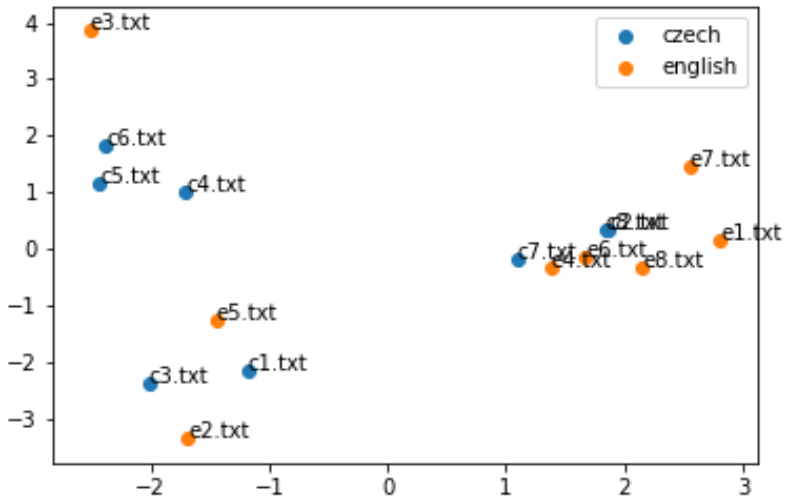

**Graf 4**: Výsledky MDS při vynechání zkrácení textů.

Při redukci počtu vlastností na dva rozměry pomocí MDS si můžeme všimnout, že obě skupiny textů jsou i zde oddělitelné. Třídy od sebe sice nejdou beze ztrát lineárně oddělit, nicméně i tak z tohoto grafu můžeme usoudit, že už samotné vlastnosti mají o obou třídách vhodné pojetí.




**Graf 5**: Výsledky t-SNE při vynechání zkrácení textů.

Ve výsledcích t-SNE se třídy nedají vizuálně oddělit tak snadno, jako u MDS. Stále se ovšem oddělit dají. Zajímavé je, že texty, které se nacházejí v pomyslné špatné skupině, jsou blízko stejné kapitole v druhém jazyce.




**Graf 6**: Výsledky PCA při zkrácení textů na 50 slov.

Poslední vizualizační metodou je PCA. Na rozdíl od MDS a t-SNE se u PCA od sebe skupiny nedají jednoduše oddělit. Nicméně jsou místa, kdy se texty drží blízko u sebe, dva české texty se i překrývají. Výsledky vizualizačních metod u této úlohy nejsou tak jednoznačné a přesné jako výsledky metod učících se s učitelem, nicméně i tak je na grafech 4-6 vidět, že jsou skupiny alespoň částečně oddělitelné.

### 3.1.2 Rozpoznání jazyka u Biblí

Na složitější určení jazyka byl použit dataset 150 Biblí. Z něj byly vybrány pouze ty, které obsahovali pouze nový zákon, jelikož jich bylo nejvíce. Z těchto 109 textů byly vytvořeno dva datasety. Z textů byl odstraněn anglický úvod a texty byly rozděleny na jednotlivé knihy. V některých případech byly kvůli délce kratší knihy spojeny. Každý z jazyků měl 17 textů, z toho 13 v trénovacím datasetu a 4 v testovacím.

První z datasetů se skládá z pěti náhodně vybraných jazyků. Jedná se o mexický jazyk Copala Triqui, o Chiquitano z Bolívie, Mborena Kam neboli Borei z Papua Nové Guinee, etiopský Gofa a Yucana z Kolumbie. Druhý dataset se skládá ze čtyř příbuzných jazyků patřících do jazykové rodiny Quechua, konkrétně Cajamarca, severní Junín, severní Conchucos Ancash a severní Pastaza.

Jelikož se jedná o paralelní korpus, je možné, že u příbuzných jazyků nastane problém. Pokud mají jazyky společnou část slovníku, která se v Bibli vyskytuje (například jména), mohlo by dojít při klasifikaci k chybě. Další problém by mohly představovat vysvětlivky a časté číslice, kterými jsou označeny jednotlivé části Bible. Pokud se ovšem jazyky oddělí, dá se poznat, že se oddělily skutečně dle jazyka, a ne díky jiným možným faktorům (např. kdyby texty jednotlivých jazyků měly žánrové rozdíly).

#### 3.1.2.1 Výsledky

##### 3.1.2.1.1 Náhodně vybrané jazyky

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 90,1 % |
| | Testovací | 100 % | 100 % | 84 % |
| LDA | Trénovací | 54,1 % | 53,2 % | 94 % |
| | Testovací | 52 % | 60 % | 92 % |
| KNN | Trénovací | 100 % | 100 % | 88,9 % |
| | Testovací | 100 % | 100 % | 80 % |
| NB | Trénovací | 100 % | 100 % | 84,9 % |
| | Testovací | 100 % | 100 % | 78 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 95 % | 97 % | 82 % |

**Tabulka 5**: Průměrná přesnost při zkrácení na 1000 slov.

Už první výsledky se ukázaly jako slibné. Při zkrácení textů na 1000 slov dokázalo SVM, KNN a NB jazyky rozdělit na 100 %. *Decision Tree* mělo průměrnou úspěšnost 95 %, LDA výrazně

nižších 52 %. Při použití pouze *bag-of-words* se výsledky změnily pouze mírně. LDA se zvýšilo o 8 % a *Decision Tree* o 2 %. Při použití samotných indexů se úspěšnost všech metod kromě LDA snížila. LDA se zlepšilo na 92 %.

| | 100 slov | 50 slov | 40 slov | 35 slov | 30 slov | 25 slov | 20 slov | 15 slov | 10 slov |
|---|---|---|---|---|---|---|---|---|---|
| SVM | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % | 98 % | 93 % |
| LDA | 72 % | 66 % | 54 % | 62 % | 60 % | 55 % | 57 % | 54 % | 42 % |
| KNN | 100 % | 100 % | 99 % | 100 % | 99 % | 96 % | 94 % | 92 % | 70 % |
| NB | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % | 100 % | 97 % | 92 % |
| Decision Tree | 87 % | 83 % | 80 % | 77 % | 66 % | 78 % | 64 % | 55 % | 52 % |

**Tabulka 6**: Průměrná přesnost testovacího datasetu při zkrácení na odlišný počet slov.

Dále bylo vyzkoušeno, kolik slov je potřeba, na správné rozpoznání jazyků. SVM a NB dosáhly 100% úspěšnosti už při zkrácení na 20 slov, KNN u zkrácení na 50 slov. Při zkrácení na 30 slov mělo KNN 99% úspěšnost. I při zkrácení na pouhých 10 slov mělo SVM úspěšnost 93 % a NB 92 %.

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 91,1 % |
| | Testovací | 100 % | 100 % | 86 % |
| LDA | Trénovací | 58,4 % | 59,4 % | 94,8 % |
| | Testovací | 58 % | 60 % | 88 % |
| KNN | Trénovací | 100 % | 100 % | 92,9 % |
| | Testovací | 100 % | 100 % | 85 % |
| NB | Trénovací | 100 % | 100 % | 85,9 % |
| | Testovací | 100 % | 100 % | 82 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 95 % | 96 % | 84 % |

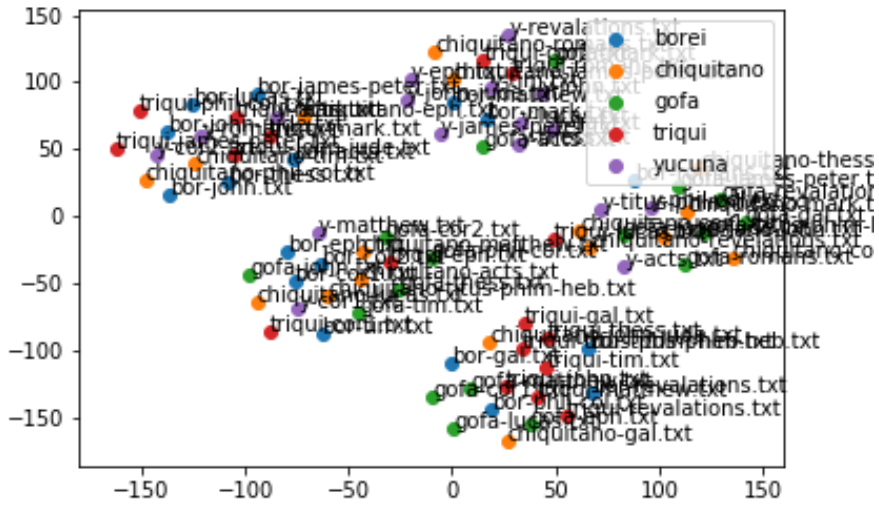
**Tabulka 7**: Průměrná přesnost při zkrácení na 2000 slov.

Výsledky se při zkrácení na 2000 slov oproti zkrácení na 1000 změnily pouze mírně. Při použití pouze indexů se výsledky mírně zlepšily u všech metod kromě LDA, kde se úspěšnost snížila o 4 %. Při použití *bag-of-words* i indexů se úspěšnost LDA zvýšila o 6 %. Při použití pouze BoW se úspěšnost *Decision tree* snížila o 1 %. Ostatní výsledky zůstaly stejné.

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 81,8 % |
| | Testovací | 98 % | 99 % | 80 % |
| LDA | Trénovací | 80,3 % | 78,5 % | 95,4 % |
| | Testovací | 100 % | 100 % | 90 % |
| KNN | Trénovací | 100 % | 100 % | 82,8 % |
| | Testovací | 100 % | 100 % | 80 % |
| NB | Trénovací | 100 % | 100 % | 80,3 % |
| | Testovací | 100 % | 100 % | 75 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 93 % | 98 % | 90 % |

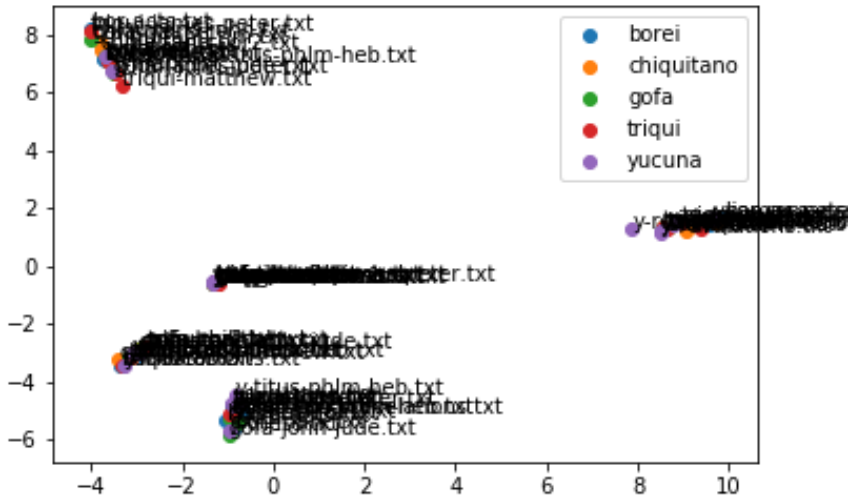
**Tabulka 8**: Průměrná přesnost při vynechání zkrácení textů.

Při vynechání zkrácení textů se přesnost testovacího datasetu LDA zvýšila na 100 % u použití *bag-of-words* a indexů i u použití pouze *bag-of-words*. U trénovacího datasetu mělo LDA výrazně nižší výsledky než u testovacího datasetu. Není to jediný výsledek, u kterého tak nastalo. Nicméně u ostatních výpočtů se jednalo o několik procent, při vynechání zkrácení textů a použití samotné *bag-of-words* byla trénovací přesnost LDA horší o 21,5 %. Stejně jako LDA, KNN a NB také dosáhly 100% úspěšnosti u *bag-of-words* a indexů a samotné *bag-of-words*. Přesnost SVM se snížila na 98 % při použití *bag-of-words* i indexů a 99 % při použití samotné *bag-of-words*. *Decision Tree* mělo při použití *bag-of-words* i indexů přesnost 93 %, ovšem při použití samotné *bag-of-words* 98 %, což je nejvyšší dosažená přesnost *Decision Tree* v této úloze. Při použití samotných indexů byla úspěšnost všech metod nižší. Došlo k tomu pravděpodobně tím, že některé indexy jsou přímo závislé na délce textu, a tudíž potřebují, aby byl počet slov u všech textů stejný.



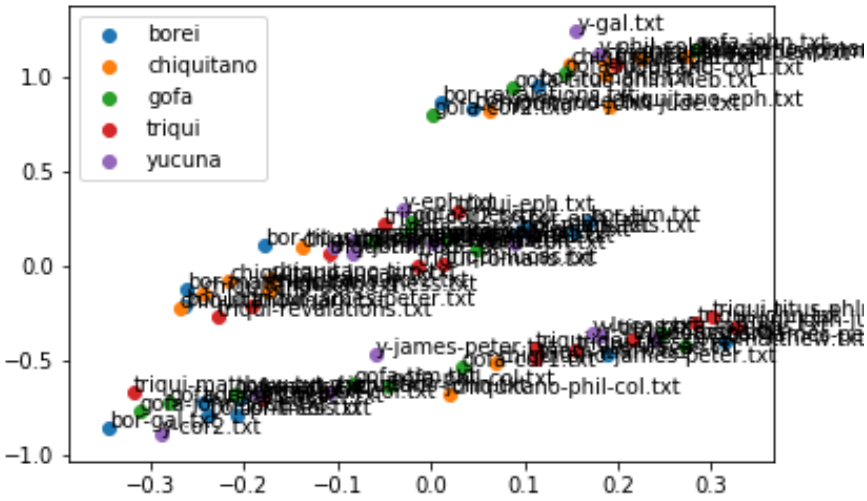

**Graf 7**: Výsledky t-SNE při vynechání zkrácení a použití pouze BoW.

U výsledků t-SNE byly ve většině případů data rozdělena na pět shluků, nicméně se v nich nacházejí texty různých tříd. V některých případech byla jistá tendence některých tříd se shlukovat. Na grafu 7 jsou všechny shluky rozděleny na 17 textů. V pravém horním shluku se nachází 9 textů v jazyku yucuna, od každého ostatního jazyku dva. Ostatní shluky jsou méně vyhraněné, v některých je rozdíl dvou nejvíce zastoupených jazyků pouze jeden text.




**Graf 8:** Výsledky PCA při zkrácení textů na 2000 slov.

PCA, podobně jako t-SNE, také rozčlenilo texty na pět shluků. Na rozdíl od t-SNE byly ovšem texty příliš blízko u sebe, a je obtížné zjistit, jestli jsou texty stejné třídy u sebe. Je jisté, že tomu tak není ve všech případech, jelikož je ve všech skupinách vidět alespoň jeden text v jazyku yucuna a v jazyku copala triqui. Nicméně přesnější informace nemohou být z grafu 8 zjištěny.




**Graf 9**: Výsledky MDS při použitích samotných indexů a zkrácení na 1000 slov.

MDS se texty rozdělilo do pěti shluků jen v několika případech. Jeden z nich je vidět na grafu 9. Stejně jako u t-SNE a PCA texty nejsou zcela rozděleny dle jazyků. V shluku v horním rohu se nachází 7 textů v jazyce chiquitano, ve shluku vpravo dole se nachází 7 textů v jazyce copala triqui. Shluk vlevo uprostřed obsahuje 5 textů v jazyce chiquitano, nicméně celkově obsahuje pouze 8 textů. Shluky vlevo dole a uprostřed nejsou čitelné kvůli překrývajícím se nápisům.

#### 3.1.2.1.2 Příbuzné jazyky

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 58,8 % |
| | Testovací | 100 % | 100 % | 56,3 % |
| LDA | Trénovací | 48,9 % | 46,9 % | 80,8 % |
| | Testovací | 43,8 % | 32,5 % | 63,8 % |
| KNN | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 100 % | 100 % | 57,5 % |
| NB | Trénovací | 100 % | 100 % | 55,8 % |
| | Testovací | 100 % | 100 % | 52,5 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 93,8 % | 92,5 % | 55 % |

**Tabulka 9**: Průměrná přesnost při zkrácení textů na 1000 slov.

Při rozpoznávání příbuzných jazyků měly při zkrácení na 1000 slov SVM, KNN a NB stále 100% úspěšnost. *Decision Tree* se mírně zhoršilo, o 1,2 % u použití *bag-of-words* i indexů a o 4,5 % u samotné *bag-of-words*. Úspěšnost LDA se snížila o 8,2 % u *bag-of-words* i indexů a o 27,5 % u samotné *bag-of-words*. Nejvíce se zhoršila úspěšnost při použití samotných indexů, a to o 22,5-28,2 %.

| | Náhodné jazyky | Příbuzné jazyky |
|---|---|---|
| SVM | 84 % | 56,3 % |
| LDA | 92 % | 63,8 % |
| KNN | 80 % | 57,5 % |
| NB | 78 % | 52,5 % |
| Decision Tree | 82 % | 55 % |

**Tabulka 10**: Průměrná přesnost testovacího datasetu u indexů a zkrácení na 1000 slov.

| | 100 slov | 75 slov | 60 slov | 50 slov | 20 slov |
|---|---|---|---|---|---|
| SVM | 100 % | 100 % | 100 % | 98,8 % | 86,3 % |
| LDA | 52,5 % | 47,5 % | 47,5 % | 45 % | 42,5 % |
| KNN | 100 % | 100 % | 100 % | 97,5 % | 81,3 % |
| NB | 100 % | 100 % | 100 % | 100 % | 96,3 % |
| Decision Tree | 73,8 % | 81,3 % | 67,5 % | 63,8 % | 48,8 % |

**Tabulka 11**: Průměrná přesnost testovacího datasetu při zkracování textů na odlišný počet slov.

U rozpoznání jazyků, které jsou bližší, bylo potřeba na získání 100% úspěšnost více slov než při rozpoznání náhodně vybraných jazyků. Přesto lze i u příbuzných jazyků dosáhnout 100% přesnosti s relativně malým počtem slov. NB dosáhlo 100% úspěšnosti při zkrácení na 50 slov, KNN a SVM při zkrácení na 60 slov.

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 63,8 % |
| | Testovací | 100 % | 100 % | 46,3 % |
| LDA | Trénovací | 55 % | 52,3 % | 86,5 % |
| | Testovací | 31,3 % | 18,8 % | 63,8 % |
| KNN | Trénovací | 100 % | 100 % | 63,8 % |
| | Testovací | 100 % | 100 % | 47,5 % |
| NB | Trénovací | 100 % | 100 % | 63,1 % |
| | Testovací | 100 % | 100 % | 46,3 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 97,5 % | 95 % | 52,5 % |

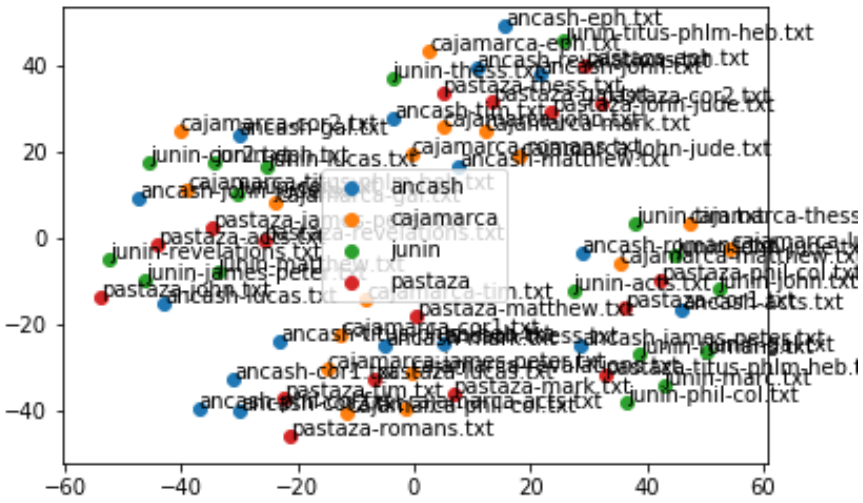
**Tabulka 12**: Průměrná přesnost při zkrácení na 2000 slov.

Při zvýšení počtu slov na 2000 se navzdory očekávání LDA nezlepšilo, naopak se zhoršilo. Při použití *bag-of-words* i indexů se úspěšnost LDA zhoršila o 12,5 % a u samotné *bag-of-words* o 13,7 % na 18,8 %, což je nižší úspěšnost než při náhodném výběru skupiny. Při náhodném hádání by při čtyřech skupinách mělo dojít k přesnosti 25 %. Pouze při použití samotných indexů zůstala úspěšnost LDA stejná, zatímco se všechny ostatní metody zhoršily. SVM, KNN a NB měly u *bag-of-words*, bez indexů i s nimi, stále úspěšnost 100 %. *Decision Tree* je jediná metoda, kterou zvýšení počtu slov na 2000 zlepšilo. Při použití *bag-of-words* i indexů se zlepšila o 3,7 % na 97,5 %. Jedná se o nejvyšší přesnost *Decision Tree* při rozpoznávání příbuzných jazyků.

| | | BoW a indexy | BoW | Indexy |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 43,9 % |
| | Testovací | 100 % | 100 % | 40 % |
| LDA | Trénovací | 76,9 % | 75,8 % | 85,4 % |
| | Testovací | 100 % | 100 % | 60 % |
| KNN | Trénovací | 100 % | 100 % | 48,9 % |
| | Testovací | 100 % | 100 % | 31,3 % |
| NB | Trénovací | 100 % | 100 % | 37,3 % |
| | Testovací | 100 % | 100 % | 40 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 93,8 % | 96,3 % | 41,3 % |

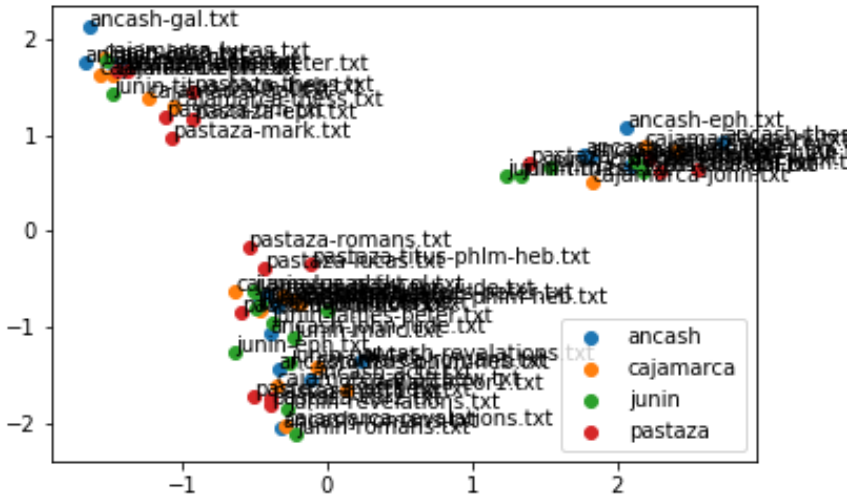
**Tabulka 13**: Průměrná přesnost při vynechání zkrácení textů.

Stejně jako u rozpoznání náhodně vybraných jazyků, vynechání zkrácení textů pomohlo *bag-of-words* a zhoršilo úspěšnost indexů. LDA se dostalo na 100% přesnost, stejně jako SVM, KNN a NB. LDA mělo při použití *bag-of-words* opět výrazně horší přesnost u trénovacího datasetu než u testovacího.



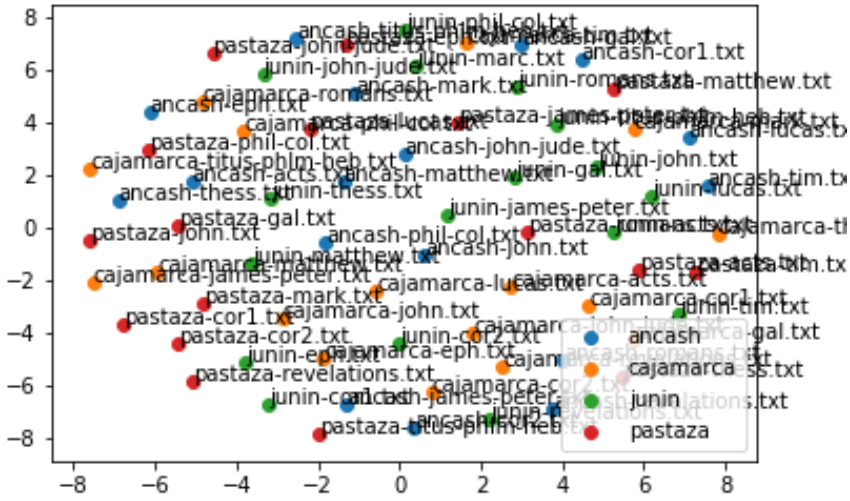

**Graf 10**: Výsledky t-SNE při zkrácení textů na 2000 slov.

Podobně jako u rozpoznání náhodně vybraných jazyků, i u rozpoznání příbuzných jazyků v některých případech jsou u výsledků t-SNE data rozdělena na čtyři skupiny. Na grafu 10 je vidět, že v levé a pravé skupině převládá severní junín, v pravé skupině se nachází osm textů a v levé sedm. Zbylé dva texty jsou v horní skupině. Texty ostatní jazyků se k sobě tolik nepřiblížily. V dolní skupině je pět textů v jazyku pastaza a šest v jazycích ancash a cajamarca. Všechny tři jazyky pak mají pět textů v horní skupině, tři v pravé skupině a zbylé texty v levé skupině.




**Graf 11**: Výsledky PCA při zkrácení textů na 75 slov.

V některých případech jsou data u výsledků PCA viditelně oddělitelná. Většinou se však rozdělila pouze na tři skupiny. Na grafu 11 je ovšem vidět, že spodní skupina by se v půlce dala rozdělit. I tak nejsou texty do skupin rozděleny podle jazyků. Některá data se překrývají, ovšem přesto je nejvíc textů stejného jazyka v jedné skupině 6, a to v jazyce severní junín ve skupině uprostřed.




**Graf 12**: Výsledek MDS při zkrácení textů na 60 slov.

Výsledky MDS u této úlohy nejsou rozdělitelné. Data mají mezi sebou mezery a jsou v jednou oválu společně. Přesto je místy vidět jistá tendence stejných jazyků být blízko sobě, například

skupina sedmi textů v jazyce severní junín vpravo nahoře, nebo na textech v jazyce pastaza, které se z většiny drží vlevo na okraji.

### 3.1.3 Závěr

| | Angličtina a čeština | Náhodné jazyky | Příbuzné jazyky |
|---|---|---|---|
| SVM | 100 % | 100 % | 100 % |
| LDA | 100 % | 100 % | 100 % |
| KNN | 100 % | 100 % | 100 % |
| NB | 100 % | 100 % | 100 % |
| Decision Tree | 100 % | 98 % | 97,5 % |

**Tabulka 14**: Nejlepší průměrná přesnost testovacího datasetu všech metod

Metody učící se s učitelem jsou schopné jazyky rozdělit, a to i s malým počtem slov. Pokud jsou si jazyky podobnější, je slov potřeba více. U všech tří datasetů však na 100% úspěšnost u SVM, KNN a NB stačilo zkrácení textů na 60 slov. LDA ke 100% úspěšnosti potřebuje ponechat texty v původní délce. Jediná metoda, která u tohoto problému testovaném na textů Biblí nedosáhla 100% úspěšnosti je *Decision Tree*. Nejvyšší přesnost *Decision Tree* byla 98 % u náhodného výběru jazyků a 97,5 % u příbuzných jazyků.

Nejlepší metodou na rozpoznání jazyků je NB, které stačilo zkrácení na 20 slov k rozpoznání náhodně vybraných jazyků a 50 slov u příbuzných jazyků. Nicméně s dostatečnou délkou textů jsou všechny metody schopny dosáhnou vysoké úspěšnosti.

## 3.2 Určení autorství u profesionálních autorů

Druhá úloha spočívá v určení autorství. Určením autorství se zabývá forenzní lingvistika, zejména u anonymních dopisů. (Musilová 2005) V této úloze budeme klasifikovat profesionální autory. Pro určení autorství bylo použito 24 děl od Jaroslava Foglara, 24 děl od Isaaca Asimovova a 24 děl od Stanisława Lema. Ve všech případech se jedná o profesionální autory beletrie. Kromě autorského stylu by se na odlišení mohl podepsat i rozdílný žánr, jelikož Asimov a Lem psali science fiction a Foglar dobrodružné příběhy pro chlapce. Také by se mohlo projevit to, že Asimovova díla jsou přeložená z angličtiny a Lemova z polštiny, zatímco Foglarova díla jsou v originálním znění. Tím by se mohly projevit rozlišné styly případných různých překladatelů. Na druhou stranu se jedná o beletrii ve stejném jazyce, a je možné, že odlišné žánry a autorské styly na určení nebudou stačit.

Nejprve se budou oddělovat Foglarovy texty od Asimovových, poté Asimovovy od Lemových a nakonec všech autorů nejednou. První úkol by měl dopadnout lépe, z důvodu rozdílných žánrů. U druhého úkolu již bude potřeba rozpoznat autory samé, a ne pouze žánrové rozdíly.

### 3.2.1 Výsledky

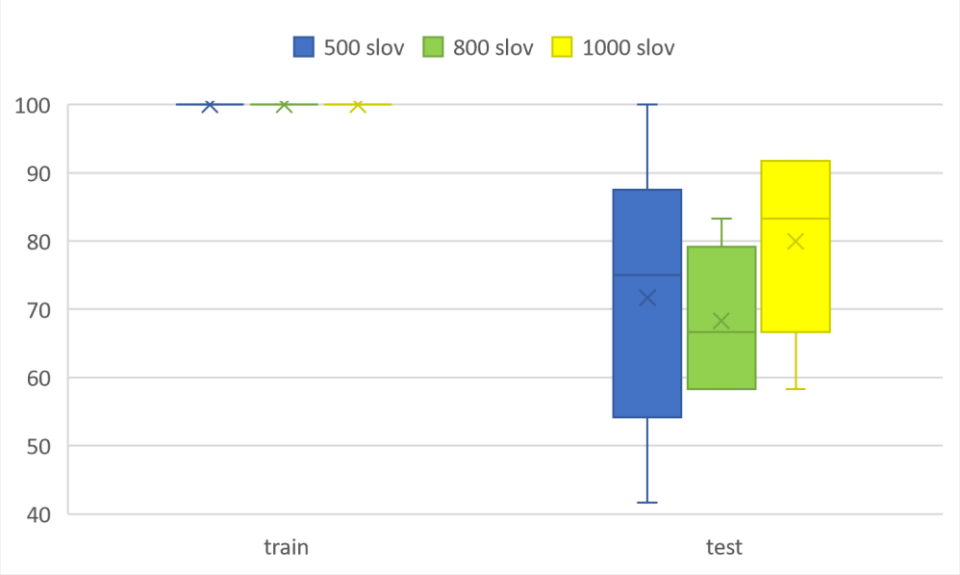
#### 3.2.1.1 Foglar – Asimov

##### 3.2.1.1.1 Metody učící se s učitelem

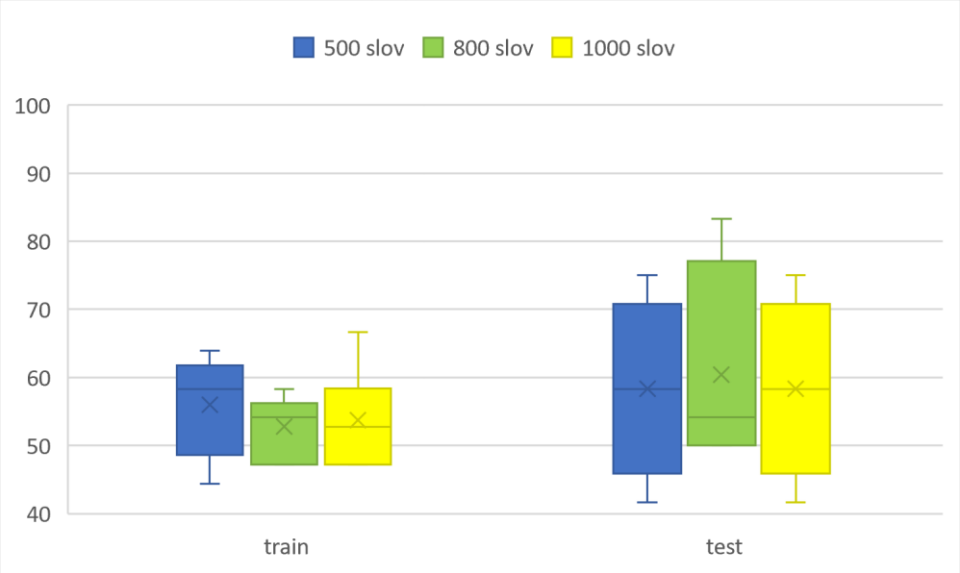
| | | 100 slov | 250 slov | 500 slov | 800 slov | 1000 slov |
|---|---|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 80 % | 96,7 % | 100 % | 100 % | 100 % |
| LDA | Trénovací | 56,7 % | 53,9 % | 55,5 % | 51,7 % | 51,1 % |
| | Testovací | 60 % | 61,7 % | 58,3 % | 60 % | 60 % |
| KNN | Trénovací | 80,6 % | 92,2 % | 100 % | 100 % | 100 % |
| | Testovací | 80 % | 93,3 % | 100 % | 98,3 % | 100 % |
| NB | Trénovací | 97,8 % | 100 % | 97,5 % | 100 % | 100 % |
| | Testovací | 85 % | 96,7 % | 100 % | 100 % | 100 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 66,7 % | 63,3 % | 71,7 % | 68,3 % | 80 % |

**Tabulka 15**: Průměrná přesnost rozpoznání autorství při zkrácení na různé počty slov.

Při testování autorství byla velmi důležitá délka textu. Počáteční testy byly provedeny se zkrácením textů na 100 slov, což se ukázalo jako nedostatečné. Při zvýšení na 1000 slov se přesnost zlepšila na 100 % u SVM, KNN a NB. Po bližším zkoumání bylo zjištěno, že na rozpoznání tohoto datasetu na 100 % u SVM a NB jsou potřeba texty o alespoň 500 slovech. KNN mělo u 500 slov také 100% úspěšnost, nicméně při zkrácení na 800 slov se přesnost snížila na 98,3 %. Při dalším zvýšení počtu slov se zlepšila přesnost *Decision Tree*, nicméně nejvyšší přesnost byla 80 % při zkrácení na 1000 slov. LDA mělo nejvyšší přesnost testovacího datasetu 61,7 % při zkrácení na 250 slov. Nejnižší přesnost mělo při zkrácení na 500 slov, u ostatního zkrácení byla přesnost vždy stejná. Ve všech případech byla přesnost testovacího datasetu u LDA vyšší, než přesnost trénovacího datasetu.




**Graf 13**: Přesnost *Decision Tree*.




**Graf 14**: Přesnost LDA.

Na grafu 13 a grafu 14 je znázorněna přesnost každého z pěti výpočtů se stejným nastavením. Zkrácení na méně, než 500 slov, byla vynechána, jelikož u nich jsou nízké výsledky. Na grafu 13 je vidět, že přesnost *Decision Tree* se pohybovala na velkém rozmezí, u zkrácení na 500 slov byl rozdíl mezi nejvyšším a nejnižším výsledkem přes 50 %. To všechno navzdory tomu, že *Decision Tree* mělo u trénovacího datasetu 100% úspěšnost. LDA mělo na druhou stranu lepší výsledky u testovacího datasetu. Na grafu 14 je vidět, že se přesnost obou datasetů překrývá, ale testovací dataset má větší rozptyl.

| | | Lemmatizace a odstranění stop slov | Odstranění stop slov | Lemmatizace |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 100 % | 100 % | 98,3 % |
| LDA | Trénovací | 73,9 % | 70 % | 76,7 % |
| | Testovací | 76,7 % | 61,6 % | 60 % |
| KNN | Trénovací | 100 % | 100 % | 99,4 % |
| | Testovací | 100 % | 100 % | 98, 3 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 100 % | 100 % | 100 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 91,7 % | 70 % | 86,6 % |

**Tabulka 16**: Průměrná přesnost při použití lemmatizace a odstranění stopových slov a zkrácení na 1000 slov.

Použití lemmatizace a odstranění stop slov přesnost *Decision Tree* a LDA mírně zlepšilo, nejvíce pokud bylo použito obojí. *Decision Tree* se při lemmatizaci a odstranění stop slov dostalo až na průměrnou přesnost 91,7 %, LDA na 76,7 %.

| | | Frekvenční BoW | Binární BoW | Binární BoW s citlivostí 2 |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 100 % | 100 % | 100 % |
| LDA | Trénovací | 50 % | 51,1 % | 60 % |
| | Testovací | 68,3 % | 51,7 % | 71,7 % |
| KNN | Trénovací | 100 % | 100 % | 96,6 % |
| | Testovací | 100 % | 100 % | 98, 3 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 100 % | 100 % | 100 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 65 % | 80 % | 71,1 % |

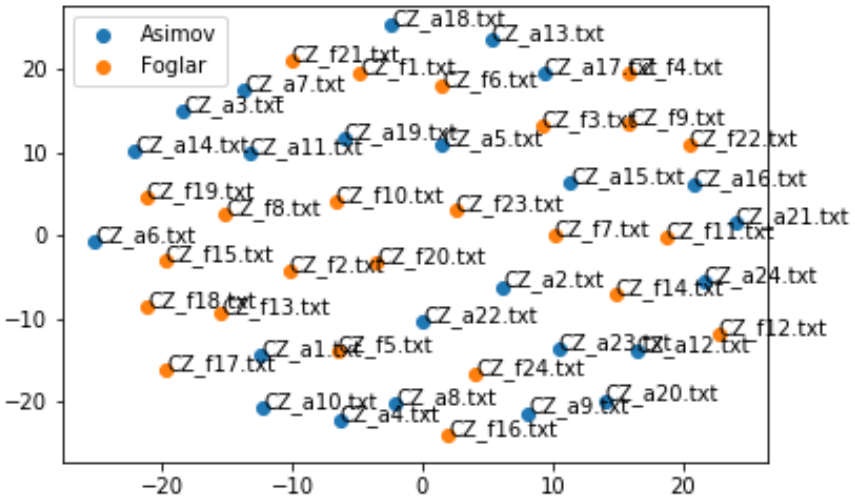
**Tabulka 17**: Průměrná přesnost při použití pouze BoW a zkrácení na 1000 slov.

| | | 1000 slov | 5000 slov | 10000 slov |
|---|---|---|---|---|
| SVM | Trénovací | 80,5 % | 90,5 % | 94,4 % |
| | Testovací | 90 % | 86,7 % | 90 % |
| LDA | Trénovací | 87,2 % | 93,3 % | 96,7 % |
| | Testovací | 76,7 % | 86,7 % | 85 % |
| KNN | Trénovací | 75 % | 85 % | 85,6 % |
| | Testovací | 76,7 % | 73,3 % | 83,3 % |
| NB | Trénovací | 68,9 % | 76,1 % | 83,3 % |
| | Testovací | 76,7 % | 73,3 % | 76,7 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 73,3 % | 81,7 % | 86,7 % |

**Tabulka 18**: Průměrná přesnost při použití pouze indexů.

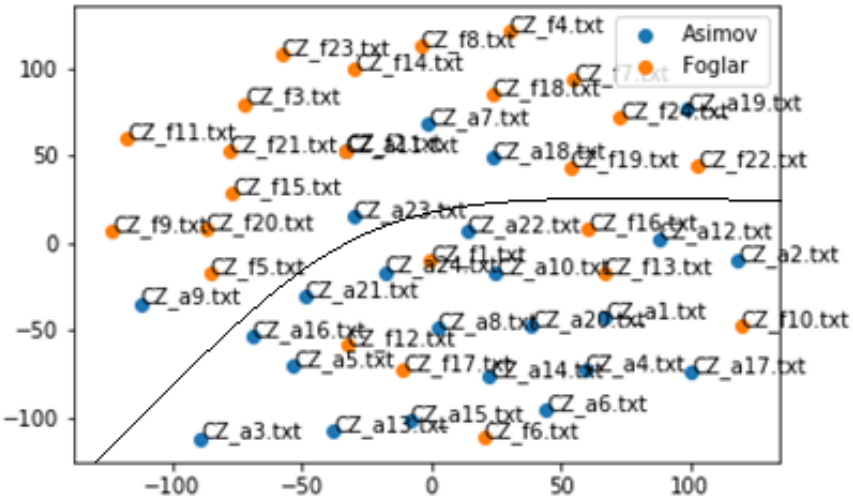
Jako velmi důležitá se ukázala metoda reprezentace textů *bag-of-words*. Při jejím vynechání a použití pouze indexů se zkrácením na 1000 slov u SVM průměrná přesnost snížila na 90 % u testovacího datasetu. Kromě toho, stejně jako KNN a NB, měla vyšší přesnost u testovacího datasetu než trénovacího. U SVM nejvíce ke klasifikaci pomohl index TTR. *Decision Tree* mělo průměrnou přesnost 73,3 %, ostatní metody 76,7 %. Při zvětšení počtu slov se průměrná úspěšnost zlepšila, nicméně nejvyšší byla stále 90 % u SVM. Přitom při použití pouze frekvenční *bag-of-words* byla přesnost u SVM, KNN a NB stále 100 %. Jediná metoda, která se zlepšila bez *Bag-of-words* je LDA. U zkrácení na 5000 slov měla průměrnou přesnost 86,7 %.

3.2.1.1.2 Vizualizační metody




**Graf 15**: Nejčastější podoba výsledků MDS.

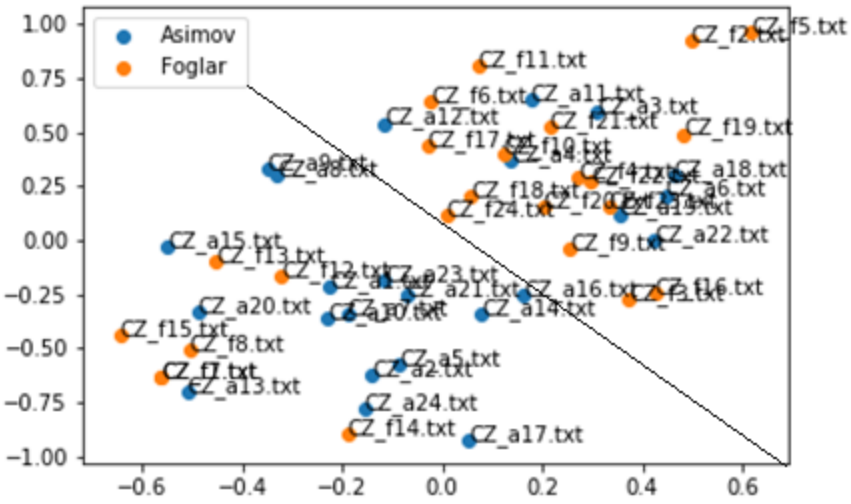
Výsledky MDS byly nejčastěji zobrazovány jako ovál, s mezerami mezi jednotlivými body. Na některých místech se k sobě shlukují texty podle skupin, například na grafu 15 je vlevo uprostřed vidět shluk několika textů od Foglara. Jednotlivé texty v rámci shluku mezi sebou ovšem stále mají mezery a jednotlivé shluky nejsou nijak odděleny od shluků druhé skupiny.




**Graf 16**: Výsledky MDS při vynechání zkrácení textu s binární BoW.

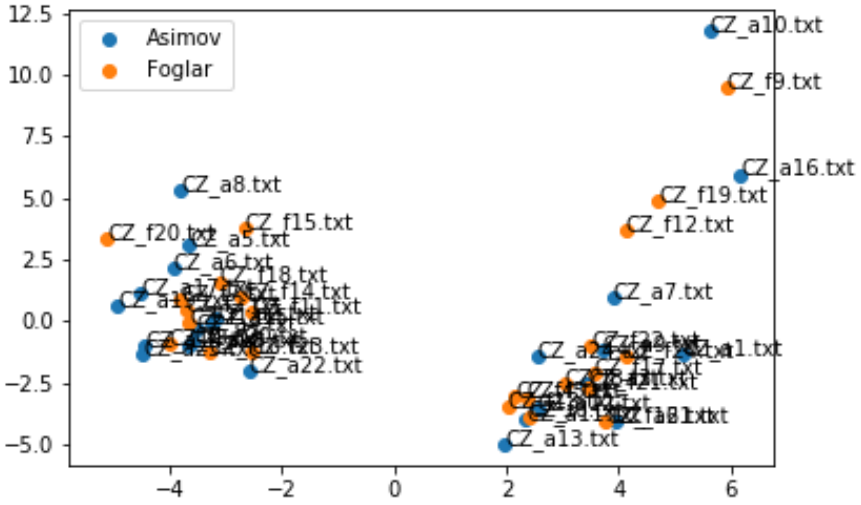
Asi nejlépe interpretovatelný graf MDS nastal při vynechání zkrácení textu a použití binární *bag-of-words*. Na první pohled je vidět, že se Foglarovy texty drží nahoře a Asimovovy dole. Navíc je mezi skupinami mezera, která byla na grafu 17 zvýrazněna. V horní skupině se nachází 17 textů od Foglara a 6 od Asimovova a ve spodní skupině se nachází 18 textů od Asimovova

a 7 od Foglara. Procentuálně se tedy v horní skupině nachází v 73,9 % texty od Foglara a v dolní skupině se v 72 % nachází texty od Asimovova. Pokud to vezmeme z druhé strany, tak se v horní skupině nachází 70,8 % ze všech použitých Foglarových textů a v dolní skupině se nachází 75 % ze všech použitých Asimovových textů.



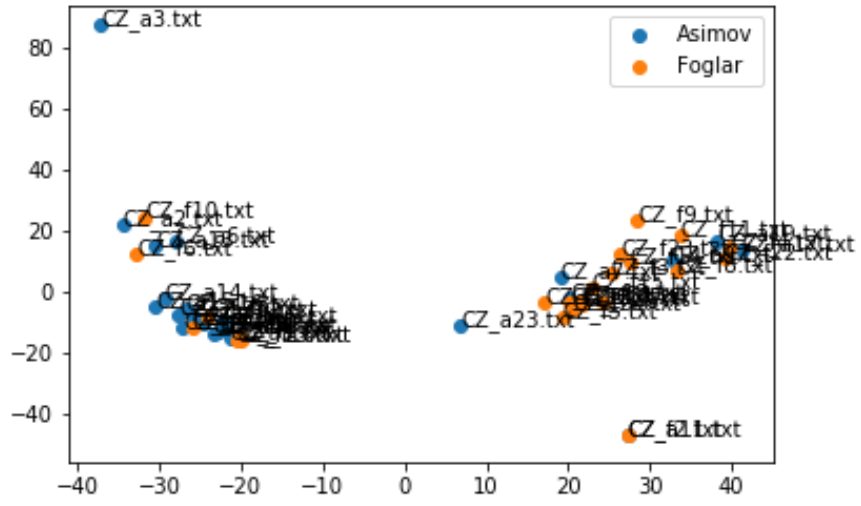

**Graf 17**: Výsledky MDS při zkrácení na 10000 slov bez použití BoW.

U grafu 17 se dvě vzniklé skupiny dokonce dají oddělit lineárně. Skupina vlevo dole ovšem není ucelená, a dala by se dále rozdělit. Nachází se v ní 16 textů od Asimovova a 7 od Foglara. Ve skupině vpravo nahoře se nachází 8 textů od Asimovova a 17 od Foglara. Ve skupině vlevo dole se tedy nachází v 69,9 % texty od Asimovova a jedná se o 66,7 % ze všech Asimovových textů. 68 % z druhé skupiny tvoří texty od Foglara, 70,8 % ze všech jeho textů.




**Graf 18**: Výsledky PCA při zkrácení textů na 1000 slov s použitím pouze BoW bez indexů.

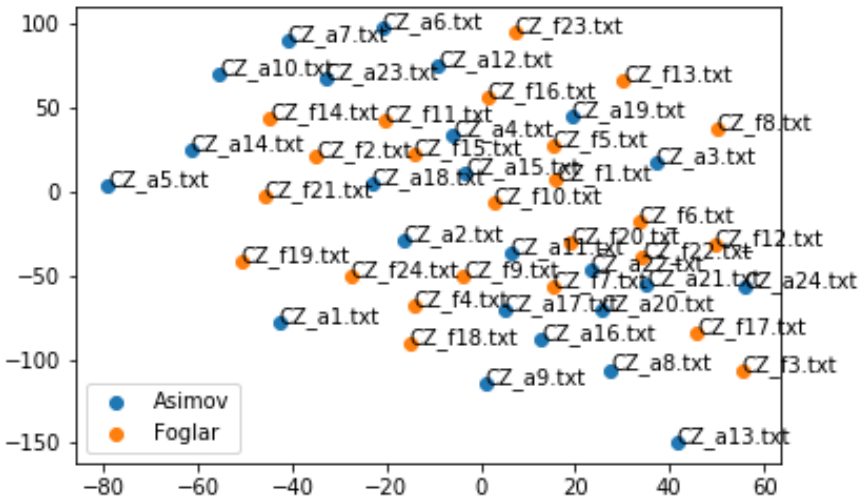
Výsledky PCA se v některých případech dají rozdělit na dvě skupiny. Nicméně skupiny nejsou rozdělitelné dle tříd. Z důvodu mnoho textů na jednom místě není možné přesně spočítat, kolik se textů v uskupeních nachází. Na levé straně grafu 18 je jedenáct textů od Foglara, na pravé straně je jich třináct. Asimovovy texty nejsou tolik viditelné. Na obou stranách je jich minimálně jedenáct, nicméně z důvodu navzájem se překrývajících dat není jistá poloha zbylých dvou textů. Každopádně se ale texty nerozdělily dle autorství.



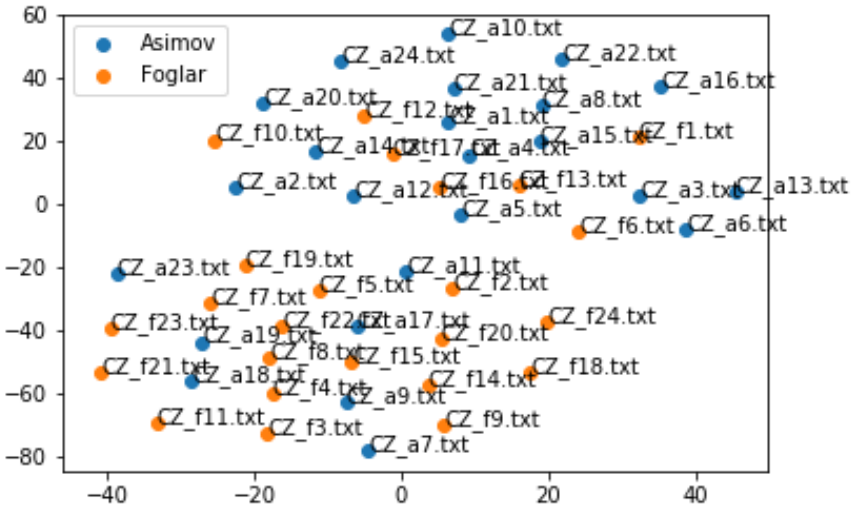

**Graf 19**: Výsledky PCA při vynechání zkrácení textů s binární BoW.

Na grafu 19 jsou skupiny také viditelně oddělitelné. Nejsou rozděleny perfektně na Asimovova a Foglara, nicméně mají lepší výsledek než graf 18. Kvůli tomu, že se data překrývají, je těžké

spočítat všechny body, nicméně to vypadá, že na levé straně se nachází alespoň 14 textů od Asimovova a 6 od Foglara a na pravé 7 od Asimovova a 16 od Foglara. Počítáme-li tedy jen texty, které lze vidět, na levé straně je ze všech textů 70 % od Asimovova a na pravé straně je 69,6 % od Foglara. Celkem se 58,3 % textů od Asimovova nachází na levé straně, 29,2 % se nachází na pravé straně a 12,5 % není vidět. Z Foglarových textů se 25 % nachází na levé straně, 66,7 % se nachází na pravé straně a 8,3 % není vidět.




**Graf 20**: Výsledky t-SNE při binární BoW.




**Graf 21**: Výsledky t-SNE při vynechání zkrácení textů.

I u výsledků t-SNE se někdy texty dají rozdělit na dvě části, ale ne vždy podle autorů. Na grafu 20 jsou vidět dvě oddělené skupiny. V obou skupinách se nachází dvanáct textů od Asimovova

a dvanáct textů od Foglara. Na grafu 21 už jsou výsledky lepší. V horní skupině se nachází 17 textů od Asimovova a 7 textů od Foglara a ve spodní skupině je tomu naopak.

#### 3.2.1.2 Asimov – Lem

##### 3.2.1.2.1 Metody učící se s učitelem

| | | Frekvenční BoW | Binární BoW | Frekvenční BoW, lemma a stop slova |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 88,3 % | 91,7 % | 90 % |
| LDA | Trénovací | 56,1 | 60 % | 56,7 % |
| | Testovací | 73,3 % | 58,3 % | 63,3 % |
| KNN | Trénovací | 74,4 | 65 % | 83,9 % |
| | Testovací | 63,3 % | 78,3 % | 78,3 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 65 % | 78,3 % | 83,3 % |
| *Decision Tree* | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 70 % | 66,7 % | 73,3 % |

**Tabulka 19**: Průměrná přesnost při zkracování textu na 1000 slov.

Oddělení textů Asimovova od textů Lema se ukázalo těžší než oddělení Asimovova od Foglara. Při zkrácení textů na 1000 slov, což u Asimovova a Foglara stačilo u SVM, KNN a NB na dosažení 100% přesnosti, mělo SVM při binární *bag-of-words* průměrnou přesnost 91,7 % a KNN a NB 78,3 %. Při použití lemmatizace a odstranění stop slov se zvýšila přesnost NB z 65 % na 83,3 % a *Decision Tree* z 70 % na 73,3 %. SVM a KNN se také zlepšily, ale ne více než při binární *bag-of-words*. LDA se zhoršilo.

| | | Lemmatizace a odstranění stop slov | Pouze tokenizace |
|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % |
| | Testovací | 96,7 % | 93,3 % |
| LDA | Trénovací | 60,7 % | 56,7 % |
| | Testovací | 70 % | 65 % |
| KNN | Trénovací | 89,5 % | 76,7 % |
| | Testovací | 91,7 % | 70 % |
| NB | Trénovací | 100 % | 100 % |
| | Testovací | 90 % | 75 % |
| Decision Tree | Trénovací | 100 % | 100 % |
| | Testovací | 85 % | 73,3 % |

**Tabulka 20**: Průměrná přesnost při zkrácení textů na 2500 slov s frekvenční BoW.

Při zkrácení textů na 2500 slov mělo použití lemmatizace a odstranění stop slov u všech metod vyšší přesnost, než když bylo vynecháno. SVM dosáhlo přesnosti 96,7 %. Kromě LDA byla u všech metod vyšší přesnost u použití lemmatizace a odstranění stop slov a zkrácení na 2500

než u jakéhokoliv výpočtu při zkrácení na 1000 slov. Při zkrácení na 2500 se samotnou tokenizací tomu tak bylo pouze u SVM.

| | | Frekvenční BoW | Frekvenční BoW, lemmatizace a odstranění stop slov | Binární BoW, lemmatizace a odstranění stop slov |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 95 % | 95 % | 93,3 % |
| LDA | Trénovací | 66,8 % | 58,9 % | 60 % |
| | Testovací | 83,3 % | 91,6 % | 75 % |
| KNN | Trénovací | 88,3 % | 97,8 % | 88,3 % |
| | Testovací | 88,3 % | 93,3 % | 83,3 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 93,3 % | 93,3 % | 83,3 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 75 % | 80 % | 75 % |

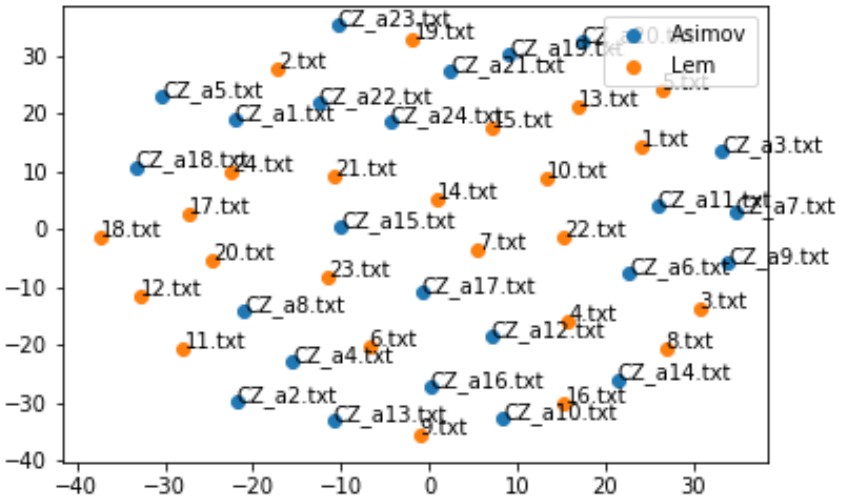
**Tabulka 21**: Průměrná přesnost při zkrácení textů na 5000 slov.

| | | Frekvenční BoW | Frekvenční BoW, lemmatizace a odstranění stop slov | Binární BoW, lemmatizace a odstranění stop slov | Binární BoW s citlivostí 2, lemmatizace a odstranění stop slov | Binární BoW |
|---|---|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 98,3 % | 95 % | 95 % | 100 % | 96,7 % |
| LDA | Trénovací | 62,7 % | 58,9 % | 60 % | 62,8 % | 63,9 % |
| | Testovací | 91,7 % | 91,6 % | 83,3 % | 74,9 % | 78,3 % |
| KNN | Trénovací | 90 % | 97,8 % | 88,3 % | 80,6 % | 80,6 % |
| | Testovací | 96,7 % | 93,3 % | 83,3 % | 88,3 % | 83,3 % |
| NB | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 96,7 % | 93,3 % | 93,3 % | 98,3 % | 90 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 78,3 % | 80 % | 75 % | 80 % | 91,7 % |

**Tabulka 22**: Průměrná přesnost při zkrácení textů na 10000 slov.

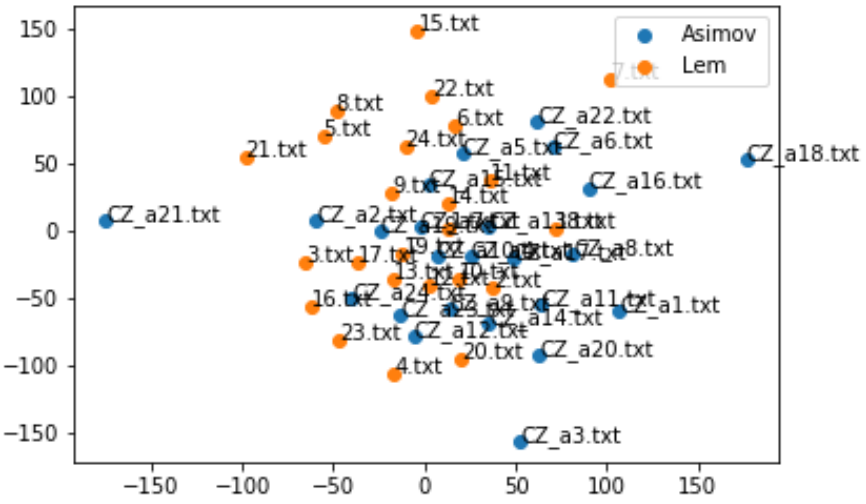
Při dalším zvyšování počtu slov se přesnost zvyšovala. Při zkrácení na 10000 slov, s binární *Bag-of-words* s citlivostí 2, lemmatizací a odstranění stop slov se podařilo získat 100% přesnost u 98,3% a NB. LDA se už při zkrácení na 5000 slov podařilo získat přesnost 91,6 %, ačkoliv mělo do té doby nejvyšší přesnost 73,3 %.

3.2.1.2.2 Vizualizační metody



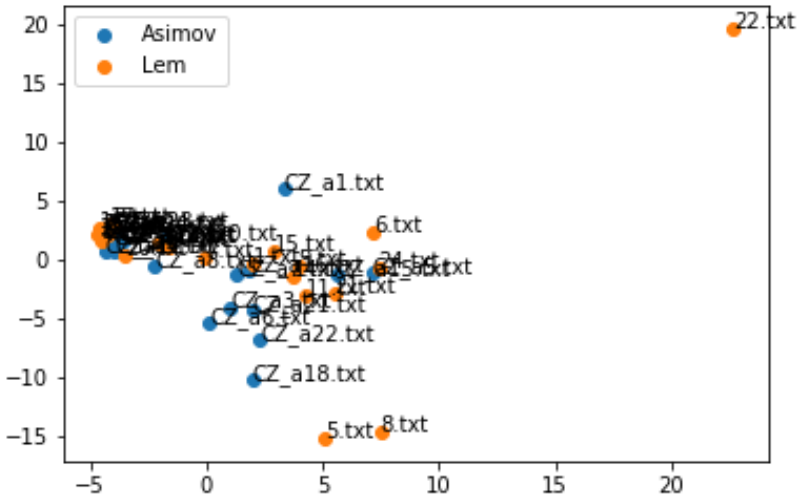

**Graf 22**: Výsledky MDS při zkrácení na 2500 slov.

Výsledky MDS od sebe při rozpoznání Asimovova od Lema třídy nijak neoddělily. Nicméně i tak je na grafu 22 vidět, že na některých místech mají texty stejných autorů tendenci být blízko u sebe.




**Graf 23**: Výsledky t-SNE při zkrácení 2500 slov.

Výsledky t-SNE byly o něco lepší. Skupiny se od sebe sice vizuálně neoddělily, nicméně se více shlukují. Na grafu 23 se Asimovovy texty drží spíše vpravo a Lemovy spíše vlevo.




**Graf 24**: Výsledky PCA při zkrácení na 2500 slov.

Graf PCA je u této úlohy nečitelný, jelikož se jednotlivé texty překrývají.

### 3.2.1.3 Foglar – Asimov – Lem

#### 3.2.1.3.1 Metody učící se s učitelem

| | | Binární BoW s citlivostí 2 | Binární BoW s citlivostí 2, lemmatizace a odstranění stop slov | Binární BoW, lemmatizace a odstranění stop slov | Frekvenční BoW | Frekvenční BoW, lemmatizace a odstranění stop slov |
|---|---|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 86,7 % | 92,2 % | 94,4 % | 93,3 % | 93,3 % |
| LDA | Trénovací | 58,2 % | 58,5 % | 47 % | 47,8 % | 49,3 % |
| | Testovací | 61,1 % | 70 % | 47,8 % | 45,5 % | 53,3 % |
| KNN | Trénovací | 88,9 % | 84,1 % | 81,1 % | 83 % | 89,2 % |
| | Testovací | 81,1 % | 84,4 % | 77,8 % | 80 % | 81,1 % |
| NB | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 78,9 % | 84,4 % | 86,7 % | 85,6 % | 86,7 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 60 % | 72,2 % | 73.3 % | 60 % | 72,2 % |

**Tabulka 23**: Průměrná přesnost se zkrácením na 1000 slov.

Oddělení všech tří autorů se ukázalo jako těžší úkol než oddělování jen dvou. Stejně jako u předchozích úloh byly výsledky lepší při použití lemmatizace s odstraněním stop slov a zvyšování počtu slov. Při zkrácení textů na 1000 slov měly všechny metody nejvyšší přesnost při použití lemmatizace a odstranění stop slov, pouze se lišilo, jaká varianta *bag-of-words* byla použita. LDA a KNN měly nejvyšší přesnost s binární *bag-of-words* o citlivosti 2, SVM

a *Decision Tree* při binární *bag-of-words* s citlivostí 1. NB mělo stejný výsledek u frekvenční *bag-of-words* i binární *bag-of-words* s citlivostí 1.

| | | Lemmatizace a odstranění stop slov | Pouze tokenizace |
|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % |
| | Testovací | 95,6 % | 95,6 % |
| LDA | Trénovací | 52,6 % | 52,2 % |
| | Testovací | 67,8 % | 53,3 % |
| KNN | Trénovací | 91,5 % | 84,1 % |
| | Testovací | 93,3 % | 78,9 % |
| NB | Trénovací | 100 % | 100 % |
| | Testovací | 94,4 % | 90 % |
| Decision Tree | Trénovací | 100 % | 100 % |
| | Testovací | 76,7 % | 75,6 % |

**Tabulka 24**: Průměrná přesnost při zkrácení na 2500 s frekvenční BoW.

Při zvýšení počtu slov na 2500 se všechny metody kromě LDA zlepšily. KNN a NB měly u testovacího datasetu, stejně jako už se tak stalo u SVM, přesnost nad 90 %. Nicméně i LDA mělo lepší výsledek než při zkrácení na 1000 slov a použití frekvenční *bag-of-words*.

| | | 7000 slov, lemmatizace a odstranění stop slov | 10000 slov |
|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % |
| | Testovací | 97,8 % | 95,6 % |
| LDA | Trénovací | 48,9 % | 52,2 % |
| | Testovací | 78,9 % | 53,3 % |
| KNN | Trénovací | 93,3 % | 84,1 % |
| | Testovací | 91,1 % | 78,9 % |
| NB | Trénovací | 100 % | 100 % |
| | Testovací | 95,5 % | 90 % |
| Decision Tree | Trénovací | 100 % | 100 % |
| | Testovací | 81,1 % | 75,6 % |

**Tabulka 25**: Průměrná přesnost s binární BoW s citlivostí 2.

Při dalším zvyšováním počtu slov se výsledky zlepšily. Při zkrácení na 7000 slov, použití lemmatizace, odstranění stop slov a binární *bag-of-words* s citlivostí 2 se SVM dostalo až na přesnost 97,8 %. Všechny metody kromě KNN s těmito parametry dosáhly svého nejlepšího výsledku v této úloze.

| | | |
|---|---|---|
| accuracy train | | 0.8703703703703703 |
| accuracy test | | 0.7777777777777778 |
| recall train | Asimov | 1.0 |
| | Foglar | 1.0 |
| | Lem | 0.6111111111111112 |
| recall test | Asimov | 1.0 |
| | Foglar | 1.0 |
| | Lem | 0.3333333333333333 |
| precision train | Asimov | 0.72 |
| | Foglar | 1.0 |
| | Lem | 1.0 |
| precision test | Asimov | 0.6 |
| | Foglar | 1.0 |
| | Lem | 1.0 |
| f1 train | Asimov | 0.8372093023255813 |
| | Foglar | 1.0 |
| | Lem | 0.7586206896551725 |
| f1 test | Asimov | 0.7499999999999999 |
| | Foglar | 1.0 |
| | Lem | 0.5 |

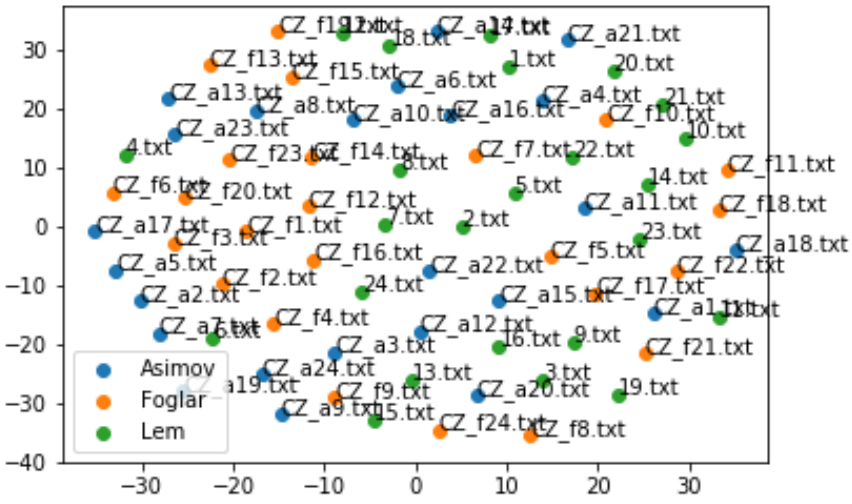
**Tabulka 26**: Výsledky KNN.

| | | |
|---|---|---|
| accuracy test | | 0.9444444444444444 |
| recall test | Asimov | 1.0 |
| | Foglar | 1.0 |
| | Lem | 0.8333333333333334 |
| precision test | Asimov | 0.8571428571428571 |
| | Foglar | 1.0 |
| | Lem | 1.0 |
| f1 test | Asimov | 0.923076923076923 |
| | Foglar | 1.0 |
| | Lem | 0.9090909090909091 |

**Tabulka 27: V**ýsledky testovacího datasetu NB.

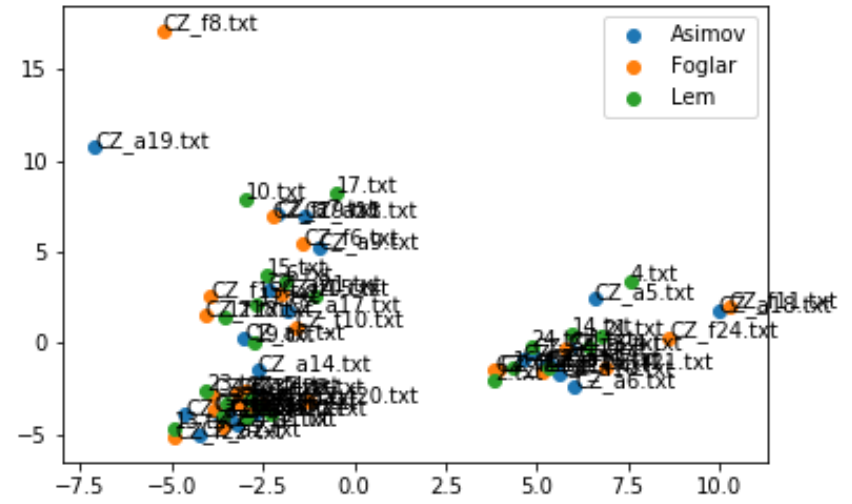
Dle očekávání se Foglar klasifikoval nejsnadněji. Pravděpodobně za to může odlišný žánr. U LDA žádná ze skupin nevyčnívala, nicméně LDA mělo ze všech metod nejhorší výsledky, a to i u trénovacích datasetů. U SVM měl Foglar nejvyšší úspěšnost v 97,9 % případů. Nižší ji měl jen jednou, jednalo se o *recall*. U Naive Bayes měl Foglar nejvyšší úspěšnost v 93,6 %, u KNN 72,3 %. U obou metod se vždy jednalo o *precision*. U *Decision Tree* měl Foglar vyšší úspěšnost v 36,2 %. Z toho v 80 % u *recall* a v 16,7 % u *precision*. V jednom případě byla nižší úspěšnost u *recall* i u *precision*, nicméně v tomto případě byla přesnost *Decision Tree* pouze 55 %.

3.2.1.3.2 Vizualizační metody



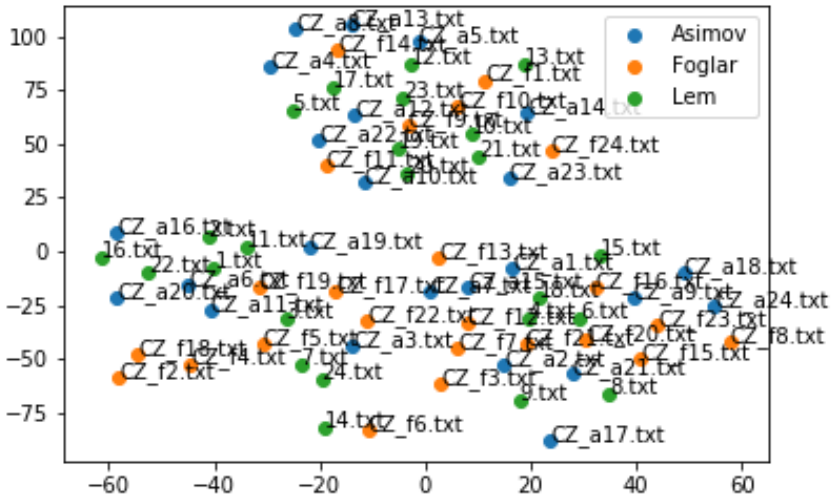

**Graf 25**: Výsledky MDS při zkrácení textů na 2500 slov.

Výsledky MDS při rozpoznání všech tří autorů vypadají velmi podobně jako při rozpoznání Asimovova od Lema. Texty nejsou rozděleny na skupiny, nicméně se místy shlukují texty stejného autora.




**Graf 26**: Výsledky PCA při zkrácení textů na 2500 slov.

Výsledky PCA jsou u této úlohy opět nečitelné.




**Graf 27**: Výsledky t-SNE při zkrácení textů na 7000 slov, použití lemmatizace, odstranění stop slov a binární BoW s citlivostí 2.

Výsledky T-SNE jsou rozdělitelné na tři skupiny. Nejsou ovšem rozděleny dle autorů. V horní skupině se nachází 9 textů od Asimovova, 9 od Lema a 6 od Foglara, v levé dolní skupině 6 od Asimovova, 9 od Lema a 8 od Foglara a v pravé dolní skupině 9 od Asimovova, 6 od Lema a 10 od Foglara.

### 3.2.2 Závěr

| | Foglar a Asimov | Asimov a Lem | Všechny texty |
|---|---|---|---|
| SVM | 100 % | 100 % | 97,8 % |
| LDA | 86,7 % | 91,7 % | 78,9 % |
| KNN | 100 % | 96,7 % | 93,3 % |
| NB | 100 % | 98,3 % | 95,5 % |
| *Decision Tree* | 91,7 % | 91,7 % | 81,1 % |

**Tabulka 28**: Nejvyšší průměrná přesnost testovacího datasetu u jednotlivých metod.

Při oddělení profesionálních autorů si nejlépe vedlo SVM. Jak u oddělení Asimovova od Foglara, tak u oddělení Asimovova od Lema se mu podařila získat 100 % přesnost. Pouze u oddělení všech tří skupin textů se přesnost snížila na 97,8 %. Ostatní metody měly také vysokou přesnost. NB a KNN u rozpoznání Foglara a Asimovova měly také přesnost 100 % a u ostatních dat měly stále přesnost nad 90 %. *Decision Tree* dosáhlo u obou binárních klasifikací stejného nejvyššího výsledku, a to 91,7 %. U rozdělení tří skupin se přesnost *Decision Tree* snížila na 85,6 %. LDA se jako jediné zlepšilo u oddělení Asimovova a Lema, a to na 91,7 %.

U prvního úkolu stačilo SVM a NB 500 slov na získání 100% úspěšnosti, KNN 1000 slov. *Decision Tree* dosáhlo svého nejlepšího výsledku při zkrácení na 1000 slov a použití lemmatizace a odstranění stop slov. LDA mělo nejvyšší přesnost při použití samotných indexů

a zkrácení na 5000 slov. Při oddělení Asimovova od Lema měly všechny metody nejvyšší přesnost při zkrácení na 10000 slov. Rozdíly u nich byly v typu *bag-of-words*. LDA a KNN měly nejvyšší přesnost při frekvenční *bag-of-words*, *Decision Tree* u binární a SVM a NB u binární s citlivostí 2 a lemmatizací a odstraněním stop slov. Při rozpoznání všech tří autorů si všechny metody kromě KNN vedly nejlépe při zkrácení na 7000 slov s binární *bag-of-words* s citlivostí 2, lemmatizací a odstraněním stop slov. KNN dosáhlo nejlepšího výsledku při zkrácení na 2500 s frekvenční *bag-of-words*, lemmatizací a odstraněním stop slov.

## 3.3 Určení autorství u příspěvků blogů

Na určení autorství u příspěvků blogů byl použit *Blog Authorship Corpus* (Schler, a další 2006). Jedná se o korpus příspěvků na blogové platformě bloger.com v roce 2004. Z něj byli vybraní tři autoři, dvě studentky ve věku patnáct a šestnáct let a třiceti devítiletý muž pracující jako moderátor v rádiu. Nejdříve budeme porovnávat jednu ze studentek, v korpusu označenou číslem 3763540, a muže číslo 3367100. Oba jsou z jiné demografické skupina a dá se předpokládat, že obsah jejich blogů bude jiný. Dále se budou porovnávat obě studentky navzájem. Druhá studentka je uvedena pod číslem 2635745. Obě studentky jsou ve stejné demografické skupině a oddělení jejich příspěvků by proto mohlo být složitější. Nakonec se budou porovnávat texty všech tří skupin.

Texty jsou různých délek, spíše však kratší. Nebudou tedy dále zkracovány. Nejmenší ze skupin má 47 textů. Vzhledem k délce textů a tomu, že se délky jednotlivých textů mohou lišit, se nedá očekávat úspěšnost stejně vysoká jako u profesionálních autorů. Nepřítomnost editorů také může znamenat více překlepů a chyb, což by se mohlo negativně projevit u lemmatizace, *stemmingu* a odstranění stop slov. Na druhou stranu na příspěvcích osobních blogů, jaké jsou zde použity, se podílí pouze jeden člověk, a tudíž se na autorském stylu nemůže podepsat styl editora či překladatele.

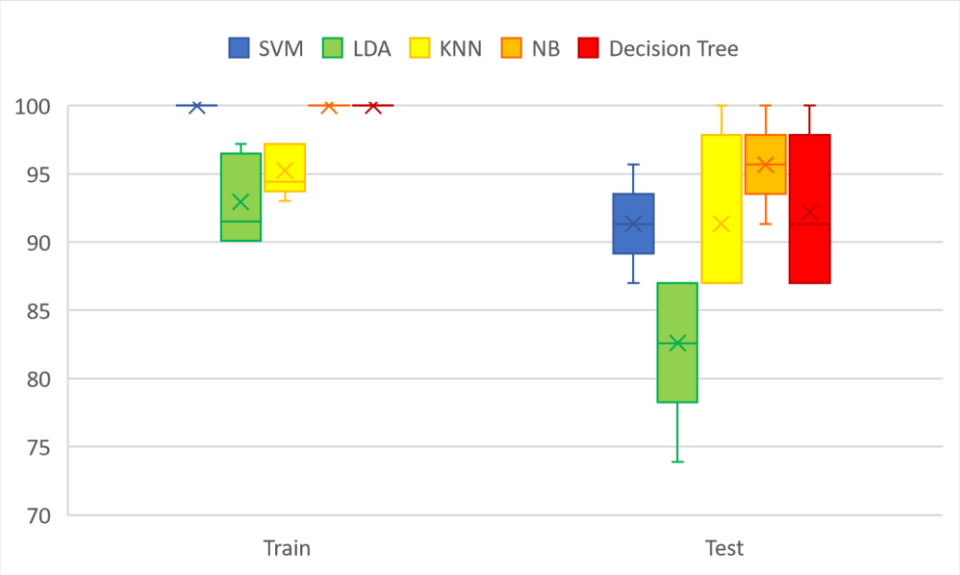
### 3.3.1 Výsledky

#### 3.3.1.1 Metody učící se s učitelem

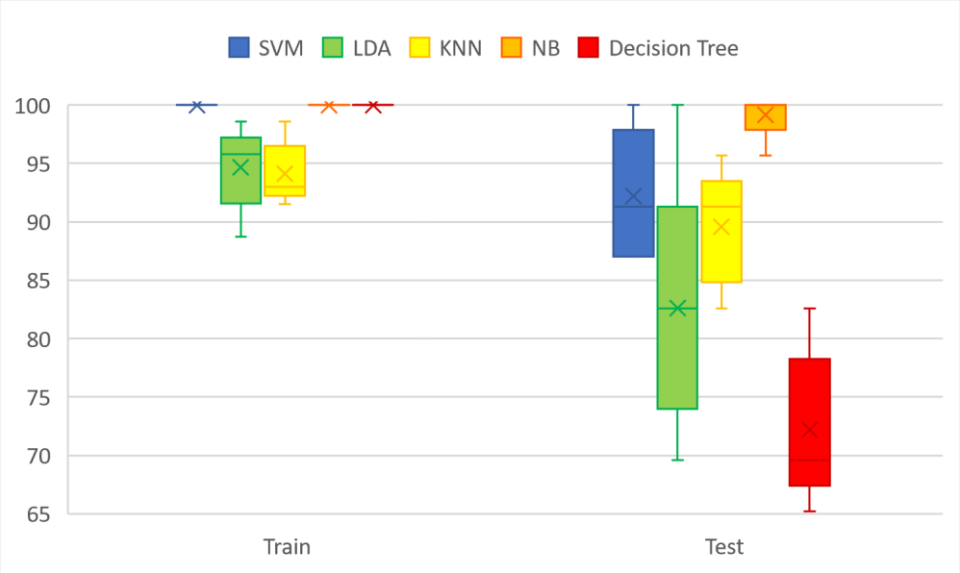
##### 3.3.1.1.1 Odlišné demografické skupiny

| | | Frekvenční BoW | Frekvenční BoW s lemmatizací a odstraněním stop slov | Binární BoW s lemmatizací a odstraněním stop slov | Frekvenční BoW se stemmingem a odstraněním stop slov |
|---|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 91,3 % | 92,2 % | 89,6 % | 95,7 % |
| LDA | Trénovací | 92,9 % | 94,7 % | 94,4 % | 93,8 % |
| | Testovací | 82,6 % | 82,6 % | 73 % | 73,4 % |
| KNN | Trénovací | 95,2 % | 94,1 % | 93,8 % | 89 % |
| | Testovací | 91,3 % | 89,6 % | 83,5 % | 85,2 % |
| NB | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 95,7 % | 99,1 % | 96,5 % | 93,9 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 92,2 % | 72,2 % | 69,6 % | 75,7 % |

**Tabulka 29**: Průměrná přesnost při klasifikaci blogů z odlišných demografických skupin.




**Graf 28**: Výsledky s frekvenční BoW.




**Graf 29**: Výsledky s frekvenční BoW, lemmatizací a odstraněním stop slov.

Při oddělování blogových příspěvků z odlišných demografických skupin mělo nejlepší výsledky NB. Při použití frekvenční BoW úspěšnost testovacího datasetu dosahovala v průměru 95,7 %. Při použití lemmatizace a odstranění stop slov se zvýšila na průměrných 99,1 %, přičemž čtyřikrát dosáhla 100% úspěšnosti a jednou 95,7%. Použití lemmatizace a odstranění stop slov nepomohlo všem metodám. SVM mělo také vyšší úspěšnost oproti samotné tokenizaci, ačkoliv v průměru o pouhých 0,9 %. LDA mělo úspěšnost stejnou, KNN a *Decision Tree* měli úspěšnost nižší. U *Decision Tree* se jednalo o 20 %. Vyzkoušeno bylo také zkrácení slov na jejich kořeny. Všechny metody kromě SVM měli nižší výsledky než při pouhé tokenizaci. U SVM se úspěšnost zvýšila z 91,3 % na 95,7 %. Při použití binární *Bag-of-words* s lemmatizací a odstranění stop slov se úspěšnost všech metod snížila.

| | | Odstranění stop slov | Lemmatizace | Stemming |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 92,2 % | 95,7 % | 89,6 % |
| LDA | Trénovací | 95,5 % | 94,9 % | 93,8 % |
| | Testovací | 70,4 % | 67,8 % | 77,4 % |
| KNN | Trénovací | 95,2 % | 95,5 % | 91 % |
| | Testovací | 87 % | 84,3 % | 82,6 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 93 % | 96,5 % | 95,7 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 73 % | 83,5 % | 82,6 % |

**Tabulka 30**: Průměrná přesnost jednotlivých úprav textu.

Aby se zjistila užitečnost lemmatizace, odstranění stop slov a *stemmingu*, byly vyzkoušeny samostatně. Ve většiny případů byla úspěšnost nižší než při pouhé tokenizaci. U odstranění stop slov mělo vyšší úspěšnost pouze SVM. U lemmatizace mělo vyšší úspěšnost SVM a NB.

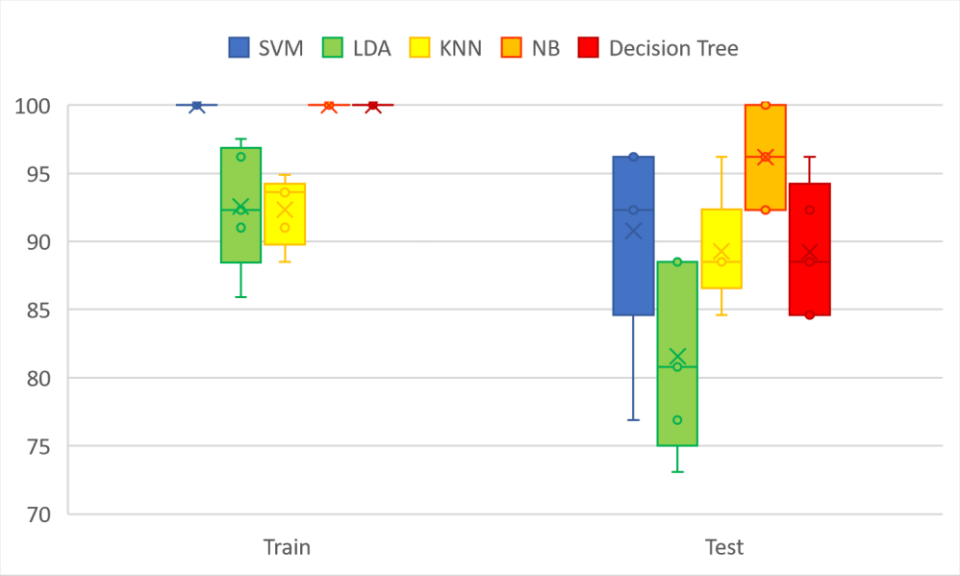
*Decision Tree* mělo sice úspěšnost nižší než při pouhé tokenizaci, ale vyšší, než při použití kombinace lemmatizace a odstranění stop slov. Stejně tomu tak bylo i u *stemmingu*. U *stemmingu* měly všechny metody úspěšnost stejnou nebo nižší než při pouhé tokenizaci.

U těchto dat byla tedy pro LDA, KNN a *Decision Tree* nejlepší pouze tokenizace, pro NB kombinace odstranění stop slov a lemmatizace a pro SVM buďto lemmatizace nebo kombinace odstranění stop slov a *stemmingu*.

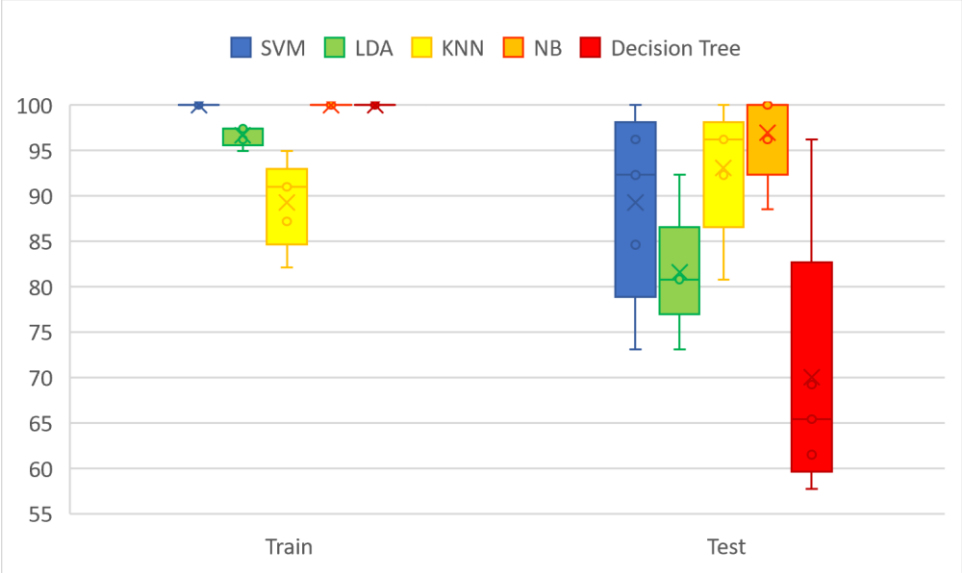
3.3.1.1.2 Stejná demografická skupina

| | | Pouze tokenizace | Stemming a odstranění stop slov | Lemmatizace a odstranění stop slov | Lemmatizace |
|---|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 90,8 % | 89,2 % | 90 % | 82,3 % |
| LDA | Trénovací | 92,6 % | 96,7 % | 96,4 % | 95,6 % |
| | Testovací | 81,5 % | 81,5 % | 83,8 % | 67,9 % |
| KNN | Trénovací | 92,3 % | 89,2 % | 91 % | 91,8 % |
| | Testovací | 89,2 % | 93 % | 89,2 % | 91,5 % |
| NB | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 96,2 % | 96,9 % | 95,4 % | 96,2 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 89,2 % | 70 % | 73,8 % | 86,2 % |

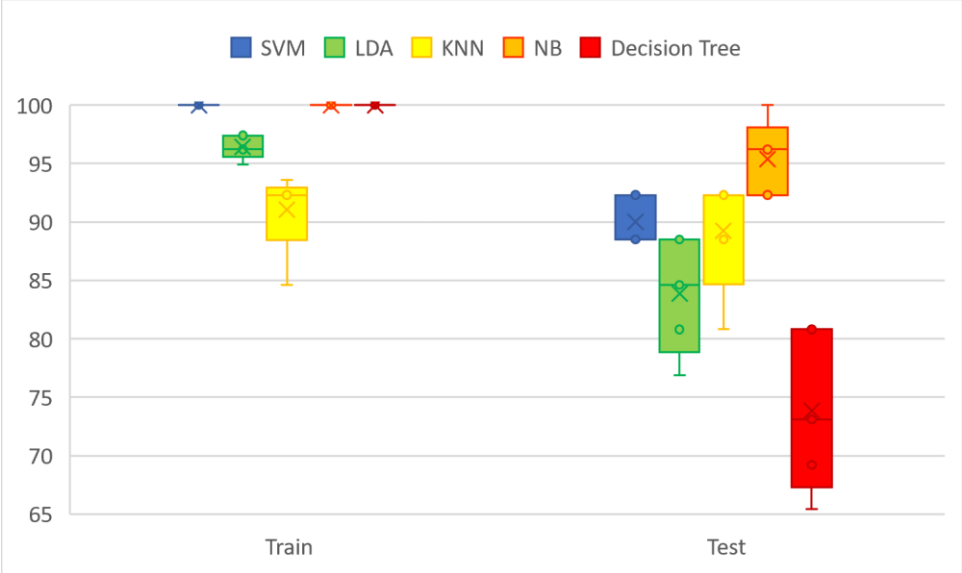
**Tabulka 31**: Průměrná přesnost blogerek ze stejné demografické skupiny.



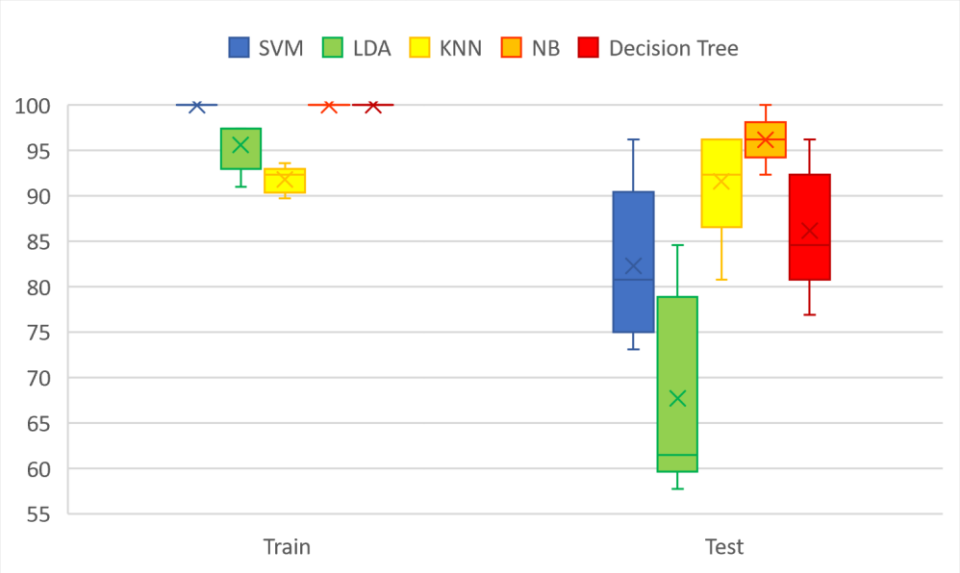

**Graf 30**: Výsledky při samotné tokenizaci.




**Graf 31**: Výsledky při *stemmingu* a odstranění stop slov.




**Graf 32**: Výsledky při lemmatizaci a odstranění stop slov.



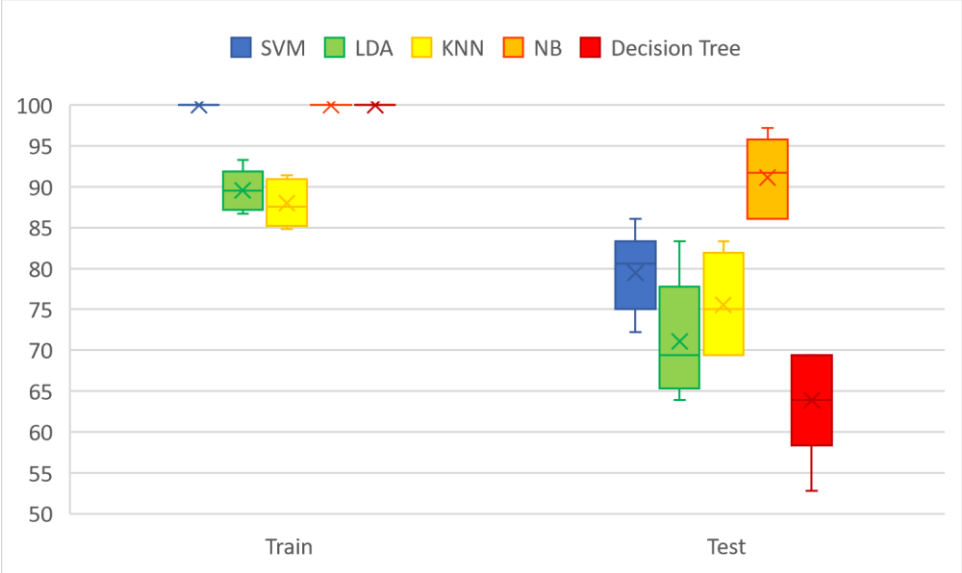

**Graf 33**: Výsledky při lemmatizaci.

Při použití textů blogerů ze stejné demografické skupiny byly výsledky testovacího datasetu SVM, NB a *Decision Tree* o něco nižší než u předchozí úlohy. Nejlepší průměrný výsledek u rozdělování blogerů ze stejné demografické skupiny byl u LDA o 1,2 % a u KNN o 1,7 % vyšší než u rozdělování textů blogerů z odlišných demografických skupin. NB, opět metoda s nejvyšší úspěšností, měla tentokrát nejlepší výsledky při použití *stemmingu* a odstranění stop slov, ačkoliv při rozdělování blogerů z odlišných demografických skupin kombinace *stemmingu* a odstranění stop slov výsledky NB zhoršila. Naopak u kombinace lemmatizace a odstranění stop slov měla NB u předchozí úlohy nejlepší výsledek a v této úloze se výsledek zhoršil. SVM měla tentokrát nejlepší výsledek bez použití dalších úprav, ačkoliv při rozdělování textů blogerů z odlišných demografických skupin bylo nejlepšího výsledku dosaženo při samotné lemmatizaci a při kombinaci *stemmingu* a odstranění stop slov.

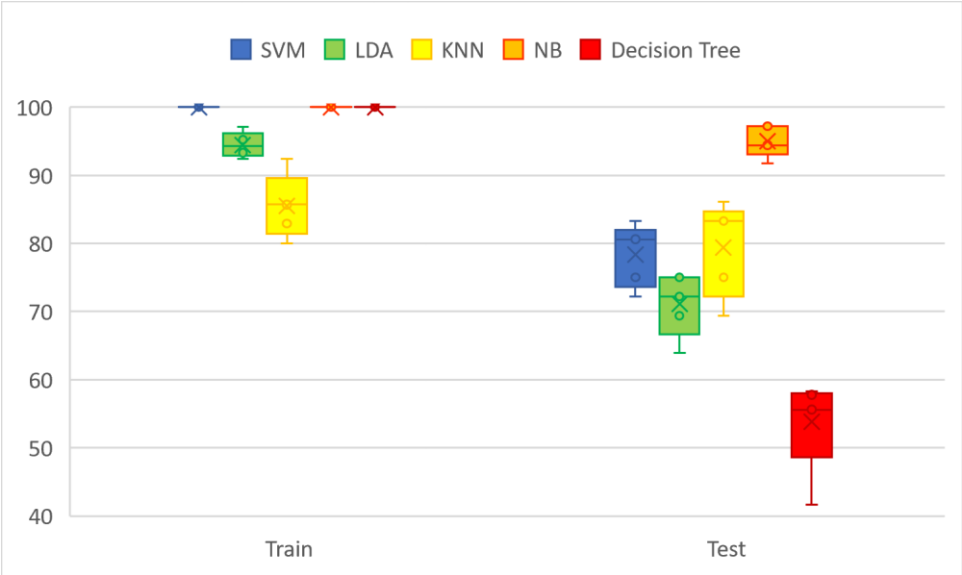
#### 3.3.1.1.3 Všechny skupiny

| | | Pouze tokenizace | Lemmatizace a odstranění stop slov | *Stemming* a odstranění stop slov |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 79,4 % | 78,3 % | 83,9 % |
| LDA | Trénovací | 89,5 % | 94,5 % | 93,5 % |
| | Testovací | 71,1 % | 71,1 % | 73,3 % |
| KNN | Trénovací | 88 % | 85,5 % | 85,9 % |
| | Testovací | 75,5 % | 79,4 % | 83,9 % |
| NB | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 91,1 % | 95 % | 92,7 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % |
| | Testovací | 63,8 % | 52,8 % | 58,9 % |

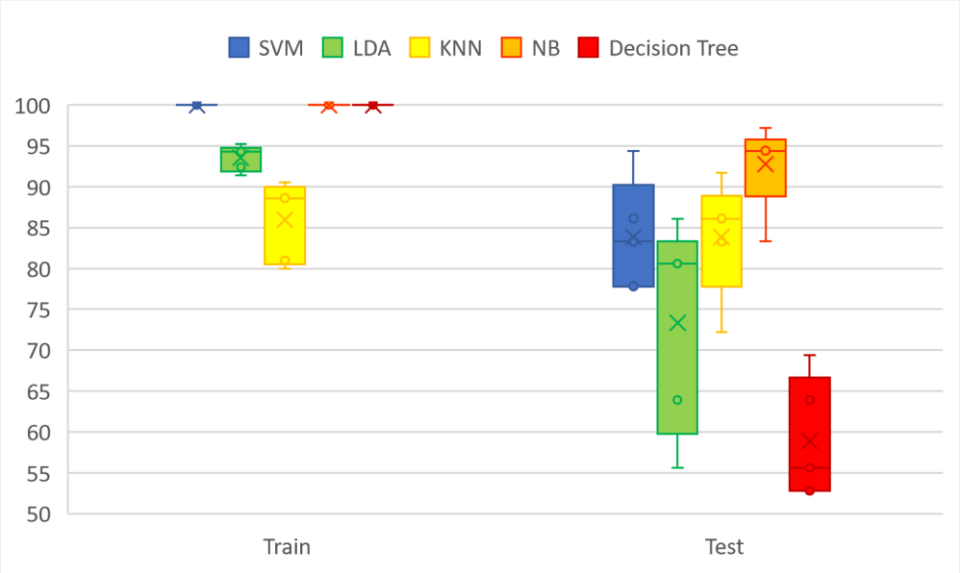
**Tabulka 32**: Průměrná přesnost oddělení všech skupin.




**Graf 34**: Výsledky při samotné tokenizaci.




**Graf 35**: Výsledky při lemmatizaci a odstranění stop slov.



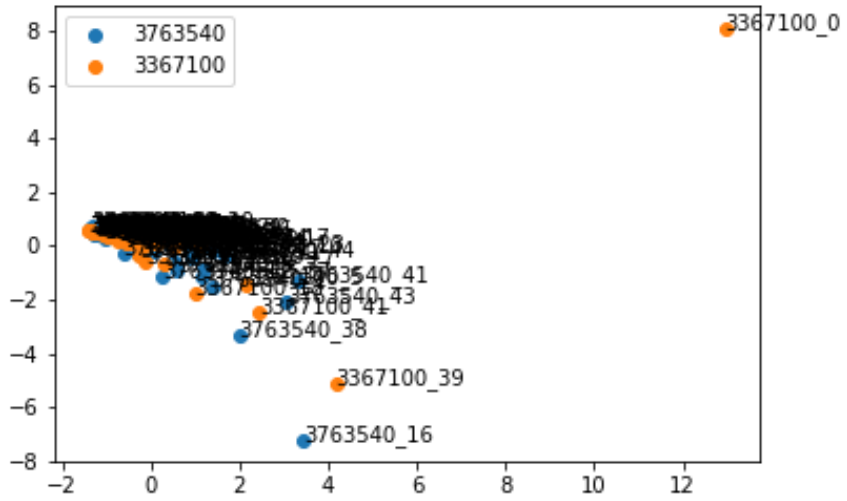

**Graf 36**: Výsledky při *stemmingu* a odstranění stop slov.

Při oddělování všech tří skupin lemmatizace a odstranění stop slov zlepšily úspěšnost pouze u NB a KNN, a to o 3,9 %. U SVM a *Decision Tree* se úspěšnost snížila, u LDA zůstala stejná. Při použití *stemmingu* a odstranění stop slov se všechny metody kromě *Decision Tree* zlepšily. Úspěšnost NB byla vyšší, než při pouhé tokenizaci, ale nižší než při lemmatizaci a odstranění stop slov. SVM, LDA a KNN měly své nejlepší výsledky.

Na grafu 35 je vidět, že při kombinaci lemmatizace a odstranění stop slov byly výsledky testovacího datasetu nejméně rozptýlené. Naproti tomu na grafu 36, u *stemmingu* a odstranění stop slov, má LDA u testovacího datasetu rozptyl 30,5 %, ačkoliv u trénovacího datasetu má relativně malý rozptyl.

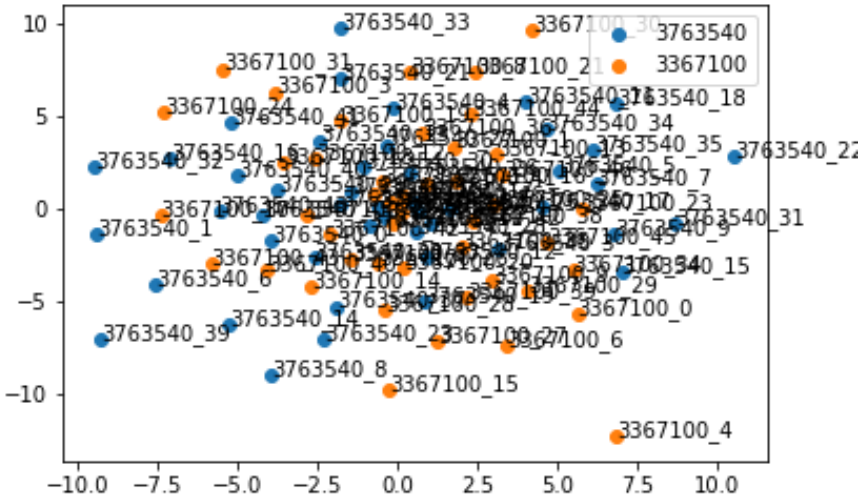
#### 3.3.1.2 Vizualizační metody

Výsledky vizualizačních metod vyšly u všech třech kombinací datasetů velmi podobně, tudíž zde nejsou zobrazeny všechny.

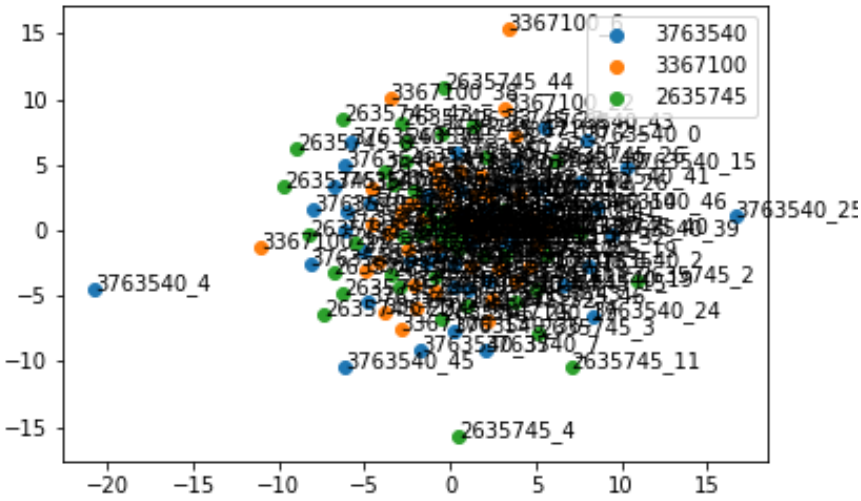



**Graf 37**: Výsledky PCA u odlišných demografických skupin.

Výsledky PCA jsou u této úlohy nečitelné. Opakuje se u nich stejný tvar, jen se liší, který text je tím nejvzdálenějším. Kvůli jednomu velmi vzdálenému textu jsou ostatní texty shromážděné u sebe a není poznat, zda se skupiny nějak rozdělily. Na první pohled vypadá, že na grafu 37 se oranžové texty v rámci shluku drží spíš vlevo a modré spíš vpravo, nicméně není možné zjistit, zda tomu tak doopravdy je.

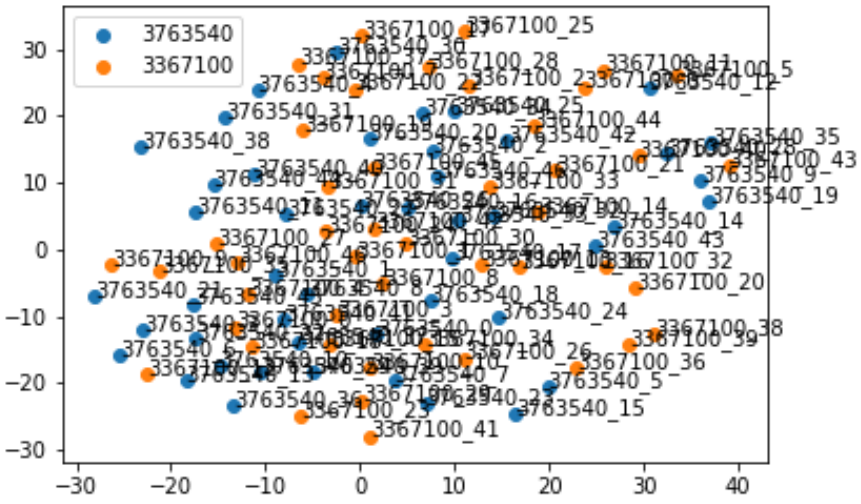



**Graf 38**: Výsledky MDS u odlišných demografických skupin.

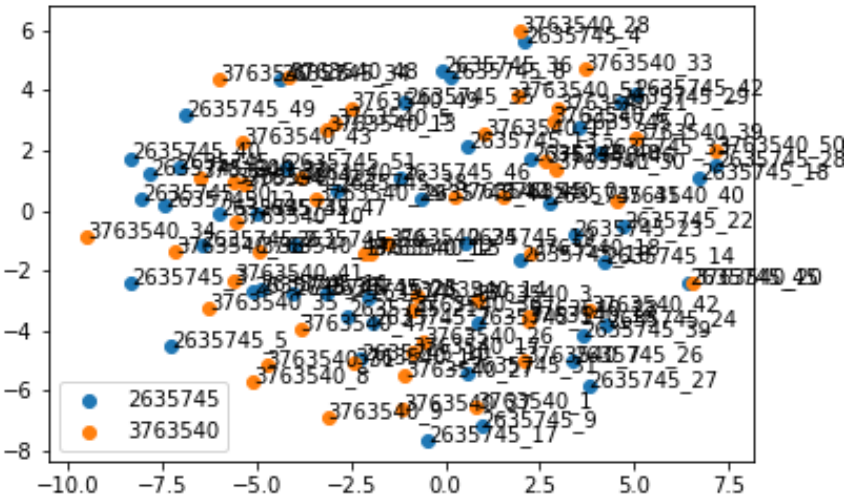



**Graf 39**: Výsledky MDS u všech tří skupin.

Výsledky MDS jsou opět ve tvaru oválu. Kromě prostředka, který je z důvodu překrývajících se názvů textů nečitelný, jsou některé texty ve skupinkách. Na grafu 38 je vpravo vidět oblast modrých bodů, uprostřed oblast bodů oranžových. Vlevo mají mezi sebou texty větší mezery a nemají takovou tendenci se shlukovat dle skupiny. Nejodlišnější jsou výsledky MDS u všech tří skupin. Nachází se tam více vzdálených bodů, kvůli kterým je zbytek bodů blíže u sebe, což činní většinu dat nečitelnými.




**Graf 40**: Výsledky t-SNE u odlišných demografických skupin.




**Graf 41**: Výsledky t-SNE u stejné demografické skupiny.

Poslední metodou je t-SNE. Na první pohled vypadají výsledky t-SNE jako náhodné umístění bodů. Při bližším zkoumání lze na některých místech najít shluky stejné skupiny, které nicméně nejsou nijak odděleny od shluků ostatních skupin. Na grafu 40 mezi sebou mají všechny body mezeru, ale u grafu 41 jsou vidět překrývající se body, mnohdy z odlišných skupin.

Ani jedna ze tří metod neoddělila skupiny od sebe. Některé texty stejné skupiny se k sobě přiblížily, nicméně stejně se tak k sobě přiblížily i texty odlišných skupin.

### 3.3.2 Závěr

| | Odlišné demografické skupiny | Stejná demografická skupina | Všechny tři skupiny |
|---|---|---|---|
| SVM | 95,7 % | 90,8 % | 83,9 % |
| LDA | 82,6 % | 83,8 % | 73,3 % |
| KNN | 91,3 % | 93 % | 83,9 % |
| NB | 99,1 % | 96,9 % | 95 % |
| Decision Tree | 92,2 % | 89,2 % | 63,8 % |

**Tabulka 33**: Nejlepší průměrná přesnost testovacího datasetu u všech metod.

Ze všech metod si při rozdělení textů neprofesionálních autorů nejlépe vedlo NB. Ačkoliv průměrná přesnost testovacího datasetu nikdy nedosáhla 100 %, u rozdělení textů blogerů z odlišných demografických skupin dosáhla průměrná přesnost 99,1 %. I u blogerů ze stejné demografické skupiny a kombinace všech tří skupin textů se výsledky NB pohybovaly vysoko. Snížily se pouze o 2,2 % v případě stejné demografické skupiny a o 4,1 % v případě kombinace textů všech tří blogerů.

Přesnost SVM, které mělo nejlepší výsledky u profesionálních autorů, zde mírně klesla. O oddělování textů z odlišných demografických skupin mělo SVM druhou nejvyšší přesnost s 95,7 %. U všech tří skupin se o druhé místo dělilo s KNN. Při textech ze stejné demografické

skupiny mělo KNN vyšší přesnost, navzdory tomu, že u textů z odlišných demografických skupin bylo až čtvrté.

KNN, stejně jako LDA, se totiž nejlépe vedlo u textů blogerů ze stejné demografické skupiny. K navýšení došlo jen o 1,2 % v případě LDA a o 1,7 % v případě KNN, nicméně i tak je to signifikantní, vzhledem k tomu, že se všechny ostatní metody zhoršily, v případě SVM až o 4,9 %. Něco podobného se již stalo u profesionálních autorů. LDA se nejlépe dařilo rozpoznat Asimovova od Lema, navzdory tomu, že jsou si podobnější. I při rozdělování všech tří autorů od sebe byly výsledky vyšší, než při rozdělování Asimovova od Foglara.

*Decision Tree* si vedlo dost odlišně u jednotlivých dat. U odlišných demografických skupin bylo třetí nejlepší s 92,2 %. U stejné demografické skupiny se přesnost snížila pod hranici 90 %, ale stále byla vyšší než u LDA. U kombinace všech tří skupin byla nejlepší průměrná přesnost *Decision Tree* pouhých 63,8 %, nejnižší ze všech metod.

SVM, NB a *Decision Tree* měly u trénovacího datasetu u rozpoznávání neprofesionálních autorů vždy 100% úspěšnost. I přesto mělo *Decision Tree* nízkou úspěšnost u testovacího datasetu. Nejmenší propad mezi trénovacím a testovacím datasetem nastal u KNN při rozdělování všech autorů při použití *stemmingu* a odstranění stop slov. KNN mělo často malý rozdíl mezi výsledky trénovacího a testovacího datasetu, v některých případech mělo dokonce vyšší úspěšnost u testovacího datasetu než u trénovacího.

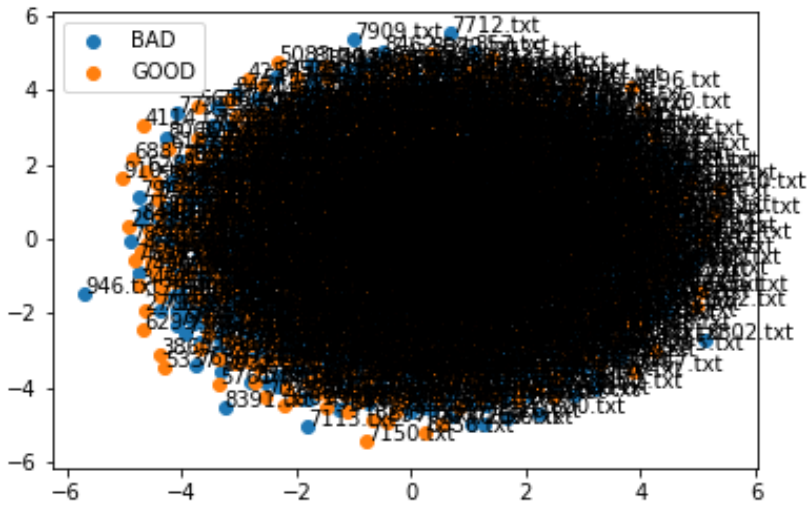
Nejlepší výsledky nastávaly při frekvenční *bag-of-words*, a to buď se samotnou tokenizací, nebo s odstraněním stop slov a lemmatizací či *stemmingu*.

## 3.4 Určení sentimentu

Další z úloh je automatické určení sentimentu, tedy citového zabarvení textu. Pro určení sentimentu bude použit dataset recenzí filmů. Jedná se o data ze *sentence polarity dataset v1.0*. (Pang a Lee 2005) Texty jsou v angličtině, v délce jedné věty a jsou rozděleny na ty s pozitivním a negativním sentimentem. Obě skupiny mají 1000 textů. Z důvodu krátké délky textů texty nebudou dále zkracovány.

Specifika tohoto datasetu, a to zejména délka jednotlivých textů, by mohla sehrát roli v úspěšnosti. Texty v datasetu se skládají z krátkých vět, často vytržených z kontextu. V jednotlivých textech se pravděpodobně nevyskytují stejná slova určující sentiment. Pokud bude použita negace, (Např. *Tenhle film není dobrý*., má negativní sentiment, ale vyskytuje se zde významové slovo s pozitivním sentimentem) mohla by být určena opačná skupina. V recenzích se také může vyskytovat sarkasmus, který může být obtížný na určení i pro lidi. Pokud se výskyt slov ukáže jako nedostačující, mohly by pomoci indexy. Nicméně jsou texty krátké, takže se v nich slova nejspíš nebudou opakovat.

Vizualizační metody nejsou u této úlohy interpretovatelné, jelikož příliš velký počet bodů je udělal nečitelnými. Také nebylo možno použít odstranění stop slov, jelikož u některých z textů docházelo k odstranění všech slov.

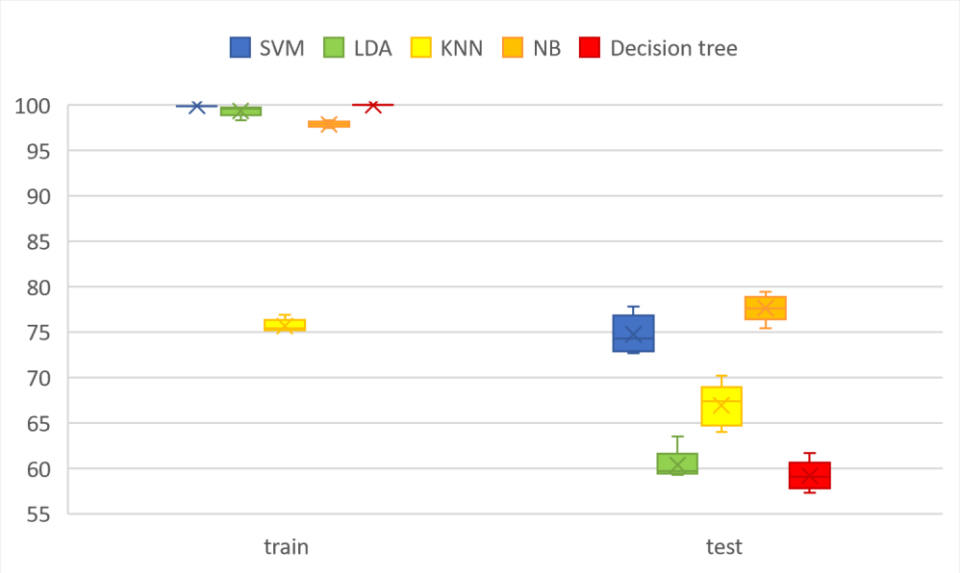



**Graf 42**: Výsledky MDS u určení sentimentu.

### 3.4.1 Výsledky

| | | Frekvenční BoW | Lemmatizace a frekvenční BoW | Binární BoW | Lemmatizace a binární BoW |
|---|---|---|---|---|---|
| SVM | Trénovací | 99,9 % | 99,9 % | 99,9 % | 99,9 % |
| | Testovací | 73,6 % | 74,7 % | 74,2 % | 72,3 % |
| LDA | Trénovací | 99,6 % | 99,4 % | 98,2 % | 99 % |
| | Testovací | 62,5 % | 60,4 % | 59,9 % | 62,1 % |
| KNN | Trénovací | 74,8 % | 75,7 % | 74,2 % | 73,1 % |
| | Testovací | 65,7 % | 66,9 % | 64,2 % | 65,5 % |
| NB | Trénovací | 97,8 % | 97,9 % | 97,5 % | 97,3 % |
| | Testovací | 75,9 % | 77,6 % | 75,8 % | 76,3 % |
| Decision Tree | Trénovací | 100 % | 100 % | 100 % | 100 % |
| | Testovací | 60 % | 59,1 % | 59,5 % | 60,2 % |

**Tabulka 34**: Průměrná přesnost určení sentimentu.

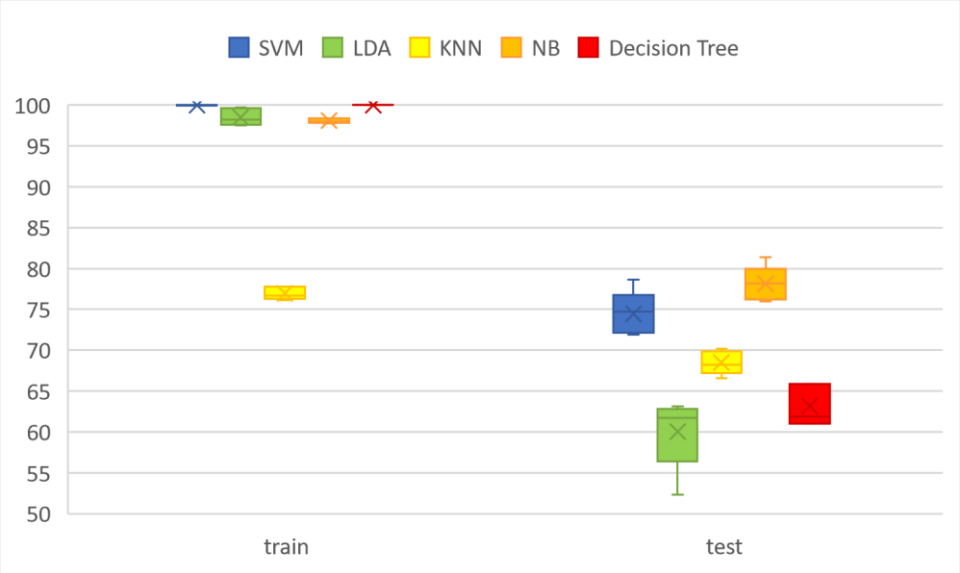



**Graf 43**: Výsledky při frekvenční BoW s lemmatizací.

U určení sentimentu se nejlépe vedlo NB. Jeho průměrná přesnost testovacího datasetu dosahovala 75,9 % u frekvenční *bag-of-words*. Při použití binární *bag-of-words* se zhoršila o pouhých 0,1 %, nicméně nastal menší nárůst při přidání lemmatizace než u frekvenční *bag-of-words*. SVM měla stejně jako NB a KNN nejlepší výsledky při frekvenční *bag-of-words* s lemmatizací, nicméně na rozdíl od NB a KNN měla při vynechání lemmatizace lepší výsledek s binární *bag-of-words*. LDA se nejlépe dařilo u frekvenční *bag-of-words* bez lemmatizace, *Decision Tree* u binární *bag-of-words* s lemmatizací.

| | | Indexy | BoW |
|---|---|---|---|
| SVM | Trénovací | 57,6 % | 99,9 % |
| | Testovací | 54,8 % | 74,5 % |
| LDA | Trénovací | 58,1 % | 98,5 % |
| | Testovací | 54,8 % | 60 % |
| KNN | Trénovací | 64,9 % | 77 % |
| | Testovací | 52,8 % | 68,5 % |
| NB | Trénovací | 55,9 % | 98,1 % |
| | Testovací | 52,8 % | 78,1 % |
| Decision Tree | Trénovací | 90,9 % | 100 % |
| | Testovací | 52 % | 63,2 % |

**Tabulka 35**: Průměrná přesnost samotných indexů a samotné BoW.




**Graf 44**: Výsledky při samotné BoW.

Při použití samotných indexů se všechny metody zhoršily. Jejich výsledky byly jen o 0,5-4,8 % lepší než náhodné hádání. Zhoršily se výsledky i u trénovacího datasetu. SVM, LDA a NB měli průměrnou úspěšnost u trénovacího datasetu jen o 2,8-3,3 % vyšší než u testovacího datasetu. KNN, které mělo u testovacího datasetu nejhorší úspěšnost ze všech metod, dosáhlo u trénovacího datasetu přesnosti 64,9 %. *Decision Tree*, které má ve většině případů 100% úspěšnost u trénovacího datasetu, dosáhlo přesnosti 90,9 %.

Použití pouze *bag-of-words* zlepšilo všechny metody kromě LDA. Stejně jako u použití lemmatizace se však výsledky změnily pouze mírně. K největšímu rozdílu došlo u *Decision Tree*, které se při použití pouze *Bag-of-words* zlepšilo o 3,2 %. Pro *Decision Tree*, KNN a NB navíc došlo k jejich nejlepšímu průměrnému výsledku testovacího datasetu u rozpoznání sentimentu.

### 3.4.2 Závěr

| | |
|---|---|
| SVM | 74,7 % |
| LDA | 62,5 % |
| KNN | 68,5 % |
| NB | 78,1 % |
| Decision Tree | 63,2 % |

**Tabulka 36**: Nejlepší průměrná přesnost testovacího datasetu,

Rozpoznání sentimentu dopadlo ze všech úloh v této práci nejhůře. Zatímco u všech ostatních úloh se alespoň některým metodám podařilo dostat přesnost nad 90 %, u rozpoznání sentimentu se jen jediné metodě podařilo dostat nad 75 %. Nejlépe si vedlo NB s 78,1 %, těsně po něm následovalo SVM s 74,7 %. NB, KNN a *Decision Tree* se nejlépe dařilo při samotné *bag-of-words*, SVM při frekvenční *bag-of-words* s lemmatizací a LDA při frekvenční *bag-of-words* bez lemmatizace.

## 3.5 Rozpoznání spamu

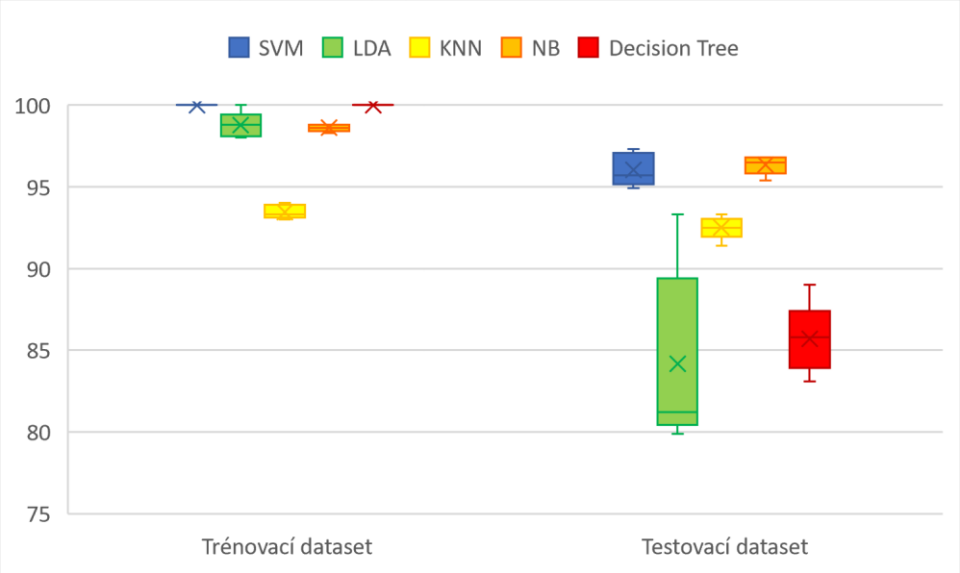
Posledním z typů klasifikace, které jsou v této práci vyzkoušeny, je rozpoznání spamu. Automatické rozpoznání spamu se běžně využívá u elektronické pošty, nicméně v této úloze jsou použity textové zprávy. Dataset se skládá z krátkých textů běžné a nevyžádané pošty v angličtině. (Almeida a Hidalgo 2012, Dua a Graff 2019) Po zkrácení počtu textů obou skupin na stejný počet má dataset 1494 textů. Stejně jako u určení sentimentu není možno u textů odstranit stop slova, jelikož u některých textů docházelo k odstranění všech slov. Vizualizační metody nejsou z důvodu množství textů interpretovatelné.

Z důvodu překlepů a zkratek, které se v některých textech vyskytují, by se mohla snížit účinnost lemmatizace a *stemmingu*. Podobně jako u určení sentimentu se jedná o velmi krátké texty, které tudíž není možné dále zkracovat. Na rozdíl od určení sentimentu jsou texty samotné více diverzifikované. U určení sentimentu se jednalo pouze o recenze filmů, kdežto u rozpoznání spamu se jedná o různé typy korespondence.

### 3.5.1 Výsledky

| | | Frekvenční BoW | Binární BoW | Binární BoW s citlivostí 2 |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 93,1 % |
| | Testovací | 96,5 % | 97,1 % | 84,8 % |
| LDA | Trénovací | 99,2 % | 98,8 % | 93,5 % |
| | Testovací | 85,8 % | 83,2 % | 83,7 % |
| KNN | Trénovací | 87 % | 93,5 % | 86,4 % |
| | Testovací | 85,6 % | 92,9 % | 82,6 % |
| NB | Trénovací | 99 % | 98,6 % | 64,8 % |
| | Testovací | 95,8 % | 96,6 % | 61,8 % |
| Decision Tree | Trénovací | 100 % | 100 % | 99 % |
| | Testovací | 83,6 % | 86,9 % | 75,4 % |

**Tabulka 37**: Průměrná přesnost rozpoznání spamu.




**Graf 45**: Výsledky trénovacího a testovacího datasetu u rozpoznání spamu s binární BoW.

Už první výsledky rozpoznání spamu jsou vysoké. Při použití frekvenční *bag-of*-words se SVM dostalo u testovacího datasetu na přesnost 96,5 % a NB na 95,8 %. Zbylé metody se pohybovaly nad 80 %. Použití binární *bag-of-words* zlepšilo průměrnou úspěšnost u všech metod kromě LDA. LDA se zhoršilo o 2,6 %. Jak může být viděno na grafu 45, LDA mělo navíc na rozdíl od ostatních metod u testovacího datasetu větší rozptyl dat. Průměrné výsledky KNN se zvýšily nejvíce, a to o 7,3 % z 85,6 % na 92,9 %. U *Decision Tree* se přesnost zvýšila o 3,4 %, u NB o 0,8 % a u SVM o 0,6 %. Naproti tomu binární *bag-of-words* s citlivostí dva zhoršila výsledky všech metod. U NB došlo k propadu o 34 % v porovnáním s frekvenční *bag-of-words*. LDA byla jediná metody, kde byly výsledky binární *bag-of-words* s citlivostí dva vyšší, než výsledky binární *bag-of-words* s citlivostí 1, ačkoliv o pouhých 0,5 %.

| | | Stemming | Lemmatizace | Tagování |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 92,2 % |
| | Testovací | 96,1 % | 96,7 % | 90 % |
| LDA | Trénovací | 99,9 % | 99,4 % | 92 % |
| | Testovací | 91,7 % | 87,6 % | 78,7 % |
| KNN | Trénovací | 87,4 % | 87,9 % | 91,6 % |
| | Testovací | 86,2 % | 86,7 % | 88,4 % |
| NB | Trénovací | 98,5 % | 99 % | 89,8 % |
| | Testovací | 95,5 % | 95,3 % | 88,3 % |
| Decision Tree | Trénovací | 100 % | 100 % | 99,9 % |
| | Testovací | 83,2 % | 84,8 % | 80,7 % |

**Tabulka 38**: Průměrná přesnost rozpoznání spamu při frekvenční BoW.

Při použití *stemmingu* se přesnost LDA zlepšila oproti samotnému tokenizování o 5,9 % na 91,7 %. KNN se také zlepšilo, ačkoliv pouze o 0,4 %. Ostatní metody se mírně zhoršily. U lemmatizace se naopak zvýšila průměrná přesnost všech metod kromě NB. NB se zhoršilo o 0,5 %. LDA se zlepšilo nejvíce, a to o 1,8 %, nicméně mělo stále lepší výsledek při

*stemmingu*. Použití tagů se projevilo zhoršením u všech metod kromě KNN, kde došlo ke zlepšení o 2,8 %.

| | | Stemming | Lemmatizace | Tagování |
|---|---|---|---|---|
| SVM | Trénovací | 100 % | 100 % | 89,5 % |
| | Testovací | 95,9 % | 95,8 % | 88,8 % |
| LDA | Trénovací | 99,9 % | 99,7 % | 90,2 % |
| | Testovací | 77,1 % | 90,3 % | 88,9 % |
| KNN | Trénovací | 95,1 % | 94,2 % | 89,7 % |
| | Testovací | 93,9 % | 92,2 % | 88,3 % |
| NB | Trénovací | 98,6 % | 98,6 % | 87,2 % |
| | Testovací | 96,6 % | 95,3 % | 85,7 % |
| Decision Tree | Trénovací | 100 % | 100 % | 99,9 % |
| | Testovací | 87,6 % | 86 % | 85,4 % |

**Tabulka 39**: Průměrná přesnost rozpoznání spamu při binární BoW.

Oproti použití binární *bag-of-words* bez dalších úprav, lemmatizace a tagování snížilo průměrnou úspěšnost všech metod, kromě LDA. To se zlepšilo o 7,1 % u lemmatizace a o 5,7 % u tagování. U *stemmingu* došlo ke zhoršení u SVM a LDA, NB mělo stejný výsledek. KNN se zlepšilo o 1 % a *Decision Tree* o 0,7 %. LDA si při použití lemmatizace i tagování mělo s binární *bag-of-words* vyšší přesnost než s frekvenční *bag-of-words*, ačkoliv bez těchto úprav mělo vyšší úspěšnost u frekvenční *bag-of-words*. Při *stemmingu* došlo s frekvenční *bag-of-words* k nejvyšší průměrné přesnost LDA, s binární *bag-of-words* k té nejnižší. *Decision Tree* má výsledky ve všech případech lepší při použití binární *bag-of-words*. U KNN vychází binární *bag-of-words* lépe při *stemmingu* a lemmatizaci. Při tagování je úspěšnost o 0,1 % nižší než při použití frekvenční *bag-of-words*. SVM a NB mají úspěšnost při *stemmingu*, lemmatizaci a tagování s binární *bag-of-words* nižší či stejnou než s frekvenční *bag-of-words*, ačkoliv bez úprav vychází binární *bag-of-words* lépe.

### 3.5.2 Závěr

| | Spam |
|---|---|
| SVM | 97,1 % |
| LDA | 91,7 % |
| KNN | 93,9 % |
| NB | 96,6 % |
| Decision Tree | 87,6 % |

**Tabulka 40**: Nejlepší průměrná přesnost u všech metod.

Navzdory krátkým textům rozpoznání spamu vyšlo podstatně lépe něž rozpoznání sentimentu. Nejlépe se vedlo SVM, kde nejvyšší úspěšnost dosáhla 97,1 % a to při použití binární *bag-of-words*. Druhá nejlepší metoda byla NB s nejvyšší průměrnou úspěšností 96,6 %. Nejhůř dopadlo *Decision Tree*, s nejvyšší úspěšností 87,6 %. LDA je jediná metoda, jejíž nejlepší úspěšnost nenastala při použití binární *bag-of-words*. Nejlepší úspěšnost LDA nastala při *stemmingu* a frekvenční *bag-of-words*. KNN a *Decision Tree* dosáhly nejlepšího výsledku při *stemmingu*

a binární *bag-of-words*. NB na tom bylo stejně, pouze mělo stejný výsledek i u binární *bag-of-words* se samotnou tokenizací.

# 4 Závěr

Záměrem této práce bylo vytvořit v programovacím jazyku Python software, která umožní trénovat klasifikaci textů a zhodnotit její výsledky. Cílem psané práce bylo přestavit, jak software funguje. V teoretické části byly popsány základní rysy programu. V první kapitole byly popsány vlastnosti, na které je text převeden. Konkrétně se jedná o indexy TTR, RR, Giniho koeficient, Shannonova entropie, průměrná délka slov a *bag-of-words*. Po představení jednotlivých vlastností byla krátce vysvětlena jejich normalizace. Druhá kapitola se zabývala možnosti předzpracování textu, jako je tokenizace, lemmatizace, *stemming*, tagování a odstranění stop slov. Třetí kapitola popisovala formát výsledků a jejich interpretaci. Poslední kapitola teoretické části uživatele seznámila s procesem instalace a následného nastavení programu. V praktické části bylo na programu vyzkoušeno několik typů klasifikace textu. Jednalo se rozpoznání jazyka, určení autorství, rozpoznání sentimentu a rozpoznání spamu. Klasifikace textů dosáhly dobrých výsledků, zejména rozpoznání jazyka. Rozpoznání jazyka, a to i u jazyků ze stejné jazykové rodiny, mělo bezchybné výsledky už při zkrácení na pouhé desítky slov.

Software byl tedy napsán a následně ověřen na praktických úkolech. Funguje dle zadání, nicméně by se dal dále rozšířit a zlepšit. Ačkoliv k použití programu není třeba znalost programování, program se stále musí spouštět v Pythonu a uživatel při zadávání nastavení musí dbát na jeho pravidla. Logickým dalším krokem by tedy bylo vytvořit aplikaci, která by byla více uživatelsky přívětivější. V uživatelsky přívětivější aplikaci by bylo zároveň možné přidat rozšířené nastavení. Nejdřív by tedy uživatel zadal pouze základní nastavení, a specifičtější nastavení by zadával pouze, pokud by chtěl. Zároveň by se mohla rozšířit možnost si klasifikaci textů přizpůsobit.

Dále je možné se zaměřit na zlepšování funkčnosti programu. U grafů vizualizačních metod by byla vhodná možnost vypnout přidávání názvů textů, což by mohlo grafy, zvláště ty, na kterých se nachází velké množství textů, zpřehlednit. Při redukci počtu slov by se mohla přidat možnost zkrátit všechny texty na počet slov nejkratšího textu, což by umožnilo zkrátit texty, u kterých si uživatel není jistý délkou. U předzpracování textů by mohly být přidány další jazyky, případně i možnost, jak předzpracovat každou třídu zvlášť, a tudíž moci texty upravit i u rozpoznání jazyků.

# Literatura a zdroje


Almeida, T.A, a Gómez Hidalgo. „UCI Machine Learning Repository: SMS Spam Collection Data Set.“ *UCI Machine Learning Repository.* 22. 6 2012. http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection (přístup získán 2. Červenec 2020).

Anaconda inc. *Anaconda Software Distribution.* Vers. 2-2.4.0. Software. 2016.

Bird, Steven, Edward Loper, a Ewan Klein. *Natural Language Processing with Python.* O'Reilly Media Inc, 2009.

Čech, Radek, Ioan-Iovitz Popescu, a Gabriel Altman. *Metody kvantativní analýzy (nejen) básnických textů.* Olomouc: Univerzita Palackého, 2014.

Dua, D., a C. Graff. *UCI Machine Learning Repository.* Irvine, California, 2019.

Han, Jiawei, Micheline Kamber, a Jian Pei. *Data Mining: Concepts and Techniques.* Třetí edice. San Francisco: Morgan Kaufmann, 2011.

Hunter, John D. „Matplotlib: A 2D Graphics Environment.“ *Computing in Science & Engineering*, 2007: 90-95.

Malhotra, Varun. „GitHub - softvar/json2html : python module for converting complex JSON object to HTML Table representation .“ *GitHub.* 2019. (přístup získán 3. Březen 2020).

Musilová, Václava. „Co je forenzní lingvistika I - Pojem a možnosti znaleckého zkoumání, předměty zkoumání.“ *Čeština doma a ve světě*, 2005: 65-70.

Oliphant, Travis E. *A guide to NumPy.* Trelgol Publishing, 2006.

Pang, Bo, a Lillian Lee. „Seeing Stars: Exploiting Class Relationships For Sentiment Categorozation With Respect To Rating Scales.“ *Proceedings of ACL.* 2005. 115-124.

Pedregosa, F. et al. „Scikit-learn: Machine Learning in Python.“ *Journal of Machine Learning Research*, 2011: 2825-2830.

Savand, Alireza. „GitHub - Alir3z4/python-stop-words: Get list of common stop words in various languages in Python.“ *GitHub.* 2018. https://github.com/Alir3z4/python-stop-words (přístup získán 21. Leden 2020).

Schler, J., M. Koppel, S. Argamon, a J. Pennebaker. *Effects of Age and Gender on Blogging in Proceedings of 2006 AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs.* 2006.

Srinivasa-Desikan, Bhargay. *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, SpaCy and Keras.* Packt Publishing Ltd, 2018.

Straka, Milan, a Jana Straková. „Czech Models (MorfFlex CZ 161115 + PDT 3.0) for MorphoDiTa 161115.“ LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, 2016.

—. „MorphoDiTa: Morphological Dictionary and Tagger.“ LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, 2014.

Thanaki, Jalaj. *Python Natural Language Processing.* Packt Publishing Ltd, 2017.

Van Rossum, G., a F. L. Drake. *Python 3 Reference Manual.* Scotts Valley, California: CreateSpace, 2009.

Virtanen, Pauli et al. „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.“ *Nature Methods*, 2020: 261-272.

# 5 Příloha

## 5.1 Obsah přiloženého media

- bakalarska_prace.pdf
    - Soubor obsahuje tuto bakalářskou práci ve formátu pdf.
- read_me.txt
    - Soubor obsahuje tento obsah a sekci 2.4 této práce, popisující instalaci a nastavení programu.
- /source_code
    - Složka obsahuje zdrojový kód programu.
- /datasets
    - Složka obsahuje data použitá při testování programu.

## 5.2 Seznam použitých textů

Použité jazyky bible:

Copala Triqui
Chiquitano
Mborena Kam
Gofa
Yucana
Quechua – Cajamarca
Quechua – North Junín
Quechua – Northern Conchucos Ancash
Quichua - Northern Pastaza

Jaroslav Foglar:

Boj o první místo
Chata v Jezerní kotlině
Devadesátka pokračuje
Dobrodružství v temných uličkách
Dobrodružství v zemi nikoho
Historie svorné sedmy
Hoši od Bobří řeky
Jestřábe vypravuj 2
Když duben přichází
Kronika hochů od Bobří řeky 2

Kronika hochů od Bobří řeky

Kronika Ztracené stopy

Modrá rokle

Pod Junáckou vlajkou

Poklad černého delfína

Přístav volá

Soví jeskyně

Stezka odvahy

Strach nad Bobří řekou

Stínadla se bouří

Tajemná Řásnovka

Tajemství velkého Vonta

Záhada hlavolamu

Závod o Modřínový srub

Isaac Asimov:

Roboti a impérium

Roboti úsvitu

Předehra k nadaci

Já, robot

Kalibán

Sny robotů

A zrodí se nadace

Zrození

Hvězdy jako prach

Dítě času

Pozitronový muž

Sbohem, země

Ani sami bohové

Nadace a říše

Povídky

Konec věčnosti

David Starr - Tulák po hvězdách

Nahé slunce

Nadace

Ocelové jeskyně

Mezi dvěma kroky

Vize robotů

Hvězdy jako prach

Druhá nadace

Stanisław Lem:

Návrat z hvězd

Lov

Mír na zemi

Nepřemožitelný

Gladiátor z Venusie

Krvavé kameny

Pánův hlas

Vzpomínky Ijona Tichého

Planeta Eden

Příběhy pilota Pirxe

Rýma

Solaris

Astronauti

Hvězdné deníky Ijona Tichého

Fiasko

Invaze z Aldebaranu

K mrakům Magellanovým

Kyberiáda

Bajka o třech strojích, které vyprávěly králi Genialonovi

Bajky robotů

Deník nalezený ve vaně

Futurologický kongres