Mendelova univerzita v Brně
Provozně ekonomická fakulta

Získávání a analýza textových dat pro oblast finančních trhů

Diplomová práce

Vedoucí práce:
doc. Ing. František Dařena, Ph.D.

Bc. Jonáš Petrovský

Brno 2016

Na tomto místě bych chtěl poděkovat vedoucímu své diplomové práce doc. Ing. Františku Dařenovi, Ph.D. Zaprvé za to, že mi poskytl možnost pracovat na zajímavém tématu, které kombinuje programování a analýzu dat, a také za možnost podílet se na výzkumném projektu. Zadruhé za to, že mi pomohl stanovit cíl a podobu experimentů, vedoucích ke splnění cíle práce. A konečně zatřetí za cenné poznámky, rady a připomínky k mé práci, díky kterým se (snad) podařilo toto poněkud specifické téma zpracovat na dostatečné úrovni. Dále bych rád poděkoval své rodině, že mě během celého studia i tvorby této práce podporovala.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Získávání a analýza textových dat pro oblast finančních trhů**
vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.
Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 22. května 2016 .....

Abstract

PETROVSKÝ, J. *Acquiring and analyzing text data for financial markets*. Brno, 2016. Master thesis. Mendel University in Brno, Faculty of Business and Economics.

The thesis examines the connection between content (and sentiment) of text documents published on the internet and direction of movements of stock prices. First, financial and stock markets, available data sources, options of storing data and relevant concepts of data and text mining and sentiment analysis are briefly described. Work methodology thoroughly describes the procedure of acquiring (Yahoo Finance, Facebook, Twitter), storing (MySQL), processing (Python, VecText) and analyzing (classification, feature selection, lexicon-based sentiment analysis) the data. In the thesis, a module for data acquiring and other modules necessary for processing and analysis of data were created. The data were collected for all companies in S&P 500 and FTSEurofirst 300 indices during a period of 8 months. Results of the classification show that if the price movement is (compared to the current trend) sufficiently substantial, there is a rather clear connection. The accuracy was 68–74 % for Yahoo and Twitter (for top 50 % and 10 % files). Results of the dictionary method were not so conclusive (62% accuracy for Yahoo). However it was discovered that the sentiment dictionary (1 000 words) that was generated automatically using the CHI method has an accuracy that is only 2 % lower for Yahoo and Twitter than the accuracy of the combined sentiment dictionary (9 412 words) which was created as a part of the thesis.

Keywords

stock market, stock price return, EWMA, social networks, financial news articles, acquiring data from web, machine learning, text mining, classification, feature selection, sentiment analysis, sentiment dictionary, VADER, scikit-learn, Python, MySQL

Abstrakt

PETROVSKÝ, J. *Získávání a analýza textových dat pro oblast finančních trhů*. Brno, 2016. Diplomová práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta.

Práce zkoumá souvislost mezi obsahem (a sentimentem) textových dokumentů publikovaných na internetu a směrem pohybů cen akcií firem na burze. V rámci rešerše jsou stručně popsány finanční resp. akciové trhy, dostupné datové zdroje a možnosti uchovávání dat a relevantní koncepty z oblasti dolování znalostí z textových dat a analýzy sentimentu. Metodika práce zevrubně popisuje postup získávání (Yahoo Finance, Facebook, Twitter), ukládání (MySQL), zpracování (Python, VecText) a analýzy (klasifikace, *feature selection*, určování sentimentu pomocí slovníku) dat. V rámci práce byl vytvořen modul pro získávání dat a další moduly nutné pro

zpracování a analýzu dat. Data byla sbírána po dobu 8 měsíců pro všechny firmy z indexů S&P 500 a FTSEurofirst 300. Výsledky klasifikace ukazují, že pokud je cenový pohyb (oproti aktuálnímu trendu) dostatečně výrazný, existuje poměrně jasná souvislost. Správnost byla pro Yahoo a Twitter 68–74 % (pro horních 50 % resp. 10 % souborů). Výsledky slovníkové metody nebyly tak průkazné (správnost 62 % pro Yahoo). Nicméně bylo zjištěno, že metodou CHI automaticky vygenerovaný slovník sentimentu (1 000 slov) má pro Yahoo a Twitter jen o 2 % nižší správnost než (v rámci práce vytvořený) kombinovaný slovník sentimentu (9 412 slov).

Klíčová slova

akciový trh, změna ceny akcie, EWMA, sociální sítě, finanční novinové články, získávání dat z webu, strojové učení, text mining, klasifikace, výběr důležitých atributů, analýza sentimentu, slovník sentimentu, VADER, scikit-learn, Python, MySQL

Obsah

Seznam obrázků

Seznam tabulek

Seznam pojmů a zkratek

adjusted close Cena upravená o rozdělení akcií a dividendy

API Application Programming Interface

blue chips Nejkvalitnější akcie dané burzy

CLI Command-line interface

DBMS Database Management System

DDL Data Definition Language

ERD Entity–Relationship Diagram

EWMA Exponentially weighted moving average

GUI Graphical user interface

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IDF Inverse Document Frequency

JSON JavaScript Object Notation

NLP Natural language processing

open-source SW Software s otevřeným zdrojovým kódem

ORM Object-relational mapping

POS Part-Of-Speech

RDBMS Relační DBMS

REST Representational State Transfer

SMA Simple moving average

TF Term Frequency

TP Term Presence

UML Unified Modeling Language

URL Uniform Resource Locator

1 Úvod a cíl práce

1.1 Úvod

Již od nepaměti si lidé mluveným slovem předávali zkušenosti, sdělovali názory a diskutovali. V určitou dobu začala být tato komunikace písemně zaznamenávána. Nejdříve v piktografické formě, později vznikaly různé abecedy. Umět číst a psát bylo dlouhou dobu výsadou jen zvolených jedinců, knihy vznikaly ručním opisováním a bylo obtížné se k nějaké vůbec dostat. S příchodem knihtisku se ovšem vše změnilo – nastala *Printing revolution* – masová výroba a rozšíření knih umožnily dosud nevídanou možnost cirkulace informací a myšlenek, což výrazně ovlivnilo společnosti, kde k této revoluci došlo (Eisenstein, 1979).

Stejnou revolucí byl vznik osobního počítače a především internetu. Jak se tyto dvě technologie rozšiřovaly a stávaly se nedílnou součástí každodenního života, vznikaly různé webové stránky či služby, nahrazující fyzické (papírové) nosiče informací těmi elektronickými. Vznikly internetové vyhledávače a zpravodajské portály. Bylo tak poprvé v historii možné automaticky (pomocí počítače) stahovat a analyzovat dokumenty publikované v čistě elektronické podobě.

Masové šíření názorů bylo ovšem stále vyhrazeno jen novinářům nebo technicky zdatným uživatelům. Samozřejmě existovala různá diskuzní fóra a člověk někdy mohl diskutovat pod publikovaným článkem či napsat recenzi k produktu na Amazonu, ale těmto textům se nedostávalo velké pozornosti a bylo obtížné je hromadně získávat. Potom přišly služby poskytující tzv. blogy, které uživatelům umožnily se zaregistrovat a snadno a v přehledné formě publikovat na webu své myšlenky. Další razantní změna a rozšíření množství publikovaných textů přišlo se vznikem sociálních sítí (Facebook, Twitter). Tyto poprvé v historii umožnily sledovat názory lidí ve velkém měřítku. To spolu se vznikem mnoha nových webových portálů a rozvojem informačních technologií způsobilo, že lze nyní komplexně studovat informace publikované o zvoleném tématu.

V rámci této práce bude zkoumáno téma akciových trhů. K tomuto tématu je na internetu dostupné velké množství zdrojů včetně těch, kde lidé publikují články a názory ohledně dané firmy či trhu. Dá se říct, že v dnešní době běžní i institucionální investoři získávají informace o vývoji situace ve firmách, na daném trhu či v celé ekonomice především z internetu. To znamená, že má smysl zkoumat vliv těchto informací na chování investorů resp. na pohyby cen akcií (nebo spíše zkoumat souvislost mezi informacemi a pohyby cen).

Dle Teorie efektivního trhu (*Efficient markets theory* – EMT) jsou ceny akcií (resp. dalších cenných papírů) blízko své fundamentální hodnoty. Efektivita trhu vyplývá z toho, že investoři jsou racionální, zpracovávají všechny dostupné informace a dle nich (racionálně – logicky) formují svá očekávání a oceňují akcie. Pokud se objeví iracionální investoři, kteří navrhnou jinou cenu, tak se navzájem vyruší nebo jsou tyto odchylky eliminovány racionálními investory, kteří provádějí arbitráž. Avšak empirická pozorování akciových trhů EMT odporují, jelikož anomálie

a přehnaná volatilita nemůže být vysvětlena změnami fundumentální hodnoty dané firmy (Shiller, 2003).

Behaviorální ekonomie nám říká, že emoce mohou hluboce ovlivnit chování a rozhodování jednotlivců. Otázkou je, zda lze toto aplikovat i na celé lidské společnosti – tedy zda společnosti mohou zažívat nálady, které ovlivňují jejich rozhodování. Provedené studie ukazují, že tomu tak je (Bollen et al., 2011).

Behavioral finance theory rozporuje základní zásady EMT a ukazuje, že neefektivitu na kapitálových trzích lze vysvětlit tím, že budou splněny dva základní předpoklady. Prvním je, že investoři jsou pod vlivem sentimentu, což může vést ke špatnému určení ceny. Sentimentem se myslí názory investorů na budoucí peněžní toky a investiční riziko, které ovšem nejsou podloženy jednoznačnými fakty. Druhý předpoklad říká, že na trhu existuje limitovaná arbitráž – oprava špatného ocenění může být nákladná a riziková, takže se do ní arbitrážní spekulanti mnohdy vůbec nepouštějí. Toto vše má za následek, že ceny na kapitálovém trhu jsou ovlivněny emocemi, náladami, přístupy a názory účastníků trhu (Kaplanski a Levy, 2010).

Na téma souvislosti sentimentu resp. obsahu textových zpráv a pohybů akciových trhů byla publikována celá řada odborných článků. Tyto empirické studie obvykle používají algoritmy strojového učení k určení kolektivního sentimentu ve společnosti a to na základě analýzy obrovského množství textových dat, která jsou nyní na internetu dostupná.

Cílem Bollen et al. (2011) bylo zjistit, zda nálada společnosti ovlivňuje její kolektivní rozhodování resp. zda má souvislost s ekonomickými indikátory. V článku je nálada zjišťována skrz texty na Twitteru a je zkoumána korelace náladu vyjadřujících měřítek s hodnotou akciového indexu DJIA. Bylo dosaženo správnosti 87,6 % v předpovídání toho, zda DJIA na konci dne klesne či stoupne.

Arias et al. (2013) měl za cíl zjistit, zda veřejný sentiment (získaný z tweetů) může zlepšit předpovídání ekonomických indikátorů. Byly zkoumány tweety ze dvou domén: akciový trh a tržby filmů v kinech. Závěrem bylo, že nelineární modely jsou schopny využít Twitter data ke svému prospěchu.

Na základě výše uvedených informací lze konstatovat, že ceny akcií jsou ovlivňovány jak publikovanými fundamentálními informacemi, tak myšlenkovými pochody v hlavách jednotlivých účastníků trhu (které obvykle nevznikají racionálně). Tyto dva vlivy se prolínají a působí současně.

Bylo by proto zajímavé zkoumat, zda a jakou souvislost mají texty a pohyby cen akcií (ty lze chápat jako zisky akcií, volatilitu nebo tržní nedokonalosti). Texty mohou vyjadřovat jak fundamentální fakta (racionalita), tak emoce a názory lidí (iracionalita) a lze tak zkoumat, jaký vliv má (i)racionalita investorů na akciový trh.

Získané poznatky mohou využít v zásadě dvě skupiny subjektů. První jsou investiční fondy, banky, centrální banky či výzkumná pracoviště, které dostanou do rukou nástroj ke sledování a vyhodnocování nálady na akciovém trhu a souvislosti této nálady s pohybem trhu. Druhou skupinou jsou investoři či spekulanti, kteří se snaží načasovat nákup akcie na nejvhodnější dobu. Pro ty by bylo užitečné vě-

dět, do jaké míry lze z publikovaných textů usoudit, jakým směrem a o kolik se v budoucnosti změní cena akcie.

Touto problematikou se, jak lze vidět z textu výše, zabývá velké množství vědců z různých oblastí (ekonomie, psychologie, neurověda, informatika). Z tohoto důvodu se jedná o aktuální téma, jehož zkoumání může přispět k rozvoji této oblasti tak, abychom dokázali lépe poznat lidské chování a jeho vliv na akciové trhy a eventuálně využít získané znalosti a vzniklé nástroje v praxi.

1.2 Cíl práce

Cílem práce je provést sběr, zpracování a analýzu textových dat pro oblast finančních trhů, se zaměřením na sentiment textů a jeho vliv na akciový trh. Tento obecně pojatý cíl bude nyní podrobněji specifikován. Předně je nutné zdůraznit, že práce bude zaměřena pouze na akciové trhy, nikoliv na finanční trhy obecně. Dále uveďme, že práce je součástí širšího výzkumu, který má za cíl zjistit, zda a jak mohou být informace obsažené v textech použity pro vysvětlení pohybu cen akcií.

Práce obsahuje v podstatě tři dílčí cíle:

1. Zvolit a posbírat zdrojová data.
2. Zpracovat získaná data do podoby vhodné pro analýzu.
3. Provést analýzu dat a vyvodit z výsledků závěry.

První dva podcíle nepotřebují další komentář. Třetím cílem je v podstatě pokusit se zjistit, zda a jak lze informace obsažené v textech (novinové články, statusy na sociálních sítích) využít k vysvětlení pohybu cen akcií. Jinak řečeno – jak texty ovlivňují cenu akcie firmy, které se týkají resp. jaká je souvislost mezi publikovanými texty a cenou akcie. Opačný směr, tedy jak cena akcie ovlivňuje publikované texty, nebude uvažován.

Texty mohou obsahovat především faktické (např. finanční výsledky) nebo názorové (např. pozitivní nebo negativní vztah k firmě či produktu) informace, případně obojí. V druhém případě tak můžeme zkoumat, zda pohyb ceny akcie souvisí s tím, jaké emoce, nálady a názory mají lidé, obchodující na akciových trzích resp. zda a jak se jimi nechají ovlivnit. Mezi těmito dvěma typy dokumentů nebudeme explicitně rozlišovat, i když hrubé rozlišení poskytnou samotné zdroje textů – dá se očekávat, že novinové články budou obsahovat faktické, zatímco statusy na sociálních sítích spíše názorové informace.

K provedení této analýzy lze využít celou řadu postupů a výsledky lze ověřovat pomocí různých metod. Zde je na místě zdůraznit, že se práce nebude zabývat statistickým ověřováním získaných výsledků, ekonometrií nebo matematickým modelováním. Vyhodnocení bude provedeno pomocí základních metrik z oblasti data miningu (*accuracy*, *precision*, *recall*), které budou zkoumat, jak často obsah či sentiment dokumentu správně předpověděl pohyb ceny akcie vzhledem k datu publikace dokumentu a danému časového okamžiku v budoucnosti.

Pro splnění cíle práce je nutné provést následující kroky (vycházejí ze zadání práce):

- Seznámit se se základním rozdělením finačních trhů a se specifiky akciových trhů.
- Seznámit se s procesem sběru a způsoby uchovávání textových dat z různých zdrojů na internetu, využitelných pro získávání informací, týkajících se firem na akciových trzích.
- Seznámit se s problematikou dolování znalostí z textových dat a metodami určování sentimentu textů.
- Posbírat, vhodně označit a uložit zprávy (dokumenty, texty) týkající se určitých firem a vybraná ekonomická data.
- Analyzovat textová data s ohledem na jejich obsah resp. sentiment a to s využitím externí informace (slovníky, ekonomická data).
- Vyvodit závěry týkající se vztahu mezi ekonomickými daty a obsahem resp. sentimentem v textových datech a diskutovat jejich využitelnost.

Nakonec bude celá práce zhodnocena.

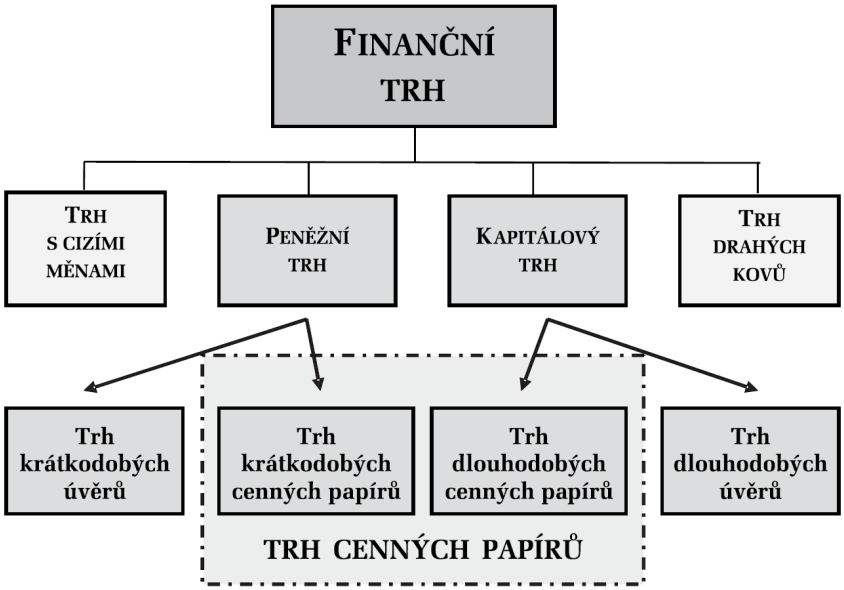
2 Rešerše

2.1 Finanční trhy

Nejdříve je nutné vysvětlit některé ekonomické pojmy, související s touto prací.

Finanční trh je nedílnou součástí finančního (resp. ekonomického) systému. Ten zajišťuje především funkce depozitní (ukládání úspor) a kreditní (získávání peněz) (Rejnuš, 2014, s. 40–41). Finanční trh se nejčastěji člení dle toho, jaké finanční investiční nástroje se na jeho dílčích trzích obchodují (viz obrázek 1).

Trhy jsou primární/sekundární, (ne)veřejné, (ne)organizované (Rejnuš, 2014, s. 66–68). Nás zajímá trh cenných papírů, na kterém se obchodují majetkové (podnikové akcie) a dluhové (dluhopisy) cenné papíry (Rejnuš, 2014, s. 229–230).



Obr. 1: Členění finančního trhu podle základních druhů investičních instrumentů (Rejnuš, 2014, s. 61)

2.2 Akciové trhy

V práci se budeme zajímat pouze o majetkové cenné papíry (akcie), obchodované na sekundárních veřejných organizovaných trzích (konkrétně na burzách).

2.2.1 Přístupy k analýze akcií

Nejdříve prozkoumáme strategie analýzy akcií, což nám poskytne základní informace o povaze akciového trhu a vlivech, které na jeho účastníky působí.

Fundamentální analýza

Fundumentální analýza je založena na předpokladu, že vnitřní hodnota akcie (spravedlivá cena resp. kurz) se liší od aktuální tržní ceny akcie – akcie tak je buď podhodnocená, nebo nadhodnocená. Vnitřní hodnota akcie je individuální názor účastníka trhu (investora) na to, jaký by měl být spravedlivý kurz akcie. Investoři na základě svých názorů (resp. očekávání) zadávají obchodní příkazy s určitými parametry a dle nich se mění kurz akcie na burze (Rejnuš, 2014, s. 238).

Technická analýza

Technická analýza se používá pro analýzu konkrétního akciového titulu (či indexu) s cílem předpovědět budoucí vývoj jeho kurzu. Jejím dalším cílem je určovat co nejvhodnější okamžiky k provádění obchodů (nákup nebo prodej). Základem je sledování vývoje kurzů a objemu obchodů v čase (Rejnuš, 2014, s. 300).

Psychologická analýza

Psychologická analýza vychází z předpokladu, že na akciové trhy má velký vliv masová psychologie – tedy že budoucí vývoj kurzů akcií závisí na impulzech, které ovlivňují chování davu (investiční, burzovní publikum). Impulzy vedou dav k nákupům nebo prodejům, což následně ovlivňuje kurzy akcií (Rejnuš, 2014, s. 372). Existuje mnoho koncepcí, zabývajících se vlivem masové psychologie na kurzy akcií. Například to je *Teorie spekulativních bublin* (Rejnuš, 2014, s. 376) nebo *Drasnarova koncepce*, která vysvětluje pokles a růst kurzů akcií jako důsledek působení dvou protichůdných lidských vlastností – chamtivosti a strachu (Rejnuš, 2014, s. 392).

2.2.2 Faktory ovlivňující akcie

Na základě předchozí sekce můžeme stanovit faktory, které mají vliv na cenu (příp. jiné vlastnosti) akcie. U každého faktoru je v závorce uveden zdroj, ze kterého pochází: FA = fundamentální analýza, TA = technická analýza, PA = psychologická analýza, BFT = teorie behaviorálních financí (Gladiš, 2005, s. 114–130).

- Vnitřní hodnota akcie (FA).
- Globální ekonomické agregáty (FA).
- Charakteristiky daného odvětví (FA).
- Minulé či budoucí odhadované ekonomické údaje o firmě (FA).
- Časová řada kurzu akcie, přičemž kurzů může být více typů (TA).
- Časová řada objemu realizovaných obchodů s danou akcií (TA).
- Pohyb celého akciového trhu případně existence různých period na trhu (TA).
- Technické psychologické indikátory (TA).
- Masová psychologie účastníků burzovního trhu (PA).
- Spekulativní bubliny (PA).

- Působení vnitřních emocí na investory – interní chyby (BFT).
- Působení okolí na investory – externí chyby (BFT).

Jaký typ faktorů má větší vliv na cenu akcií lze hrubě určit dle Kostolanyho burzovní psychologie tak, že zjistíme, jaká ze dvou skupin investorů (hráči, investoři) aktuálně drží většinu akcií. Pokud to budou investoři, měly by hrát hlavní roli fundementální faktory. Naopak pokud budou převažovat hráči, větší vliv budou mít technické a především psychologické faktory (Rejnuš, 2014, s. 372–373). Jelikož psychologické faktory ovlivňují jak amatérské, tak profesionální investory, je jisté, že mají na akciový trh velký vliv a proto má význam se jimi zabývat.

V práci se zaměříme na psychologické faktory, které mají spojitost s náladou a názory reálných i potenciálních investorů. Budeme tedy zkoumat, jak na investory a potažmo celý akciový trh působí okolní prostředí (důsledkem čehož jsou externí chyby) ve formě textových dokumentů (resp. informací).

2.2.3 Akciové indexy

Každá burza má svůj index (číselnou hodnotu), který určuje aktuální stav burzy resp. vývoj cen akcií na burze. Tabulka 1 zobrazuje nejvýznamnější indexy pro Evropu a USA. Jedině *DJIA* je cenově vážený, všechny ostatní jsou hodnotově vážené. Do výběrových indexů se zařazují především tzv. *blue chip* společnosti, což jsou nejkvalitnější resp. nejprestižnější akcie dané burzy nebo oblasti (Rejnuš, 2014, s. 452). Z těchto indexů vybereme společnosti (firmy), které budou v práci zkoumány.

Tab. 1: Nejvýznamnější akciové burzovní indexy v Evropě a USA

Název indexu	Oblast	Počet společností
Dow Jones Industrial Average	USA	30 *blue chip* společností z NYSE
Nasdaq Composite	USA	2 592 (Yahoo!, 2009)
Standard & Poor's 500	USA	500
FTSE 100	Velká Británie	100 akcií z indexu FTSE-250
DAX 30	Německo	30 nejobchodovanějších *blue chips* z *Frankfurt Stock Exchange*
CAC 40	Francie	40 nejvýznamnějších *blue chips* z *Euronext Paris*
EURO STOXX 50	Eurozóna	50 *blue chip* největších společností v Eurozóně dle *free float value*
FTSEurofirst 300 Index	Evropa	300 největších společností dle tržní kapitalizace z indexu *FTSE Developed Europe Index* (FTSE Russell, 2016)

Zdroj: (Rejnuš, 2014, s. 452–454)

2.3 Získávání dat

V rámci práce je nutné získat určitá data o vybraných společnostech (firmách). Konkrétně se jedná o vybraná ekonomická data a především textová data, obsahující názory, emoce a nálady lidí ohledně firem na akciovém trhu. V této kapitole jsou představeny potenciální zdroje textových dat, dostupné na internetu. Jelikož dat v textové formě je dostatečné množství, nebudeme se zabývat audio ani video nahrávkami.

2.3.1 Zpravodajské servery

Jako první logický zdroj dat se nabízejí zpravodajské servery (webové portály), zaměřující se na akciové trhy. Tyto servery publikují různé novinové články, pojednávající o jednotlivých firmách i o akciovém trhu celkově. Nutným požadavkem je, aby obsahovaly pro každou firmu (akciový titul) jakousi „domovskou stránku“, kde jsou zobrazeny všechny informace (články), související s touto firmou.

Tabulka 2 ukazuje vybrané servery, které byly získány vyhledáním na *Google Search* pomocí fráze „stock market news“. Sloupec ALEXA udává globální *ALEXA rank*[1] daného serveru (resp. související domény druhého řádu). Sloupec Evropa značí, zda server nabízí informace o akciích mimo USA (konkrétně v Evropě). Toto bylo zjišťováno vyhledáváním akciového symbolu *FER.MC* (firma FERROVIAL na *Madrid Stock Exchange*). Sloupec DA říká, zda server nabízí k akciím doporučení analytiků (koupit, držet, prodat).

Všechny servery kromě *Reuters.com* nabízí na stránce firmy agregátor zpráv z jiných serverů. Z tabulky 2 lze vyčíst, že pro sledování akcií firem obchodovaných na burzách mimo USA lze použít pouze servery Reuters.com, Yahoo Finance a Google Finance. Z těchto tří serverů se jako nejlepší jeví Yahoo Finance, jelikož Reuters.com neobsahuje agregátor zpráv a Google Finance zase nenabízí doporučení analytiků.

Jelikož žádný ze zmíněných serverů neposkytuje API[2], je nutné získat data pomocí *web scrapingu*, což je proces získávání textu z webové stránky zpracováním HTML kódu dané stránky (Russell, 2013, s. 183). K tomuto účelu existuje pro každý „běžný“ programovací jazyk mnoho nástrojů (knihoven).

Další možností je použít API, která nabízejí komerční poskytovatelé finančních dat. Jedná se například o známý Bloomberg a jeho službu *Bloomberg Professional* (k té se přistupuje přes Windows aplikaci *Bloomberg Terminal*), která umožňuje přes *Bloomberg Open API* číst data z Bloomberg databáze – ta obsahuje mj. novinové články. Nevýhodou je vysoká cena tohoto řešení, která dle Wikipedia (2016a) činí 24 000 dolarů ročně pro jednoho uživatele.

[1] Jedná se o žebříček návštěvnosti webových stránek, ve kterém jsou stránky navzájem porovnávány podle hodnoty, počítané na základě odhadovaného průměrného počtu unikátních návštěvníků za 1 den a odhadovaného počtu zobrazení za poslední 3 měsíce (Alexa Internet, 2016).

[2] Application Programming Interface – rozhraní pro programový přístup k dané aplikaci

Tab. 2: Vybrané zpravodajské servery, zaměřující se na akciové trhy

Název	URL	ALEXA	Evropa	DA
MarketWatch	`http://www.marketwatch.com`	784	ne	ano
CNBC International	`http://www.cnbc.com`	704	ne	ne
Wall Street Journal	`http://www.wsj.com/news/markets`	384	ne	ano
Reuters.com	`http://www.reuters.com/finance/markets`	398	ano	ano
FINVIZ.com	`http://finviz.com`	4 857	ne	ano
Bloomberg Business	`http://www.bloomberg.com`	338	ne	ne
Yahoo Finance	`http://finance.yahoo.com`	5	ano	ano
CNNMoney	`http://money.cnn.com`	85	ne	ano
Google Finance	`https://www.google.com/finance`	1	ano	ne
MSN Money	`http://www.msn.com/en-us/money/markets`	15	ne	ne
Motley Fool	`http://www.fool.com`	1 224	ne	ano
Seeking Alpha	`http://seekingalpha.com`	1 402	ne	ne
Morningstar	`http://www.morningstar.com`	2 660	ne	ano

2.3.2 Sociální sítě

Tabulka 3 ukazuje počet aktivních uživatelů (v milionech, sloupec PU, k lednu 2016) největších sociálních sítí a komunikačních aplikací (pro leden 2016), oblast kde je daná služba převážně používána a jakého je služba typu[3].

První místo zaujímá s velkým náskokem Facebook, který používá již více než 1,5 mld. lidí. 2.–6. místo zaujímají komunikační aplikace. Na 7. místě se umístila blogovací platforma Tumlbr, následovaná aplikací pro sdílení fotografií Instagram a konečně další sociální sítí Twitter.

Jaké sociální sítě by tedy měly být zdrojem dat? Lze vidět, že velké množství služeb je zaměřeno na uživatele z Číny (resp. z Asie). Tyto rovnou vyřadíme, jelikož se budeme zaměřovat především na „západní“ akciové trhy. Dále nebudeme brát

[3] IM = Instant Messaging = „Online chat“ služba umožňují posílat textové zprávy (případně i audio, video nebo jiné soubory) mezi dvěma nebo více uživateli skrz internet.
Microblogging = Vystavování krátkých textových zpráv (příp. obrázků či videí) na veřejném profilu na internetu.
VoIP = Voice over IP = přenášení hlasu (telefonování) přes internet.

Tab. 3: Největší sociální sítě dle počtu uživatelů k lednu 2016

Č.	Název	PU	Oblast	Typ
1	Facebook	1 550	celý svět	sociální síť
2	WhatsApp	900	celý svět	IM
3	QQ	860	Asie	IM
4	Facebook Messenger	800	celý svět	IM
5	Qzone	653	Čína	sociální síť
6	WeChat	650	Čína	IM
7	Tumblr	555	celý svět	microblogging mediální
8	Instagram	400	celý svět	foto a video sdílení
9	Twitter	320	celý svět	microblogging textový
10	Baidu Tieba	300	Čína	komunikační platforma (fórum)
11	Skype	300	celý svět	video, voip, IM
12	Viber	249	celý svět	voip, IM
13	Sina Weibo	222	Čína	microblogging
14	LINE	212	celý svět	video, voip, IM
15	Snapchat	200	celý svět	foto a video sdílení, IM
16	YY	122	Čína	sociální síť zaměřená na video
17	VKontakte	100	Rusko a okolí	sociální síť
18	Pinterest	100	celý svět	foto sdílení, objevování
19	BlackBerry Messenger	100	celý svět	video, IM
20	LinkedIn	100	celý svět	sociální síť pro profesionály

Zdroj: (Statista, 2016)

v úvahu všechny IM služby, jelikož ty neposkytují veřejně dostupná a nám užitečná data. Pokud vyřadíme služby pouze pro sdílení fotek a videí, zbudou nám následující: Facebook, Tumblr, Twitter, LinkedIn a Google+[4]. Tyto služby budou prozkumány z hlediska toho, jaké možnosti nabízí jejich veřejné API a jaký obsah nabízejí.

Tabulka 4 zobrazuje ve sloupci PU počet aktivních uživatelů dané sociální sítě v milionech a další charakteristiky. Všechny služby poskytují API založené na REST a (kromě Tumblr) omezují počet požadavků, které je možné do API zaslat za jednotku času. LinkedIn poskytuje nějaký obsah pouze přes partnerský, zřejmě placený, program (LinkedIn, 2016). Tumblr nabízí velmi omezené možnosti vyhledávání (Tumblr, 2016) a většinu jeho obsahu tvoří „zábavné“ příspěvky. Google+ umožňuje globální vyhledávání (Google, 2016), ale jeho uživatelská základna není příliš široká. Z těchto důvodů budou v práci použity pouze služby Facebook a Twitter.

[4]Počet uživatelů byl získán z informací v Barrie (2015) a Enge (2015).

Tab. 4: Vybrané sociální sítě a jejich charakteristiky

Název	PU	Globální vyhledávání	Obsah dat	Neomezená historie
Facebook	1 550	ne	stránky, skupiny, místa aj.	ano
Tumblr	555	částečně (tagy)	údaje a příspěvky blogu, uživatelské údaje	ano blog, ne tagy
Twitter	320	ano	textové příspěvky	ne
LinkedIn	100	ne	nedostupný	?
Google+	50–100	ano	lidé, aktivity, komentáře	ano

2.3.3 Diskuzní fóra a jiné zdroje

Dalším místem, kde lidé vyjadřují názory na akcie, jsou diskuzní fóra nebo komunitní weby, zabývající se akciemi. I když budeme brát v potaz pouze veřejně dostupné informace (zdarma, bez potřeby registrace), je takových služeb obrovské množství. Proto jsou níže uvedeny pouze vybrané služby, kde je hlavním jazykem angličtina. Základem pro tabulku 5 byl článek (Stack Exchange, 2015).

Tab. 5: Diskuzní fóra a jiné zdroje pro hodnocení akcií

Název	URL	Funkce
The Motley Fool	`http://www.fool.com/`	Tipy na akcie, diskuzní fórum, masové hodnocení akcií
Yahoo message boards	`http://finance.yahoo.com/mb/ABT/`[5]	Diskuzní fórum
Wikinvest	`http://www.wikinvest.com/wiki/ABT`[6]	Důvody pro vzestup nebo pokles
InvestorPlace	`http://investorplace.com/`	Tipy na akcie
Seeking Alpha	`http://seekingalpha.com/`	Tipy na akcie, fórum pouze pro registrované
Profit.ly	`http://profit.ly/trades`	Sociální síť pro tradery

Dalším zdrojem může být známý diskuzní web *reddit*, který obsahuje 9 800 aktivních komunit a měsíčně ho navštíví přes 230 mil. lidí (reddit, 2016). Konkrétně jsou zde komunity Reddit Investing (`https://www.reddit.com/r/investing`) a Reddit StockMarket (`https://www.reddit.com/r/StockMarket/`).

[5]Příklad pro firmu Abbott Laboratories. Odkaz na fórum je dostupný ze stránky firmy: `http://finance.yahoo.com/q?s=ABT`, sloupec vlevo, oddíl *News & Info* a odkaz *Message Boards*.

[6]Příklad pro firmu Abbott Laboratories. Názory se skrývají pod odkazy *Bulls* a *Bears*.

2.3.4 Ekonomická data

V rámci práce bude nutné získat určitá ekonomická data – konkrétně data o akciovém trhu resp. o akciích jednotlivých firem. Nebudou nás zajímat fundamentální údaje, ale pouze metriky (vlastnosti) dané akcie – cena a zobchodovaný objem, příp. další. Nepotřebujeme aktuální data pro daný okamžik (tzv. *live feeds*), ale pouze historická data. Pro získání těchto dat lze využít jak placené, tak neplacené zdroje.

Tabulka 44 v příloze B ukazuje zdroje nalezené především v Caltech (2015) a lukebuehler (2013). Zdroje se liší v tom, kolik stojí pořízení dat, v jakých intervalech měří metriky – tj. denně (na konci dne – *end of day*) nebo během dne (*intraday* – každou minutu, sekundu nebo tick[7]), zda obsahují i firmy (symboly) vyřazené z burzy, do jak daleké minulosti sahají, zda upravují symboly dle toho jak se případně mění, akcie z jakých burz obsahují, zda zahrnují také akciové indexy a celkově jak kvalitně jsou data upravená. Důležité je, že všechny uvedené služby upravují ceny dle splitu (rozdělení akcií) a dividend. Poznámka: NYSE TAQ zpoplatňuje historická data částkou 500 dolarů za jeden měsíc.

Lze vidět, že pořídit kvalitní (obzvláště *intraday*) data není levné. Navíc jak uvádí lukebuehler (2013), získat dobrá data je obecně obtížné – dokonce tak, že „hedge fondy a banky utrácejí stovky tisíc dolarů měsíčně, aby získaly data, kterým mohou věřit“. Naštěstí existuje možnost stahovat *end of day* data zdarma z *Yahoo Finance*. Ačkoliv Caltech (2015) podotýká, že tato data relativně často obsahují chyby, lze tuto službu považovat za vhodnou, jelikož data dle lukebuehler (2013) nejspíš pocházejí od *CSI data*, což se dá považovat za solidní zdroj.

2.4 Uchovávání dat

Dále je nutné rozhodnout, kde a jak ukládat získaná data. Existují v zásadě dvě možnosti, jak data ukládat: v běžných souborech nebo v databázi.

Přístup založený na souborech

Tímto přístupem se myslí situace, kdy jsou data uložena jako prosté soubory (ať už binární či textové) a aplikace k těmto souborům přistupuje přímo (pomocí příkazů souborového systému) s cílem číst/zapisovat data.

Tento přístup má potenciální výhodu v tom, že pokud je aplikace, která ho používá, specificky zaměřená na jeden problém, může dosahovat vysokého výkonu. Další výhodou je, že není potřeba speciální software, který bude souborový systém spravovat. Naopak nevýhod je velké množství (Connolly a Begg, 2005, s. 12–14):

- Oddělení a izolace dat do jednotlivých souborů → duplikace dat.
- Závislost programu na datech → nekompatibilní souborové formáty.
- Uživatel nemůže sám sestavovat dotazy nad daty, musí to dělat vývojář.

[7]Minimální pohyb (nahoru nebo dolů) ceny akcie – pro akcie obchodované za více než 1 dolar to je (od roku 2001, kdy se zavedla decimalizace) 1 cent tj. 0,01 dolaru (Investopedia, 2016).

Databázový přístup

Pro odstranění limitů souborového přístupu musely být zajištěny dvě vlastnosti:

1. Definice dat je uložena odděleně a nezávisle od aplikačního programu.
2. Kontrola přístupu a manipulace s daty je zajištěna mimo aplikační program.

Databáze je úložiště logicky souvisejících dat, přičemž její součástí je také popis těchto dat (tzv. systémový katalog nebo datový slovník). To zapříčiňuje, že data jsou nezávislá na programu, který je využívá (Connolly a Begg, 2005, s. 14).

Přístup uživatelů (resp. programů) k databázi řídí DBMS, což je softwarový systém, který dle Connolly a Begg (2005, s. 16–17):

1. umožňuje uživatelům definovat datový slovník.
2. umožňuje uživatelům získávat/vkládat/aktualizovat/mazat data z databáze.
3. poskytuje kontrolovaný přístup k databázi: bezpečnost, integrita dat, souběžný přístup mnoha uživatelů, obnova po havárii, přístup k datovému slovníku.

2.4.1 Typy databázových systémů

Existuje mnoho různých typů DBMS. Často se dělí dle toho, jaký databázový (logický datový) model podporují (tedy do jakých datových struktur ukládají data).

První DBMS byl navrhnut za účelem organizace informací pro projekt Apollo (přistání na Měsíci), jmenoval se GUAM resp. IMS a používal **hiearchický** model (Connolly a Begg, 2005, s. 24). Zde je hlavní objekt rozložen na dílčí objekty, což je reprezentováno stromovou strukturou, ve které má jeden objekt nejvýše jednoho rodiče a neomezeně potomků. Jedná se tedy (dle teorie grafů) o kořenový strom (Silberschatz et al., 2010, Appendix E).

V 60. letech se objevil také systém IDS založený na **síťovém** modelu. Data jsou zde reprezentována kolekcemi záznamů a vztahy jsou reprezentovány vazbami mezi záznamy. Kolekce zde jsou obecné grafové struktury (nikoliv stromy jako u hierarchického modelu) (Silberschatz et al., 2010, Appendix D).

V roce 1970 publikoval E. F. Codd z IBM článek o **relačním** modelu. Tento model přišel ve správný čas a zanedlouho se stal de facto standardem pro komerční DBMS. V dnešní době existuje více než 100 relačních DBMS (Connolly a Begg, 2005, s. 25). Databáze je v relačním modelu tvořena relacemi – tabulkami s unikátním názvem, kde řádky odpovídají jednotlivým záznamům a sloupce představují atributy. Každý atribut má název a definovanou doménu – množinu povolených hodnot (Connolly a Begg, 2005, s. 72). Jedna tabulka (relace) představuje jednu entitu (jeden typ objektu), přičemž vztahy jsou reprezentovány speciálními atributy (cizími klíči).

Nicméně i relační model má nevýhody – především omezené modelovací schopnosti pro popis některých problémů reálného světa. Proto již od publikace původního článku byl prováděn výzkum s cílem tento problém vyřešit. Tyto pokusy se

dají nazvat snahou o sémantické modelování. V roce 1976 Chen prezentoval *Entity–Relationship* model, což je nyní široce používaná technika pro návrh databází. Sám Codd navrhl dva nové modely (RM/T a RM/V2). V návaznosti na vzrůstají komplexnost databázových aplikací vznikly **objektově-orientované** DBMS a **objektově-relační** DBMS (Connolly a Begg, 2005, s. 25).

S rozvojem internetu a nástupem Web 2.0 se zvětšoval objem dat generovaný uživateli a bylo obtížné s těmito daty pracovat v tradičních DBMS. Proto se v roce 2009 objevily tzv. **NoSQL** („not only sql“) databáze, což jsou DBMS nové generace s některými z následujcích vlastností: používají nerelační datový model, jsou distribuované, open-source a horizontálně škálovatelné. Jsou vhodné pro aplikace, vyžadující DBMS, který je bez schématu, nabízí snadnou podporu replikace a jednoduché API, umožňuje extrémně vysokou rychlost čtení/zápisu (obvykle výměnou za spolehlivost – nejsou ACID[8]) a ukládání obrovského množství dat (Edlich, 2009).

NoSQL DBMS je mnoho různých typů. Dle použitého datového modelu je lze rozdělit následovně: sloupcové, dokumentové (obvykle používají JSON formát; patří sem také historicky dříve se objevivší XML databáze), key-value, grafové (v podstatě se jedná o síťové databáze), multimodel (více modelů dohromady).

Jelikož pro uvažovou aplikaci určenou pro stahování a ukládání dat nejsou výše zmíněné charakteristiky NoSQL DBMS klíčové, zaměříme se dále na porovnání tradičních relačních DBMS.

2.4.2 Porovnání relačních DBMS

Jak bylo uvedeno v sekci 2.4.1, relačních DBMS (RDBMS) existuje velké množství (více než 100). Tabulka 6 ukazuje 15 nejpopulárnějších[9] k březnu 2016. Lze vidět, že více než polovina (přesněji 9) systémů je komerčních (placených). Tyto rovnou vyřadíme, jelikož pro zamýšlenou aplikaci plně dostačují open-source řešení. Zbývající systémy porovnává tabulka 7, přičemž jejich popularitu zobrazuje obrázek 2.

Je nutné podotknout, že se (kromě Hive) jedná o klasické RDBMS vhodné pro správu obecných dat. *Apache Hive* je naopak software pro datové sklady, který umožňuje prohledávat a spravovat velké balíky dat umístěných na distribuovaném úložišti. Je postavený na Apache Hadoop (Apache Software Foundation, 2014).

Specialitou *SQLite* je, že nepoužívá klient-server architekturu, ale proces aplikačního programu přistupuje skrz použitou knihovnu přímo k souborům na disku. Pro jeho zprovoznění tedy není potřeba žádná instalace ani konfigurace databázového serveru. Pro své vlastnosti je používán v širokém spektru programů a zařízení a jedná se tak o nejpoužívanější DBMS na světě (Hipp, 2016).

[8] Atomicity, Consistency, Isolation, Durability – vlastnosti transakce (sekvence příkazů) v DBMS nutné pro to, aby byla spolehlivě provedena (Connolly a Begg, 2005, s. 575–576).

[9] Popularita se určuje dle relativního skóre, které porovnává jednotlivé DBMS mezi sebou. Skóre je založeno na následujících faktorech: počet výsledků vyhledávání na frázi „<jméno DBMS> database“; frekvence hledání na Google Trends; frekvence technických diskuzí o DBMS; množství pracovních nabídek a profilů na LinkedIn, ve kterých je DBMS uveden; počet tweetů, ve kterých je DBMS zmíněn (solid IT, 2016a).

Tab. 6: Seznam 15 nejpopulárnějších RDBMS k březnu 2016

Pořadí	Název	Licence	03-2016	02-2016	03-2015
1	Oracle	komerční	1 472,01	-4,13	2,93
2	MySQL	open-source	1 347,71	26,59	86,62
3	Microsoft SQL Server	komerční	1 136,49	-13,73	-28,31
4	PostgreSQL	open-source	299,62	10,97	35,19
5	DB2	komerční	187,94	-6,55	-10,91
6	Microsoft Access	komerční	135,03	1,95	-6,66
7	SQLite	open-source	105,77	-1,01	4,06
8	SAP Adaptive Server	komerční	76,64	-3,39	-8,72
9	Teradata	komerční	74,07	0,69	1,29
10	Hive	open-source	50,51	-2,26	11,18
11	FileMaker	komerční	47,93	0,90	-4,41
12	SAP HANA	komerční	39,99	1,91	7,82
13	Informix	komerční	31,87	-1,15	-5,95
14	MariaDB	open-source	29,88	1,11	7,79
15	Firebird	open-source	20,88	0,76	-1,09

Zdroj: (solid IT, 2016b)

MariaDB vznikla jako klon MySQL, když tento DBMS převzal Oracle, aby bylo zajištěno, že bude navždy dostupná jako open-source. MariaDB je vyvíjena originálními autory MySQL a jedná se o vylepšenou a plně kompatibilní náhradu MySQL (MariaDB Foundation, 2016).

Jelikož je MySQL v popularitě s velkým náskokem na prvním místě (1 347,71 bodů), je na obrázku 2 rozmezí osy y sníženo na hodnotu 400, aby byly dobře pozorovatelné i hodnoty pro ostatní DBMS. Na druhém místě je se ztrátou více než 1 000 bodů (22 % hodnoty MySQL) PostgreSQL a na třetím místě je s rozdílem asi 200 bodů (35 % PostgreSQL) SQLite. Lze také vidět, že kromě Firebird se popularita všech uvedených DBMS oproti roku 2015 zvýšila.

Porovnání MySQL a PostgreSQL

Na základě výše uvedených informací lze pro použití v naší aplikaci uvažovat o MySQL (resp. MariaDB) a PostgreSQL. Tyto dva DBMS budou nyní stručně porovnány. Zdrojem informací pro toto porovnání byly následující webové stránky:

- `http://db-engines.com/en/system/MySQL%3BPostgreSQL`,
- `https://www.wikivs.com/wiki/MySQL_vs_PostgreSQL`,

Tab. 7: Charakteristiky vybraných open-source RDBMS

Pořadí	Název	Poslední aktualizace	Rok vzniku	Vývojář
2	MySQL	02-2016	1995	Oracle
4	PostgreSQL	02-2016	1989	PostgreSQL Global Development Group
7	SQLite	03-2016	2000	Dwayne Richard Hipp
10	Hive	02-2016	2012	Apache Software Foundation
14	MariaDB	02-2016	2009	MariaDB Foundation
15	Firebird	11-2015	2000	Firebird Foundation

- `https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems`,
- `http://www.brightball.com/postgresql/why-should-you-learn-postgresql`.

Oba DBMS podporují ACID, referenční integritu (cizí klíče), uzamykání řádků, Unicode kódování, replikaci, trigery, kurzory, funkce a procedury, instalaci rozšíření (pluginů). Oba DBMS také běží na široké škále operačních systémů (BSD, Linux, OS X, Solaris, Windows). Tabulka 42 v příloze B zobrazuje základní rozdíly mezi těmito DBMS. V případě MySQL se údaje vztahují k defaultnímu úložišti InnoDB.

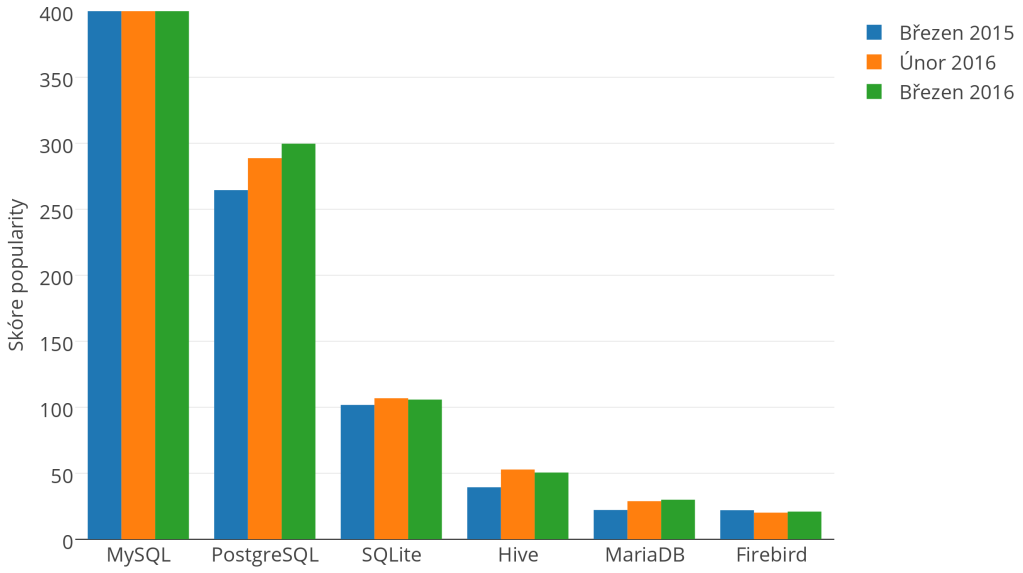
Ale nejdůležitější zřejmě je, jak se liší výkonnost (rychlost) těchto DBMS při čtení a zápisu dat. Zde v minulosti platilo, že MySQL je zaměřená na rychlost při jednoduchých operacích, zatímco PostgreSQL se zaměřuje na komplexní funkce a dodržování standardů. V dnešní době, pokud jsou správně nastaveny konfigurační parametry, je jejich výkonnost na podobné úrovni (WikiVS, 2016).

V práci se předpokládá ukládání velkého množství textových i jiných dat. Proto je kromě rychlosti také důležité, kolik místa na disku zaujímají uložená data. Zde musíme brát v potaz jak velikost samotných dat (která je určena použitými datovými typy a délkou dat), tak velikost indexů, nutných k rychlému vyhledávání v databázi.

MySQL (resp. jeho defaultní úložiště InnoDB) ukládá řádky tabulky na disk dle pořadí primárního klíče (PK). To má tři důsledky (Prelert, 2014):

1. Primární klíče nezabírají žádné místo: Jelikož jsou data uspořádána podle PK, není potřeba samostatný indexový soubor.
2. Potenciálně pomalejší vkládání: Pokud nejsou data vkládána dle pořadí PK, může vkládání trvat déle, protože se musí data na disku přeuspořádat.
3. Rychlejší čtení rozmezí dat: Pokud jsou vybrána data z určitého rozmezí dle PK, je diskový přístup sekvenční.

Naproti tomu PostgreSQL automaticky nesoustřeďuje data na disku dle primár-



Obr. 2: Popularita vybraných open-source RDBMS v letech 2015–2016

ního klíče. Místo toho je pro primární klíč vytvořen samostatný indexový soubor, který se během zpracování dotazu používá k určení toho, kde v hlavním datovém souboru je daný řádek uložen (Prelert, 2014).

Prelert (2014) provedl experiment s datovou sadou tvořenou jednou tabulkou se 4 číselnými sloupci, z nichž dva tvořily PK. Tabulka obsahovala asi 732 mil. řádků, naplněných čísly. Výsledek byl jasný: Velikost souboru tabulky byla v MySQL 57 GB, zatímco v PostgreSQL to bylo 73 GB pro tabulku a 46 GB pro primární klíč – tedy celkem 119 GB (dvakrát tolik). Toto zjištění potvrzuje i Kumar (2012), když uvádí, že jeho databáze měla v MySQL (zde bylo použito úložiště MyISAM) velikost 13 GB oproti 36 GB v PostgreSQL (konkrétně se jednalo o denní přírůstek).

Jak již bylo řečeno, předpokládáme, že v rámci práce bude ukládáno velké množství dat. Půjde jak o textová data, tak o číselná data – ekonomické údaje a statistické údaje o uložených textech. Tyto statistiky budou sledovány také v průběhu času, což způsobí vysoký nárůst množství dat v dané tabulce. Jelikož nemáme k dispozici výkonný server s rozsáhlým diskovým prostorem, je otázka velikosti dat uložených na disku poměrně zásadní. Z tohoto důvodu se jeví jako lepší varianta zvolit pro ukládání dat MySQL. Pro použití tohoto DBMS mluví také vyšší rychlost při sekvenčním čtení záznamů, které bude jistě využito při následném zpracování a analýze dat. A v neposlední řadě má autor práce s tímto DBMS (na rozdíl od PostgreSQL) praktické zkušenosti, což spolu s jeho vyšší popularitou (a tedy vyšším počtem návodů a diskuzí na internetu) přispěje k tomu, že případná konfigurace resp. vyřešení problémů budou snažší.

2.5 Dolování znalostí z dat

V této kapitole je popsán současný stav problematiky, týkající se dolování znalostí z dat obecně (data mining) resp. z textových dokumentů (text mining).

2.5.1 Data mining

V rámci práce je nutné analyzovat textová data. K tomu lze využít mj. metody pro dolování znalostí z *jakýchkoliv* dat. Nejdříve vysvětlíme pojmy, které zde budou často používány: data, informace, znalosti. Tyto pojmy mají (z hlediska algoritmů strojového učení) následující hierarchický vztah (Žižka, 2009):

1. Data + šum – Reálný svět obsahuje obrovské množství dat a okolního šumu.
2. Data – Odfiltrujeme šum a zbudou nám data (signály, čísla, text).
3. Informace – data, kterým přiřadíme určitý význam. Vybereme část dat, která jsou relevantní k řešení určitého problému.
4. Znalost – zobecnění informace.
5. Meta znalost – znalost o znalosti (říká nám, jak využít znalost).

Data mining je „proces objevování zajímavých vzorů a znalostí ve velkém množství dat“. Ačkoliv se v podstatě jedná se o jeden z kroků v KDD[10] (kterému předchází příprava dat a následuje po něm hodnocení nalezených vzorů a reprezentace získaných znalostí), který má za cíl nalézt vzory v datech, je tento pojem s KDD často zaměňován a proto platí výše uvedená definice (Han et al., 2012, s. 8).

Existují dva základní typy úloh pro data mining:

- Deskriptivní – cílem je charakterizovat vlastnosti zdrojových dat (popsat vzory v datech).
- Prediktivní – cílem je pomocí indukce na zdrojových datech (na základě nalezených vzorů v datech) provést predikci pro neznámá data.

Data mining systém je schopen nalézt obrovské množství vzorů, ale jen malá část z nich je něčím zajímavá (užitečná). Vzor je zajímavý, pokud jej může snadno pochopit člověk, je validní na nových (testovacích) datech, je potenciálně užitečný a něčím nový nebo neobvyklý. Případně pokud validuje hypotézu, kterou chtěl uživatel systému potvrdit (Han et al., 2012, s. 21).

Úlohy data mingu jsou řešeny pomocí funkcí data miningu, které se liší dle toho, jaké typy vzorů jsou schopny odhalit (Han et al., 2012, s. 15–21).

- Popis třídy/konceptu: *Data characterization* nebo *Data discrimination*.
- Časté vzory, asociace, korelace: výstupem jsou asociační pravidla.

[10] Knowledge Discovery from Data – objevování znalostí v datech

- Predikce: klasifikace (pro diskrétní hodnoty), regrese (pro spojité hodnoty).
- Shluková analýza: seskupování podobných objektů do skupin.
- Analýza odlehlých objektů.

Pro data mining se používají metody z různých oblastí: umělá inteligence, strojové učení, statistika, datové sklady a databázovové systémy, rozpoznávání vzorů, vizualizace, *information retrieval* aj. (Han et al., 2012, s. 23).

Strojové učení

Strojové učení je základní metodou pro data mining. Zabývá se tím, jak se mohou počítače učit na základě dat. Cílem je tedy vytvořit počítačový program, který se bude automaticky učit a rozpoznávat komplexní vzory v datech. Existují tři základní způsoby, jak se algoritmy učí. Liší se v tom, zda používají ohodnocená (označkovaná) data – tedy data, pro která je známa jejich příslušnost k určité třídě.

- Učení s učitelem (*Supervised learning*) – klasifikace – Základem jsou ohodnocená trénovací data, na kterých se algoritmus natrénuje a následně je schopen určovat třídu pro neznámá data.
- Učení bez učitele (*Unsupervised learning*) – shluková analýza – Trénovací data nejsou ohodnocená. Algoritmus nalezne v datech nějakou zajímavou strukturu resp. data rozdělí do tříd.
- *Semi-supervised learning* – Cílem je klasifikace, přičemž jako trénovací data jsou použita ohodnocená i neohodnocená data (těch je více). Základní přístup spočívá v tom, že ohodnocené příklady jsou použity k vytvoření modelů tříd a neohodnocené příklady ke zpřesnění hranice mezi třídami (Han et al., 2012, s. 24–25). Dalším způsobem je *co-training*, kdy můžeme data zkoumat z více nezávislých pohledů – na základě ohodnocených dat vytvoříme modely a pomocí nich zpracujeme neohodnocená data (Witten et al., 2011, s. 296).

Mezi další způsoby učení patří:

- Aktivní učení (*Active learning*), kdy se člověk aktivně účástní učícího procesu (Han et al., 2012, s. 25).
- Zpětnovazebné učení (*Reinforcement learning*), kde je cílem programu naučit se, jak reagovat na nastalé situace akcemi tak, aby maximalizoval dosaženou odměnu (Sutton a Barto, 1998, s. 4).

2.5.2 Text mining

Data mining může být aplikován na jakákoliv data. Základními typy dat jsou data v (relační) databázi, data v datovém skladu a transakční data. Tato data jsou dobře strukturovaná a pro jejich zpracování postačují základní metody (resp. algoritmy) pro data mining. Pro zpracování jiných typů dat (proudová, sekvenční, síťová, prostorová, multimediální aj.) je nutné použít speciální metody nebo upravit data tak, aby je bylo možné zpracovat základními metodami (Han et al., 2012, s. 8).

V práci budeme zkoumat textová data – hledat v nich skryté vzory a souvislosti s ekonomickými daty. K tomu lze použít text mining – proces obdobný data miningu, jehož cílem je získat užitečné informace (resp. znalosti) nalezením zajímavých vzorů v textových datech. Tato data jsou ovšem nestrukturovaná – tvoří je kolekce dokumentů, psaných přirozeným jazykem. Data je tedy nutné upravit do lépe strukturované podoby a následně zpracovat pomocí metod data miningu, strojového učení či statistiky. Text mining také využívá poznatků oborů, zaměřujících se na zpracování přirozeného jazyka (*NLP – Natural language processing*): *Information retrieval/extraction* a *Computational linguistics* [11] (Feldman a Sanger, 2007, s. 1).

Text mining lze využít pro tyto základní úlohy:

- **Klasifikace dokumentů** (*Document classification*): Cílem je zařadit určitý dokument do jedné z předem definovaných kategorií (tříd). Typickým příkladem je rozhodnutí, zda příchozí e-mail je či není SPAM. Lze kombinovat textová a numerická data – např. tak, že třídu pro dokumenty v trénovací množině určíme na základě nějakého externího souvisejícího údaje (venkovní teplota, pohyb ceny akcie) (Weiss et al., 2010, s. 6).
  Formálně lze tento problém zapsat následovně (Hotho et al., 2005): Nechť $D = (d_1, \ldots, d_n)$ je trénovací množina dokumentů, které již mají přiřazenou třídu $L \in \mathbb{L}$. Úkolem je definovat klasifikační model (funkci)

$$f : D \to \mathbb{L} \qquad f(d) = L\,, \tag{1}$$

  který je schopen novému (neznámému) dokumentu d přiřadit správnou třídu L.

- **Shlukování dokumentů** (*Document clustering*): Cílem je rozdělit kolekci dokumentů do (předem neznámých) tříd tak, aby každá obsahovala podobné dokumenty. Příkladem může být zjištění typů problémů, se kterými se zákaznící často obracejí na kontaktní centrum firmy (Weiss et al., 2010, s. 8).

- **Vyhledávání informací** (*Information Retrieval*): Cílem je na základě zadaného dotazu najít relevantní (dotazu nejlépe odpovídající) dokumenty. Dotazem mohou být jak jednotlivá slova, tak kompletní dokument. Tento postup je využit ve vyhledávačích (Weiss et al., 2010, s. 7).

[11] Počítačová lingvistika – disciplína na pomezí lingvistiky (věda zkoumající přirozený jazyk) a matematické informatiky (*computer science*), zabývající se výpočetními aspekty lidského (přirozeného) jazyka (Uszkoreit, 2000).

- **Extrakce informací** (*Information Extraction*): Cílem je převést nestrukturovaný dokument do strukturované formy. To lze udělat nalezením a vyplněním určitých hodnot předem definovaných atributů. Jako příklad můžeme použít výroční zprávu firmy, z které chceme zjistit ekonomické ukazatele dané firmy (např. zisk, tržby, počet zaměstnanců) (Weiss et al., 2010, s. 9).
  Dalším způsobem může být nalézt entity, které se v dokumentu vyskytují a jejich vlastnosti, vztahy mezi nimi nebo události, kterých se účastní. První krok se nazývá *Named Entity Recognition* (Feldman a Sanger, 2007, s. 96).
- **Síťová analýza** (*Link Analysis*): Výstupy předešlých metod poskytují informace k sestavení grafového modelu vztahů mezi entitami, identifikovanými v dokumentech. Výsledný, vhodně uspořádaný, vizuální model je poté možné využít k nalezení nových vzorů v datech (Feldman a Sanger, 2007, s. 242).
- **Analýza sentimentu** (*Sentiment analysis*): Podrobně popsána v sekci 2.6.
- **Automatická sumarizace** (*Automatic summarization*): Cílem je zredukovat rozsáhlý dokument (vytvořit souhrn) a zachovat jeho hlavní význam. Extraktivní metody vytvoří souhrn vybráním slov, frází a vět z původního dokumentu. Abstraktivní metody vytvoří interní sémantickou reprezentaci a potom pomocí technik pro generování přirozeného jazyka vytvoří požadovaný souhrn. Existují dva typy extraktivní sumarizace: Prvním je *keyphrase extraction*, jejímž cílem je zvolit určitá slova nebo fráze pro označení dokumentu. Druhým je *document summarization*, která ma za úkol vybrat z dokumentu věty a vytvořit z nich odstavec, vyjadřující hlavní význam původního dokumentu (Goldberg, 2007, s. 1).

Vektorová reprezentace dokumentů

Algoritmy strojového učení neumí zpracovávat dokumenty v textové formě. Text dokumentu je nutné převést do číselné reprezentace, určující jeho vlastnosti. Dokument je obvykle popsán vektorem vlastností (*feature vector*). Nejčastější je tzv. *bag-of-words* model, který jako vlastnosti dokumentu používá všechny tokeny (slova) či n-tice tokenů (obojímu se říká termy), které se vyskytují ve zkoumané kolekci dokumentů. Počet dimenzí prostoru vlastností (tedy velikost vektoru) je tedy roven počtu unikátních termů (ty tvoří slovník kolekce) (Feldman a Sanger, 2007, s. 68).

Kolekci dokumentů si lze představit jako tabulku, kde řádky představují jednotlivé dokumenty a sloupce termy. Dokument (řádek) je popsán vektorem, jehož každý prvek určuje, zda se daný term v dokumentu vyskytuje nebo ne (resp. má přiřazenou váhu 0 nebo jinou). Tento model tedy nebere v úvahu pořadí termů v dokumentu.

Z výše uvedené definice je zřejmé, že drtivá většina prvků vektorů bude obsahovat nuly, protože jeden dokument obsahuje velmi málo termů ze slovníku kolekce. Tato reprezentace je tedy paměťově velmi nehospodárná. Lepší je reprezentace ve formě řídkých vektorů (*sparse vectors*), kde je dokument popsán sekvencí uspořádaných dvojic, jejichž první prvek značí číslo termu a druhý jeho (nenulovou) váhu.

Formálně lze tento vektorový model zapsat následovně (Hotho et al., 2005): Nechť D je množina dokumentů a $T = \{t_1, \ldots, t_m\}$ slovník (množina všech termů, které se vyskytují v dokumentech z D). Pak má každý term t v dokumentu d danou váhu w, která je spočítána zvolenou funkcí f. Dokument d_x je tedy tvořen vektorem $(w_{x1}, \ldots, w_{xm})$, který pro každý term t_i obsahuje jeho váhu w_{xi}.

Každý prvek vektoru má tedy svou váhu, která určuje jeho důležitost. Váhy jsou lokální (v daném dokumentu) nebo globální (v celé kolekci), případně se jedná o kombinaci obou hodnot. Váhy lze určovat (definice funkce f) těmito základními způsoby (Weiss et al., 2010, s. 21–26):

- Přítomnost termu (*Term Presence – TP*) – lokální – váha je 1, pokud je daný term přítomen v dokumentu, jinak je 0.
- Frekvence termu (*Term Frequency – TF*) – lokální – váha značí, kolikrát se daný term vyskytuje v dokumentu.
- IDF (*Inverse Document Frequency*) – globální – váha určuje, jak je term v kolekci „vzácný“ a tedy důležitý (kolik informace poskytuje).

  $$\mathrm{IDF}(t, D) = log\left(\frac{N}{\mathrm{df}(t, D)}\right), \tag{2}$$

  kde N je počet všech dokumentů v kolekci D (tedy $|D|$) a $\mathrm{df}(t, D)$ je počet dokumentů z kolekce D, obsahujících term t.
- TF-IDF (*Term Frequency-Inverse Document Frequency*) – kombinace – frekvence termu t v dokumentu d je upravena faktorem, který určuje důležitost termu v kolekci D.

  $$\mathrm{TFIDF}(t, d, D) = \mathrm{TF}(t, d) \cdot \mathrm{IDF}(t, D) \tag{3}$$

  Použití IDF způsobí, že pokud se term objevuje v mnoho dokumentech (je běžný), váha termu v dokumentu se výrazně sníží, a naopak pokud je term vzácný (objevuje se v málo dokumentech), váha termu se výrazně zvýší.

Jaký způsob výpočtu váhy použít závisí především na algoritmu, který má s výslednou reprezentací pracovat. Nicméně nejčastěji se používá metoda TF-IDF.

Pokud se délka dokumentů v kolekci značně liší, může při výpočtu TF u nadprůměrně dlouhého dokumentu dojít k tomu, že bude hodnota TF určitého termu vzhledem k ostatním dokumentům absolutně vysoká. Tomu lze předejít provedením normalizace, která tuto hodnotu převede na relativní. Existují různé druhy normalizace (suma výskytů všech termů v dokumentu, kosinová).

Vytvořená reprezentace kolekce dokumentů obsahuje obrovský počet dimenzí – pro velké kolekce dokumentů to mohou být tisíce či sta tisíce. To může způsobit, že některé algoritmy strojového učení budou data zpracovávat neprakticky dlouhou dobu. Většina těchto dimenzí (termů) je ovšem pro daný úkol (např. klasifikaci) nepodstatná a může být odstraněna. Někdy se tím dokonce i zlepší výsledky algoritmu

z důvodu snížení šumu. Odstranění irelevantních termů se nazývá *Feature selection* – výběr vlastností (Feldman a Sanger, 2007, s. 68).

Metody Feature selection

Odstranění irelevantních termů lze docílit předzpracováním zdrojových dokumentů. Budou uvedeny tři typy metod.

První typ se zabývá frekvencemi jednotlivých termů v kolekci dokumentů.

- Odstranění slov, které mají nízkou *document frequency* (vyskytují se v málo dokumentech): Experimenty ukazují, že použití pouze 10 % nejčastějších termů nesnižuje úspěšnost klasifikátorů (Feldman a Sanger, 2007, s. 69).
- Odstranění *stop slov* (resp. termů): Jsou to slova, která mají v daném jazyce určitou funkci (resp. se v něm vyskytují velmi často) a nemají žádný význam z hlediska sémantiky dokumentů. Například pro angličtinu to jsou slova *the, a, he, has, is* apod. (Feldman a Sanger, 2007, s. 68)

Druhý typ upravuje slova tak, aby se ve slovníku vyskytovalo co nejméně, z hlediska poskytnuté informace, v podstatě stejných slov.

- Lematizace (*Lemmatisation*): Převedení slov do základního tvaru, nezávislého na skloňování nebo časování. Pro to je nutné určit slovní druh daného slova – pomocí metody *Part-of-speech* (POS) *tagging*. (Weiss et al., 2010, s. 19).
- Stematizace (*Stemming*): Převedení slov na jejich kmen odstraněním předpon, přípon a morfologických koncovek (Weiss et al., 2010, s. 20).
- *Case folding*: Převod všech slov v dokumentu na malá nebo velká písmena.
- Kontrola pravopisu: Detekce a opravení překlepů a gramatických chyb v dokumentu.

Třetí typ metod produkuje metriky relevantnosti termů, které berou v úvahu vztahy mezi termy a třídami dokumentů.

- *Information gain* (IG): Měří počet bitů, získaných pro predikci tříd tím, že víme, zda se v dokumentu nachází nebo nenachází daný term.
- *Chi-square* (CHI): Měří maximální sílu závislosti mezi termem a třídami.

Experimenty ukazují, že tyto metody mohou zredukovat počet dimenzí až 100× bez snížení úspěšnosti klasifikace či dokonce i s jejím mírným zvýšením (Feldman a Sanger, 2007, s. 69). Krupník (2014) tyto metody použil pro nalezení a odstranění stop slov. Výsledky ukázaly, že správnost zůstala stejná či se mírně zvýšila, ale významným efektem byla kratší (až o 30 %) doba konstrukce modelu (pro algoritmus C5.0). Práce na stranách 18–19 popisuje dvě výše zmíněné (a další) metody a její součástí je také skript, schopný vygenerovat seznam nejdůležitějších resp. nejméně důležitých slov.

2.6 Analýza sentimentu

Analýza sentimentu (*sentiment analysis*, *opinion mining*) je obor studia, jehož cílem ja „analyzovat názory, nálady, hodnocení, postoje a emoce lidí vůči určitým entitám (produkty, služby, firmy, události, témata) a jejich vlastnostem“ (Liu, 2012, s. 7). Pro všechny výše uvedené pojmy budeme používat slovo názor (mínění – sentiment).

2.6.1 Základní dělení a pojmy

Obecně existují tři základní úrovně (typy) analýzy sentimentu (Liu, 2012, s. 10–12):

1. Úroveň dokumentu (*Document-level sentiment classification*): Úkolem je určit, zda celý dokument vyjadřuje pozitivní nebo negativní názor. Zde předpokládáme, že každý dokument obsahuje názor na jednu entitu. Z toho plyne, že tuto analýzu nelze použít pro dokumenty, hodnotící více entit.

2. Úroveň věty (*Sentence-level*): Úkolem je určit, zda daná věta vyjadřuje pozitivní, negativní nebo neutrální (žádný) názor. Tato úroveň analýzy je úzce spjata s klasifikací subjektivity (*subjectivity classification*), která rozlišuje objektivní (vyjadřující faktické informace) a subjektivní věty (vyjadřující subjektivní mínění). I objektivní věta může vyjadřovat názor a naopak subjektivní věta nemusí.

3. Úroveň entit a aspektů (*Entity and Aspect level*): Úkolem je určit, co konkrétně se autorovi dokumentu líbí či nelíbí. Základem je myšlenka, že názor se skládá ze sentimentu (pozitivního nebo negativního) a cíle. Cíle jsou obvykle představovány entitami a jejich různými aspekty (vlastnostmi). Úkolem je objevit mínění ohledně nalezených entit a jejich aspektů, z čehož lze poté vytvořit shrnutí.

   Názor lze formálně definovat jako pětici

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l) \,, \tag{4}$$

kde e_i je název entity, a_{ij} je aspekt, s_{ijkl} je sentiment o aspektu a_{ij} entity e_i, h je původce názoru a t je čas, kdy byl názor vyjádřen. Sentiment s_{ijkl} je pozitivní, negativní nebo neutrální (případně vyjádřený na nějaké škále) (Liu, 2012, s. 18–19).

2.6.2 Klasifikace dokumentů na základě sentimentu

V rámci analýzy sentimentu je zřejmě nejčastěji studována klasifikace sentimentu na úrovni dokumentů. Jejím cílem je určit, jaký má daný dokument celkový sentiment (jaký názor vyjadřuje). Obvykle se používá binární škála (pozitivní nebo negativní), nicméně existuje mnoho prací, kde se používá i třetí, neutrální třída – např. dále zmíněný článek (Hutto a Gilbert, 2014b). Lze také určovat číselnou hodnotu sentimentu z daného rozpětí (např. 1–5), potom se ale jedné o problém regrese. Vždy předpokládáme, že dokument d vyjadřuje názory na jedinou entitu e a tyto názory jsou od jediného autora názoru h. Čas nebereme v potaz. Tudíž nás z pětice ve vzorci

4 zajímá pouze proměnná s, určující celkový sentiment (aspektu GENERAL), jejíž hodnotu máme zjistit (Liu, 2012, s. 30).

Existují dva základní přístupy k analýze sentimentu: přístup založený na slovnících a přístup založený na strojovém učení (s učitelem). Případně lze využít kombinaci obou přístupů (např. pro vygenerování slovníku pro danou doménu). Tyto přístupy jsou v další části textu stručně představeny.

2.6.3 Slovníky sentimentu

Nejdůležitějšími indikátory sentimentu jsou tzv. názorová slova, což jsou slova, která jsou často používána k vyjádření pozitivního nebo negativního sentimentu. Také sem patří různé fráze a idiomy[12]. Seznam takových slov a frází se nazývá slovník sentimentu (*sentiment lexicon/dictionary*).

Existuje velké množství slovníků sentimentu, které se liší svými vlastnostmi. Tabulka 8 zobrazuje všechny slovníky, které byly autorem práce nalezeny. Odkazy na umístění daných slovníků lze nalézt v tabulce 43 v příloze B. Sloupec „Počet termů“ znamená počet unikátních slov nebo frází ve slovníku. Sloupec „Váhy“ říká, zda mají termy přiřazené dvě (ne – pozitivní nebo negativní tj. *polarity*) nebo více (ano – např. interval $[-1...1]$ – *valence*) hodnot sentimentu. Sloupec „Stemmed“ značí, zda slovník obsahuje také slova v základním tvaru (jsou nějak označená). Sloupec „N-gramy“ říká, zda jsou ve slovníku pouze jednotlivá slova (unigramy) nebo i víceslovné fráze (2,3,...,n-gramy).

Některé z uvedených slovníků podrobně popisuje Hutto a Gilbert (2014b) v kapitole 2.1. Jaký slovník použít záleží především na doméně (oblasti), ze které pochází texty, které budeme analyzovat. Je nutné si uvědomit, že pokud použijeme obecně zaměřený slovník, vždy jím zjišťujeme explicitní (subjektivní) sentiment. Pokud chceme (alespoň zčásti) zjišťovat i implicitní (objektivní) sentiment, musíme pro danou doménu vytvořit zvláštní slovník.

Pro novinové články se jako užitečné jeví finančně zaměřené slovníky. Zde se dá za nejlepší označit *Loughran and McDonald Financial sentiment dictionary*. Pro Facebook komentáře a tweety bude vhodnější obecný slovník jako *WordStat sentiment dictionary* nebo slovník od profesora *Bing Liu*.

Některé práce, jako např. (Arias et al., 2013), použily pro určení sentimentu dokumentů fakt, zda a jaká emotikona[14] byla v dokumentu (obvykle tweetu) přítomna. Nedávno byl dokonce vytvořen lexikon sentimentu pro emotikony – *Emoji Sentiment Ranking*[15] (Novak et al., 2015).

[12] Ustálené slovní spojení typické jen pro určitý jazyk (Mikuláš, 2007, s. 219).

[13] Slova jsou rozdělena do dvou kategorií – slabě a silně subjektivní.

[14] Jedná se v podstatě o zkratku pro výraz obličeje – vyjadřuje autorovy pocity, nálady a emoce (Novak et al., 2015).

[15] `http://kt.ijs.si/data/Emoji_sentiment_ranking/`

Tab. 8: Přehled slovníků sentimentu

Název	Typ	Počet termů	Váhy	Stemmed	N-gramy
AFINN	obecný	2 477	ano	ne	ano
Bing Liu opinion lexicon	obecný	6 786	ne	ne	ne
Hajek, Myskova – Novel multi-word lists	finanční	256	ne	ne	ano
Henry word list	finanční	189	ne	ne	ne
LabMT	obecný	10 222	ano	ne	no
Lexicoder Sentiment Dictionaryten	politický	4 566	ne	ano	ano
Loughran and McDonald Financial sentiment dictionary	finanční	2 709	ne	ne	ne
MICRO-WNOP corpus	obecný	1 105	ano	ne	ano
OpinionFinder's Subjectivity Lexicon	obecný	6 886	částečně[13]	ano	ne
SenticNet 1.0	obecný	5 726	ano	ne	ano
SentiWordNet 3.0	obecný	147 306	ano	ne	ano
VADER	obecný	7 502	ano	ne	ne
Warriner affective ratings	emoční	13 915	ano	ne	ano
WordStat sentiment dictionary 1.2	obecný	15 074	ne	ano	ne

2.6.4 Určování sentimentu pomocí slovníků

V předchozí kapitole byly popsány slovníky sentimentu, což jsou základní stavební kameny pro analýzu sentimentu založenou na slovnících. Existují různé (méně či více pokročilé) algoritmy, jak pomocí slovníku určit sentiment daného textu.

Základní algoritmus a jeho nevýhody

Pro základní analýzu na úrovni celého dokumentu lze využít jednoduchý algoritmus, který popisuje např. Petrovský (2015). Algoritmus získá četnosti všech slov v dokumentu, ty vynásobí odpovídající hodnotou sentimentu ze slovníku a nakonec vše sečte. Pokud je součet kladný, je sentiment pozitivní, jinak negativní.

Algoritmus lze rozšířit o normalizaci celkové hodnoty ts tak, aby nebyla závislá na délce dokumentu a ležela vždy (pokud možno) v určeném intervalu (např. $[-1, +1]$). Provedeme to vydělením hodnoty ts určitou hodnotou: $n, \sqrt{n}$ nebo např.

$ts_{norm} = ts/\sqrt{n^2 + \alpha}$, kde n je počet slov v dokumentu a α představuje maximální očekávanou hodnotu ts (Hutto a Gilbert, 2014a).

Tento algoritmus tedy bere dokument jako množinu nezávislých slov (*bag-of--words*). To má ovšem řadu nevýhod, plynoucích z toho, že nezkoumáme sekvenci slov, tvořících dokument. Tím pádem nebereme v úvahu intenzitu (stupňovací příslovce, velikost písmen, interpunkci) nebo fráze a idiomy. Zřejmě nejzávažnější je ale problém negace. K jeho řešení existují dva přístupy.

První přístup stále používá model *bag-of-words*, ale při čtení dokumentu sleduje, zda je před daným slovem umístěný negovací výraz (tzv. *valence shifter*). Pokud tomu tak je, tak dané slovo změní na „NOT“ verzi, umístí je do slovníku a přiřadí mu opačný sentiment. Příklad: *This car is not good.* → *This car is not NOT_good.* Je tedy nutné mít k dispozici slovník negovacích výrazů. Existuje také komplexnější přístup, beroucí v potaz slovní druhy (POS) (Arias et al., 2013).

Druhý přístup vyžaduje, aby byla dostupná původní struktura dokumentu, kterou lze dále analyzovat. Uvedeme příklad pro měření negativního sentimentu (analogicky se postupuje pro pozitivní). Za negativní považujeme jak negativní slova, kterým nepředchází negace (v rámci 3 slov v dané větě), tak pozitivní slova, kterým naopak negace předchází (v rámci 3 slov v dané větě) (Provalis Research, 2016).

Pro všechny uvedené problémy existuje řada postupů a algoritmů – např. VADER (Hutto a Gilbert, 2014b) nebo SO-CAL (Taboada et al., 2011). V následující části bude představen první z nich, jelikož související článek je o 3 roky novější, dle všeho je použitý algoritmus pokročilejší a veřejně přístupný. Ještě byl nalezen algoritmus v Kaushik a Mishra (2014), ale daný článek nezvzbuzuje velkou důvěru.

Algoritmus VADER

Hutto a Gilbert (2014b) vytvořili algoritmus VADER (*Valence Aware Dictionary and sEntiment Reasoner*), který implementuje jednoduchý, na pravidlech založený model pro analýzu sentimentu na úrovni vět. Model je zaměřený na analýzu textů na sociálních médiích (tweety), ale je dostatečně obecný, takže (po případné výměně použitého slovníku) lze použít pro jakoukoliv oblast.

V rámci článku bylo provedeno následující:

- Vytvořen slovník sentimentu, obsahující 7 500 slov, emotikon, akronymů[16] (např. LOL, WTF) a slangových internetovových výrazů (např. „nah“, „meh“).
- Identifikovány gramatické a syntaktické heuristiky, pomocí kterých lidé vnímají intenzitu sentimentu a byl ohodnocen jejich dopad na sentiment textu.
- Vytvořeny 4 *ground truth* korpusy (tweety, recenze filmů/produktů, editorialy).

Z provedených experimentů vyplývá, že VADER lexikon je jednoznačně nejlepší mezi všemi lexikony a klasifikuje dokonce lépe než lidé. Algoritmus VADER ve srovnání s algoritmy strojového učení poskytuje (mimo recenze filmů) lepší výsledek (dle *F1 score*), přičemž je daleko rychlejší a lze jej použít v široké škále domén.

[16]Slovo utvořené z počátečních písmen několika slov (Mikuláš, 2007, s. 20).

2.6.5 Určování sentimentu pomocí učení s učitelem

Klasifikace sentimentu na úrovni dokumentu má za cíl určit, zda daný dokument vyjadřuje pozitivní nebo negativní názor. Jedná se v podstatě o klasický problém textové klasifikace – v té se obvykle zjišťuje, o jakém tématu dokument pojednává, přičemž klíčové prvky jsou tématická slova. V klasifikaci sentimentu jsou důležitější tzv. názorová slova, indikující pozitivní nebo negativní sentiment.

Lze použít jakýkoliv algoritmus učení s učitelem. Je ale nutné vybrat vhodnou množinu efektivních příznaků. První výzkumy používaly jako příznaky jednoduše slova (unigramy – model *bag-of-words*). Ukázalo se, že tento přístup funguje poměrně dobře (pro algoritmy SVM a Naive Bayes). V dalších výzkumech byly použity také jiné příznaky: termy a jejich prekvence, slovní druhy, názorová slova a fráze, pravidla názorů, posunovače sentimentu, syntaktické závislosti (Liu, 2012, s. 31–32).

Přehled *supervised* algoritmů

Tabulka 45 v příloze B zobrazuje přehled výsledků *supervised* algoritmů strojového učení, používaných pro určování sentimentu. Všechny výsledky platí pro příznaky ve formě unigramů a binární klasifikaci. U většiny publikací není uveden způsob vážení termů a použitá normalizace u vektorů. Pang et al. (2002) uvádí pro algoritmus SVM jako nejlepší způsob vážení TP (přítomnost termu), jelikož podává lepší výsledek než TF (frekvence termu). Dále používá délkovou normalizaci a provedl úpravu negace (NOT_slovo). Maas et al. (2011) používá kosinovou normalizaci, vážení TP a *smoothed delta idf*, které je blíže popsané v Pak et al. (2014).

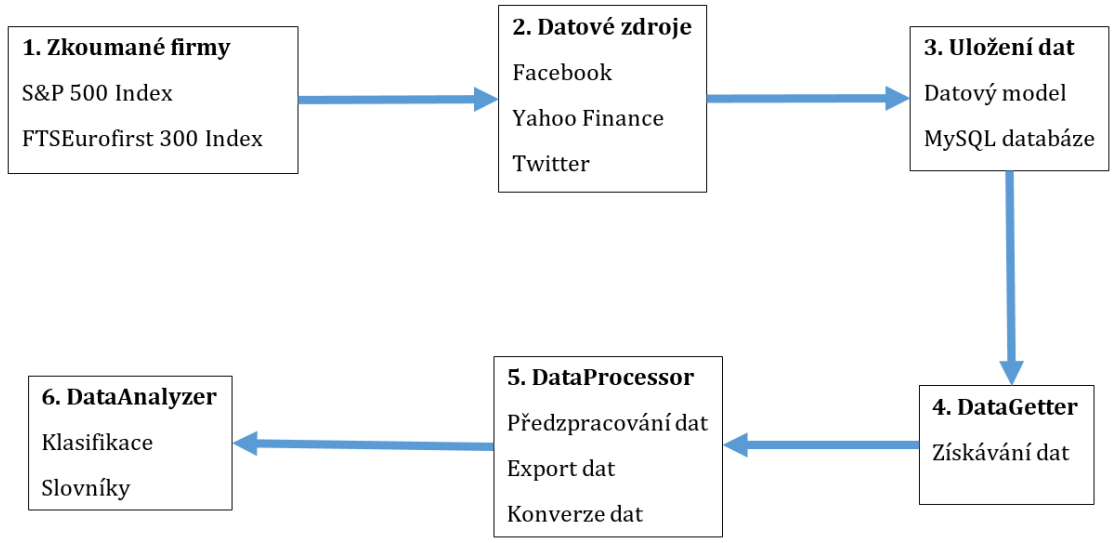
Z tabulky 45 vyplývá, že nejvyšší správnost má RNTN (*Recursive Neural Tensor Network* – 87,60), který se ale zaměřuje na věty a není jasné, jak by si poradil s celými dokumenty. Ze zbylých „klasických“ algoritmů má průměrnou nejvyšší správnost SVM (*Support Vector Machines* – 84,93), následovaný NB (*Naive Bayes* – 81,58) a ME (*Maximum Entropy* – 80,45). Je zajímavé, že ačkoliv je NB daleko jednodušší (a tudíž rychlejší) než ostatní algoritmy, dosahuje porovnatelných výsledků (Narayanan et al., 2013).

Pang a Lee (2004) používá pro klasifikaci pouze subjektivní věty z dokumentu (zůstává asi 60 % původního obsahu). To u algoritmu NB zvýšilo správnost o 4 % na 86,4 % (u SVM to nemělo žádný vliv). Výběr subjektivních vět byl proveden natrénováním NB a SVM na korpusu vět. Je zde také popsána technika hledání minimálního řezu v grafu (*Minimum cut*), kterou lze použít pro vložení informace o kontextu (konkrétně blízkosti vět) do klasifikátoru, což má také mírně pozitivní vliv na správnost (+1 %).

Některé publikace kromě klasického *bag-of-words* modelu používají také jiné typy příznaků. V Go et al. (2009) to jsou unigramy, bigramy, unigramy + bigramy, unigramy + POS. V Pang et al. (2002) jsou navíc přídavná jména a unigramy se zohledněním pozice. Nicméně typ příznaků má vliv na správnost klasifikace maximálně v rozmezí 2–3 %. Bigramy mají obvykle nižší správnost než unigramy – důvodem je vysoká řídkost výsledkých vektorů (Go et al., 2009).

3 Metodika

V této kapitole je popsáno, jak budeme při práci postupovat. Obrázek 3 zobrazuje zjednodušený přehled kroků, které budou v rámci práce provedeny.



Obr. 3: Přehled postupu v práci (metodika)

Konkrétně je tedy nutné provést následující kroky:

1. Vybrat sledované firmy.
2. Získat data:
   a) Zvolit datové zdroje.
   b) Vytvořit datový model.
   c) Navrhnout a implementovat modul pro získávání a ukládání dat.
3. Analyzovat data:
   a) Vytvořit modul pro předzpracování, export a konverzi dat pro klasifikaci.
   b) Vytvořit modul pro klasifikaci připravených dat.
   c) Provést klasifikaci a vyhodnotit výsledky.
   d) Na základě výsledků klasifikace provést výběr důležitých atributů.
   e) Vytvořit nové slovníky sentimentu.
   f) Vytvořit modul pro určování sentimentu pomocí slovníku.
   g) Provést a vyhodnotit analýzu sentimentu pomocí slovníků.

3.1 Volba sledovaných firem

Nejdříve je nutné vybrat firmy, které budou použity pro analýzu. Zaměříme se na velké a známé firmy (*blue chips*), jelikož lze předpokládat, že o nich bude dostupných hodně textů. Bude sledováno velké množství firem, aby šly výsledky dobře zobecnit.

Tabulka 1 v sekci 2.2.3 zobrazuje nejvýznamnější akciové indexy v Evropě a USA – můžeme je využít jako základ pro to, jaké firmy sledovat. Zajímají nás pouze indexy, obsahující více než 100 firem a zahrnující více burz – jsou to pouze dva a to S&P 500 a FTSEurofirst 300. Firmy z těchto indexů tedy budeme analyzovat – celkem přibližně 800 firem (jejich seznam se nachází v el. příloze A).

S tímto přístupem se pojí dva hlavní problémy. Zaprvé, seznam firem, obsažených v obou indexech, se z jejich podstaty neustále mění. Tyto změny nemá význam sledovat – pokud se firma zvládla dostat do indexu, jedná se o dostatečně silný indikátor toho, že patří mezi *blue chips* a je pro nás relevantní. Navíc pokud bychom sledovali vždy aktuální seznam firem v indexu, historie některých sledovaných firem by mohla být příliš krátká. Budeme tedy sledovat firmy, které byly v indexu k určitému datu, přičemž nás nezajímá, zda byly následně z indexu vyřazeny.

Zadruhé, firma může zbankrotovat, stáhnout se z burzy nebo být pohlcena jinou firmou. Toto se během získávání dat stalo mnohokrát. Tento problém vyžaduje okamžité řešení. Daná firma je jednoduše označena jako neaktivní – její historie zůstává v databázi, ale nová data se pro ni již nestahují.

Firmy pro analýzu byly vybrány na základě obsahu indexů S&P 500 a FTSEurofirst 300 k 9. 7. 2015. Na počátku jich bylo 784. Firmy jsou každý den kontrolovány a co nejdříve označovány jako neaktivní. K 4. 4. 2016 bylo v databázi 775 aktivních firem – to znamená, že již 9 jich je neaktivních. Pro ilustraci poměrně živelného vývoje ohledně bankrotů a (zejména) akvizic firem uveďme, že k 2. 10. 2015 byly neaktivní jen 3 firmy (Petrovský, 2015).

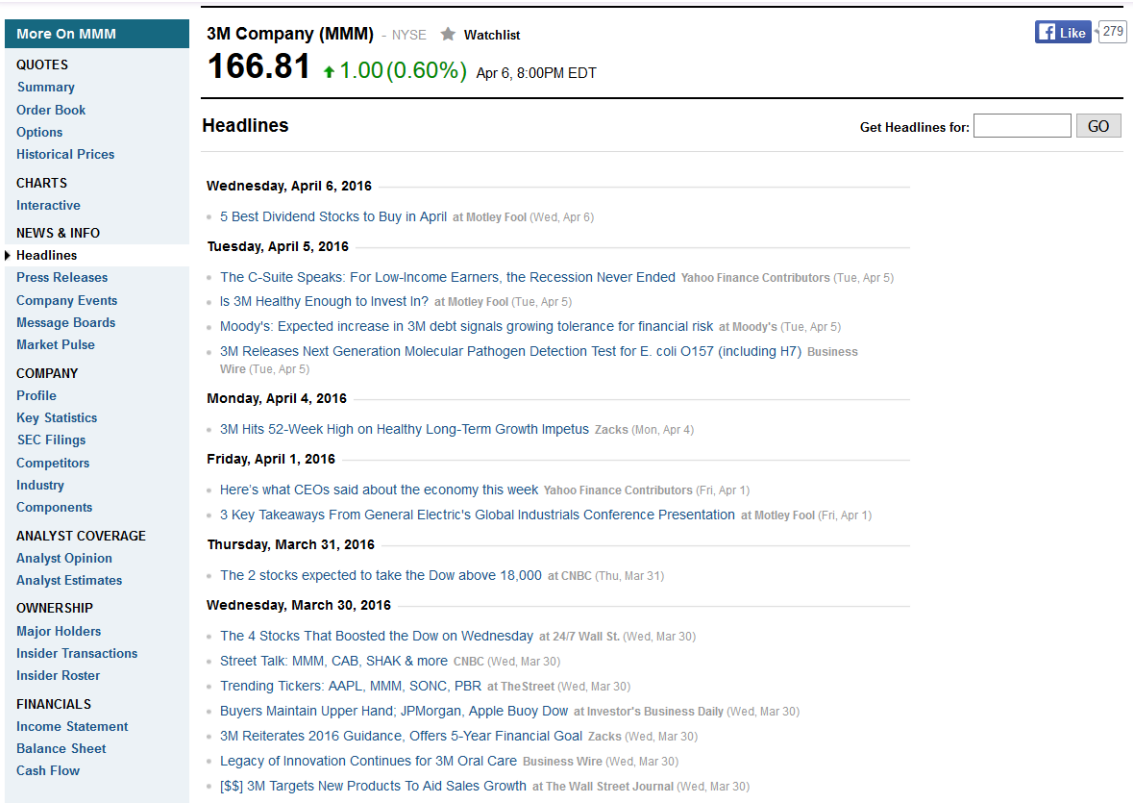
3.2 Použité datové zdroje

V sekci 2.3 byly představeny možné datové zdroje pro textové dokumenty, které obsahují názory, emoce a nálady lidí ohledně jednotlivých firem. Jedná se především o zpravodajské servery a sociální sítě, případně diskuzní fóra. Na základě provedeného srovnání byly zvoleny následující zdroje dat, které jsou dále blíže popsány.

3.2.1 Yahoo Finance

Jako zdroj pro novinové články byl zvolen server Yahoo Finance (viz sekce 2.3.1). Pro každou aktivní firmu (vedenou na nějaké burze) je k dispozici webová stránka[17]. Ta obsahuje velké množství informací (viz levý sloupec na obrázku 4): analýza ceny, finanční ukazatele a výkazy, doporučení analytiků aj. Nás bude zajímat odkaz *Headlines*, obsahující seznam článků, souvisejících s danou firmou.

[17]Například pro firmu 3M Company to je `http://finance.yahoo.com/q?s=MMM`



Obr. 4: Novinové články na Yahoo Finance (pro 3M Company)

Nadpis článku představuje odkaz na plný text článku. Ten může být umístěn buď přímo na serveru Yahoo Finance, nebo míří na jiný server (takových serverů je celkem 42). V prvním případě je struktura vygenerované HTML stránky vždy stejná, takže lze text pomocí *web scrapingu* snadno získat. V druhém případě je situace složitější, jelikož každý server má jinou HTML strukturu a tudíž by bylo nutné pro 42 serverů vytvořit speciální parser. To bylo vyhodnoceno jako příliš pracné a ve výsledku nadbytečné. Jak se později ukázalo, i pro Yahoo Finance bylo nutné vyřešit několik speciálních případů, takže se dá domyslet, jaká by byla situace u jiných webů, jejichž struktura se samozřejmě může občas změnit. Udržovat 43 různých verzí parsovací metody je poněkud nepraktické. Zřejmě by šlo tento problém vyřešit pomocí nějakého méně rigidního resp. pravděpodobnostního přístupu či dokonce pomocí strojového učení, ale to přesahuje rámec této práce.

Každý článek může mít určitý počet sdílení na Twitteru[18] a Facebooku – zjistím jej dotazem na veřejné API: `http://graph.facebook.com/<url článku>`.

Yahoo Finance také poskytuje informace o historických cenách akcií (odkaz

[18]Pro Twitter bylo možné do 20. 11. 2015 použít API `http://cdn.api.twitter.com/1/urls/count.json?url=<url>`. Nyní to bohužel již možné není. K danému účelu je potřeba využít autentizovaný požadavek přes Search API Twitteru – vyhledám danou URL a spočítám počet tweetů, které ji obsahují. Případně existují externí služby (např. `https://opensharecount.com`).

Historical prices). Na dané stránce[19] je také možné stáhnout CSV soubor (odkaz *Download to Spreadsheet*), obsahující ceny akcie v zadaném časovém rozmezí (nebo všechny dostupné ceny od uvedení firmy na burzu). Konkrétně je potřeba zaslat HTTP GET dotaz ve tvaru `http://real-chart.finance.yahoo.com/table.csv?s=ABT&d=3&e=9&f=2016&g=d&a=3&b=6&c=1983&ignore=.csv` kde s určuje ticker firmy, d, e, f měsíc, den a rok DO a a, b, c měsíc, den a rok OD. Číslo měsíce d je z neznámého důvodu o jedna menší než to skutečné – uvedený dotaz totiž získá ceny akcií od 6. 3. 1983 do 9. 4. 2016. V datech jsou přítomny pouze ceny pro pracovní dny – víkendy a svátky tam uvedeny nejsou.

K dispozici jsou *Open* (otevírací), *High* (nejvyšší), *Low* (nejnižší), *Close* (uzavírací) a *Adjusted Close* (uzavírací cena upravená o dividendy a splity – rozdělení akcií) ceny akcie pro daný den, doplněné o *Volume* – počet akcií dané firmy zobchodovaných na burze za daný den. Budeme ukládat všechny tyto údaje.

3.2.2 Facebook

Jak je uvedeno v sekci 2.3.2, Facebook používá v současnosti asi 1,5 mld. lidí. Tím pádem představuje potenciálně obrovský zdroj názorových textových dat. Problémem je, že tato data jsou většinou osobní a tudíž veřejnosti skrytá. Jelikož Facebook nenabízí žádné globální vyhledávání, tak bylo rozhodnuto, že se zaměříme na tzv. „fanouškovské stránky“ (*fan pages*) jednotlivých zkoumaných firem. Pro každou ze sledovaných firem bylo tedy nutné ručně ověřit, zda má stránku na Facebooku (FB) a pokud ano, tak její název uložit do Excelu.

Na počátku práce (červenec 2015) bylo nalezeno 398 stránek, což znamená, že asi polovina (50,8 %) ze 784 sledovaných firem má Facebook stránku. Později byly nalezeny další stránky, takže celkově jich je 431, což dává asi 55 %. Každá FB stránka obsahuje časovou osu (*timeline*) – sekvenci příspěvků (*posts*), vytvořených danou firmou (resp. administrátorem spravujícím danou stránku). Uživatelé mohou k těmto příspěvkům přidávat komentáře a „likovat“ (dávat to se mi líbí) jednotlivé příspěvky i komentáře ostatních uživatelů.

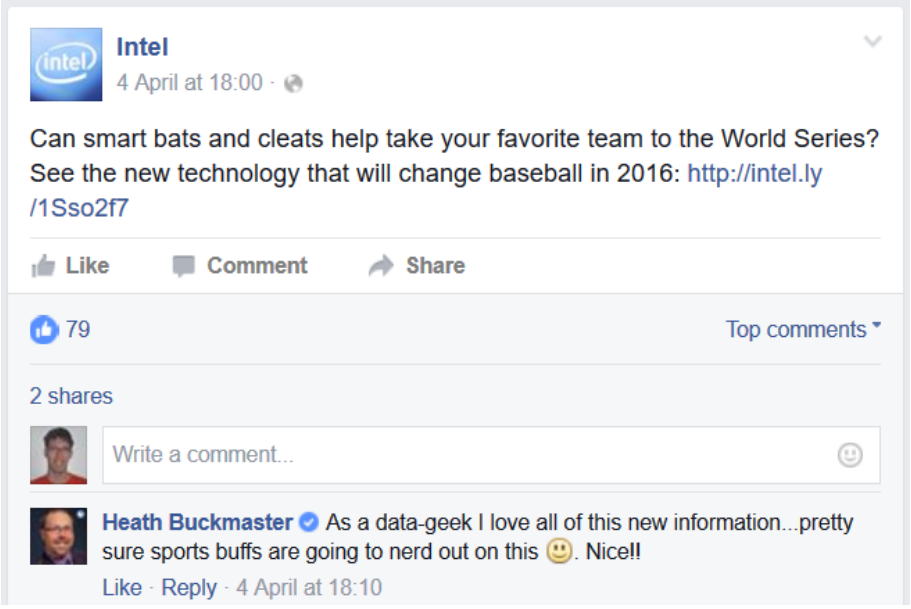
Na FB stránce jsou tedy přítomny dva základní typy textů – příspěvky firmy a komentáře uživatelů. Oba dva typy mají přiřazený počet „líků“, přičemž příspěvek má také počet sdílení. Tyto počty se v průběhu času mění a bylo by vhodné tyto změny sledovat.

Je nutné si uvědomit, co je obsahem těchto textů. Příspěvky jsou obvykle firemní oznámení („Podařilo se nám XY.“) nebo prezentace zajímavostí/lidí, souvisejících s produkty firmy („Podívejte se, co náš produkt umí.“). Je jasné, že zástupci firmy zřídkakdy (pokud vůbec) publikují statusy s negativním sentimentem. Nicméně data jsou i tak použitelná – např. pro zjištění, zda pozitivní texty ovlivňují názory lidí a tím pádem i cenu akcií.

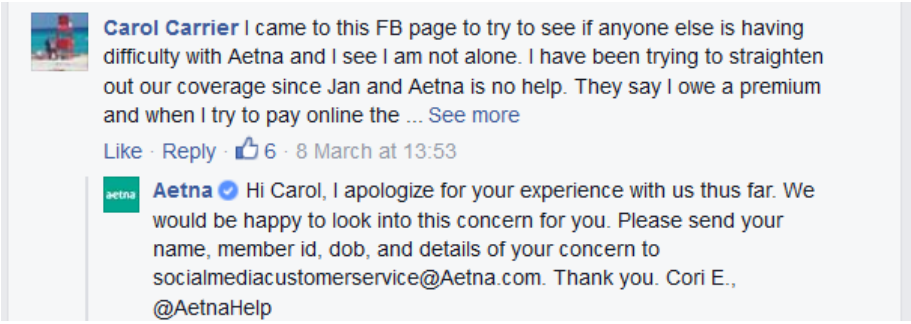
Komentáře jsou dvojího typu: Buď (více či méně) souvisí s daným příspěvkem (obrázek 5), nebo vůbec ne. V tom druhém případě obecně chválí nebo haní da-

[19] Např. `http://finance.yahoo.com/q/hp?s=ABT`

nou firmu – obvykle zde lidé píšou negativní zkušenosti s produkty/službami či se zákaznickou podporou a případně žádají nápravu defektního stavu (obrázek 6).



Obr. 5: Příspěvek a komentář na Facebook stránce (pochvala)



Obr. 6: Komentář na Facebook stránce (zákazník)

Data z Facebooku lze získat pomocí jeho *Graph API*, které aplikacím umožňuje číst a zapisovat do „sociálního grafu“ Facebooku. API je založené na REST[20] a vrací data ve formátu JSON. Jak lze vidět v dokumentaci (Facebook, 2016), API nabízí mnoho možností (odkaz zobrazuje „kořenové uzly“ sociálního grafu). Nás bude zajímat uzel Page, reprezentující FB stránku. Uzel nabízí mnoho polí včetně pole *posts*, obsahujícího odkaz na příspěvky publikované na stránce. V jednom požadavku je možné přečíst nejvýše 100 příspěvků. Ke čtení je potřeba přístupový token (*access token*), který slouží k autentizaci (resp. autorizaci). Ten získáme po vytvoření vývojářské aplikace u Facebooku na základě její `APP_ID` a `APP_SECRET`. Tento token poté při každém požadavku na API zasíláme spolu s požadovanými parametry.

[20] Representational State Transfer

Příspěvky lze získat zasláním HTTP GET požadavku na URL `https://graph.facebook.com/<version>/<page-id>/posts`, kde `<page-id>` je ID dané stránky (lze použít také textový název). Konkrétně byl použit následující dotaz:

```
https://graph.facebook.com/2.3/intel/posts/?limit=90&date_format=u
&since=<timestamp>&fields=id,created_time,message,shares,
likes.limit(0).summary(true),comments.limit(100).summary(true)
```

Význam: Získej 90 (`limit`) nejnovějších příspěvků ze stránky „intel“, které vznikly od doby `<timestamp>` (`since` – UNIX timestamp[21]). Vrať ID, čas publikování, text, počet sdílení a liků každého příspěvku. Dále vrať počet komentářů a obsah 100 nejpopulárnějších komentářů (řazených dle počtu liků). Časy jsou uloženy jako UNIX timestamp (`date_format`). Jsou stahovány pouze komentáře první úrovně, nikoliv komentáře komentářů, jelikož k tomu je nutné zaslat dotaz na API pro každý komentář, což by trvalo neúnosně dlouho.

3.2.3 Twitter

Jako další sociální síť pro získávání dat byla zvolena mikro-blogovací služba Twitter. Jak je uvedeno v sekci 2.3.2, tuto síť používá asi 320 mil. lidí. Základní funkcionalita služby spočívá v tom, že registrovaný uživatel může publikovat krátké textové zprávy (tzv. „tweety“, s maximální délkou 140 znaků), ostatní uživatelé ho mohou „sledovat“ (číst jeho zprávy) a interagovat s ním (odpovídat na jeho tweety nebo mu posílat zprávy). Hlavní výhoda Twitteru je, že nabízí globální vyhledávání. Pro získávání dat souvisejících se sledovanými firmami byly použity následující vyhledávací dotazy. Příklad je pro firmu Google, jejíž uživatelské jméno (už. jm.) je `google`.

1. `@google` ... tzv. *mentions* – tweety, obsahující už. jm. firmy.
2. `Google -@google` ... tweety, obsahující název firmy, ale bez už. jm. firmy.
3. `to:google` ... tzv. *replies* – tweety, které jsou odpověďmi na tweety firmy.

Navíc jsou stahovány tweety z „timeline“ (profilu) firmy. Tweety jsou velmi různorodé - od ohlášení nového produktu, zpráv o budoucích plánech či aktuálních problémech firmy, přes názory zákazníků a novinářů na firmu, až po různé zajímavosti, týkající se firmy. Příklad tweetu ukazuje obrázek 7.

Tweety lze získávat pomocí dvou typů API. Prvním je REST API, které má určité limity týkající se toho, kolik dotazů za určitý časový interval může aplikace zaslat na servery Twitteru. Pro hledání tweetů to je 450 dotazů za 15 minut. Druhou možností je použít Streaming API, které je zaměřeno na zpracování tweetů v reálném čase. Je založeno na dlouhodobém HTTP spojení a umožňuje tak obejít limity REST API ohledně množství stahovaných dat (Twitter, 2016a).

Bylo rozhodnuto, že zpočátku budou stahována Twitter data pouze pro malé množství firem (21). Důvodem je, že se chceme zaměřit také na jiné zdroje dat a pro

[21] Počet sekund, které uplynuly od 00:00 UTC 1. 1. 1970.

Obr. 7: Příklad tweetu

stahování tweetů o všech firmách by byl potřeba obrovský úložný prostor. Vybrány byly známé, reprezentativní firmy z různých oborů. Každopádně je tedy možné v tomto případě použít REST API, aniž by nás omezovaly výše uvedené limity.

Jak lze vidět v dokumentaci (Twitter, 2016b), API nabízí mnoho různých koncových bodů (tzv. *endpoints*). Obecně se k API přistupuje přes URL `https://api.twitter.com/<version>/<endpoint>`, kde `<version>` udává verzi API a `<endpoint>` určuje koncový bod, ze kterého chceme data číst. Nás zajímá především to, jaké existují možnosti pro vyhledávání (tzn. Twitter Search API). Toto API poskytuje tweety staré maximálně 1 týden, přičemž odpověď na jeden dotaz může obsahovat až 100 tweetů. Přistupuje se k němu přes HTTP GET dotaz na koncový bod `search/tweets.json?q=<query>`, kde `<query>` je vyhledávací výraz. Konkrétně byl použit dotaz s následujícími parametry (příklad pro firmu Google):

```
https://api.twitter.com/1.1/search/tweets.json?
q=Google&lang=en&result_type=mixed&count=100&since_id=123
```

Význam: Získej 100 (`count`) tweetů, odpovídajících zadanému datazu (`q`), napsaných v angličtině (`lang`). Zahrnuty jsou jak populární, tak nedávné tweety (`result_type`), jejichž ID je větší než 123 (`since_id` – vyhledávání dle času).

Pro stahování tweetů z profilu firmy je použit API endpoint `statuses/user_timeline.json`. Ten umožňuje získat posledních 3 200 tweetů, přičemž odpověď na jeden dotaz může obsahovat až 200 tweetů. Limit dotazů je 300 za 15 minut. Použitý GET dotaz:

```
https://api.twitter.com/1.1/statuses/user_timeline.json?
count=200&since_id=123&screen_name=google
```

Význam: Získej 200 (`count`) nejnovějších tweetů z profilu firmy Google (`screen_name`), které mají ID větší než 123.

Obě dvě metody vrací data (tweety) ve formátu JSON, obsahující mnoho různých polí (viz `https://dev.twitter.com/overview/api/tweets`). Budeme ukládat co nejvíce informací o daném tweetu, které by nám mohly být k něčemu užitečné. Jedná se o text, ID, čas vytvoření, počet liků a retweetů (sdílení), zda to je odpověď na nějaký jiný tweet, lokace tweetu a informace u autorovi: ID, počet followerů (sledujících uživatelů), počet uživatelů které sleduje, počet publikovaných tweetů, lokace. Pro autentizaci se používá protokol OAuth buď na úrovni uživatele nebo aplikace, přičemž pro aplikaci jsou limity vyšší a bude tedy použit druhý způsob.

3.3 Návrh modulu pro získávání dat

Pro splnění cíle práce je nutné získat (stáhnout) data, specifikovaná v sekci 3.2. K tomu bude vytvořen softwarový modul. Pro vývoj softwaru existuje celá řada metodik. Ty obvykle používají jako základní strukturu tzv. vodopádový model. Ten je založen na obecném životním cyklu vývoje softwaru, který obsahuje tyto etapy:

1. specifikace problému – definice požadavků,
2. analýza a návrh,
3. implementace,
4. zavedení a testování,
5. provoz a údržba.

Waterfall model přidává možnost během vývoje specifikovat nové požadavky a vrátit se do bodu 2 (Rábová, 2008, s. 40). Nebude použita rozsáhlá metodika vývoje (jako např. RUP), jelikož modul není extrémně složitý a autor ho vyvíjí sám. Bude použita metodika založená na vodopádovém modelu s tím, že výsledné řešení se bude testovat a průběžně upravovat dle nalezených problémů a nových požadavků. V podstatě tedy bude použita (v malém měřítku) iterativní metoda vývoje.

3.3.1 Funkční požadavky

Účelem modulu je získávat vybraná data z internetu a ukládat je do databáze.

1. Proces získávání a ukládání dat musí probíhat automaticky, bez zásahu uživatele, pro všechny sledované firmy. Respektive pro ty, pro které to je možné – jsou známa jména jejich Facebook a Twitter účtů resp. jsou přítomné na burze (pro Yahoo články).
2. Požadovaná data jsou textové dokumenty (novinové články, texty z Facebooku a Twitteru) obsahující dodatečné atributy a ekonomická data (ceny akcií).
3. Pro každý datový zdroj musí existovat zvláštní procedura, zajišťující správnou práci s ním – tedy získávání, zpracování a ukládání dat.
4. Čas (resp. interval) spouštění procedur může být libovolně nastaven.
5. Pokud z nějakého důvodu nebude možné data z určitého zdroje stáhnout, bude administrátorovi zaslán e-mail, sdělující problém a jeho detaily. Především je potřeba sledovat přítomnost firmy na burze, HTML strukturu novinových článků a změny názvů profilů firem na Facebooku a Twitteru.
6. V databázi by se neměly vyskytovat duplicitní dokumenty.
7. Po doběhnutí procedury je zaznamenán aktuální čas a zda se během jejího běhu vyskytla chyba.

Dále je nutné vytvořit funkci pro jednorázový import firem z Excel souboru do tabulky v databázi.

3.3.2 Nefunkční požadavky

Modul musí pracovat co nejefektivněji, jelikož má získávat data pro téměř 800 firem. To představuje potenciálně obrovské množství dat, která bude (v rámci určitého časového intervalu) nutné stáhnout z internetu a uložit do databáze. Dá se předpokládat, že většinu strojového času zabere stahování dat pomocí protokolu HTTP. Je tedy nutné minimalizovat počet posílaných dotazů a velikost vracených odpovědí. Další čas bude vynaložen na zpracování získaných dat. Ten ovšem bude vzhledem k rychlosti dnešních procesorů minimální. Nakonec budou data uložena do databáze, což v případě mnoha tisíců vkládaných řádků může zabrat značné množství času. Z tohoto důvodu bude nutné minimalizovat počet dotazů posílaných do databáze a zřejmě také obětovat některé kontrolní mechanismy robustních RDBMS.

Modul je vytvořen pomocí objektově orientovaného přístupu (OOP). Procedury pro jednotlivé datové zdroje jsou implementované pomocí tříd (resp. jejích metod) a spustitelné nezávisle na sobě. Proceduru lze spustit zavoláním vhodné veřejné metody dané třídy. Pro automatické spouštění jsou k dispozici výkonné soubory (skripty), zajišťující spuštění procedury.

Modul neposkytuje žádné GUI. Pro jeho vývoj je použit vhodný interpretovaný jazyk, což zajistí vysokou rychlost vývoje a umožní snadné modifikace. To, že program poběží pomaleji, než při použití kompilovaného jazyka není podstatné, jelikož daleko více času zabere síťová a databázová komunikace.

3.3.3 Výběr programovacího jazyka

V této části práce bude zvolen vhodný jazyk pro implementaci modulu. Jak je uvedeno v následující sekci 3.4, pro implementaci databáze bude použit relační DBMS MySQL. Jelikož MySQL podporuje průmyslové databázové standardy ODBC a JDBC, lze jej používat z velkého množství programovacích jazyků. Kromě toho je k dispozici několik MySQL konektorů, vytvořených pro konkrétní platformu resp. jazyk: .NET, Java, Python, C++, C, PHP (Oracle, 2016).

Dle nefunkčních požadavků se zaměříme pouze na interpretované jazyky. Důležitá je podpora stahování a zpracování dat z internetu. V podstatě každý jazyk umožňuje zasílat HTTP požadavky a zpracovávat odpovědi. Liší se ovšem v tom, jak jednoduše to lze provádět příp. zda je nutné instalovat speciální rozšíření.

Webová aplikace Carbonnelle (2015) sleduje popularitu programovacích jazyků – na základě toho, jak často jsou o nich na Google vyhledávány tutoriály (zjišťováno pomocí Google Trends). Tabulka 9 ukazuje intepretované jazyky, které patří mezi 17 nejpopulárnějších k dubnu 2016, jejich podíl na celkovém počtu vyhledávání a trend v porovnání s dobou před rokem.

Lze vidět, že na prvním místě je Python, následovaný PHP a JavaScriptem. Autor práce nemá rád PHP ani JS. Ještě zde jsou R, Ruby nebo Perl: první jazyk

Tab. 9: Popularita interpretovaných programovacích jazyků (duben 2015–2016)

Pořadí	Jméno	Podíl 04-2016 [%]	Trend 2016/2015 [%]
2	Python	12,3	1,6
3	PHP	10,6	-1,0
5	JavaScript	7,5	0,4
9	R	3,1	0,4
11	Matlab	2,9	-0,1
12	Ruby	2,3	-0,3
15	Perl	1,1	-0,2
17	lua	0,5	0,1

Zdroj: (Carbonnelle, 2015)

je vhodný pro statistické výpočty, druhý je podobný Pythonu a třetí se hodí pro zpracování textu. Nicméně jejich popularita a tedy i množství návodů a knihoven je menší (a v případě Ruby a Perlu mírně klesá). Vzhledem k tomu, že Python je velmi populární a autor práce s ním již má zkušenosti, bude vybrán právě tento jazyk.

Otázkou je, zda použít Python ve verzi 2 nebo 3. Python 3 je novější a obsahuje spoustu syntaktických a jiných vylepšení. Nicméně existují populární knihovny, které pro něj nejsou dostupné (viz `http://py3readiness.org`). Navíc na unixových OS je standardem stále Python 2. Z tohoto důvodu byl zvolen Python ve verzi 2.7.10.

3.4 Návrh databáze

V rámci práce je nutné navrhnout a implementovat databázi, která bude schopna pojmout všechny požadované informace.

Existují tři hlavní fáze (úrovně) návrhu databáze (Connolly a Begg, 2005, s. 293):

1. Konceptuální: Cílem je (na základě požadavků uživatelů) vytvořit model dat (používaných v daném projektu/organizaci), který bude zcela nezávislý na fyzické implementaci.
2. Logická: Cílem je vytvořit model dat, který je založen na konkrétním typu datového modelu (např. relační), je ale nezávislý na konkrétním použitém DBMS. V této fázi je tedy konceptuální model převeden na logický model.
3. Fyzická: Cílem je přesně popsat implementaci databáze na datovém úložišti (tabulky, organizace souborů, indexy, integritní omezení, bezpečnost) při použití konkrétního DBMS. V podstatě tedy musíme popsat, jak fyzicky implementovat logický model.

3.4.1 Požadavky na informace v databázi

Nejdříve je nutné určit, jaké informace má databáze uchovávat. Lze k tomu použít různé techniky pro hledání faktů – viz Connolly a Begg (2005, s. 317). Nicméně v našem případě bude postačovat použít názory autora práce, dosavadní text práce a konzultace s vedoucím práce.

Účelem databáze bude uchovávat textová a ekonomická data, týkající se určitých firem. Pro textová data byly zvoleny tři datové zdroje:

- Yahoo Finance – jeden typ dokumentu – články.
- Facebook – dva typy dokumentů – příspěvky, komentáře.
- Twitter – jeden typ dokumentu – tweety (4 různé variace).

O získaných dokumentech je potřeba uchovávat následující informace: datum (resp. čas) vytvoření, samotný text, ID nebo URL originálního textu, popularita dokumentu (počet sdílení apod.). Pro Yahoo články ještě název zdrojového serveru a pro FB komentáře navíc jméno a ID autora komentáře. Pro tweety všechny vlastnosti uvedené na konci sekce 3.2.3. A samozřejmě také to, k jaké firmě dokument patří. Chceme také sledovat, jak se vlastnosti (zejména popularita) dokumentu mění v čase. U Facebooku včetně toho, když k příspěvku přibude nový komentář.

Dále je nutné uchovávat informace o akciích všech firem – pro každý den nás bude zajímat cena (různé typy) a objem obchodů.

Je logické, že potřebujeme udržovat seznam firem, a pro každou: ticker na burze, název firmy, název Facebook stránky, Twitter uživatelské jméno a jméno firmy pro vyhledávání na Twitteru. Užitečné může být také vědět, do jakého indexu firma patří a na jaké burze je obchodovaná.

Z požadavků na související aplikaci vyplývá, že dokumenty se budou stahovat každý den. Proto je nutné udržovat pro každý typ dokumentu a každou firmu údaj o tom, jakého data je poslední stažený dokument. Dále je vhodné zaznamenávat proběhlé procesy stahování – čas dokončení a zda během něj nastala chyba.

3.4.2 Logický datový model

Pomocí *Entity–Relationship* modelování bude vytvořen ER diagram – diagram entit a jejich vztahů (ve zvolené notaci). Entita představuje skupinu objektů se stejnými vlastnostmi. Tyto objekty mohou být reálné nebo abstraktní. Entita je identifikována jménem a má definované atributy (vlastnosti). Každý atribut má svou doménu (množinu povolených hodnot).

Vztah je množina asociací mezi (obvykle dvěma) entitami. Má jméno, popisující jeho funkci. Na vztah jsou obvykle kladena dvě omezení. Prvním je kardinalita, která určuje maximální počet výskytů první resp. druhé entity v daném vztahu (1:1, 1:n, m:n). Druhým je parcialita, která určuje, zda se vztahu účastní všechny výskyty první resp. druhé entity – tedy zda je pro danou entitu povinný či nepovinný (Connolly a Begg, 2005, s. 342–363).

Na základě analýzy požadavků byly identifikovány následující entity:

- Firma (*company*), cena akcie (*stock price*).
- Yahoo článek (*article*), Facebook příspěvěk (*fb post*), Facebook komentář (*fb comment*), tweet (*tw status*).
- Zdrojový server článku (*article server*).
- Historie sdílení článků (*article history*), historie Facebook příspěvků (*fb post history*), historie Facebook komentářů (*fb comment history*).
- Poslední stažení dokumentů pro firmu (*last download*), záznam proběhlého stahování (*log exec*).

Následně byly definovány atributy jednotlivých entit a vztahy mezi entitami. Tímto vznikl konceptuální resp. logický model dat. Pro implementaci databáze byl (na základě výsledků porovnání v sekci 2.4.2) vybrán relační DBMS MySQL. Výslednou podobu databáze ukazuje obrázek 8 v sekci 4.1.1.

3.4.3 Nahrání dat

Nejdříve ovšem bylo nutné načíst do databáze data z Excel souboru, který obsahoval seznam firem a jejich název, ticker a jméno Facebook stránky a Twitter účtu. K tomuto účelu byla vytvořena třída `CompanyCsvParser`, která na vstupu dostane Excel soubor a vygeneruje SQL soubor, obsahující příkazy pro vložení záznamů do tabulky `Company`. Tento soubor byl následně spuštěn v MySQL.

3.5 Analýza dat

Získávání dat je již vyřešeno a předpokládáme, že máme k dispozici dostatečné množství dat pro analýzu. Cílem analýzy je zjistit, zda (a případně jak) lze informace obsažené v textových dokumentech použít k vysvětlení pohybu cen akcií. Informace je v dokumentu uložená ve formě slov, frází, vět a odstavců. Můžeme zkoumat jak obsah, tak i sentiment dokumentu.

3.5.1 Analýza č. 1 – obsah dokumentů a pohyby cen akcií

Nejdříve se pokusíme odpovědět na otázku: „Jak souvisí obsah dokumentů a pohyby cen akcií?“. Konkrétně budeme řešit tento problém: Na základě obsahu dokumentu urči, zda cena akcie (firmy, o které dokument pojednává) na konci $1, 2, \ldots, N$ dalších dnů klesne nebo stoupne oproti ceně na konci dne publikace dokumentu. Jedná se tedy o problém binární klasifikace, kdy cílem je zařadit dokument do třídy *up* nebo *down* dle toho, zda jeho publikace na internetu způsobí, že cena související akcie (s určitým zpožděním) stoupne nebo klesne.

Pro klasifikaci lze použít algoritmy strojového učení, založené na učení s učitelem. Potřebujeme tedy množinu označkovaných objektů, které budou použity pro

natrénování a vyhodnocení algoritmu. Objekty je nutné kvantitativně charakterizovat – je vytvořena množina vlastností objektů (atributy, příznaky – *features*). Na základě těchto příznaků následně algoritmus vytvoří model, který každému objektu přiřadí jednu třídu. Je vhodné, aby byla množina označkovaných objektů vyvážená – tj. aby pro každou třídu obsahovala stejný počet objektů.

V našem případě jsou objekty textové dokumenty. Jak bylo popsáno v sekci 2.5.2, ty se obvykle reprezentují pomocí vektoru, obsahujícího jednotlivé termy (nejčastěji slova – unigramy), získané z textu všech dokumentů. Tyto termy tedy tvoří množinu příznaků. Vektor dokumentu obsahuje pro každý term z množiny prvek, který určuje, zda se term v dokumentu vyskytuje nebo ne (resp. váhu tohoto výskytu).

Ještě zbývá stanovit, jak určovat třídu, do které dokument patří. Jak bylo uvedeno, budeme sledovat relativní rozdíl cen mezi dvěma časovými okamžiky (tedy zisky akcie – *returns*). Formálně lze zisk akcie R v čase t zapsat jako

$$R_t = \frac{p_t - p_{t-1}}{p_{t-1}} \,, \tag{5}$$

kde p_t je cena akcie v čase t. Prvním okamžikem $(t-1)$ je čas publikace dokumentu. Druhým okamžikem (t) je nějaký čas v budoucnosti, kdy by se měl projevit vliv obsahu dokumentu na cenu akcie.

Máme k dispozici pouze *end-of-day* ceny akcií (nikoliv *intra-day* – během dne). Jako první okamžik stanovíme konec obchodního dne publikace dokumentu. Jako druhý okamžik bude zvolen konec dalšího (resp. dalších $2, \ldots, N$ dnů) obchodního dne. Rozdíl mezi těmito okamžiky udává zpoždění od publikace dokumentu (získání informace) po vliv informace na cenu akcie. Zatím nevíme, jak daleký druhý okamžik bude přinášet nejlepší výsledky, proto musí být otestováno více variant (1, 2, 3 dny).

Stejně jako v Arias et al. (2013) bude použita cena *Adjusted close*, což je koncová cena daného dne, upravená o korporátní události, které se udály do otevření burzy následujícího dne (zejména rozdělení akcií a vyplacení dividend).

Místo prosté ceny je možné použít klouzavý průměr cen. Ten vyhladí křivku ceny tak, aby zobrazovala dlouhodobější cenový trend. Existuje mnoho typů klouzavého průměru: jednoduchý, kumulativní, vážený, exponenciální (Wikipedia, 2016b).

Jako první bude vyzkoušen jednoduchý klouzavý průměr (SMA – *Simple Moving Average*). Je to prostý aritmetický průměr posledních n hodnot. Pro den t je

$$\mathrm{SMA}_t = \frac{1}{n} \sum_{i=0}^{n-1} p_{t-i} \,. \tag{6}$$

Hodnota n určuje velikost tzv. *sample window* – zkoumaného okna. Byla zvolena hodnota $n = 5$, která se používá jako nejnižší v technické analýze (Mitchell, 2016).

Dále bude použit exponenciální klouzavý průměr (EWMA – *Exponentially weighted moving average*), který hodnotám v oknu přiřazuje váhy, klesající exponenciálně. Novější hodnoty tedy mají mnohem vyšší váhu, než ty starší. Pro den t je

$$\mathrm{EWMA}_t = \lambda \cdot p_t + (1 - \lambda) \cdot \mathrm{EWMA}_{t-1} \qquad 0 < \lambda \leq 1 \qquad t = 2, \ldots, n \,, \tag{7}$$

kde p_t je cena v čase t, n je velikost okna, λ je vyhlazovací konstanta (určuje jak rychle jsou znehodnocovány starší hodnoty) a $\mathrm{EWMA}_1 = p_1$ (NIST, 2012). Arias et al. (2013) používá pro EWMA $n = 30$, my ale zvolíme $n = 20$, což je běžná hodnota v technické analýze (Mitchell, 2016). Hodnota n je použita pouze pro určení λ. Pokud zvolíme $\lambda = 2/(n + 1)$, tak pro dostatečně velké n reprezentuje prvních n hodnot asi 86 % celkového součtu vah. Ten je vždy jedna, přičemž vah je nekonečně mnoho (Wikipedia, 2016b). V našem případě $\lambda = 2/(20 + 1) = 2/21 \doteq 0,095238$.

Nakonec získáme číslo, udávající procentuální změnu ceny. Je nutné rozhodnout, zda je tato změna významná či bezvýznamná. K tomuto účelu bude stanoveno neutrální pásmo změny – např. $(-3; +3)$, tedy ± 3 %. Pokud je výsledné hodnota v tomto intervalu, bude změna označena za konstantní a s ní související dokumenty budou vyřazeny ze zkoumání.

3.5.2 Postup pro analýzu č. 1

Obecný popis postupu při zpracování analýzy 1 poskytuje sekce výše. Nyní budou potřebné kroky přesně specifikovány. Konkrétně je tedy nutné:

1. Zvolit dokumenty.
2. Každý dokument předzpracovat a přiřadit mu třídu.
3. Dokumenty exportovat do textového souboru vhodné velikosti a formátu, který bude obsahovat třídu a text dokumentu.
4. Převést exportované dokumenty do vektorové reprezentace – vznikne soubor určitého formátu, který je možné zpracovat zvoleným nástrojem pro data mining. Pro minimalizaci velikosti souboru bude použit řídký (*sparse*) formát vektorů.
5. Zajistit vyváženost tříd ve vektorových souborech.
6. Nahrát vektorový soubor do zvoleného nástroje pro data mining a provést klasifikaci.
7. Získat a vyhodnotit výsledky klasifikace.

Krok 1 – volba zdrojových dokumentů

Nejdříve je nutné zvolit dokumenty, které budeme analyzovat. Dokumenty lze rozdělit podle dvou základních kritérií – typ dokumentu a firma, ke které se dokument váže. Bylo rozhodnuto, že budeme zkoumat každý typ dokumentu zvlášť. Pokud je dokumentů relativně málo (Yahoo články a Facebook příspěvky), je postup jasný – budou zvoleny všechny dokumenty.

Pokud je dokumentů obrovské množství (tweety a Facebook komentáře), je nutné určit způsob jejich výběru. Výhodou je, že tyto dokumenty obsahují informace o popularitě – pro tweety to je tzv. *retweet count* (počet sdílení), pro FB komentáře to je počet liků.

Pro každou firmu a každý den ze sledovaného období (bylo zadáno 2. 8. 2015 až 2. 4. 2016 – tedy 244 dnů resp. 8 měsíců) bude vybráno 40 nejpopulárnějších FB komentářů a 200 tweetů. FB komentáře ze zkoumají pro každou firmu, která má zadaný název FB stránky. Tweety se zkoumají pro 10 firem, vybraných z celkového počtu 21 firem, majících zadaný název Twitter účtu.

Pro firmy Apple, Google, Intel, Microsoft, General Electric a Bank of America byly zkoumány všechny typy tweetů. Pro firmy AT&T, Hewlett-Packard, United Parcel Service a Wells Fargo byly zkoumány pouze tweety typu 1, 3, 4 a nikoliv typu 2 – vyhledávání dle názvu firmy. Důvodem bylo, že vyhledávací výrazy „AT&T, HP, UPS, Wells Fargo“ obsahují velké množství irelevantních tweetů. Byly vybrány pouze firmy působící v technologickém sektoru, aby tweety neobsahovaly protichůdné zprávy, které v jednom sektoru mají pozitivní, zatímco v jiném negativní sentiment. Dvě banky byly zahrnuty z toho důvodu, že zprávy o nich mohou obsahovat i obecné informace, týkající se ostatních firem.

Dalším krokem může být zkontrolovat, zda se ve vybraných dokumentech nevyskytují duplicity. Bylo zjištěno, že pro Facebook dokumenty existuje mizivý podíl duplicit (0,1 %). V případě tweetů duplikáty existovat nemohou, jelikož primární klíč tweetu je také jeho reálné ID – DBMS sám zajistí, že nebude podruhé vložen tweet se stejným ID. Yahoo článků se stejným URL je v databázi 12,%, což je logické, jelikož jeden článek může být spojen s více firmami. Výhodou je, že tak lze lépe zobecnit, jak souvisí text článku a ceny akcií.

Krok 2 – předzpracování a třída dokumentu

Dokumentům je nutné přiřadit správnou třídu. Nejdříve jsou pro danou firmu získány ceny od určitého data. Výsledkem je asociativní pole formátu `datum => cena`. V základu je použita cena typu *Adjusted close*. Místo ní lze použít jinou proměnnou – klouzavý průměr či objem obchodů. Potom se zpracovává každý dokument:

1. Urči datum publikace dokumentu (různé typy dokumentů mají různě reprezentovaná data).
2. Zjisti procentuální změnu (h) ceny akcie pro zvolený počet dnů zpoždění.
3. Stanov, zda je změna h růst, pokles nebo konstantní. Pokud je konstantní, ukonči zpracování dokumentu (přeskoč jej).
4. Ulož dokument do pomocného pole dokumentů.

Kritickým místem je zde krok 2, tedy zjištění procentuální změny ceny akcie. Hlavním problémem je, že ceny akcií nejsou sledovány pro nepracovní dny (víkendy a svátky). V datech tedy máme „díry“. K vyřešení tohoto problému byl použit přístup spočívající v tom, že pro zadaný časový interval (3. 3. 2008–5. 4. 2016) jsou stáhnuty ceny z Yahoo Finance a v cyklu je pro každý den zkontrolováno, zda je pro něj dostupná cena. Pokud není, tak je pro daný den i vytvořena umělá hodnota p_i

dle vzorce

$$p_i = \frac{p_{i-1} + p_{i+1}}{2} , \tag{8}$$

kde p_{i-1} je *Adjusted close* nejbližšího předchozího a p_{i+1} je otevírací cena (*open*) nejbližšího následujícího pracovního dne. Jako cenu akcie pro nepracovní den tedy zvolíme aritmetický průměr dvou nejbližších dostupných cen. Kód (1) metody, která kvůli efektivitě pracuje na základě indexování pole, se nachází v příloze D.

Všechny hodnoty jsou uloženy do databáze, která tedy pro každý den ze zadaného intervalu obsahuje cenu akcie na konci dne. Následně můžeme pro každý dokument získat cenu k datu publikace (čas $t - 1$). K tomuto datu poté přičteme požadovaný počet dnů zpoždění (1, 2 nebo 3). Tím získáme cenu v čase t. Potom spočítáme procentuální změnu ceny d_p jako

$$d_p = \frac{p_t - p_{t-1}}{p_{t-1}} \cdot 100 . \tag{9}$$

V dalším kroku už jen zbývá vytvořit z d_p správnou třídu. Pokud hodnota d_p spadá do zadaného neutrálního intervalu $(-a, a)$, je vrácena hodnota „const“. Pokud leží mimo interval a je větší než nula, je vráceno „up“, jinak „down“. Je jasné, že rozmezí neutrálního intervalu je zásadní. Proto bude vyzkoušeno více hodnot horní (resp. dolní) hranice intervalu – konkrétně $a \in \{1, 2, 3, 4, 5\}$. Pokud má dokument přiřazenou neutrální třídu „const“, je vyřazen z dalšího zpracování.

Následně zpracujeme text zbylých dokumentů. Pro všechny typy dokumentu:

- Nahraď URL odkazy ekvivalentní třídou: „http://neco.cz“ → „XURL“. Je k tomu využit regulární výraz `https?://\S+` (inspirace z Go et al. (2009)).
- Nakonec převeď text na malá písmena (*lowercase*).

Pro Yahoo články se navíc odstraní tagy `<p> a </p>`. Tweety, Facebook příspěvky a komentáře vyžadují speciální zpracování (v uvedeném pořadí):

- Odstraň symboly @ (značí uživatelské jméno na Twitteru) a # (znak čísla, tzv. *hashtag* – označuje téma dokumentu): „#lifewith3m“ → „lifewith3m“.
- Nahraď emotikony ekvivalentní (pozitivní nebo negativní) třídou.
- Odstraň nadbytečné bílé znaky (`\t`, `\n` apod.) – zbudou pouze jednotlivé mezery.

Celý postup ukazuje kód 2 v příloze D.

Krok 3 – export souborů

Tento krok souvisí s krokem 1 a 2 (zvolení dokumentů) a krokem 4 (transformace dokumentů na vektory) resp. 5 (provedení klasifikace). Musíme zvážit dva faktory: Kolik dokumentů budeme mít k dispozici a jak velký soubor dokáží zpracovat programy použité v krocích 4 a 5. Z provedených experimentů vyplynulo, že pro Yahoo články je limitní množství 50 000 a pro Facebook dokumenty a tweety 100 000.

Export probíhá následovně: Jsou zvoleny zdrojové dokumenty (krok 1). Poté jsou zjištěny třídy těchto dokumentů, přičemž jsou vyřazeny ty neutrální (třída *const*), a text dokumentů je zpracován (krok 2) a uložen do pomocné proměnné. Nakonec je potřeba takto zpracované dokumenty zapsat do textového souboru, kde každý řádek představuje jeden dokument a má následující formát:

```
<třída>\tab<text>
...
2    giffgaff to introduce 4g cap on unlimited data tariff xurl apple
```

Pro třídu „up“ je zadána číselná hodnota 1, pro „down“ hodnota 2.

Výše uvedený postup probíhá postupně pro každou firmu. Na začátku je tedy vytvořen prázdný soubor, do kterého jsou poté zapisovány dokumenty. V tomto místě by bylo možné zajistit zajistit vyváženost tříd (aby soubor obsahoval pro každou třídu stejný počet dokumentů) pro danou firmu. Nicméně toto nezajistí, že výsledný soubor bude také vyvážený, jelikož ten obsahuje dokumenty různých firem. Tudíž nemá smysl v této fázi vyvážení provádět.

Krok 4 – převod dokumentů na vektory

Pro aplikaci metod strojového učení na textové dokumenty je nutné tyto převést do vektorové reprezentace. K tomu je použit program VecText[22], implementovaný v jazyce Perl. Program poskytuje jak grafické (GUI), tak i textové rozhraní (CLI). Nabízí mnoho možností pro vstupní a výstupní soubory, filtrování a vážení atributů a podporuje např. odstranění stop slov a stemming.

Vstupem do programu je soubor z kroku 3. Tento soubor je celý přečten, přičemž z textu každého dokumentu je odstraněno vše, co není součástí slova – interpunkce (mezery, tečky, pomlčky, uvozovky, závorky aj.), HTML tagy, čísla. Co zůstane, je považováno za slova a vytvoří slovník termů (atributů). Z dokumentů jsou poté (dle parametrů v oddílu filtrování) odstraněny určité termy. Následně jsou dokumenty převedeny na vektory, kde prvky mají hodnoty dané nastavením vah. Nakonec jsou vektory zapsány do souboru ve zvoleném výstupním formátu – na výběr je ARFF, C5, CLUTO, SVMlight, YALE plus obecné formáty CSV a SPARSE.

Nastavení filtrování má vliv na to, kolik a jakých atributů budou výsledné vektory obsahovat. Je žádoucí, aby toto množství nebylo příliš velké. V každém případě bude nastavena minimální délka slova (*minimal word length*) na 2. Otázkou je, zda použít *document* (v kolika dokumentech se term vyskytuje) nebo *global term* (kolikrát se term vyskytuje ve všech dokumentech) frekvenci. Logicky vždy platí $document \leq global$. Lepší se zdá být použití *document* frekvence. Důvodem je, že pokud se bude v jednom dokumentu term vyskytovat vícekrát a v ostatních ani jednou, bude tento zahrnut do výsledného modelu, i když nebude přispívat pro zobecnění výsledků na větší počet dokumentů.

[22] `https://akela.mendelu.cz/~darena/VecText/`

Klíčovou otázkou je volba vah. Jak je uvedeno v sekci 2.6.5, pro klasifikaci sentimentu se často používá váha Term Presence a délková nebo kosinová normalizace. Nicméně pro obecné aplikace (zejména v *Information retrieval*) se používá váha TF-IDF. Bude tedy použita jak váha TP, tak TF-IDF (bez normalizace). Třetím typem vektoru bude TF-IDF s kosinovou normalizací – to znamená, že každá hodnota termu i v dokumentu j se vydělí hodnotou n_j, vypočítanou pro každý dokument j následovně:

$$n_j = \sqrt{\sum_{i=1}^{m} (gw_i \cdot lw_{ij})^2} \,, \tag{10}$$

kde gw_i je globální váha termu i, lw_{ij} je lokální váha termu i v dokumentu j, m je počet termů v dokumentu j (Dařena, 2016).

Lze také nastavit, jaké termy bude program zkoumat – zda jednotlivá slova (unigramy) nebo uspořádané n-tice slov (n-gramy). Použití bigramů (n=2) nezvyšuje (výrazně) správnost klasifikace sentimentu (viz sekce 2.6.5). Pokud není použita filtrace, použití jiné možnosti než 1 (unigramy) výrazně zvyšuje počet atributů. Na druhou stranu, pokud je nastavená filtrace i relativně „slabá“, počet atributů se přibližuje počtu pro unigramy. Nicméně budou zkoumány pouze unigramy ($n = 1$).

Výstupní soubor bude mít formát SVMlight:

```
<třída> (<ID termu>:<váha>)+
...
2 830:1 844:1 1409:1 1411:1 1645:1 1805:1 1910:1 2016:1
```

Ten používá řídké vektory – atributy s váhou 0 zde nejsou zapsány. To výrazně snižuje velikost výsledného souboru. Formát SVMlight lze načíst jak v programu SVM$^{\text{light}}$, tak např. v knihovně scikit-learn, která bude v práci využita.

Následně vyvážíme třídy v souborech tak, aby pro každou třídu existoval stejný počet dokumentů.

Krok 5 – klasifikace

Po převedení dokumentů do vektorové reprezentace můžeme konečně na daný soubor číselných dat aplikovat vybrané algoritmy strojového učení. V sekci 2.6.5 je uveden přehled algoritmů, používaných pro určování sentimentu. Ukazuje se v něm, že dobrých výsledků dosahují Naive Bayes a SVM a používá se také Maximum Entropy.

Existuje mnoho komerčních i nekomerčních programů pro data mining. Mezi komerční programy patří IBM SPSS Modeler, SAS Enterprise Miner nebo Oracle Data Mining. Nekomerčními programy jsou Weka, RapidMiner nebo KNIME. Patří sem v podstatě i programovací jazyk R, který obsahuje mnoho integrovaných funkcí pro statistické výpočty a zobrazování grafů. Jednou z nejlepších implementací SVM je C program SVM$^{\text{light}}$, který podporuje mnoho tisíců podpůrných vektorů, stovky tisíc trénovacích příkladů a používá řídkou reprezentaci vektorů (Joachims, 1999).

Kromě toho pro řadu programovacích jazyků existují knihovny, implementující vybrané algoritmy strojového učení. Pro Python to je především knihovna scikit-learn. Ta ke správné funkčnosti potřebuje knihovny Numpy (zajišťuje práci s vícerozměrnými poli a maticemi) a SciPy (pomocí těchto polí poskytuje rychlé numerické metody). Pro zobrazení grafů lze použít knihovnu *matplotlib*. Práci s těmito nástroji popisují ve své knize Coelho a Richert (2015).

Bylo rozhodnuto, že pro provedení analýzy bude použita výše uvedená Python knihovna scikit-learn. Ta má následující výhody (Lorica, 2013):

- Kvalitní dokumentace a přehledná veřejná API.
- Algoritmy jsou vybírány a implementovány týmem expertů.
- Poskytuje nástroje pro většinu úloh strojového učení (shlukování, klasifikace, regrese atd.), přičemž zde není více soupeřících implementací jednoho algoritmu.
- Pomocí Numpy a Cython dosahuje při číselných úlohách podobné rychlosti jako kompilované jazyky.
- Je použitelná (na dostatečně výkonném serveru) pro analýzu velkých dat (Big Data). Jinak řečeno „snadno škáluje“.

Dokumentace (scikit-learn, 2013) uvádí celou řadu algoritmů pro učení s učitelem. Nás budou zajímat pouze ty pro klasifikaci. Pro provedení analýzy byly vybrány:

- Multinomial Naive Bayes,
- Bernoulli Naive Bayes,
- Logistic Regression (Maximum Entropy),
- Rozhodovací strom CART,
- Random Forest Classifier,
- Linear SVC – speciální implementace SVM s lineárním kernelem,
- SVM – RBF a polynomiální kernel (stupně 3).

Tyto algoritmy jsou podrobně popsány v mnohé literatuře resp. na výše uvedeném webu scikit-learn. Nicméně pro úplnost je v příloze C uveden jejich stručný popis. Každý zdrojový soubor bude rozdělen na trénovací (65 %) a testovací (35 %) data.

Krok 6 – vyhodnocení výsledků klasifikace

Po konci trénování je nutné vyhodnotit kvalitu vytvořených modelů na testovacích datech. Ty jsou tvořeny vzorovými objekty (instance, vzorky, případy, příklady), u kterých je známa jejich příslušnost ke třídě. Tu ale klasifikátor nezná a naopak se snaží pro tyto, jemu neznámé, objekty určit správnou třídu.

Tab. 10: Vzorová matice záměn pro dvě třídy

		Predikovaná třída	
		pozitivní	negativní
Skutečná třída	pozitivní	TP (true positive)	FN (false negative)
	negativní	FP (false positive)	TN (true negative)

Prvním krokem vyhodnocení je vytvoření tzv. matice záměn (*confusion matrix*) – viz tabulka 10 (Witten et al., 2011, s. 164). Součet $TP+FN$ určuje počet skutečně pozitivních instancí ve zkoumané množině. Součet $TP + FP$ určuje počet instancí, které byly klasifikátorem označeny jako pozitivní. To stejné platí pro negativní instance. V našem kontextu znamená pozitivní třída to, že cena akcie (firmy, o které dokument pojednává) stoupla a negativní třída to, že cena akcie klesla. Na základě tabulky 10 lze zkonstruovat metriky, kterou budou v práci použity (Manning a Schütze, 1999, s. 268–269):

- *Accuracy* (správnost): $A = \frac{TP+TN}{TP+TN+FP+FN}$ ... Podíl správných předpovědí na všech instancích.
- *Precision* (přesnost): $P = \frac{TP}{TP+FP}$ ... Podíl pozitivních správných předpovědí na všech pozitivně predikovaných instancích. Je to schopnost klasifikátoru neoznačit jako pozitivní instanci, která je negativní.
- *Recall* (míra úplnosti): $R = \frac{TP}{TP+FN}$ ... Podíl pozitivních správných předpovědí na všech skutečně pozitivních instancích. Je to schopnost klasifikátoru nalézt všechny pozitivní instance.
- *F-measure* (F1 skóre): $F = 2 \cdot \frac{P \cdot R}{P+R}$. Jedná se o harmonický průměr Precision a Recall, reprezentující celkovou výkonnost systému.

Popis provedení experimentů

Na základě možností uvedených v krocích výše lze vidět, že existuje velké množství parametrů, které mohou ovlivnit výsledky experimentů. Abychom našli ty nejlepší hodnoty parametrů, je nutné je všechny otestovat. Každý typ dokumentu bude zpracován zvlášť.

Textové soubory budou tedy exportovány na základě třech parametrů:

- Proměnná použitá jako cena akcie: konečná cena (`adjclose`), jednoduchý klouzavý průměr (`sma`), exponenciální klouzavý průměr (`ewma`).
- Počet dnů zpoždění: $1, 2, 3$.
- Horní (resp. spodní) hranice intervalu pro konstantní změnu ceny akcie: $\{1, 2, 3, 4, 5\}$.

Každý textový soubor bude poté převeden na vektory, přičemž budou použity tři různé typy vah: TP a TF-IDF bez normalizace, TF-IDF s kosinovou normalizací. Při tvorbě vektorů bude vždy nastavena minimální *document frequency* na 10 pro Yahoo články a na 5 pro ostatní typy dokumentů.

Nakonec bude každý soubor zpracován pomocí algoritmů, uvedených v kroku 5 této sekce. Pro každý experiment budou spočítány metriky uvedené v kroku 6.

V první fázi vznikne pro každý typ dokumentu až 45 TEXT souborů. Soubor vznikne, jen pokud alespoň jedna akcie (firma) z výběru bude splňovat stanovené parametry, určující vývoj ceny. Ve druhé fázi vzniknou pro každý tento soubor tři různé soubory, celkem tedy $3 \cdot 45 = 135$ DAT souborů. Každý tento soubor bude otestován výše uvedenými algoritmy, což dává celkem $135 \cdot 6 = 810$ experimentů (plus hypotetických 270 pro algoritmus SVM, který ale nakonec nebyl použit).

Parametry a výsledky experimentů budou průběžně zapisovány do CSV souboru, pomocí kterého pak bude provedeno výsledné vyhodnocení. Nabízí se otázka, jak se v tomto velkém množství dat orientovat. Bylo by možné použít tzv. *summary tree*, který popisuje Arias et al. (2013). Nicméně bude dostačující zpracovat data pomocí Excelu, který poskytuje funkce pro řazení a filtrování a podporuje vzorce.

3.5.3 Analýza č. 2 – zjištění významných slov

V sekci 2.5.2 bylo uvedeno, že metodami *feature selection* lze odstranit termy, které jsou irelevantní (bezvýznamné) pro klasifikaci. Nicméně lze takto také nalézt termy, které jsou naopak informačně významné. Bylo by proto zajímavé aplikovat zmíněné metody na soubory, vygenerované v analýze 1, a získat tak termy, které souvisí s pohybem ceny akcie. Samozřejmě je nutné zvolit soubory, které poskytly ten nejlepší výsledek klasifikace. Získaná slova poté rozdělíme dle třídy (nahoru/dolů) na pozitivní/negativní a můžeme je porovnat se slovníky, uvedenými v sekci 2.6.3, nebo je použít pro obohacení těchto slovníků.

Touto problematikou (i když z opačného směru – z hlediska stopslov) se ve své diplomové práci zabýval Krupník (2014). Autor tam zjistil, že pro generování stopslov je nejúspěšnější metoda CHI (*Chí Statistic*), následovaná GSS (koeficient *Galavotti-Sebastiani-Simi*) a IG (*Information Gain*). Byl zde implementován program (stopwords.pl), který analyzuje zvolené TEXT soubory (obsahují třídu a text dokumentu) a vygeneruje zvolený počet slov, které mají nejmenší nebo naopak největší informační hodnotu. Jak píše Krupník (2014), metoda CHI (χ^2) měří nezávislost mezi termem a třídou. Nejnižší hodnotou je nula – pro případ, kdy jsou term a třída naprosto nezávislé. Nejdůležitější jsou tedy termy s nejvyšší hodnotou.

Pro každý typ dokumentu budou vybrány soubory, poskytující nejlepší správnost (alespoň 0,69). Nad všemi těmito soubory bude poté spuštěn program `stopwords.pl` (metoda `generuj_seznam_reverse`)), který vygeneruje textový soubor, obsahující slova seřazená sestupně dle hodnoty CHI. Takto bude získáno 1 000 slov s přiřazenou třídou 1 nebo 2. Program byl upraven tak, aby se choval požadovaným způsobem

(vstup/výstup z/do určitého adresáře, výstup slova, hodnoty a třídy ve formátu CSV, reset obsahu souborů). Se svolením autora je přítomen v příloze A.

Program `stopwords.pl` bude spuštěn s následujícím konfiguračním souborem:

```
vystup=best1000.csv
zdroj=../input
log=logger.txt
metoda=kompilace
pocet=1000
min_delka_slova=2
min_globalni_vyskyt=5
```

3.5.4 Analýza č. 3 – sentiment dokumentů a pohyby cen akcií

V další analýze budeme chtít odpovědět na otázku: „Jak souvisí sentiment dokumentů a pohyby cen akcií?“ Konkrétně budeme zkoumat, zda počet pozitivních, negativních a neutrálních zpráv souvisí s tím, zda se cena akcie další den (resp. po N dnech) posune nahoru, dolů, nebo zůstane konstantní. K určení sentimentu využijeme slovníkovou metodu – konkrétně algoritmus VADER a zvolený slovník.

Algoritmus VADER je implementován v jazyce Python. Jak funguje, je dostatečně jasně popsáno v jeho zdrojovém kódu, který je v aktualizované podobě dostupný zde: `http://www.nltk.org/_modules/nltk/sentiment/vader.html`. Hlavní metoda `polarity_scores` vrací sentiment zadaného textu jako asociativní pole hodnot (negativní, neutrální, pozitivní, složená). Celkový sentiment určuje pole „compound“, které je získáno sečtením sentimentu každého slova z lexikonu (které je přítomné v dokumentu), upraveno dle pravidel a normalizováno, aby bylo od -1 do $+1$. Pole „pos, neu, neg“ určují podíly textu, které spadají do dané kategorie sentimentu (součet by měl být 1). Metrika compound tedy měří sentiment jednorozměrně, zatímco metriky Pos/Neu/Neg jej měří vícerozměrně.

Následující příklad zobrazuje určení sentimentu pro relativně pozitivní větu.

```
vader = SentimentIntensityAnalyzer()
sentence = 'At least it isn't a horrible book.'
print vader.polarity_scores(sentence)
>> {'neg': 0.0, 'neu': 0.637, 'pos': 0.363, 'compound': 0.431}
```

Nejdříve je nutné určit, do kolika tříd budeme rozdělovat dokumenty dle sentimentu (ten je reprezentován skórem „compound“). Bylo rozhodnoto, že stejně jako Hutto a Gilbert (2014b), použijeme tři třídy: pozitivní, neutrální, negativní. Je tedy nutné stanovit dvě hodnoty, dělící skóre $[-1; 1]$ na tři intervaly. Jak vyplynulo z korespondence s autorem VADERu (C. J. Hutto), on ve svém článku používal hodnoty $-0,05$ a $+0,05$, tudíž je použijeme také.

VADER je vyladěn a optimalizován pro určování sentimentu na úrovni vět. Je možné jej aplikovat i na celý dokument, což poskytne hrubý odhad sentimentu. Nicméně lepších výsledků dosáhneme, pokud text rozdělíme do vět a spočítáme

a sečteme sentiment každé věty. Aby tento součet byl v intervalu $[-1; 1]$, musíme jej normalizovat. Můžeme použít prostý průměr (počet vět) nebo vážený průměr (např. počet slov v každé větě). Pro jednoduchost bude použit první způsob.

Pro rozdělení textu na věty bude použita funkce `tokenize.sent_tokenize` knihovny NLTK. Ovšem bude to provedeno pouze pro Yahoo články. Facebook dokumenty a tweety budou zpracovány celé, jelikož jsou většinou tvořeny jednou nebo dvěma větami a obsahují speciální znaky (emotikony), kvůli kterým nelze jasně odlišit hranici vět. Získaný součet sentimentu (s) bude normalizován dle vzorce

$$s_n = \frac{s}{\sqrt{s^2 + \alpha}}\,, \tag{11}$$

kde α je nejvyšší očekávaná hodnota. Bude použita výchozí hodnota VADERu (15).

Algoritmus používá ve výchozím nastavení slovník VADER, uzpůsobený pro sociální média. Jak vyplývá z Hutto a Gilbert (2014b), byl s (relativním) úspěchem použit i pro recenze filmů a technických produktů a pro názorové novinové články. Z toho plyne, že tento slovník je zcela jistě vhodný pro analýzu tweetů a Facebook komentářů. Firemní Facebook příspěvky jsou různorodé (viz sekce 3.2.2), jedná se ale obvykle o krátké obecné texty, takže pro ně lze slovník použít také.

Nicméně Yahoo články jsou něco úplně odlišného. Jsou plné ekonomických a finančních výrazů, které ve slovníku VADER nejsou zastoupeny. Pro analýzu tedy bude nutné využít jiné slovníky, uvedené v sekci 2.6.3. Finanční slovníky jsou tři: *Loughran and McDonald Financial sentiment dictionary* (2 709 slov), *Henry word list* (189 slov) a *Hajek multi-word list* (256 slov a slovních spojení).

Loughran a McDonald (2011) uvádí, že většina slov, která jsou v obecných slovnících označená jako negativní, ve finančním kontextu negativní nejsou. Jako příklady uvádí slova *depreciation, liability, foreign, board*. Ve slovníku VADER je přítomné pouze „liability“ (pasivum), které zde opravdu má lehce negativní sentiment $(-0,8)$. Uvedený článek zkoumal *10-K fillings* – výroční zprávy akciových společností z USA. Je jasné, že v těchto zprávách je uvedena mj. účetní rozvaha a v komentářích se termín pasivum vyskytuje často. Nicméně novinové články se zabývají konkrétním tématem, týkajícím se firmy, a nemají většinou charakter pouhého souhrnu účetních výkazů. Navíc termín *liability* může znamenat i překážku či břímě, což se dá považovat za negativní slovo.

Bylo rozhodnuto, že bude vytvořen nový slovník resp. tři různé slovníky. Nejdříve tři výše uvedené finanční slovníky zkombinujeme do jednoho (ten bude obsahovat asi 3 000 slov). Vzniklý slovník sloučíme s VADER slovníkem, čímž vznikne první slovník. Bude provedena kontrola, zda se slova z různých slovníků nepřekrývají. Pokud tato situace nastane, bude ponecháno pouze slovo z finančního slovníku. Druhý slovník bude vytvořen přidáním slov získaných pomocí *Feature selection* do prvního slovníku. Třetí slovník budou tvořit pouze slova z *Feature selection*.

3.5.5 Postup pro analýzu č. 3

Algoritmus a slovník využijeme v následujícím postupu, který bude aplikován pro každou firmu a pro každý den (ze zadaného intervalu).

1. Získáme dokumenty (týkající se firmy), které byly publikovány v daný den.
2. Pro každý typ dokumentu provedeme vhodné předzpracování. Následně zjistíme počet pozitivních, neutrálních, negativních dokumentů. Nakonec vybereme maximum z tohoto počtu a to bude určovat celkový sentiment pro daný typ.
3. Můžeme také spočítat celkový sentiment pro daný den tak, že pro každý typ dokumentu odečteme počet pozitivních od počtu negativních dokumentů a tyto výsledky sečteme.
4. Určíme směr pohybu ceny akcie pro zpoždění 1, 2, 3 dny od daného dne. Opět je nutné zadat rozmezí pro neutrální třídu.
5. Všechny tyto údaje vložíme do pomocné proměnné.

Předzpracování dokumentů je v podstatě stejné jako u analýzy 2, až na to, že URL odkazy byly úplně odstraněny a emotikony byly ponechány. Po zpracování všech dnů jsou všechny údaje pro firmu uloženy do souboru.

Nakonec jsou na základě získaných údajů spočítány metriky *Accuracy*, *Precision* a *Recall*. Ty (pro danou firmu) určují, zda platí předpoklad, že pokud v daný den převládaly pozitivní dokumenty, cena akcie stoupla, pokud negativní, tak klesla a pokud neutrální, tak zůstala (přibližně) stejná. Nevýhodou je, že pro spočítání metrik máme k dispozici omezený počet instancí, daný počtem sledovaných dnů.

Jelikož mohou dokumenty nabývat tří možných tříd, má matice záměn $3 \cdot 3 = 9$ polí. Nejdříve je nutné tuto matici vytvořit a naplnit správnými údaji. Následně můžeme spočítat dané metriky. V případě *Accuracy* existuje pro celou matici jediná hodnota – jednoduše sečteme všechny správně klasifikované instance a vydělíme je celkovým počtem instancí v matici. Ale *Precision* a *Recall* je nutné spočítat pro každou třídu zvlášť. Pro celou matici nemá smysl tyto dvě metriky počítat.

Metriky se počítají pro každou hodnotu zpoždění a pro každý typ dokumentu, případně také pro celkový sentiment dne. Po zpracování firmy jsou metriky zapsány do CSV souboru. Na jeho základě poté bude provedeno celkového vyhodnocení.

Z výše uvedených informací vyplývá, že je nutné zvolit rozmezí pro neutrální třídu pohybu ceny akcie a vhodný typ cenové proměnné. Hodnoty budou určeny po vhodném zvážení důsledků dané volby, přičemž využijeme výsledků analýzy 1.

4 Výsledky

Tato kapitola obsahuje výsledky, dosažené na základě postupu, uvedeného v metodice.

4.1 Modul pro získávání dat (Data Getter)

Pro splnění cíle práce bylo nutné získat definovaná data. Bylo tedy nutné navrhnout databázi, implementovat modul pro získávání dat, ten následně nainstalovat na server a začít získávat data.

4.1.1 Databáze

Po zvážení požadavků na výsledný modul a informací, které chceme uchovávat (viz sekce 3.3 a 3.4), byla nejdříve navrhnuta databáze. Pro implementaci databáze byl vybrán relační DBMS MySQL (viz sekce 2.4.2). Proto byl pro návrh databáze použit program *MySQL Workbench* (verze 6.3), umožňující vytvořený fyzický ER diagram vyexportovat do SQL souboru, obsahujícího příslušné DDL příkazy pro MySQL.

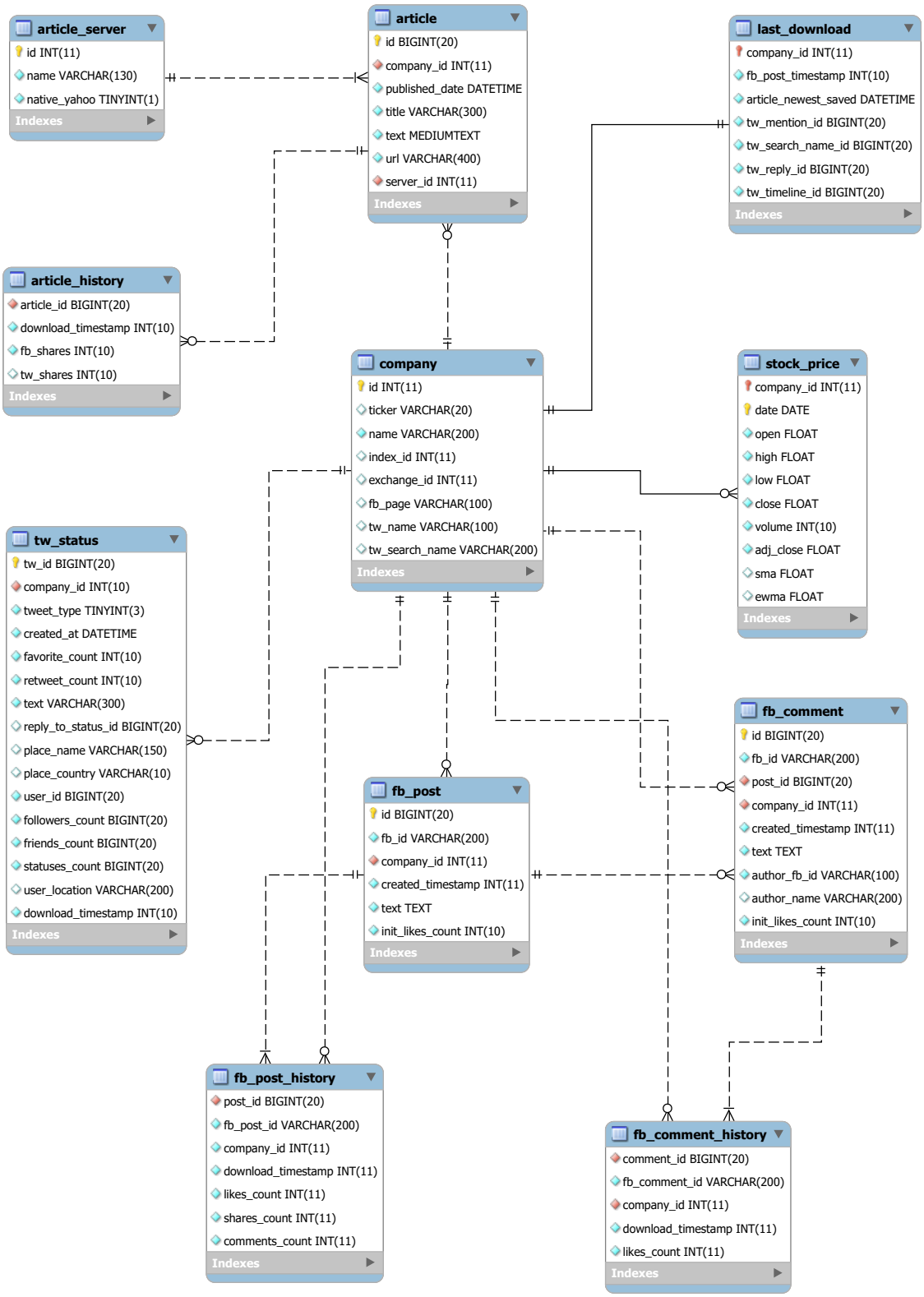
Obrázek 8 zobrazuje (v notaci *Crow's foot*) všechny entity (zde tabulky) uvedené v sekci 3.4, doplněné o znázornění vztahů (včetně kardinality a parciality), primární a cizí klíče a atributy (včetně datových typů).

Jako primární klíč byl u (téměř) každé tabulky zvolen umělý klíč jednotného jména *id*, což usnadní následnou implementaci (například bude možné bez problémů používat ORM). Výjimkou je tabulka `tw_status`, jejíž primární klíč není umělý, ale tvoří ho skutečné ID tweetu (64-bitové číslo). To mj. zajistí, že nebudeme uchovávat duplicitní tweety. Nyní budou jednotlivé entity resp. tabulky stručně popsány.

Dokumenty jsou uchovávány v tabulkách `article`, `fb_post`, `fb_comment` a `tw_status`. Jak lze vidět, obsahují mnoho atributů (především pro Twitter), popisujících dané dokumenty. Pro *FB post* jsou těmi hlavními Facebook ID příspěvku, jeho text, kdy byl vytvořen a počet liků v době, kdy byl stažen. *FB comment* má dva dodatečné atributy, identifikující jeho autora – ID a plné jméno. Tabulka `article_server` ukládá informace o zdrojových serverech pro Yahoo články (název serveru a jestli lze zpracovat nativní metodou).

Tabulky `fb_post_history` a `fb_comment_history` slouží ke sledování toho, jak se v čase měnily hodnoty určitých atributů (počet liků, komentářů, sdílení). Tabulka `article_history` udržuje informaci o tom, kolikrát byl článek sdílen na Facebooku a Twitteru. Doba sledování je nastavena na 14 dnů.

Tabulka `company` zaznamenává sledované firmy. Tabulka `stock_price` obsahuje informace o cenách akcií firmy a objemu obchodů pro daný den. Tabulka `last_download` ukládá pro každou firmu čas naposledy staženého FB příspěvku resp. článku a ID naposledy staženého tweetu daného typu. Tabulka `log_exec` v diagramu není uvedená, jelikož nemá žádné vazby a byla by tam matoucí. Obsahuje tři sloupce (`timestamp, script (int), was_error (boolean)`) a zaznamenává pro každý spuštěný skript čas jeho dokončení a zda během jeho běhu nastala chyba.



Obr. 8: Fyzický datový model – ERD pro modul DataGetter

Byl použit MySQL ve verzi 5.6.26 se *storage engine* InnoDB. Databázové schéma bylo upraveno tak, aby umožňovalo vyšší rychlost čtení a zapisování. Za prvé zde není použita kontrola cizích klíčů, což snižuje dobu vkládání nového záznamu do databáze. Toto by bylo možné nahradit zadáním příkazu `SET FOREIGN_KEY_CHECKS=0` před vkládáním dat a následným zadáním příkazu `SET FOREIGN_KEY_CHECKS=1` po ukončení vkládání. Tato možnost nebyla využita z důvodu, že toto řešení by bylo složitější na implementaci a také z důvodu, že autor práce o této možnosti při tvorbě modelu, která probíhala minulý rok v červenci, nevěděl. Za druhé byla provedena denormalizace, která umožní rychlejší a snadnější analýzu uložených dat. To lze vidět např. v tabulce `fb_comment`, ve které se nachází atribut `company_id`, který je již přítomen v tabulce `fb_post`, spojené s ní pomocí atributu `post_id`.

4.1.2 Struktura modulu

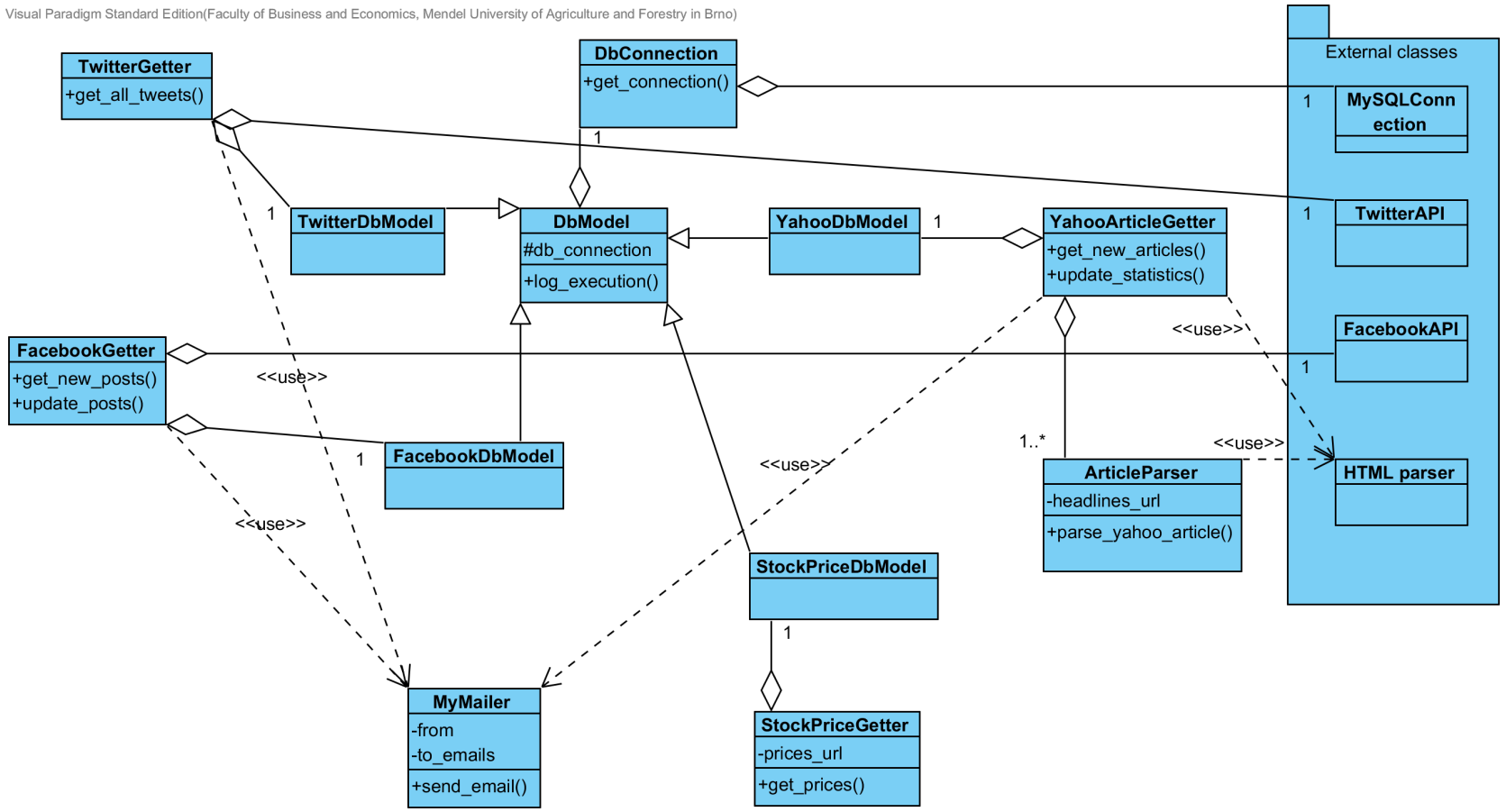
Po vytvoření databáze bylo dalším krokem vytvořit modul pro získávání dat. Nejdříve byl pomocí jazyka UML[23] vytvořen diagram tříd (*Class diagram*). Na obrázku 9 lze vidět, že se modul skládá z 12 tříd (plus je v plánu využít 4 externí moduly). Třídy lze rozdělit dle jejich funkce:

- Získávání dat: `FacebookGetter`, `YahooArticleGetter`, `TwitterGetter`, `StockPriceGetter`.
- Práce s databází: Třída `DbConnection` poskytuje pomocí návrhového vzoru Singleton připojení k databázi. Rodičovská třída `DbModel` ve svém konstruktoru toto připojení (resp. odkaz na něj) uloží jako svůj atribut. Připojení k databázi (resp. i další metody třídy) mohou využívat potomci `FacebookDbModel`, `TwitterDbModel`, `YahooDbModel`, `StockPriceDbModel`.
- Třída `MyMailer` slouží k odesílání informačních e-mailů.
- Třída `ArticleParser` slouží k parsování HTML článků, přičemž se počítá s možností, že případně budou vytvořeni potomci, specializující se na určité servery.
- Balík `External classes` obsahuje obecné definice tříd, které budou v modulu použity, ale nebudeme je implementovat.

V diagramu jsou uvedeny pouze vybrané veřejné metody a důležité atributy. Třídy pro práci s databází mají metody pro čtení a zapisování údajů do databáze. Údaje pro připojení k Twitter a Facebook API jsou zadány ve výkonných skriptech, které vytvoří objekt dané externí třídy a nastaví jej jako atribut instanci třídy typu Getter.

Modul je implementován v jazyce Python. Pro použití modulu v jiném Python skriptu je nutné načíst hlavní třídu pro požadovaný datový zdroj, vytvořit její instanci a následně na ní zavolat správnou metodu.

[23] Unified Modeling Language – univerzální vizuální jazyk pro modelování (nejen objektově orientovaných softwarových) systémů. UML se skládá ze tří základních stavebních kamenů: artefakty (prvky), vztahy mezi prvky, diagramy (Arlow a Neustadt, 2005, s. 5–11).



Obr. 9: Diagram tříd pro modul DataGetter

4.1.3 Facebook a Twitter

Facebook a Twitter mají REST API, se kterým lze poměrně snadno pracovat. Nicméně byly použity Python balíky[24], které práci s nimi ještě ulehčují: „twython“ pro Twitter a „facebook-sdk“ pro Facebook.

Facebook API vrací příspěvky v pořadí od nejnovějšího. Získané pole je tedy nejdříve obráceno. Potom se zkontroluje, zda se jedná o „běžný“ příspěvěk:

```
if 'message' not in post or 'shares' not in post or 'likes' not in post:
        continue
```

Pokud ne, je tento přeskočen. Při práci s Facebook API může nastat několik problémů, kdy API vrátí místo dat chybový kód. Zaprvé – občas se stává, že daná Facebook stránka změní jméno či je zrušena (chyby 803 a 21). V tomto případě přijde administátorovi chybový e-mail a je nutné manuálně upravit jméno stránky v databázi. Zadruhé – z některých stránek nelze vždy číst data (chyby 100 a 1). Je jich asi 10 a jsou především firem z EU. Je to pravděpodobně způsobeno nastavením v administraci, které zamezuje poskytování dat. Dále existuje limit, kolik dat může Facebook server poslat v jedné odpovědi. Pokud tento limit překročíme (požadavek by navrátil příliš mnoho dat), dojde k chybě -3. Toto se stává, pokud chceme číst příspěvky s mnoha komentáři z daleké historie. Při prvotním stahování dat musely být parametry upraveny tak, aby bylo možné stáhnout co nejstarší příspěvky s ještě dostatečným počtem komentářů (10–50 dle konkrétní stránky).

Twitter API vrací minimum chyb. Jedna nastane, když chceme číst tweety z profilu firmy, který je označen jako „protected“. K těmto tweetům se prostě nedostaneme a tak je nutné tento profil v databázi zneplatnit (zadat hodnotu NULL do pole `tw_name`). Výjimečně se změní Twitter jméno firmy. Potom na základě e-mailového upozornění musíme najít aktální jméno nebo profil zneplatnit.

Z tweetů jsou odstraněny všechny nadbytečné „bílé“ znaky (vícenásobné mezery, tabulátory, znaky nového řádku aj.) – zůstanou jen prosté mezery. FB dokumenty se ukládají v původní podobě (pro zachování eventuálně přítomné informace).

4.1.4 Yahoo Finance

Yahoo Finance nemá žádné API pro stahování článků, týkajících se dané firmy. Proto bylo potřeba stáhnout danou webovou stránku a získat data ručně. Pro parsování byla použita knihovna „beautifulsoup4“[25] s parserem „lxml“.

Nejdříve je prozkoumána stránka s titulky článků – ta vždy obsahuje datum a pod ním titulky. Při parsování jsou přeskočeny články z dnešního dne (v časové zóně EDT). Včerejší články nemají údaj o čase publikování. Můžeme tedy stahovat dva dny staré články (což umožní ukládat i přesný čas publikace) nebo i jeden den staré články – to ovšem znamená, že budeme znát jen datum publikace. Na začátku práce byla zvolena první možnost, přičemž toto rozhodnutí bylo později změněno.

[24] `https://pypi.python.org/pypi/facebook-sdk` a `https://pypi.python.org/pypi/twython`
[25] `https://www.crummy.com/software/BeautifulSoup/`

Po rozhodnutí o tom, jaký den se má zkoumat, jsou zpracovány jednotlivé články z daného dne. Článek je stažen z dané URL a je nalezen blok ( ) obsahu článku. Většinou se to podaří na základě atributu `class="mw_release"`. Pokud není takový blok nalezen, je vyhledán blok s atributem `itemtype="http://schema.org/Article"`.

Z HTML kódu bloku jsou vybrány pouze odstavcové ( ) tagy. V odstavci přítomné tagy jsou odstraněny a nahrazeny mezerami. Dokument tedy obsahuje text, oddělený znaky odstavce (text1 text2 …). To může být užitečné pro pozdější zpracování. Nakonec jsou z textu odstraněny „bílé“ znaky.

Problém nastal při zjišťování počtu sdílení článků přes veřejné API Facebooku. Po zaslání požadavků pro asi 100 firem začala služba odpovídat s chybou – byl zřejmě aplikován limit počtu volání z jedné IP adresy za určitý čas. Tudíž bylo nutné mezi voláními pro jednotlivé firmy nastavit pauzu (6 sekund). Pokud je počet sdílení článku nula, do tabulky `article_history` není vložen žádný záznam.

4.1.5 Funkčnost modulu

Jak již bylo uvedeno, modul slouží pro stahování, zpracování a ukládání dat z internetu. Samotné stahování zajišťují metody daných tříd, které jsou spouštěny výkonnými soubory (skripty). V seznamu níže je uveden název skriptu, volaná metoda a jaké dokumenty (resp. údaje) se při každém běhu skriptu stáhnou:

- `run_yahoo_new.py`: `YahooArticleGetter:get_new_articles()`: Yahoo články publikované od data naposledy uloženého článku (aktuální den je přeskočen).
- `run_yahoo_update.py`: `YahooArticleGetter:update_article_stats(14)`: Statistika počtu sdílení článků (pro články staré <= 14 dnů).
- `run_fb_new.py`: `FacebookGetter:get_new_posts()`: Facebook příspěvky, publikované od data naposledy uloženého příspěvku. S nimi se stáhne až 100 komentářů (vybraných dle nejvyššího počtu liků).
- `run_fb_update.py`: `FacebookGetter:update_posts(14)`: Statistika pro FB příspěvky a komentáře, přidání nových komentářů (pro příspěvky staré <= 14 dnů).
- `run_tw.py`: `TwitterGetter:get_all_tweets()`: Tweety, jejichž ID je větší než ID naposledy uloženého tweetu daného typu.
- `save_prices.py`: `StockPriceGetter:save_prices_for_all_companies (datetime.date(2000, 1, 15), yesterday(), True)`: Ceny akcií z daného časového období.

Pokud je pole `ticker`, `fb_page`, `tw_name` v tabulce `company` označeno jako NULL, nestahují se pro danou firmu články / Facebook dokumenty / tweety. Výkonné soubory jsou spouštěny automaticky pomocí programu *cron* v časech, které

ukazuje tabulka 11. „Update“ skript musí být spuštěn před „new“ skriptem, aby v tabulce historie nebyl pro daný den žádný duplikátní záznam. V rámci „new“ skriptu jsou totiž do tabulky historie vloženy aktuální údaje pro nově získané dokumenty.

Tab. 11: Časy spouštění skriptů pro získávání dat (DataGetter)

Soubor	Nastavení cronu	Popis
run_yahoo_new.py	30 6 * * *	každý den v 6:30
run_yahoo_update.py	30 3 * * *	každý den v 3:30
run_fb_new.py	0 2 * * *	každý den v 2:00
run_fb_update.py	20 0 * * *	každý den v 0:20
run_tw.py	0 */6 * * *	každých 6 hodin (0,6,12,18)
save_prices.py	0 10 * * *	každý den v 10:00

Po doplnění Twitter jmen pro všechny firmy byl interval spouštění `run_tw.py` nastaven na 1krát denně ve 12:00. Také byl přidán parametr údávající počet sekund, jak dlouho bude skript čekat mezi stahováním pro jednotlivé firmy. Bylo vypočítáno, že pro splnění limitů Twitter REST API musí být pauza 8 sekund.

4.1.6 Získaná data

Modul je nainstalován na virtuálním serveru, který je spravován Ústavem informačních technologií Mendelovy univerzity v Brně. Stahuje data každý den dle tabulky 11. Stahování započalo 1. 8. 2015 a k 4. 4. 2016 (celková doba je tedy 8 měsíců resp. 247 dnů) je velikost databáze asi 4 GiB, přičemž databáze obsahuje asi 25 milionů řádků. Tabulka 12 zobrazuje pro každou tabulku počet řádků, PVR (průměrnou velikost řádku v bytech) a velikost dat a indexů tabulky (v MiB).

Lze vidět, že v databázi jsou přes dva miliony Facebook komentářů a téměř 4 miliony tweetů. To představuje mnoho dat pro analýzu. Počet Facebook příspěvků se pohybuje okolo 135 tisíc. Množství novinových článků (asi 82 tisíc) není příliš vysoké, ale je nutno vzít v potaz, že článek je mnohem delší než komentář nebo tweet a obsahuje více dat (a potenciálně informací) ohledně názoru autora článku. Výhodou je, že články lze získat zpětně – Yahoo Finance obsahuje archiv článků, sahající až do roku 2013. Ceny akcií byly uloženy od 3. 3. 2008 do 7. 4. 2016.

Tabulka 13 zobrazuje, kolik dokumentů je průměrně stahováno pro všechny firmy a pro jednu firmu za den a měsíc. Pro výpočet těchto údajů byl použit časový interval od 20. 8. do 19. 9. 2015 (celkem 31 dnů tedy jeden měsíc).

Z tabulky 13 mohou být odvozeny některé zajímavé statistiky. Průměrný počet komentářů na jeden Facebook příspěvěk je 24. Zkušenost ukazuje, že mnoho příspěvků nemá žádné komentáře, ale některé mají i 100 či víc. Je zřetelné, že průměrná firma píše na svou Facebook stránku méně než jednou denně a je o ní publikován

článek asi každý druhý den. Nicméně tyto hodnoty se pro jednotlivé firmy značně liší – závisí na jejich velikosti, oblasti podnikání a popularity u veřejnosti.

Tab. 12: Celkové statistiky získaných dat (1. 8. 2015 až 4. 4. 2016)

Název tabulky	Počet řádků	PVR	Velikost dat	Velikost indexů
article	81 519	10 209	615,84	3,03
article_history	196 084	50	9,52	4,52
article_server	77	212	0,02	0,02
company	784	104	0,08	0,02
fb_comment	2 222 362	222	488,00	194,97
fb_comment_history	14 774 405	86	1 173,00	433,00
fb_post	134 941	292	36,56	9,03
fb_post_history	874 472	91	75,61	28,58
last_download	784	104	0,08	0,00
log_exec	1 871	52	0,09	0,00
stock_price	2 817 093	50	108,64	0,00
tw_status	3 887 527	232	785,00	137,27
Celkem	24 990 274	–	3 292,44	810,44

Tab. 13: Průměrný počet pravidelně stahovaných dokumentů

Typ dokumentu	Denně (celkově)	Měsíčně (celkově)	Denně (1 firma)	Měsíčně (1 firma)
FB příspěvky	261	8 086	0,71	21,97
FB komentáře	6 263	194 174	17,59	545,43
Yahoo články	266	8 244	0,40	12,55
Twitter statusy	16 260	504 062	774,29	24 003

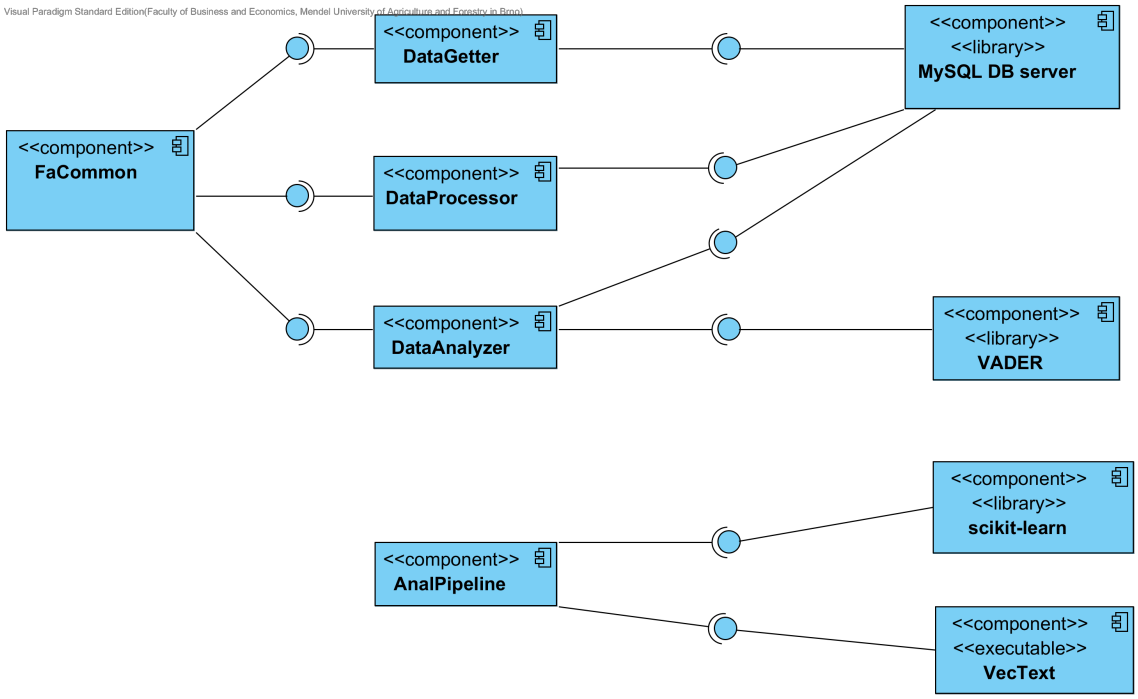
Jak bylo uvedeno v sekci 3.1, k 4. 4. 2016 je v databázi 775 aktivních firem (tj. těch, které jsou aktuálně obchodované na burze). Z toho 398 má Facebook stránku (jsou pro ně stahovány Facebook příspěvky a komentáře). Tweety se stahují pro 21 firem.

Tyto počty platí pro starou verzi aplikace, která byla aktivní do 4. 4. 2016. Od 5. 4. 2016 se sbírají Facebook data pro 431 firem a tweety pro všech 784 firem. Byla totiž nalezena a zadána Twitter jména pro všechny firmy v databázi. Pro aktualizaci databáze byl využit skript (`/DataGetter/helper_scripts/update_companies.py`), který bere na vstupu Excel soubor, jehož základ je do CSV exportovaná tabulka `company`, do které byla ručně zadána Twitter jména pro všechny firmy v databázi.

Nicméně v práci budeme zpracovávat data pouze z původní databáze, která byla (před aktivací nové verze aplikace) vytvořena exportem ze serveru Sosna.

4.2 Další implementované moduly

Pro realizaci úkolů uvedených v metodice bylo nutné navrhnout a implementovat další moduly: modul pro zpracování a export dat (DataProcessor), konverzi (převod textů na vektory) a klasifikaci dat (AnalPipeline) a pro analýzu sentimentu pomocí slovníků (DataAnalyzer). Všechny jsou naprogramovány v jazyce Python a jejich kód je zdokumentován standardním způsobem, což by mělo zajistit snadnou pochopitelnost a usnadnit případné budoucí úpravy. Na obrázku 10 je zobrazen komponentový diagram celého projektu FinaceAnalyzer. Komponenty vpravo jsou externí knihovny a programy, komponenty vlevo interní moduly.



Obr. 10: Komponentový diagram projektu FinanceAnalyzer

4.2.1 Modul DataProcessor

Modul pro zpracování dat zajišťuje kroky 1, 2 a 3 z metodiky (viz sekce 3.5.2). Stručně řečeno, získá dokumenty z databáze, přiřadí jim třídu a upraví text a vygeneruje TEXT soubor, určený pro převod na vektory. V modulu je 5 hlavních tříd. Třída `DocumentsExporter` je hlavní výkonná třída a obsahuje veřejné metody pro zpracování různých typů dokumentů.

Její privátní metoda `_process_given_documents` přijímá na vstupu vybrané dokumenty z databáze (pro danou firmu) a vrací pole upravených dokumentů. Pro každý dokument je získáno datum jeho publikace a směr pohybu ceny. Pokud je tento konstantní, přejde se na další dokument. Následně je upraven text. Pokud

je dokument prázdný nebo obsahuje pouze jeden znak, je přeskočen. Nakonec je dokument uložen do pomocného pole.

Privátní metoda `_write_docs_to_file` přijímá pole dokumentů, které následně pomocí instace třídy `TextWriter` zapíše do souboru, jehož název má formát:

```
<typ dokumentu>_<použité firmy>_<cenová proměnná>_<počet dnů zpoždění>
_<hranice konstantního intervalu>.text
```

Třída **`DocumentsExporterDbModel`** obsahuje metody pro čtení firem a dokumentů z databáze. Problémem při získávání tweetů byla skutečnost, že jako primární klíč tweetů bylo zvoleno reálné ID tweetu. Výhodou tohoto přístupu je, že se v databázi nemohou vyskytovat duplicity. Nevýhodou je, že tweety jsou ve *storage enginu* InnoDB uloženy dle pořadí primárního klíče, tudíž tweety z různých firem a dní se nacházejí ve velmi vzdálených blocích dat na disku a jejich čtení tak trvá velmi dlouho. Na toto bohužel nebyl při návrhu databáze brán zřetel. Z tohoto důvodu bylo nutné vymyslet způsob, jak čtení urychlit.

MySQL naštěstí podporuje cache (vyrovnávací paměť) pro SQL dotazy. Po správném nastavení se poté každý dotaz, který přesně (*byte-to-byte*) odpovídá již provedenému dotazu, neprovádí, ale pouze se získá jeho výsledek. Tohoto lze dosáhnout nastavením `query_cache_type=2` a následným spuštěním (pro každou z 10 firem) požadovaného SQL dotazu s příznakem `SQL_CACHE`. Bylo zjištěno, že export tweetů obsahuje duplicity, proto byla přidána klauzule GROUP BY, která je odstraní. Metoda pro tweety (kód 3) je uvedena v příloze D.

Třída **`StockPriceProcessor`** slouží pro získání třídy dokumentu na základě rozdílu cen akcie firmy. Třída **`StockPriceTransformer`** zajišťuje tvorbu klouzavých průměrů z původních cen v tabulce `stock_price`. Jsou zde dvě metody (SMA a EWMA) pro tvorbu a uložení hodnot průměru do databáze pro jednu firmu, plus metoda provádějící toto pro všechny firmy. Pro SMA byl vytvořen vlastní algoritmus (viz kód 4 v příloze D). Pro EWMA byl využit Python modul „pyma“[26]. Konečně třída **`DbModel`** obsahuje připojení k databázi a metody pro získání a aktualizaci cen.

Výkonný soubor `export_docs.py` obsahuje definici výstupního adresáře, která je zadána do konstruktoru třídy `DocumentsExporter`. Poté jsou pomocí funkce `itertools.product` vytvořeny všechny kombinace třech parametrů, uvedených v sekci 3.5.2, a jsou vyexportovány požadované dokumenty.

4.2.2 Modul AnalPipeline

Po vyexportování dat do formátu TEXT je nutné soubory převést do vektorové reprezentace a provést nad nimi klasifikaci. K tomu slouží modul pro konverzi a analýzu dat jménem AnalPipeline (zkratka z Analyzing Pipeline).

[26] `https://bitbucket.org/aideaucegep/pyma`

Převod textů na vektory pomocí VecText

Skript `docs_to_vectors.py` používá program VecText k převodu TEXT souborů ze zadaného adresáře do DAT souborů ve výstupním adresáři. Pro každý typ dokumentu je definována `min_document_frequency` – pro Yahoo články je 10, pro ostatní 5. Minimální délka slova je nastavena na 2. Jelikož převod trvá velmi dlouho (22h pro FB komentáře), bylo nastaveno, že se bude zpracovávat pouze prvních 100 000 řádků vstupního souboru. Tento počet v drtivé většině případů není dosažen.

Byly definovány tři konfigurace pro tři typy vah vektorů. Dle formátu `<lokální váha>-<globální váha>-<normalizace>` se jedná o:

- `tp-no-no`: Lokální váha TP, žádná globální váha ani normalizace.
- `tf-idf-no`: Lokální váha TF, globální váha IDF, normalizace žádná.
- `tf-idf-cos`: Stejně jako výše s kosinovou normalizací.

Výstupem je soubor ve formátu řídké matice SVMlight. Při převodu je také vytvořen soubor se statistikami. Všechny výše uvedené parametry jsou zaznamenány v názvu souboru, který má následující formát:

```
<typ dokumentu>_<all|company ID>_<cenová proměnná>_<počet dnů zpoždění>
_<hranice konst. intervalu>_<lokální-globální-normalizace>.SVMlight.dat
```

Kód níže ilustruje podobu příkazu pro VecText pro Yahoo článek:

```
perl vectext-cmdline.pl --logarithm_type="natural" --min_document_frequency="10"
--input_file="article_all_ewma_1_1.text" --output_format="SVMlight"
--min_word_length="2" --encoding="utf8" --normalization="none" --case="lower case"
--subset_size="100000" --output_decimal_places="6" --create_dictionary="no"
--output_file="article_all_ewma_1_1_tp-no-no" --class_position="1"
--local_weights="Binary (Term Presence)" --global_weights="none" --n_grams="1"
--output_dir="\outputs\vec_text\article" --sort_attributes="none" --print_statistics
```

Vyvážení tříd v souborech

Pro zajištění objektivních výsledků bylo provedeno vyvážení tříd v souborech vektorů – tedy aby v daném souboru byl pro každou třídu přítomen stejný počet dokumentů. K tomu byl použit skript `balance_files.py`. Ten využívá metodu `file_processing.balance_files`, která pro každý soubor ze vstupního adresáře provede následující: Nejdříve přečte ze STAT souboru (vygenerovaného při převodu textů na vektory) počty tříd a zjistí tak počet méně zastoupené třídy (`min_class_count`). Tento počet poté předá do metody `_balance_given_file`, která postupně čte zdrojový soubor, počítá počet dokumentů třídy 1 a 2 a pokud je tento $<=$ `min_class_count`, zapíše řádek do nového souboru ve výstupním adresáři. Pokud soubor obsahuje pouze jednu třídu, je vyřazen ze zpracování. Zmíněné metody se nacházejí v kódu 5 v příloze D.

Klasifikace pomocí scikit-learn

Posledním krokem bylo samotné provedení klasifikace (`scikit_anal_bulk.py`). K té byla použita Python knihovna scikit-learn a zde implementované algoritmy:

- Multinomial Naive Bayes (`NB-multi`): $\alpha = 1.0$ ... tzv. *Laplace smoothing*,
- Bernoulli Naive Bayes (`NB-berno`),
- Logistic Regression (`MaxEnt`),
- Rozhodovací strom CART (`CART`),
- Random Forest Classifier (`RandForest`),
- Linear SVC (`LinearSVC`).

Při prvotním testování bylo zpozorováno, že algoritmus SVM s RBF a polynomiálním kernelem poskytuje překvapivě špatné výsledky a jeho trénování trvá ve srovnání s ostatními algoritmy mnohem déle. Z tohoto důvodu nebyl vůbec použit.

Ve skriptu jsou definovány vstupní adresáře, obsahující SVMlight soubory. Nejdříve je vytvořen souhrnný soubor výsledků a je do něj zapsána hlavička (viz níže). Poté je každý datový soubor načten a data jsou pomocí funkce `cross_validation.train_test_split` rozdělena na trénovací (65 %) a testovací (35 %). Toto rozdělení je náhodné, ale byl specifikován parameter (`random_state=47`) pro tzv. „semínko“ pseudonáhodného generátoru čísel, což zajistí, že při každém spuštění metody (na stejných datech a stejné platformě) bude toto rozdělení stejné. Následně je každý klasifikátor natrénován a otestován (metoda `classification.classify_data` – viz kód 6 v příloze D). Je také možné uložit výsledný natrénovaný modul na disk – zda se tak stane, určuje konstanta `SAVE_CLF_MODEL_TO_DISK` (výchozí hodnota je *False*). Výsledky testování jsou zkoumány metrikami *accuracy, precision, recall, F1 score*, kromě prvního váženými dle *support* (počtu instancí dané třídy).

Výstupem je soubor výsledků `*.csv`, kde řádky představují výsledky algoritmů, spuštěných na datovém souboru, a sloupce představují charakteristiky experimentu:

- `timestamp` – čas spuštění experimentu.
- `input_file` – název vstupního souboru (bez přípony).
- `doc_type` – typ zkoumaného dokumentu: `article` (Yahoo článek), `fb-post` (Facebook příspěvěk), `fb-comment` (Facebook komentář), `tweet` (Twitter status).
- `company` – zkoumané firmy: `all`, `fb-com-40pd`, `twitter-10`.
- `variable_type` – zkoumaná cenová proměnná: `adjclose`, `sma`, `ewma`.
- `days_delay` – počet dnů zpoždění: $1, 2, 3$.
- `const_border_top` – horní hranice konstantního intervalu: $1, 2, 3, 4, 5$.

- `vector_type` – typ vektorů: `tp-no-no`, `tf-idf-no`, `tf-idf-cos`.
- `total_samples` – celkový počet dokumentů v souboru.
- `n_features` – počet atributů souboru.
- `class_1_test_samples` – počet testovacích dokumentů třídy 1.
- `class_2_test_samples` – počet testovacích dokumentů třídy 2.
- `algo_name` – název použitého algoritmu.
- `accuracy, precision, recall, f1_score` – hodnoty metrik (vážené).
- `train_time` – čas trénování v sekundách (vytvoření modelu – metoda *fit*).
- `test_time` – čas testování v sekundách (metoda *predict*).

Dále je ve výstupním adresáři v adresáři `logs` vytvořen logovací soubor, obsahující navíc *classification report* a *confusion matrix*, vygenerované pomocí funkcí scikitu.

4.2.3 Modul DataAnalyzer

Tento modul slouží pro určování sentimentu pomocí slovníku. Původně měl zajišťovat i funkce modulu AnalPipeline. Ten byl ale nakonec vydělen do samostatné jednotky, aby byl lépe přenositelný (na virtuální server Sosna).

Hlavní třídou je `DocumentsAnalyzer`. Její metoda `analyze_company` zajišťuje získání, předzpracování a analýzu sentimentu všech dokumentů pro danou firmu. Jejím výstupem jsou tři CSV soubory:

1. Hlavní soubor, obsahující pro každou firmu a každý den počet pozitivních, negativních a neutrálních dokumentů každého typu. Dále zde je pohyb ceny za 1, 2, 3 dny a celkový sentiment pro typy dokumentů a pro celý den.
2. Soubor s metrikami dle typu (zdroje) dokumentu, který pro každou firmu obsahuje pro každý zdroj a zpoždění hodnotu *accuracy*, průměr pro *precision* a *recall* a hodnoty těchto dvou metrik pro každou ze tří tříd.
3. Soubor s celkovými metrikami, obsahující pro každou firmu a každé zpoždění řádek s *accuracy*, *precision* a *recall*.

Soubor `dict_anal_docs.py` zajišťuje samotné spuštění analýzy sentimentu dokumentů pomocí slovníků. Je zde definován výstupní adresář a vytvořena instance třídy `DocumentsAnalyzer`. Dále jsou definovány parametry: typ cenové proměnné, názvy slovníků a hranice konstantního intervalu pro pohyb ceny akcie. Nakonec je v cyklu, pro každý slovník, vytvořen základ názvu výstupních souborů a je zavolána metoda `analyze_all_companies`. Ta nejdříve resetuje tři uvedené soubory, zapíše do nich hlavičku a provede analýzu pro všechny firmy, přičemž do souborů postupně zapisuje zjištěné výsledky.

Určení sentimentu dokumentu zajišťuje třída `LexiconSentimentAnalyzer` a její metoda `calculate_vader_sentiment`, která přijímá jméno slovníku, vstupní text a parameter, zda text rozdělit na věty (výchozí je *False*). Tato metoda volá metodu `polarity_scores` třídy `SentimentIntensityAnalyzer` z knihovny VADER a získanou číselnou hodnotu zkonvertuje (s použitím nastaveného neutrálního intervalu) na řetězec „neu“, „pos“ nebo „neg“. Konstruktor této třídy byl upraven tak, aby při zadání názvu slovníku byl jeho obsah načten a uložen do proměnné objektu. Díky tomu lze zajistit, aby při každém volání metody `calculate_vader_sentiment` byla, pokud se nezměnilo jméno aktuálního slovníku, vrácena jen uložená struktura, a soubor slovníku tak nemusel být čten znovu.

Načtení zvoleného slovníku má na starost třída `LexiconReader` resp. její metoda `get_dictionary`, která dle názvu slovníku v parametru zavolá konkrétní metodu, zajišťující přečtení souboru z disku a uložení slov do asociativního pole tvaru `slovo => polarita`. Byly implementovány metody pro čtení všech získaných slovníků a samozřejmě také metoda pro čtení slovníků, vytvořených v této práci.

Třídy `TotalMetricsCalculator` a `SourceMetricsCalculator` přijímají ve své hlavní metodě především data, která zapisuje třída `DocumentsAnalyzer` do hlavního souboru. Na jejich základě poté „metrická“ třída připraví matici záměn, tu poté naplní, spočítá metriky a ty nakonec vypíše do výstupního souboru.

4.3 Analýza č. 1 – klasifikace

Cílem analýzy 1 bylo zjistit, jak souvisí obsah dokumentů a pohyby cen akcií. Dle metodiky (viz 3.5.2) bylo nutné zvolit zdrojové dokumenty, přezpracovat je, exportovat, převést na vektory a nakonec klasifikovat. Následuje krátké shrnutí.

Třídy byly 1 (pohyb ceny nahoru) a 2 (dolů). Pohyb ceny byl zkoumán na základě rozdílu mezi cenou *adjusted close* v den publikace dokumentu a cenou za 1, 2, 3 dny (parametr zpoždění). Místo samotné ceny byl použit také jednoduchý (SMA, $n = 5$) a exponenciální (EWMA, $n = 20$) klouzavý průměr (parametr cenová proměnná). Důležitý byl parametr udávající rozmezí konstantního pohybu ceny, který byl (-1,1) až (-5, 5). Pro nepracovní den byla cena akcie stanovena jako $(close_{t-1} + open_{t+1})/2$. Pro každou kombinaci tří výše uvedených parametrů vznikl jeden testovací TEXT soubor (celkem tedy až 45 souborů pro každý typ dokumentu).

Volba zdrojových dokumentů závisela na jejich celkovém dostupném množství. Pro Yahoo články byly vybrány všechny dokumenty (resp. do limitu 50 000 v jednom souboru, který nebyl nikdy překročen). Pro Facebook příspěvky taktéž (do limitu 100 000). Facebook komentáře byly zpracovány pro všechny firmy (398), přičemž pro každý den a každou firmu bylo vybráno 40 komentářů (od nejvyššího počtu liků). Twitter statusy byly zkoumány pro 10 vybraných firem a pro každý den bylo ukládáno 200 tweetů (od nejvyššího počtu retweetů). V rámci exportu byl pro každý dokument spočítán směr pohybu ceny (nahoru, dolů, konstantní). Dokumenty s konstantním směrem pohybu byly vyřazeny z dalšího zpracování. Zbylé dokumenty byly předzpracovány, exportovány, převedeny na vektory a nakonec klasifikovány.

4.3.1 Postup zpracování výsledků

Pro každý typ dokumentu byl vygenerován CSV soubor, obsahující výsledky provedených experimentů. V následující části jsou tyto výsledky prezentovány a vyhodnoceny. Bude nás v podstatě zajímat, jaké parametry vedou k nejlepším výsledkům (měřeno správností). Především budeme zkoumat typ cenové proměnné, zpoždění a hranici konstantního intervalu (plus použité algoritmy a typy vektorů).

Konkrétně nás bude zajímat:

- Jaká proměnná / zpoždění / hranice intervalu poskytuje nejlepší výsledky?
- Jaký algoritmus a typ vektorů poskytuje nejlepší výsledky?
- Jaké výsledky jsou u souborů s více než 500 dokumenty?
- Jak se mění výsledky s počtem dokumentů / atributů v souboru?
- Jak obecně použitelné jsou dosažené výsledky?

CSV soubor bude otevřen v Excelu, kde budou vytvořeny následující statistiky:

1. Celková analýza: charakteristiky základních metrik souboru (správnost, F1 skóre, počet atributů a dokumentů, čas trénování).
2. Počet experimentů pro zvolené intervaly počtu dokumentů.
3. Počet dokumentů a atributů vs. průměrná správnost resp. čas trénování.

Následně budou ze zdrojových dat vybrány experimenty s alespoň 500 dokumenty. Experimenty s méně než 500 dokumenty nebudou zkoumány, jelikož by výsledky měly malou zobecnitelnost. Experimenty budou zpracovány, přičemž je potřeba:

1. Spočítat charakteristiky základních metrik souboru.
2. Seřadit soubor sestupně podle správnosti.
3. Stanovit hranice pro 5, 10, 25, 50 a 100 % nejlepších výsledků – toto budou hodnocené skupiny.
4. Pro každou skupinu určit, jak procentuálně zastoupená je určitá hodnota parametru experimentu: cenová proměnná, zpoždění, hranice intervalu, typ vektoru, algoritmus.
5. Vytvořit shrnující tabulku, která pro každou skupinu a parametr zobrazuje nejčastější hodnoty.

Nakonec budou výsledky slovně okomentovány. První typ dokumentu bude popsán velmi detailně, další typy budou popsány méně detailně. Kompletní výsledky se nacházejí v Excel sešitech v el. příloze A. Poznámka: Z důvodu úspory místa bude dále místo „Graf na obrázku X ukazuje…“ používáno pouze „Obrázek X ukazuje…“

4.3.2 Analýza č. 1 – Yahoo články

Zde jsou zanalyzovány a okomentovány výsledky experimentů pro Yahoo články.

Yahoo články – celková analýza

Tabulka 14 ukazuje charakteristiky hlavních metrik. Sloupec průměr zobrazuje aritmetický průměr, sloupec vážený průměr jej váží počtem dokumentů. Lze vidět, že všechny 4 metriky úspěšnosti klasifikace jsou téměř stejné. Tudíž je možné detailně zkoumat pouze jednu – konkrétně byla vybrána *Accuracy* (správnost).

Data jsou (téměř stejnoměrně) vyvážená. Důvodem mírných odchylek mezi počtem testovaných dokumentů třídy 1 a 2 je, že testovací data jsou vybrána náhodně z vyváženého souboru dat a tedy není možné zaručit naprosto stejný počet.

Tab. 14: Yahoo – hlavní metriky (analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5479	1,0000	0,6526	0,6265	0,6140
Precision	0,5479	1,0000	0,6565	0,6303	0,6163
Recall	0,5479	1,0000	0,6526	0,6265	0,6140
F1 skóre	0,5333	1,0000	0,6504	0,6250	0,6119
Počet dokumentů	46	45 342	10 949	6 212	–
Počet atributů	226	32 827	13 314	11 967	–
Čas trénování [s]	0,0007	210,8243	6,8202	0,1853	–

Byl vytvořen graf závislosti průměrné správnosti na počtu dokumentů. Pro jeho tvorbu bylo nutné rozdělit počet dokumentů do intervalů a pro každý najít, jaká byla průměrná správnost experimentů. Tabulka absolutních četností byla vytvořena následujícím postupem (Otipka a Šmajstrla, 2013):

1. Určíme variační rozpětí: $R = max - min = 45\,342 - 46 = 45\,296$
2. Určíme počet tříd: $k = \sqrt{n} = \sqrt{810} \doteq 28$
3. Určímě šířku třídy: $h = R/k = 45\,296/28 \doteq 1618$
4. Zvolíme horní mez prvního intervalu: $x_0 = min + h$
5. Vytvoříme intervaly ve tvaru $\langle x, y)$, kde y je vždy hodnota x zvýšená o h.
6. Pro každý interval spočítáme absolutní četnost hodnot (počet dokumentů v experimentu), které do něj spadají.

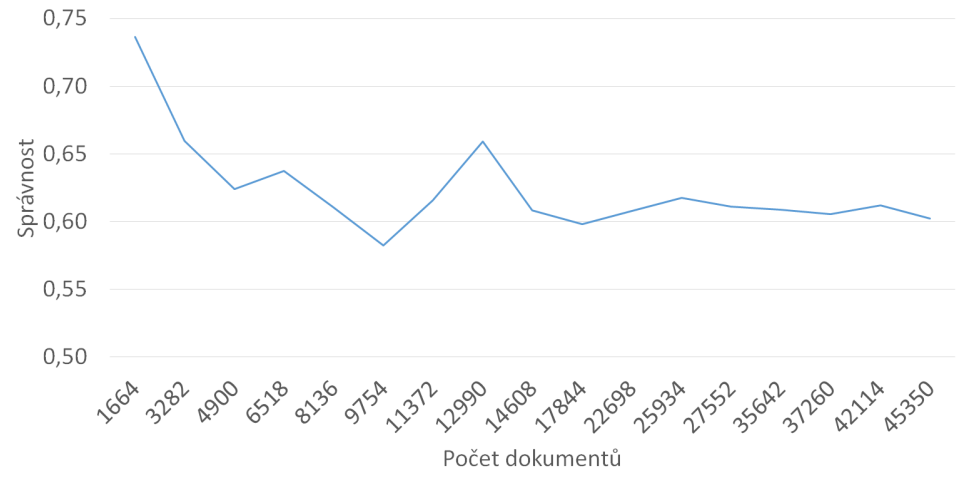
Výsledný obrázek 11 ukazuje, že průměrná správnost nejdříve s počtem dokumentů klesá (s výkyvem v hodnotě 12 990), přičemž od hodnoty asi 15 000 se stabilizuje

na konstantní úrovni okolo 0,6. Byl také vytvořen obrázek 12, zobrazující mírně exponenciální nárůst průměrného času trénování na počtu dokumentů (se zvláštním zalomením na konci). Dále byl zkoumán vliv počtu atributů (slov) na průměrnou správnost – postup byl stejný jako pro počet dokumentů (šířka třídy byla 1165). Výsledný obrázek 13 ukazuje, že průměrná správnost s počtem atributů klesá, přičemž od hodnoty asi 14 000 se stabilizuje na konstantní úrovni těsně nad 0,6. Obrázek 14 zobrazuje jasnou roustoucí exponenciální závislost času trénování na počtu atributů.

Čtyři výše uvedené grafy ukazují poměrně očekávané výsledky, tudíž pro další typy dokumentů nebudou prezentovány.

Průměrnou správnost pro jednotlivé algoritmy a typy vektorů ukazují obrázky 38 a 39 v sekci 4.3.7 (pro všechny typy dokumentů – toto dále nebude uváděno).

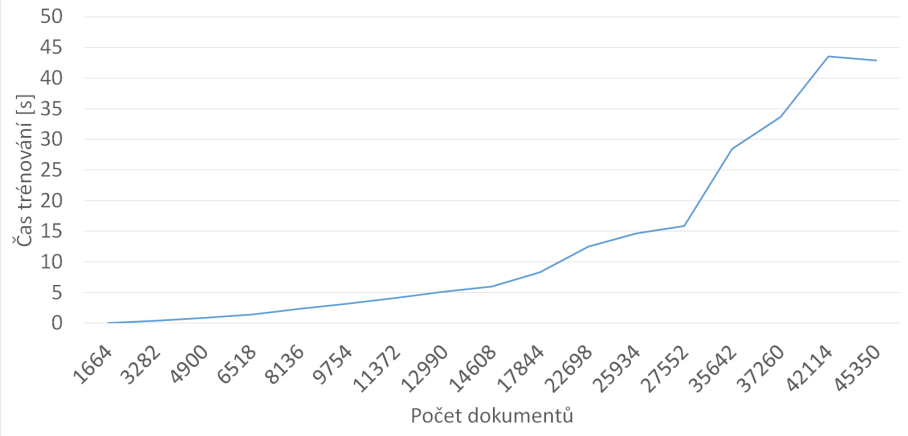
Tabulka 15 zobrazuje počet experimentů pro zvolené intervaly počtu dokumentů. Lze vidět, že experimentů s méně než 500 dokumenty bylo 8,89 %. Průměrná správnost těchto dokumentů je vysoká (0,84) a jejich zobecnitelnost nízká (obsahují 8–49 instancí jedné třídy), tudíž v detailní analýze nebudou zkoumány.



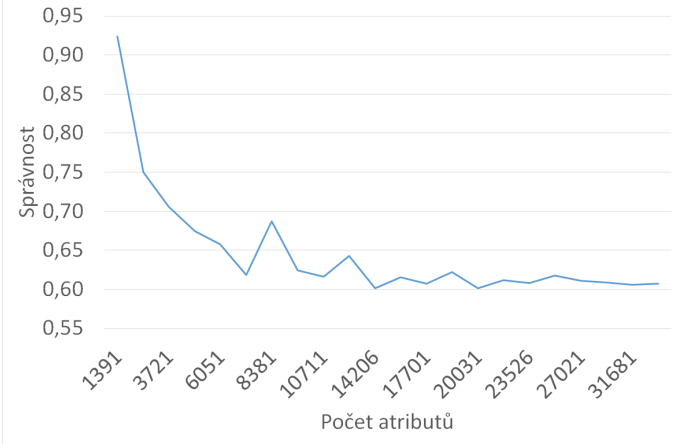
Obr. 11: Průměrná správnost v závislosti na počtu dokumentů (Yahoo)

Tab. 15: Yahoo – počty dokumentů (analýza 1)

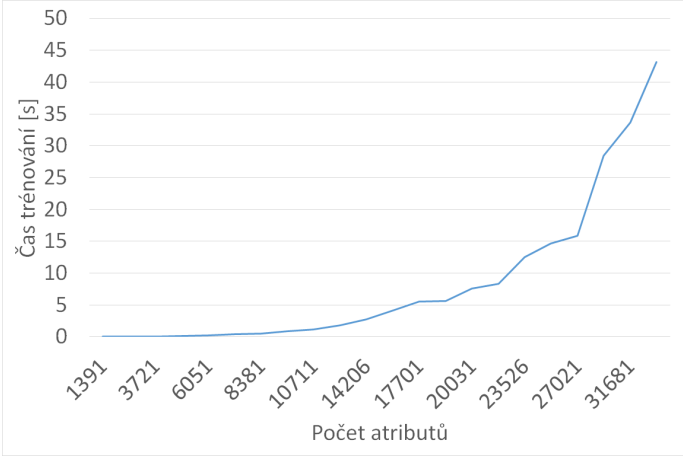
Počet dokumentů	Počet experimentů	Podíl [%]
vše	810	100,00
$(0; 500)$	72	8,89
$[500; 1000)$	90	11,11
$[1000; 10000)$	342	42,22
$[10000; \infty)$	306	37,78



Obr. 12: Průměrný čas trénování v závislosti na počtu dokumentů (Yahoo)



Obr. 13: Průměrná správnost v závislosti na počtu atributů (Yahoo)



Obr. 14: Průměrný čas trénování v závislosti na počtu atributů (Yahoo)

Yahoo články – nejlepší výsledky

Pro všechny zastoupené zdrojové soubory (bylo jich 45) byla vybrána hodnota maximální správnosti a algoritmus a typ vektoru, pomocí kterého byla dosažena.

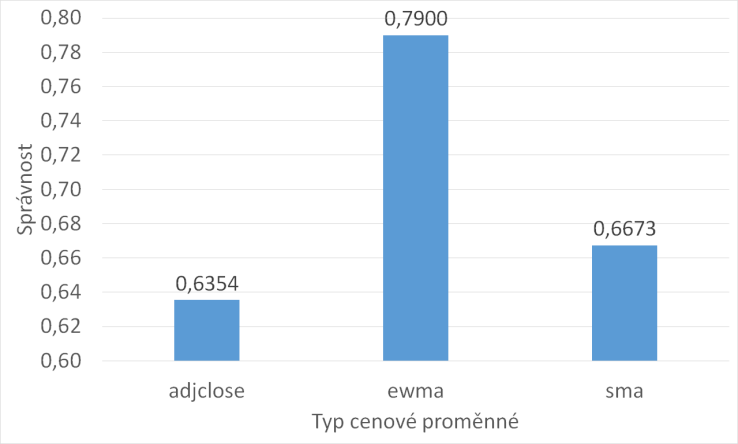
Tabulka 16 zobrazuje 15 zdrojových souborů s nejlepší správností (další v tabulce 54 v příloze G). Sloupec PD značí počet dokumentů, sloupec PA počet atributů. Typ souboru je ve formátu `<cenová proměnná>_<počet dnů zpoždění>_<hranice konst. intervalu>`. Lze vidět, že i cenová proměnná SMA je schopná dosáhnout více než 70% správnosti, i když v drtivé většině případů převládá EWMA.

Tab. 16: Yahoo – nejlepší soubory (analýza 1)

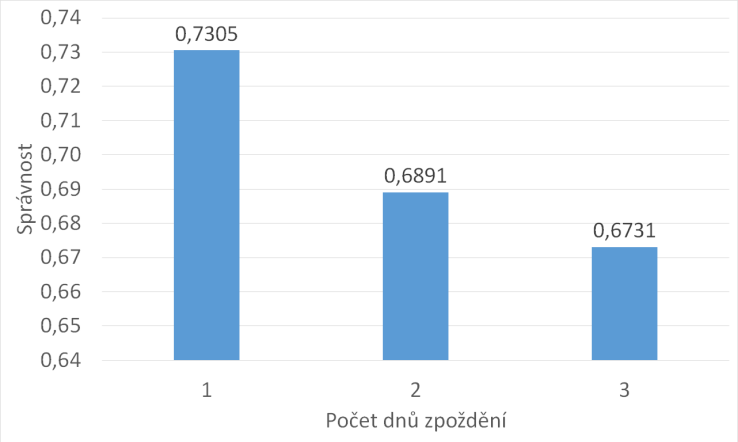
Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
ewma_1_5	1,0000	tf-idf-cos	LinearSVC	46	226
ewma_1_4	0,9643	tf-idf-no	LinearSVC	80	566
ewma_1_3	0,8261	tf-idf-no	LinearSVC	196	1 559
ewma_2_4	0,8142	tf-idf-no	NB-multi	522	2 811
ewma_2_5	0,8132	tf-idf-cos	LinearSVC	258	1 948
ewma_1_2	0,8122	tp-no-no	NB-multi	654	3 063
ewma_3_5	0,7774	tp-no-no	MaxEnt	808	3 605
ewma_2_2	0,7497	tf-idf-no	MaxEnt	2 772	7 868
ewma_1_1	0,7468	tf-idf-cos	LinearSVC	2 706	8 288
ewma_2_3	0,7454	tf-idf-cos	LinearSVC	1 234	4 539
ewma_3_2	0,7428	tf-idf-cos	LinearSVC	6 476	12 915
ewma_3_3	0,7428	tf-idf-no	MaxEnt	2 686	7 450
sma_1_4	0,7238	tf-idf-cos	NB-multi	900	4 111
ewma_3_4	0,7158	tf-idf-no	MaxEnt	1 568	5 216
ewma_2_1	0,7140	tf-idf-no	MaxEnt	12 728	17 925

Mezi všemi soubory byl nejčastěji nejlepší algoritmus MaxEnt (22×) a vektor typu tf-idf-no (21×). Průměrná maximální (prům. max.) správnost byla určena tak, že pro každý zdrojový soubor byl vybrán experiment s nejvyšší správností (nezávisle na použitém algoritmu nebo typu vektoru, který ho dosáhl) a byl vytvořen průměr těchto správností pro soubory, mající zkoumanou hodnotu parametru.

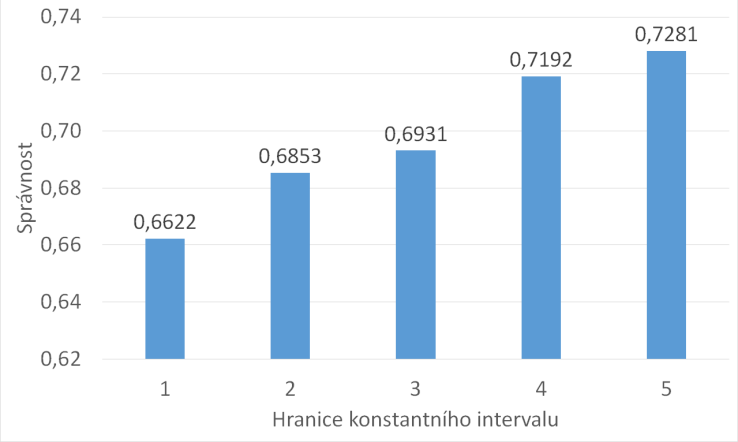
Obrázky 15, 16 a 17 ukazují závislost této hodnoty na parametrech: typ cenové proměnné, počet dnů zpoždění a hranice konstantního intervalu. Lze vidět, že nejlepší správnost poskytuje proměnná EWMA, zpoždění 1 den a hranice 5 %.



Obr. 15: Yahoo – prům. max. správnost pro typ cenové proměnné



Obr. 16: Yahoo – prům. max. správnost pro počet dnů zpoždění



Obr. 17: Yahoo – prům. max. správnost pro hranici konst. intervalu

Yahoo články – detailní analýza

Tabulka 50 v příloze F zobrazuje metriky experimentů, vybraných pro detailní analýzu. Tabulka 17 zobrazuje průměrné hodnoty hlavních metrik pro skupiny experimentů, seřazených dle správnosti. Sloupec PE značí počet experimentů. Lze vidět, že pro horních 10 % se správnost pohybuje okolo 0,75.

Tab. 17: Yahoo – průměrné metriky (detailní analýza 1)

Skupina	PE	Správnost	F1 skóre	PD	PA	Čas trénování [s]
vše	738	0,6346	0,6324	12 003	14 508	7,4881
top 50 %	369	0,6718	0,6705	7 600	11 038	5,9243
top 25 %	185	0,7059	0,7048	3 162	7 150	0,8920
top 10 %	74	0,7406	0,7398	1 930	5 538	0,4941
top 5 %	37	0,7627	0,7617	1 109	4 091	0,1470

Tabulka 18 zobrazuje nejvíce zastoupené hodnoty parametrů pro skupiny experimentů. Lze vidět, že zdaleka nejlepší výsledky poskytuje cenová proměnná EWMA se zpožděním 2 dny a hranicí 2 %. Nejúspěšnější je typ vektoru tf-idf-no a algoritmy MaxEnt a LinearSVC. Poslední sloupec zobrazuje nejvyšší správnost dosaženou danou kombinací parametrů.

Obrázek 50 v příloze F zobrazuje detailní analýzu parametrů.

Tab. 18: Yahoo – souhrnné zhodnocení parametrů (detailní analýza 1)

Skupina	Proměnná	Zpoždění	Hranice	Typ vektoru	Algoritmus	Správnost
top 5 %	ewma	2	2	tf-idf-no	MaxEnt	0,7497
top 10 %	ewma	2	2	tf-idf-no	LinearSVC	0,7199
top 25 %	ewma	2	2	tf-idf-no	MaxEnt	0,7497
top 50 %	sma	3	2	tf-idf-no	LinearSVC	0,6144
vše	adjclose/sma	3	1/2	–	–	–

4.3.3 Analýza č. 1 – Facebook komentáře

Zde jsou prezentovány výsledky experimentů pro Facebook komentáře (FB-com).

Facebook komentáře – celková analýza

Tabulka 19 ukazuje charakteristiky hlavních metrik. Sloupec průměr zobrazuje aritmetický průměr, sloupec vážený průměr jej váží počtem dokumentů.

Tabulka 20 zobrazuje počet experimentů pro zvolené intervaly počtu dokumentů. Lze vidět, že experimentů s méně než 1 000 dokumenty je celkem 8,89 %. Průměrná

Tab. 19: FB-com – hlavní metriky (analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5229	0,9091	0,6092	0,5868	0,5652
Precision	0,5228	0,9242	0,6132	0,5896	0,5664
Recall	0,5229	0,9091	0,6092	0,5868	0,5652
F1 skóre	0,5136	0,9091	0,6075	0,5865	0,5637
Počet dokumentů	30	94 870	44 830	35 804	–
Počet atributů	16	19 618	10 563	11 191	–
Čas trénování [s]	0,0004	74,7695	7,7360	0,1335	–

Tab. 20: FB-com – počty dokumentů (analýza 1)

Počet dokumentů	Počet experimentů	Podíl [%]
vše	810	100,00
$(0, 500)$	54	6,67
$[500; 1000)$	18	2,22
$[1000; 10000)$	180	22,22
$[10000; \infty)$	558	68,89

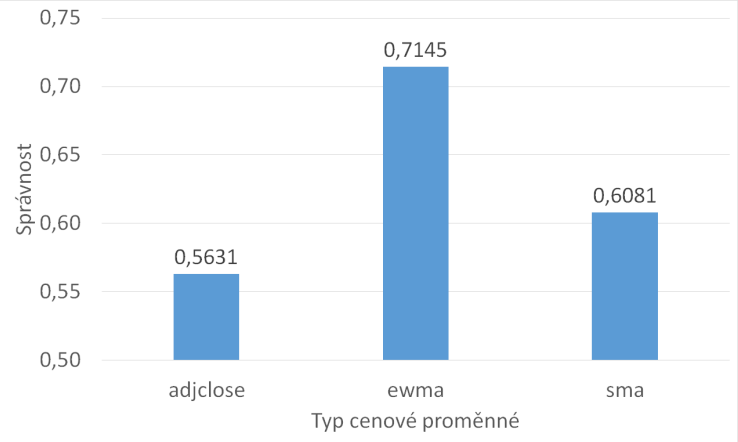
správnost těchto dokumentů je vysoká (0,79) a jejich zobecnitelnost nízká (obsahují 5–109 testovacích instancí jedné třídy), tudíž v detailní analýze nebudou zkoumány.

Facebook komentáře – nejlepší výsledky

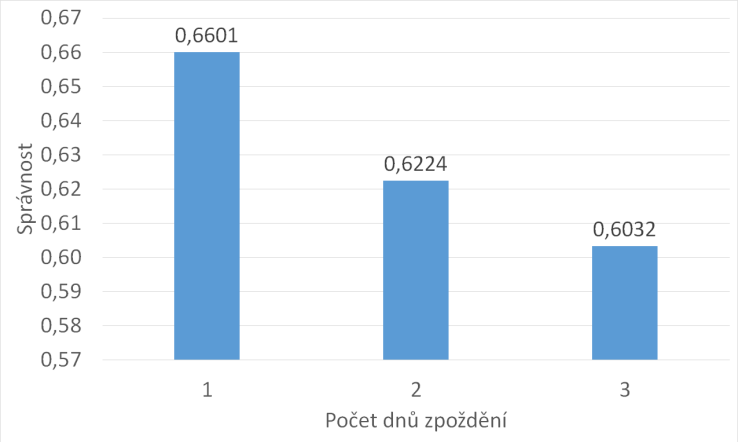
Pro všechny zastoupené zdrojové soubory (bylo jich 45) byla vybrána hodnota maximální správnosti a algoritmus a typ vektoru, pomocí kterého byla dosažena.

Tabulka 21 zobrazuje 15 zdrojových souborů s nejvyšší správností (další viz tabulka 55 v příloze G). Sloupec PD značí Počet dokumentů, sloupec PA počet atributů. Typ souboru je ve formátu `<cenová proměnná>_<počet dnů zpoždění>_<hranice konst. intervalu>`. Lze vidět, že správnost je až na tři případy menší než 0,7 a blíží se hranici 0,6.

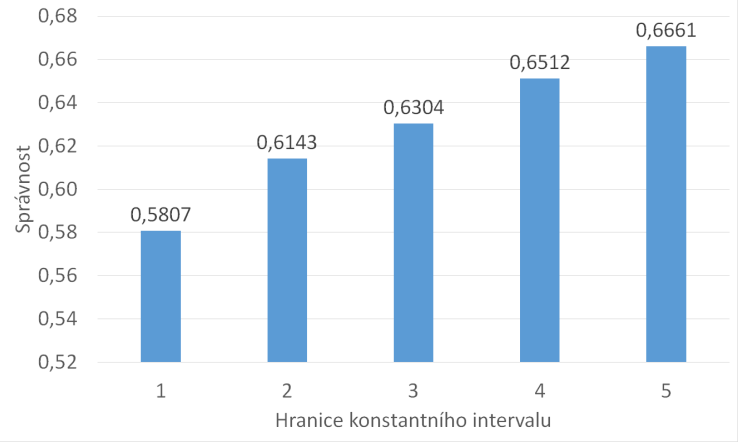
Mezi všemi soubory byl nejčastěji nejlepší algoritmus NB-multi (25×) a vektor typu tf-idf-cos (21×). Obrázky 18, 19 a 20 ukazují závislost průměrné maximální správnosti na parametrech: typ cenové proměnné, počet dnů zpoždění a hranice konstantního intervalu. Lze vidět, že nejlepší správnost poskytuje proměnná EWMA, zpoždění 1 den a hranice 5 %.



Obr. 18: FB-com – prům. max. správnost pro typ cenové proměnné



Obr. 19: FB-com – prům. max. správnost pro počet dnů zpoždění



Obr. 20: FB-com – prům. max. správnost pro hranici konst. intervalu

Tab. 21: FB-com – nejlepší soubory (analýza 1)

Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
ewma_1_5	0,9091	tf-idf-cos	NB-multi	30	16
ewma_1_4	0,8421	tf-idf-cos	MaxEnt	54	101
ewma_1_3	0,8070	tf-idf-cos	LinearSVC	162	437
ewma_2_5	0,7861	tf-idf-cos	LinearSVC	572	517
ewma_1_2	0,7352	tf-idf-cos	NB-multi	1 876	1 003
ewma_2_4	0,7311	tf-idf-cos	NB-multi	1 508	816
ewma_3_5	0,6962	tf-idf-no	NB-multi	1 740	1 027
ewma_1_1	0,6836	tp-no-no	MaxEnt	13 480	5 404
ewma_2_2	0,6722	tp-no-no	NB-multi	11 806	4 841
ewma_3_4	0,6662	tf-idf-cos	NB-multi	4 228	1 847
sma_1_2	0,6639	tf-idf-cos	LinearSVC	18 124	7 169
ewma_2_3	0,6563	tp-no-no	NB-multi	3 472	1 650
ewma_3_2	0,6562	tf-idf-cos	LinearSVC	35 804	11 810
sma_2_4	0,6557	tp-no-no	MaxEnt	15 758	6 176
ewma_3_3	0,6503	tf-idf-cos	NB-multi	9 672	3 998

Facebook komentáře – detailní analýza

Tabulka 51 v příloze F zobrazuje metriky experimentů, vybraných pro detailní analýzu. Tabulka 22 zobrazuje průměrné hodnoty hlavních metrik pro skupiny experimentů, seřazených dle správnosti. Sloupec PE značí počet experimentů. Lze vidět, že pro horních 10 % se správnost pohybuje okolo 0,69.

Tab. 22: FB-com – průměrné metriky (detailní analýza 1)

Skupina	PE	Správnost	F1 skóre	PD	PA	Čas trénování [s]
vše	738	0,5917	0,5901	49 183	11 567	8,4922
top 50 %	369	0,6300	0,6283	24 878	7 392	1,6000
top 25 %	185	0,6581	0,6562	13 453	4 870	0,3999
top 10 %	74	0,6860	0,6843	7 373	3 033	0,0744
top 5 %	37	0,7080	0,7059	2 660	1 285	0,0243

Tabulka 23 zobrazuje nejvíce zastoupené hodnoty parametrů pro skupiny experimentů. Lze vidět, že zdaleka nejlepší výsledky poskytuje proměná EWMA se zpožděním 1–2 dny a hranicí 2 %. Nejúspěšnější je typ vektoru tp-no-no a algoritmus NB-multi. Obrázek 51 v příloze F zobrazuje detailní analýzu parametrů.

Tab. 23: FB-com – souhrnné zhodnocení parametrů (detailní analýza 1)

Skupina	Proměnná	Zpoždění	Hranice	Typ vektoru	Algoritmus	Správnost
top 5 %	ewma	1	2	tp-no-no	NB-multi	0,7199
top 10 %	ewma	1	2	tp-no-no	MaxEnt	0,7093
top 25 %	ewma	2	2	tp-no-no	MaxEnt	0,6707
top 50 %	ewma	2	2/4	tp-no-no	NB-multi	0,7235
vše	adjclose/sma	3	1/2	–	–	–

4.3.4 Analýza č. 1 – Facebook příspěvky

Zde jsou prezentovány výsledky experimentů pro Facebook příspěvky (FB-post).

Facebook příspěvky – celková analýza

Tabulka 24 ukazuje charakteristiky hlavních metrik. Sloupec průměr zobrazuje aritmetický průměr, sloupec vážený průměr jej váží počtem dokumentů.

Tab. 24: FB-post – hlavní metriky (analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5020	0,8488	0,5888	0,5808	0,5722
Precision	0,5021	0,8528	0,5956	0,5821	0,5776
Recall	0,5020	0,8488	0,5888	0,5808	0,5722
F1 skóre	0,5020	0,8479	0,5847	0,5769	0,5677
Počet dokumentů	162	69 038	15 373	6 916	–
Počet atributů	224	23 650	7 256	4 762	–
Čas trénování [s]	0,0007	57,5003	1,7997	0,0446	–

Tab. 25: FB-post – počty dokumentů (analýza 1)

Počet dokumentů	Počet experimentů	Podíl [%]
vše	810	100,00
$(0, 500)$	36	4,44
$[500; 1000)$	18	2,22
$[1000; 10000)$	432	53,33
$[10000; \infty)$	324	40,00

Tabulka 25 zobrazuje počet experimentů pro zvolené intervaly počtu dokumentů. Lze vidět, že experimentů s méně než 500 dokumenty je 4,44 %. Průměrná správnost

těchto dokumentů je vysoká (0,75) a jejich zobecnitelnost nízká (obsahují 28–45 testovacích instancí jedné třídy), tudíž v detailní analýze nebudou zkoumány.

Facebook příspěvky – nejlepší výsledky

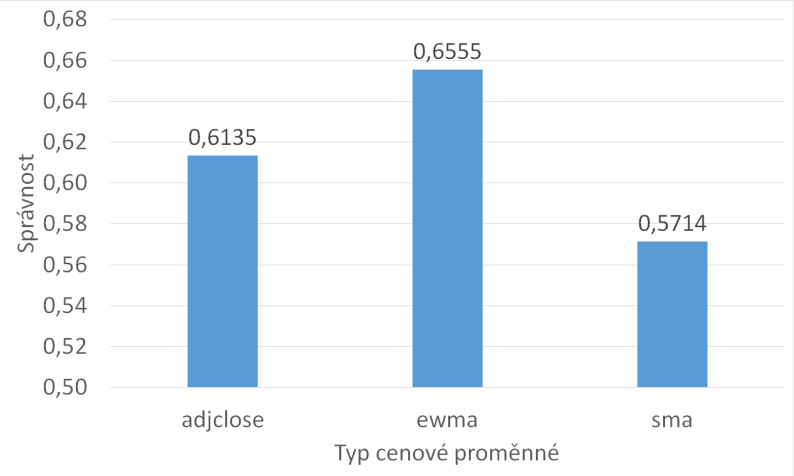
Pro všechny zastoupené zdrojové soubory (bylo jich 45) byla vybrána hodnota maximální správnosti a algoritmus a typ vektoru, pomocí kterého byla dosažena.

Tabulka 26 zobrazuje 15 zdrojových souborů s nejvyšší správností (další viz tabulka 56 v příloze G). Sloupec PD značí Počet dokumentů, sloupec PA počet atributů. Typ souboru je ve formátu `<cenová proměnná>_<počet dnů zpoždění>_<hranice konst. intervalu>`. Lze vidět, že první dva soubory poskytují vysokou správnost (okolo 0,8), ale jelikož obsahují nízký počet dokumentů, nelze je brát příliš v potaz. Ostatní soubory poskytují správnost 0,6–0,7. Překvapivé je dobré umístění proměnné Adjclose, zatímco SMA zde vůbec není přítomna.

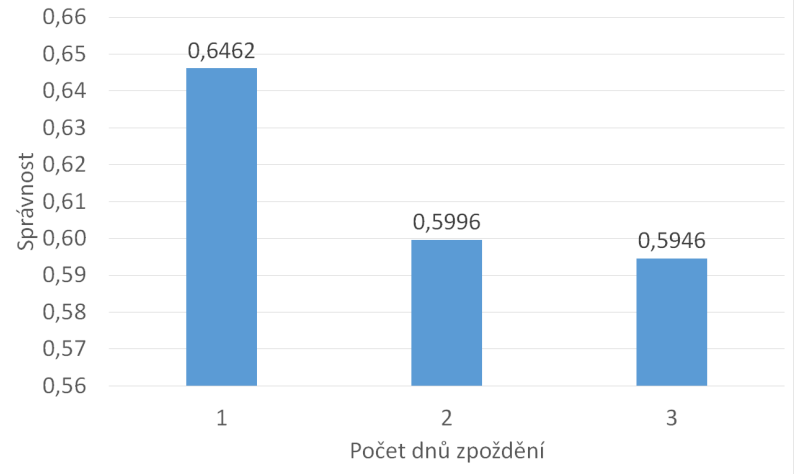
Tab. 26: FB-post – nejlepší soubory (analýza 1)

Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
ewma_1_4	0,8488	tf-idf-cos	NB-berno	244	289
ewma_1_5	0,7895	tf-idf-no	MaxEnt	162	224
ewma_1_3	0,6940	tf-idf-no	MaxEnt	904	789
ewma_2_5	0,6913	tf-idf-cos	NB-multi	1 044	875
ewma_3_5	0,6782	tf-idf-cos	NB-berno	1 242	1 052
ewma_1_2	0,6753	tf-idf-no	NB-multi	1 654	1 188
ewma_1_1	0,6553	tf-idf-cos	NB-berno	3 430	2 840
adjclose_1_5	0,6453	tp-no-no	NB-multi	2 448	2 541
ewma_2_4	0,6390	tf-idf-cos	LinearSVC	1 272	1 048
adjclose_2_5	0,6332	tp-no-no	NB-multi	6 432	5 363
adjclose_2_1	0,6236	tf-idf-no	NB-multi	61 324	22 934
adjclose_2_4	0,6233	tf-idf-no	NB-multi	10 708	7 779
adjclose_3_5	0,6196	tf-idf-cos	NB-berno	9 230	6 897
adjclose_3_2	0,6168	tf-idf-cos	NB-berno	40 904	17 819
adjclose_2_2	0,6158	tf-idf-no	NB-multi	34 668	16 397

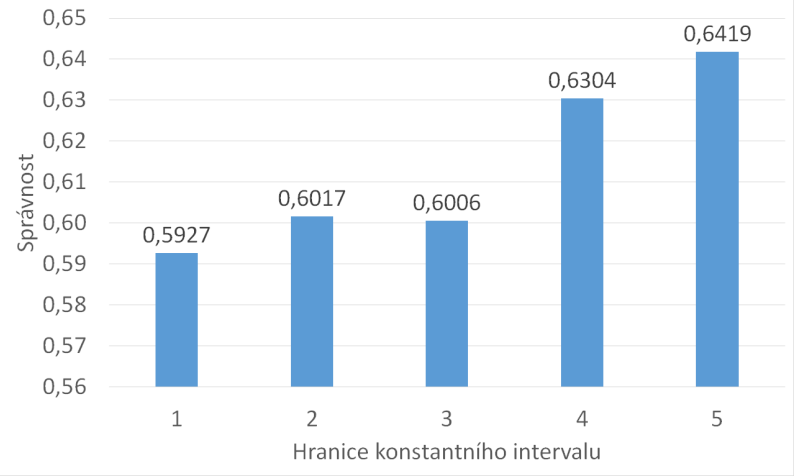
Mezi všemi soubory byl nejčastěji nejlepší algoritmus NB-multi (29×) a vektor typu tf-idf-cos (21×). Obrázky 21, 22 a 23 ukazují, že nejlepší správnost opět poskytuje proměnná EWMA, zpoždění 1 den a hranice 5 %.



Obr. 21: FB-post – prům. max. správnost pro typ cenové proměnné



Obr. 22: FB-post – prům. max. správnost pro počet dnů zpoždění



Obr. 23: FB-post – prům. max. správnost pro hranici konst. intervalu

Facebook příspěvky – detailní analýza

Tabulka 52 v příloze F zobrazuje charakteristiky hlavních metrik experimentů, vybraných pro detailní analýzu (obsahují alespoň 500 dokumentů).

Tabulka 27 zobrazuje průměrné hodnoty hlavních metrik pro skupiny experimentů, seřazených dle správnosti. Sloupec PE značí počet experimentů. Lze vidět, že pro horních 10 % se správnost pohybuje okolo 0,66.

Tabulka 28 zobrazuje nejvíce zastoupené hodnoty parametrů pro skupiny experimentů. Lze vidět, že nejlepší výsledky poskytuje proměná EWMA se zpožděním 1 den a 5% hranicí, přičemž pro skupiny 25 a 50 % je častější Adjclose. Nejúspěšnější je typ vektoru tp-no-no a algoritmus NB-berno. Obrázek 52 v příloze F zobrazuje detailní analýzu parametrů.

Tab. 27: FB-post – průměrné metriky (detailní analýza 1)

Skupina	PE	Správnost	F1 skóre	PD	PA	Čas trénování [s]
vše	774	0,5814	0,5773	16 079	7 582	1,8835
top 50 %	387	0,6104	0,6039	13 848	6 981	0,6099
top 25 %	194	0,6297	0,6215	11 854	6 004	0,0890
top 10 %	77	0,6561	0,6482	1 607	1 363	0,0140
top 5 %	39	0,6715	0,6606	1 305	1 073	0,0075

Tab. 28: FB-post – souhrnné zhodnocení parametrů (detailní analýza 1)

Skupina	Proměnná	Zpoždění	Hranice	Typ vektoru	Algoritmus	Správnost
top 5 %	ewma	1	5	tp-no-no	NB-berno	0,7719
top 10 %	ewma	1	5	tf-idf-no	NB-multi	0,7368
top 25 %	adjclose	1	5	tf-idf-cos	NB-berno	0,6231
top 50 %	adjclose	1	5	tf-idf-no	NB-multi	0,6418
vše	adjclose/sma	2/3	1/2/3	–	–	–

4.3.5 Analýza č. 1 – Twitter statusy

Zde jsou prezentovány výsledky experimentů pro Twitter statusy (tweety).

Twitter – celková analýza

Tabulka 29 ukazuje charakteristiky hlavních metrik. Sloupec průměr zobrazuje aritmetický průměr, sloupec vážený průměr jej váží počtem dokumentů.

Tab. 29: Twitter – hlavní metriky (analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5652	0,7995	0,6557	0,6560	0,6460
Precision	0,5652	0,8120	0,6569	0,6569	0,6470
Recall	0,5652	0,7995	0,6557	0,6560	0,6460
F1 skóre	0,5652	0,7976	0,6552	0,6554	0,6454
Počet dokumentů	1 998	99 022	41 288	28 228	–
Počet atributů	1 566	17 114	9 325	8 185	–
Čas trénování	0,0015	45,7299	4,6358	0,2947	–

Tab. 30: Twitter – počty dokumentů (analýza 1)

Počet souborů	Počet experimentů	Podíl [%]
vše	738	100,00
$[1000; 10000)$	36	4,88
$[5000; 10000)$	126	17,07
$[10000; \infty)$	576	78,05

Tabulka 30 zobrazuje počet experimentů pro zvolené intervaly počtu dokumentů. Lze vidět, že experimentů s méně než 10 000 dokumenty bylo 4,88 %. Maximální správnost těchto dokumentů nicméně není nijak vysoká (pohybuje se okolo 0,7), takže je můžeme pro detailní analýzu ponechat.

Twitter – nejlepší výsledky

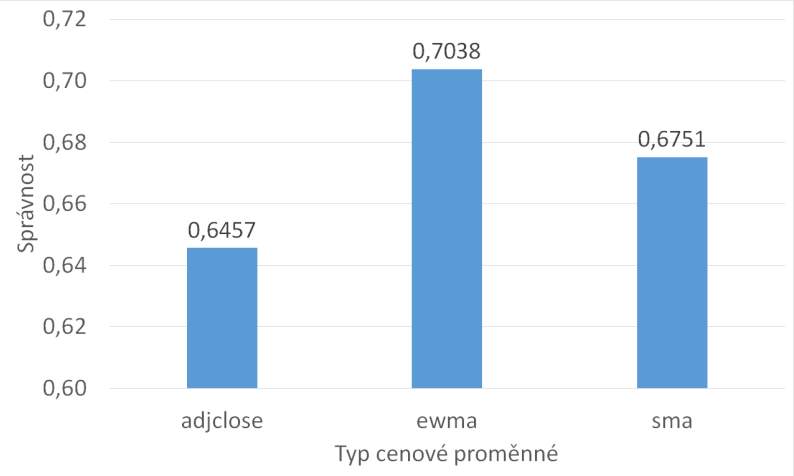
Pro všechny zastoupené zdrojové soubory (bylo jich 41) byla vybrána hodnota maximální správnosti a algoritmus a typ vektoru, pomocí kterého byla dosažena.

Tabulka 31 zobrazuje 15 zdrojových souborů, které poskytly nejlepší správnost (další viz tabulka 57 v příloze G). Sloupec PD značí Počet dokumentů, sloupec PA počet atributů. Typ souboru je ve formátu `<cenová proměnná>_<počet dnů zpoždění>_<hranice konst. intervalu>`. Lze vidět, že i proměnná SMA je schopná dosáhnout více než 70% správnosti.

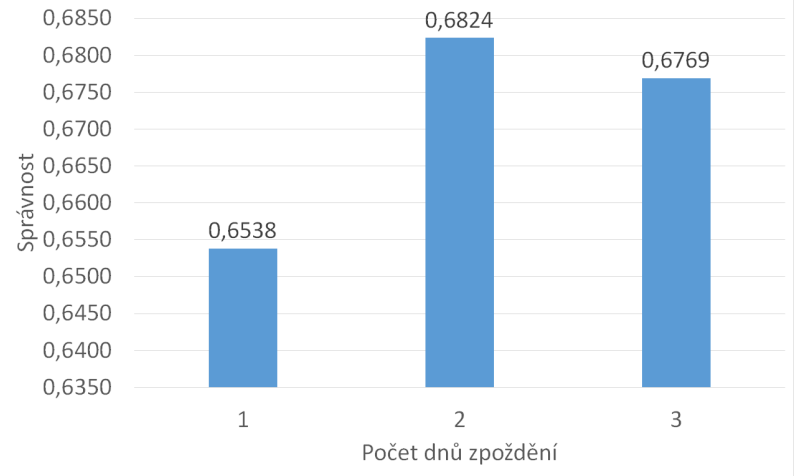
Tab. 31: Twitter – nejlepší soubory (analýza 1)

Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
ewma_3_2	0,7995	tf-idf-no	RandForest	22 398	8 185
ewma_2_1	0,7640	tp-no-no	RandForest	45 596	12 589
sma_2_3	0,7479	tp-no-no	RandForest	20 798	7 348
ewma_2_2	0,7257	tp-no-no	LinearSVC	10 800	5 186
sma_2_4	0,7190	tp-no-no	RandForest	14 400	5 347
ewma_1_1	0,7148	tp-no-no	RandForest	11 600	5 897
ewma_3_1	0,7098	tp-no-no	RandForest	91 072	17 021
sma_3_4	0,7040	tp-no-no	RandForest	28 398	8 775
ewma_2_4	0,6957	tf-idf-cos	LinearSVC	1 998	1 566
sma_3_3	0,6897	tp-no-no	RandForest	50 798	11 898
sma_2_2	0,6890	tp-no-no	MaxEnt	51 598	11 957
sma_1_1	0,6876	tp-no-no	MaxEnt	53 598	12 501
ewma_1_2	0,6863	tf-idf-no	LinearSVC	3 196	2 580
sma_1_2	0,6853	tp-no-no	CART	14 800	5 475
sma_3_5	0,6783	tp-no-no	RandForest	20 800	6 820

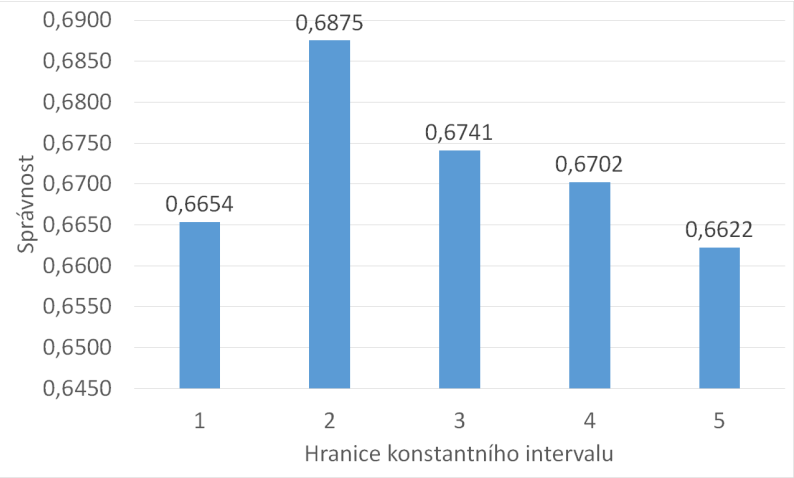
Mezi všemi soubory byl nejčastěji nejlepší algoritmus RandForest (26×) a vektor typu tp-no-no (25×). Obrázky 24, 25 a 26 ukazují závislost průměrné maximální správnosti na parametrech: typ cenové proměnné, počet dnů zpoždění a hranice konstantního intervalu. Lze vidět, že nejlepší správnost poskytuje proměnná EWMA, zpoždění 2 dny a hranice 2 %.



Obr. 24: Twitter – prům. max. správnost pro typ cenové proměnné



Obr. 25: Twitter – prům. max. správnost pro počet dnů zpoždění



Obr. 26: Twitter – prům. max. správnost pro hranici konst. intervalu

Twitter - detailní analýza

Tabulka 32 zobrazuje průměrné hodnoty hlavních metrik pro skupiny experimentů, seřazených dle správnosti. Lze vidět, že pro horních 10 % se správnost pohybuje okolo 0,75. Sloupec PE značí počet experimentů.

Tab. 32: Twitter – průměrné metriky (detailní analýza 1)

Skupina	PE	Správnost	F1 skóre	PD	PA	Čas trénování [s]
vše	738	0,6557	0,6552	41 288	9 325	4,6358
top 50 %	369	0,6871	0,6866	32 245	8 441	3,2518
top 25 %	185	0,7100	0,7093	31 400	8 676	3,0637
top 10 %	74	0,7426	0,7416	25 726	8 439	1,9662
top 5 %	37	0,7654	0,7643	30 289	9 597	2,6416

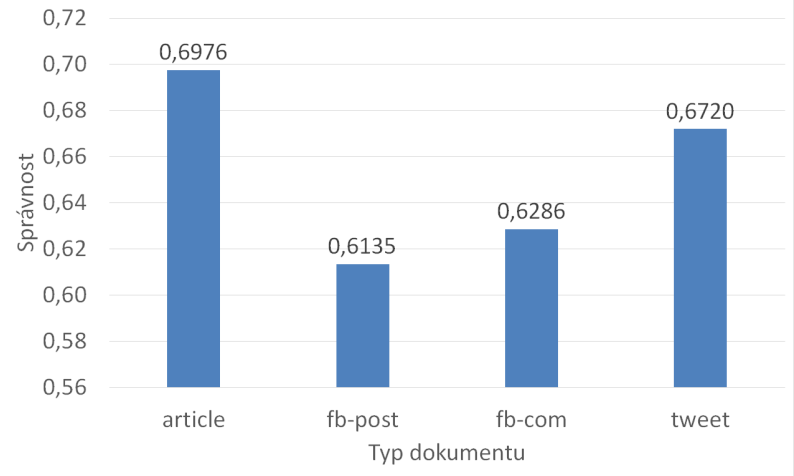
Tabulka 33 zobrazuje nejvíce zastoupené hodnoty parametrů pro skupiny experimentů. Lze vidět, že zdaleka nejlepší výsledky poskytuje proměná EWMA se zpožděním 2 dny a hranicí 1–2 %. Úspěšné jsou typy vektorů tp-no-no a tf-idf-no a algoritmy RandForest a LinearSVC. Obrázek 53 v příloze F zobrazuje detailní analýzu parametrů.

Tab. 33: Twitter – souhrnné zhodnocení parametrů (detailní analýza 1)

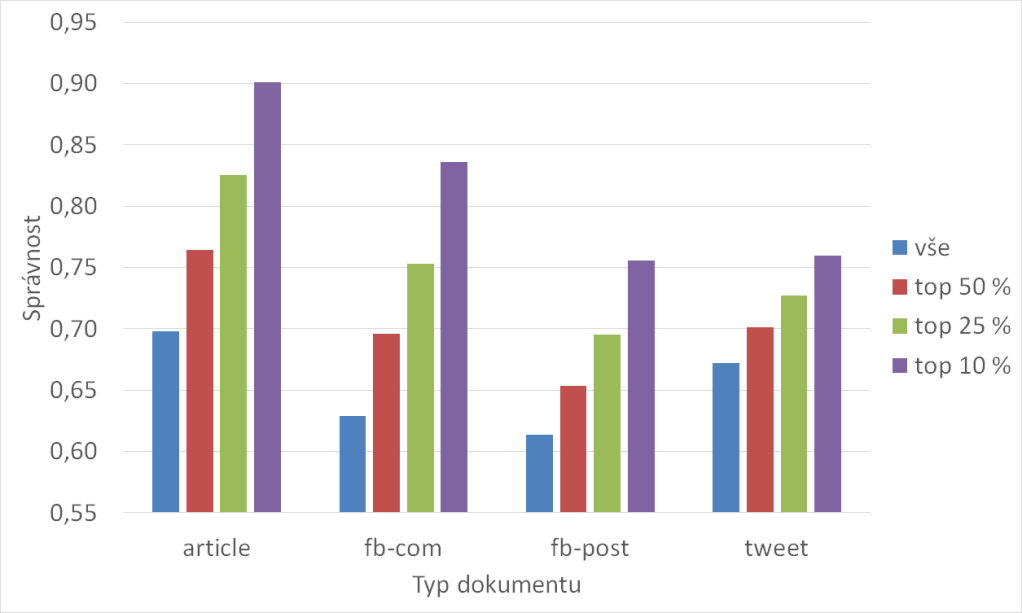
Skupina	Proměnná	Zpoždění	Hranice	Typ vektoru	Algoritmus	Správnost
top 5 %	ewma	2	2	tp-no-no/tf-idf-no	RandForest	0,7066
top 10 %	ewma	2	2	tp-no-no	LinearSVC	0,7257
top 25 %	ewma	2	1	tf-idf-no	RandForest	0,7512
top 50 %	sma	2	2	tf-idf-no	RandForest	0,6608
vše	adjclose/sma	3	1/2	–	–	–

4.3.6 Celková analýza všech typů dokumentů

Pro celkovou analýzu byly zvoleny všechny zdrojové soubory a nejlepší výsledky pro jednotlivé typy dokumentů. Celkem je k dispozici 176 souborů – pro Twitter 41 a pro ostatní typy 45. Obrázek 27 ukazuje, že nejlepší průměrné správnosti dosahují Yahoo články a tweety $(0,68)$, zatímco Facebook příspěvky a komentáře mají správnost okolo $0,62$.



Obr. 27: Průměrná maximální správnost pro typy dokumentů

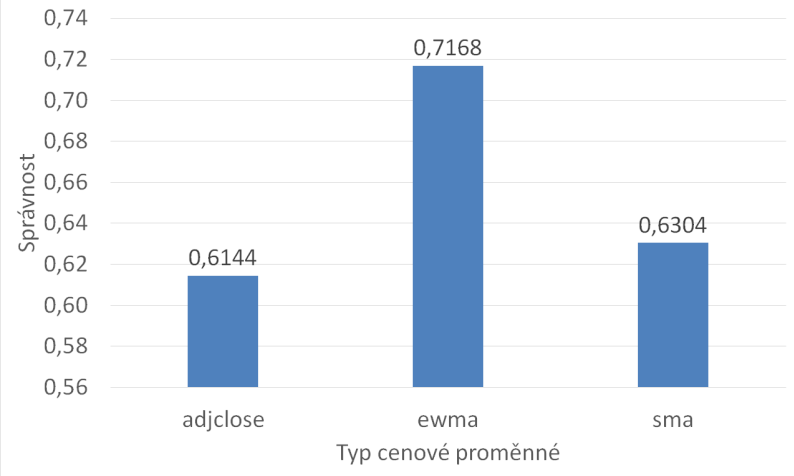


Obr. 28: Prům. max. správnost pro skupiny souborů a typy dokumentů

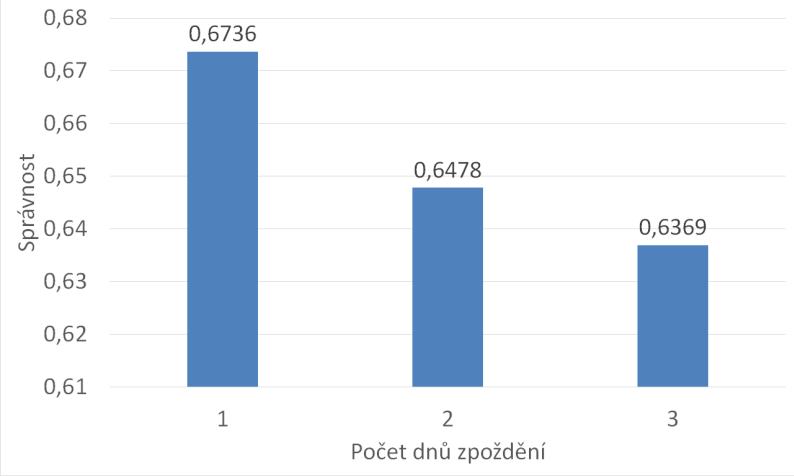
Obrázek 28 zobrazuje průměrnou maximální správnost pro horních 100, 50, 25 a 10 procent souborů. Lze vidět, že nejvyšší správnost pro horních 25 % mají Yahoo

články (0,83), následují FB komentáře (0,75) a tweety (0,73) a poslední jsou FB příspěvky (0,69). To znamená, že v 1/4 zkoumaných situací dosahovala správnost pro všechny typy dokumentů alespoň 0,69, což je dobrý výsledek. Pro horních 10 % je správnost samozřejmě ještě vyšší, v případě tweetů asi 0,77.

Obrázky 29, 30 a 31 ukazují závislost průměrné maximální správnosti na parametrech: typ cenové proměnné, počet dnů zpoždění a hranice konstantního intervalu. Lze vidět, že nejlepší správnost poskytuje proměnná EWMA, zpoždění 1 den a hranice 5 %.

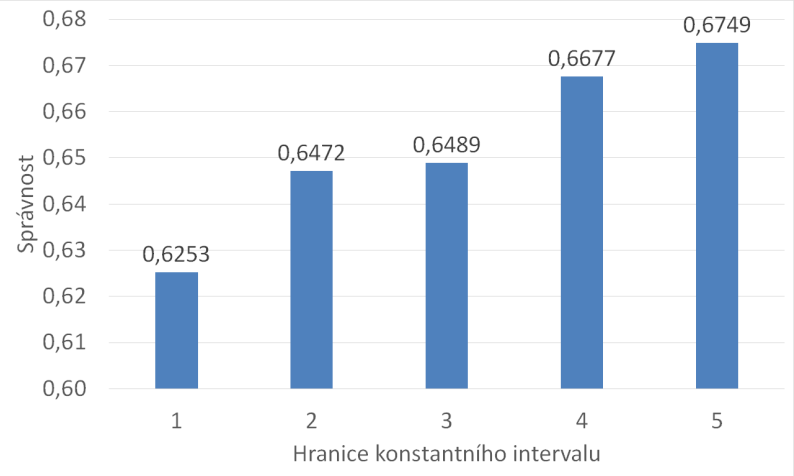


Obr. 29: Vše – prům. max. správnost pro typ cenové proměnné

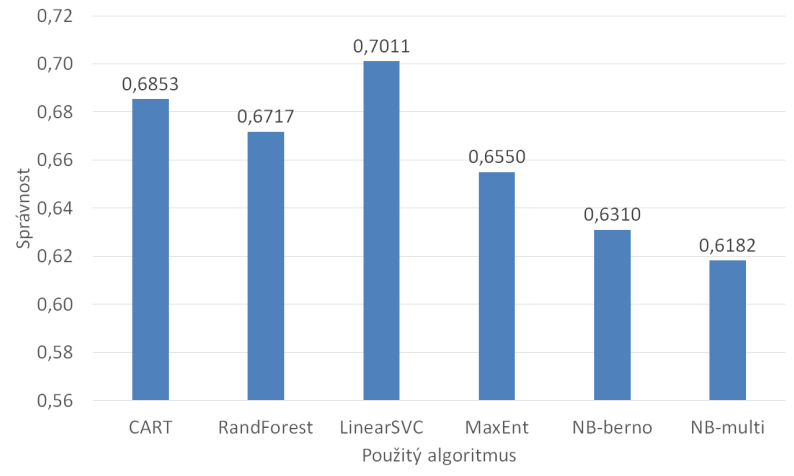


Obr. 30: Vše – prům. max. správnost pro počet dnů zpoždění

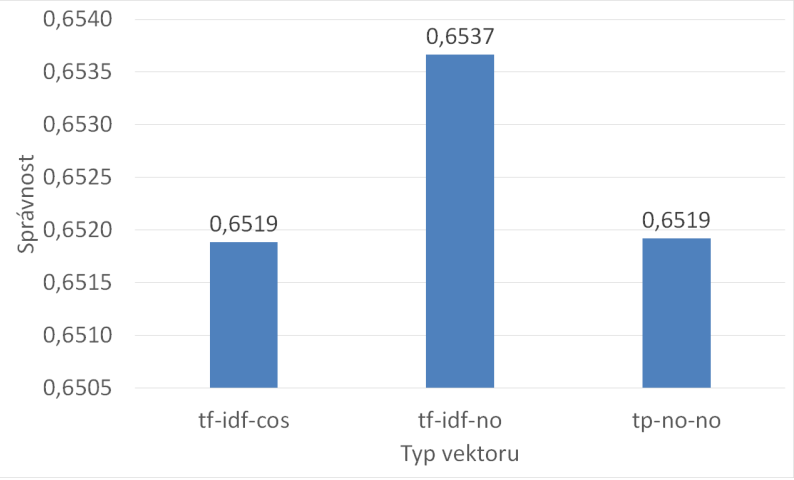
Obrázky 32 a 33 zobrazují průměrnou maximální správnost pro jednotlivé typy algoritmů a vektorů. Lze vidět, že průměrně nejlepší je algoritmus LinearSVC a vektor tf-idf-no. Mezi všemi soubory byl nejčastěji nejlepší algoritmus NB-multi (60×) a vektor tf-idf-cos (60×).



Obr. 31: Vše – prům. max. správnost pro hranici konst. intervalu



Obr. 32: Vše – prům. max. správnost pro algoritmy



Obr. 33: Vše – prům. max. správnost pro typy vektorů

Nakonec byly prozkoumány všechny typy zdrojových souborů (na základě zmíněných tří parametrů) a jejich maximální správnosti pro každý typ dokumentu. Tabulka 34 ukazuje 16 (z 45) nejlepších souborů, seřazených dle aritmetického průměru maximálních správností. Lze vidět, že jasně dominuje cenová proměnná EWMA. Ta poskytuje nejlepší výsledky pro zpoždění 1 den a hranici 5, 4 a 3 %. To značí, že pokud se exponenciální trend za jeden den změní o více než 2 %, je tato změna velmi významná. Kompletní seznam souborů se nachází v tabulce 53 v příloze G.

Je zajímavé, že první 4 soubory poskytují nejlepší správnost, ale pro tweety nejsou vůbec přítomné. Fakt, že se tweety zpracovávaly pouze pro 10 firem, nám říká, že tato kombinace parametrů je vzácná a nastává výjimečně. Ale právě proto se dá předpokládat, že tyto soubory obsahují text, pro klasifikaci velmi relevantní.

Tab. 34: Nejlepší soubory dle průměrné správnosti pro všechny typy dokumentů

č.	typ souboru	article	fb-post	fb-com	tweet	průměr
1	ewma_1_5	1,0000	0,7895	0,9091		0,8995
2	ewma_1_4	0,9643	0,8488	0,8421		0,8851
3	ewma_1_3	0,8261	0,6940	0,8070		0,7757
4	ewma_2_5	0,8132	0,6913	0,7861		0,7635
5	ewma_1_2	0,8122	0,6753	0,7352	0,6863	0,7273
6	ewma_2_4	0,8142	0,6390	0,7311	0,6957	0,7200
7	ewma_3_2	0,7428	0,6156	0,6562	0,7995	0,7035
8	ewma_3_5	0,7774	0,6782	0,6962	0,6529	0,7012
9	ewma_1_1	0,7468	0,6553	0,6836	0,7148	0,7001
10	ewma_2_2	0,7497	0,5879	0,6722	0,7257	0,6839
11	ewma_2_1	0,7140	0,6026	0,6350	0,7640	0,6789
12	sma_1_2	0,7083	0,6056	0,6639	0,6853	0,6658
13	ewma_2_3	0,7454	0,5843	0,6563	0,6668	0,6632
14	ewma_3_3	0,7428	0,5795	0,6503	0,6781	0,6627
15	ewma_3_4	0,7158	0,6067	0,6662	0,6484	0,6593
16	sma_2_3	0,6800	0,5622	0,6386	0,7479	0,6572

Byla také provedena detailní analýza nejlepších výsledků souborů (pro všechny typy dokumentů), které jsou seřazeny dle správnosti a rozděleny do skupin. Tabulka 35 ukazuje nejčastější hodnoty parametrů pro dané skupiny. Lze vidět, že jasně dominuje cenová proměnná EWMA, zpoždění 1 den a hranice 4–5 %. Z hlediska vektorů je zajímavé vítězství tf-idf-cos. Naopak přítomnost algoritmu LinearSVC překvapivá není.

Obrázek 34 zobrazuje samotnou detailní analýzu provedou v Excelu. Ta pro každý parameter a skupinu souborů zobrazuje, jak je konkrétní hodnota parametru

Tab. 35: Souhrnné zhodnocení parametrů pro všechny typy dokumentů

Skupina	Proměnná	Zpoždění	Hranice	Typ vektoru	Algoritmus
top 5 %	ewma	1	4	tf-idf-cos	LinearSVC
top 10 %	ewma	1	5	tf-idf-cos	LinearSVC
top 25 %	ewma	1	2/4	if-idf-cos	LinearSVC
top 50 %	ewma	3	5	tp-no-no	LinearSVC
vše	adjclose/sma	2/3	1/2	tf-idf-cos	NB-multi

zastoupena v dané skupině. Rozdíl oproti předchozím grafům tedy spočívá v tom, že nepočítáme průměrnou hodnotu pro každý parametr, ale počet výskytů této hodnoty v určité hladině správnosti. Z obrázku lze jasně vidět údaje z tabulky 34.

Dá se říct, že údaje v obrázku 34 potvrzují výše uvedené závěry. Průměrně nejlepší cenová proměnná EWMA je zde přítomna, stejně tak jeden den zpoždění a vysoké hranice konst. intervalu. V podstatě nic neříkající obrázek 33 je nahrazen jasným vítězstvím vektoru tf-idf-cos, zatímco obrázek 32 je potvrzen přítomností algoritmu LinearSVC. Co se týče cenové proměnné Adjclose, tak ta se v horních 25 % nevyskytuje ani jednou a v horních 50 % má 17% podíl, zatímco SMA má podíl 25 resp. 31 %.

Nakonec se v obrázku 34 podíváme na situaci pro horních 50 % souborů. Zde opět dominuje cenová proměnná EWMA (52% podíl). Nicméně počet dní zpoždění je vyrovnaný, rozdíly jsou v jednotce procenta. Pro hranici konst. intervalu je na prvním místě hodnota 5 (24 %), i když ostatní hodnoty jsou v malém odstupu. Nejúspěšnější je vektor tp-no-no (36 %), který o dvě procenta poráží vektor tf-idf-cos z tabulky 34. Téměř 30% podíl algoritmu LinearSVC hodnocení uzavírá.

Dále byly prozkoumány pouze soubory, obsahující alespoň 500 dokumentů. Ty byly rozděleny do skupin dle hodnot správnosti na horních 5, 10, 25 a 50 % souborů. Obrázek 42 (umístěný z důvodu úspory místa na straně 114) oproti obrázku 28 věrohodněji ukazuje skutečnou správnost. Lze na něm vidět, že pro horních 25 % je správnost pro Twitter a Yahoo 0,71, zatímco pro FB komentáře je 0,66 a pro FB příspěvky 0,63.

Cenová proměnná																	
all	count	percent		top50%	count	percent		top25%	count	percent		top10%				top5%		
adjclose	60	34,09091		adjclose	15	17,04545		adjclose	0	0		adjclose	0	0		adjclose	0	0
ewma	56	31,81818		ewma	46	52,27273		ewma	33	75		ewma	17	94,44444		ewma	9	100
sma	60	34,09091		sma	27	30,68182		sma	11	25		sma	1	5,555556		sma	0	0
	176	100			88	100			44	100			18	100			9	100
Počet dnů zpoždění																		
all				top50%				top25%				top10%				top5%		
1	93	32,86219		1	44	32,35294		1	26	37,68116		1	12	46,15385		1	7	63,63636
2	95	33,5689		2	45	33,08824		2	25	36,23188		2	9	34,61538		2	3	27,27273
3	95	33,5689		3	47	34,55882		3	18	26,08696		3	5	19,23077		3	1	9,090909
	283	100			136	100			69	100			26	100			11	100
Horní hranice konstantního intervalu																		
all				top50%				top25%				top10%				top5%		
1	36	20,45455		1	13	14,77273		1	7	15,90909		1	2	11,11111		1	0	0
2	36	20,45455		2	17	19,31818		2	10	22,72727		2	3	16,66667		2	1	11,11111
3	35	19,88636		3	18	20,45455		3	8	18,18182		3	3	16,66667		3	1	11,11111
4	35	19,88636		4	19	21,59091		4	10	22,72727		4	4	22,22222		4	4	44,44444
5	34	19,31818		5	21	23,86364		5	9	20,45455		5	6	33,33333		5	3	33,33333
	176	100			88	100			44	100			18	100			9	100
Typ vektoru																		
all				top50%				top25%				top10%				top5%		
tp-no-no	54	30,68182		tp-no-no	32	36,36364		tp-no-no	14	31,81818		tp-no-no	4	22,22222		tp-no-no	1	11,11111
tf-idf-no	56	31,81818		tf-idf-no	26	29,54545		tf-idf-no	13	29,54545		tf-idf-no	6	33,33333		tf-idf-no	3	33,33333
tf-idf-cos	66	37,5		tf-idf-cos	30	34,09091		tf-idf-cos	17	38,63636		tf-idf-cos	8	44,44444		tf-idf-cos	5	55,55556
	176	100			88	100			44	100			18	100			9	100
Algoritmus																		
all				top50%				top25%				top10%				top5%		
NB-multi	60	34,09091		NB-multi	16	18,18182		NB-multi	8	18,18182		NB-multi	3	16,66667		NB-multi	3	33,33333
NB-berno	12	6,818182		NB-berno	4	4,545455		NB-berno	1	2,272727		NB-berno	1	5,555556		NB-berno	1	11,11111
MaxEnt	42	23,86364		MaxEnt	21	23,86364		MaxEnt	12	27,27273		MaxEnt	4	22,22222		MaxEnt	1	11,11111
CART	1	0,568182		CART	1	1,136364		CART	1	2,272727		CART	0	0		CART	0	0
RandFores	27	15,34091		RandFores	20	22,72727		RandFores	8	18,18182		RandFores	3	16,66667		RandFores	0	0
LinearSVC	34	19,31818		LinearSVC	26	29,54545		LinearSVC	14	31,81818		LinearSVC	7	38,88889		LinearSVC	4	44,44444
	176	100			88	100			44	100			18	100			9	100

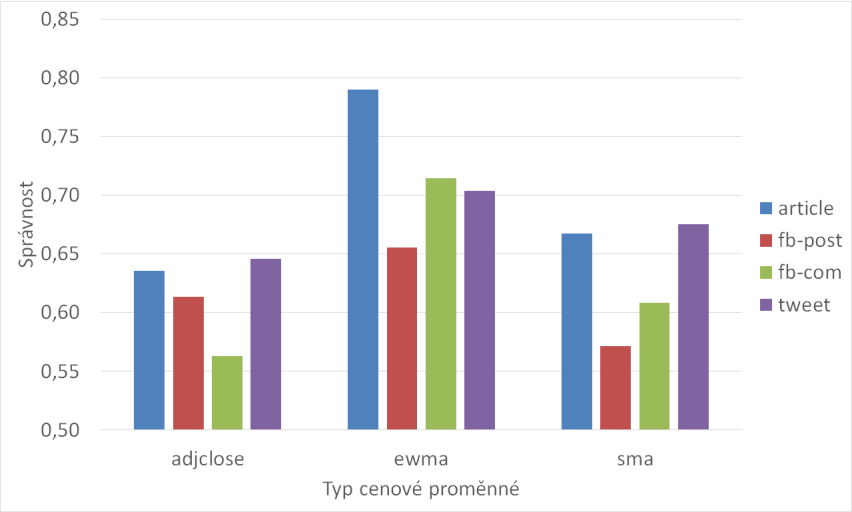
Obr. 34: Vše – detailní analýza parametrů

4.3.7 Porovnání parametrů pro jednotlivé typy dokumentů

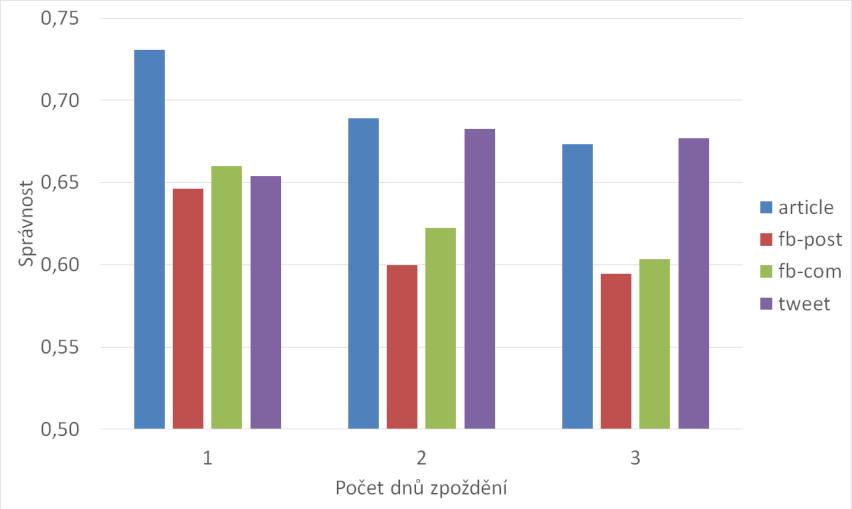
Tato sekce v první části přehledně porovnává typy dokumentů pomocí maximální hodnoty správnosti pro různé parametry.

Obrázek 35 ukazuje, že pro všechny typy dokumentů poskytuje nejlepší výsledky cenová proměnná EWMA, další je SMA a nejhorší Adjclose. Výjimkou jsou FB příspěvky, pro které je lepší Adjclose než SMA.

Obrázek 36 nám říká, že nejlepší je zpoždění jeden den, následovaná dvěma dny a nejhorší jsou tři dny. Výjimkou jsou tweety, pro které je nejlepší hodnota 2, následovaná 3 a 1.

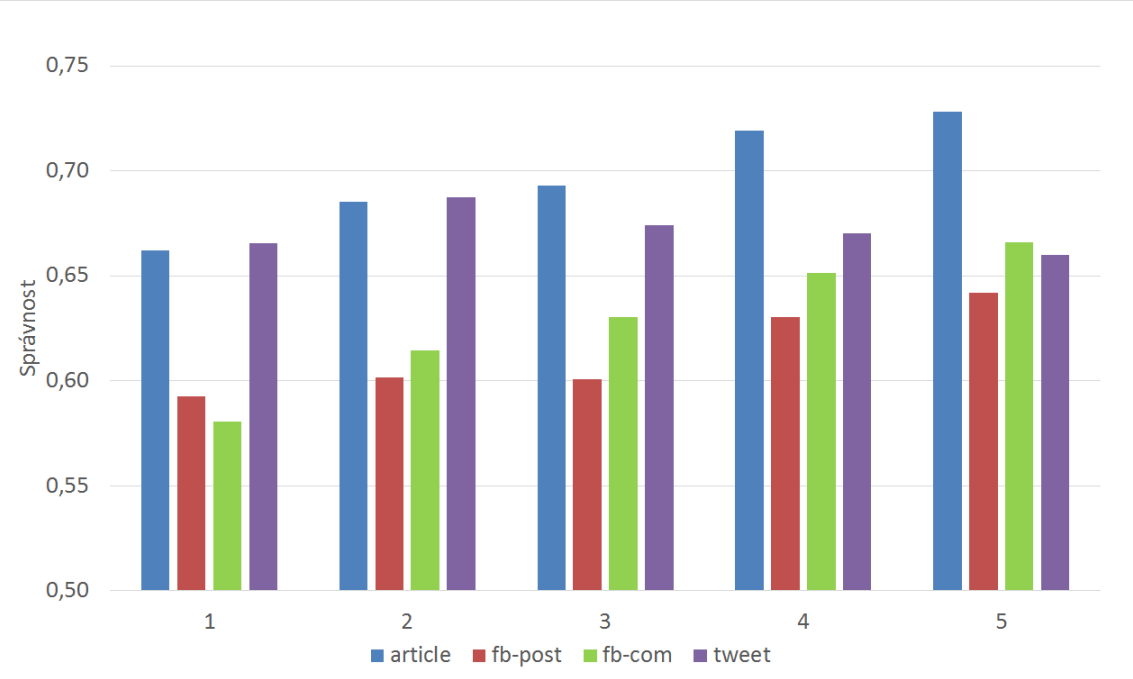


Obr. 35: Porovnání typů dokumentů – prům. max. správnost pro typ cenové proměnné



Obr. 36: Porovnání typů dokumentů – prům. max. správnost pro počet dnů zpoždění

Obrázek 37 zobrazuje roustoucí správnost pro vyšší hodnotu hranice konstantního intervalu. Výjimkou jsou opět tweety, pro které je správnost nejvyšší u hodnoty 2, přičemž vlevo a vpravo od ní postupně klesá.

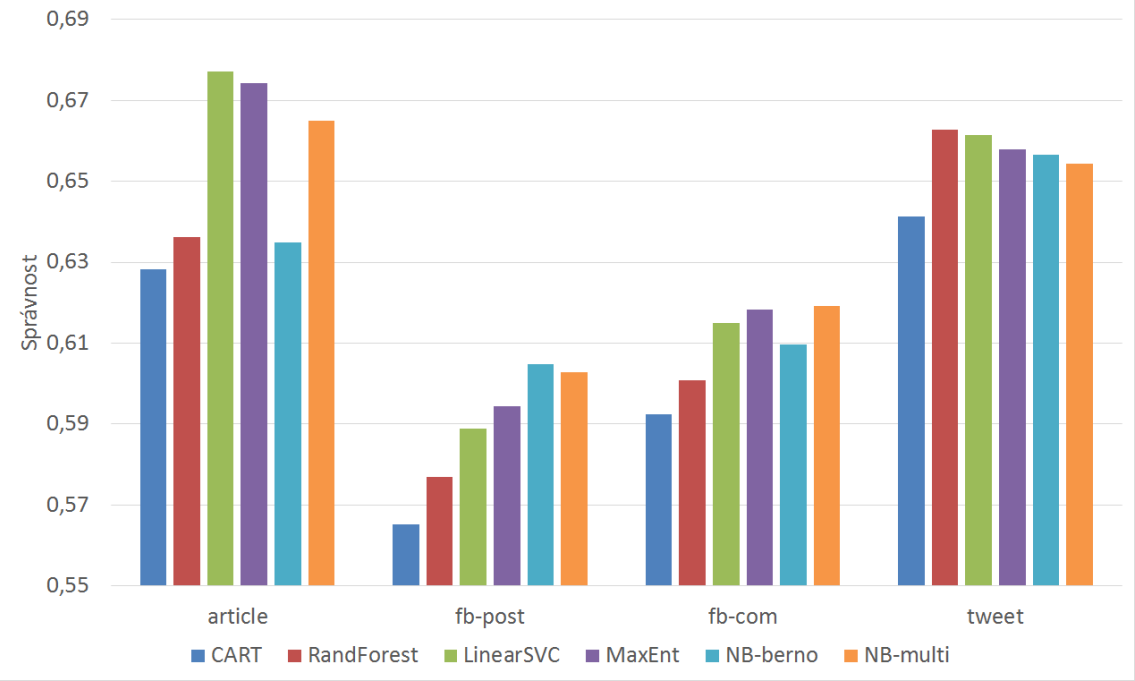


Obr. 37: Porovnání typů dokumentů – prům. max. správnost pro hranici konst. intervalu

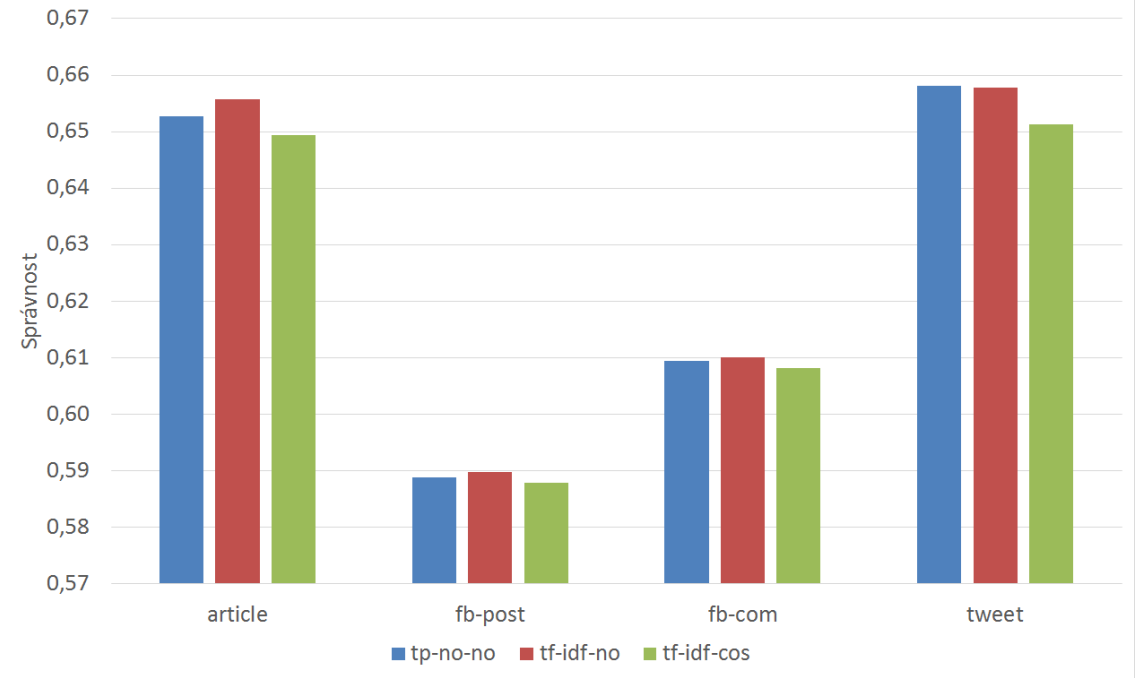
Dále byla zkoumána průměrná správnost všech experimentů pro jednotlivé typy dokumentů, algoritmů a vektorů.

Obrázek 38 ukazuje, že nejhorší je vždy algoritmus CART. Pro Yahoo články je nejlepší LinearSVC, pro FB příspěvky NB-berno, pro FB komentáře NB-multi a pro tweety RandForest.

Z obrázku 39 plyne, že průměrná správnost pro různé typy vektorů je velmi podobná (v rozmezí 0,01). Nicméně nejlepší je pro každý typ dokumentu tf-idf-no a nejhorší tf-idf-cos. Výjimkou jsou tweety, pro které je nejlepší tp-no-no. Tento typ vektoru se v případě FB komentářů blíží prvnímu místu.



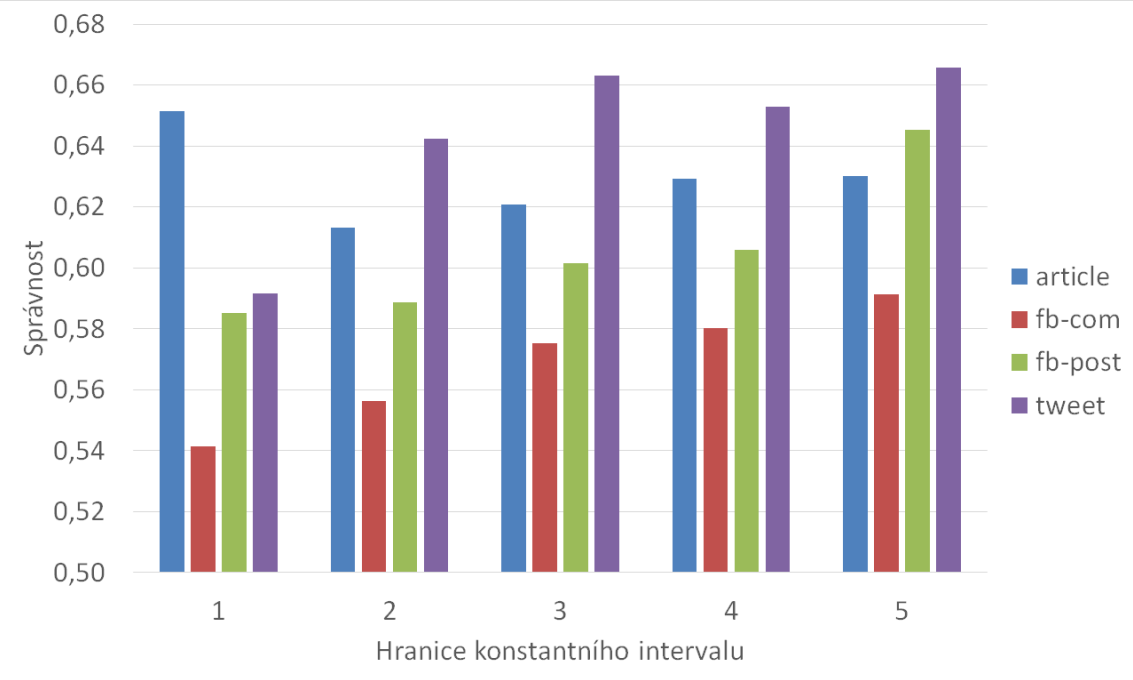
Obr. 38: Porovnání typů dokumentů – průměrná správnost pro algoritmy



Obr. 39: Porovnání typů dokumentů – průměrná správnost pro typy vektorů

4.3.8 Analýza č. 1 pro Adjclose a zpoždění jeden den

V poslední části analýzy 1 se zaměříme na to, jak dobré výsledky modely podávají pro nejjednodušší případ - cenovou proměnnou Adjclose a zpoždění jeden den. Budeme tedy zkoumat, nakolik lze (dle obsahu dokumentů, publikovaných daný den) určit, jestli cena akcie na konci dalšího dne klesne nebo stoupne. Zdrojovými daty pro toto srovnání jsou nejlepší výsledky pro odpovídající zdrojové soubory.

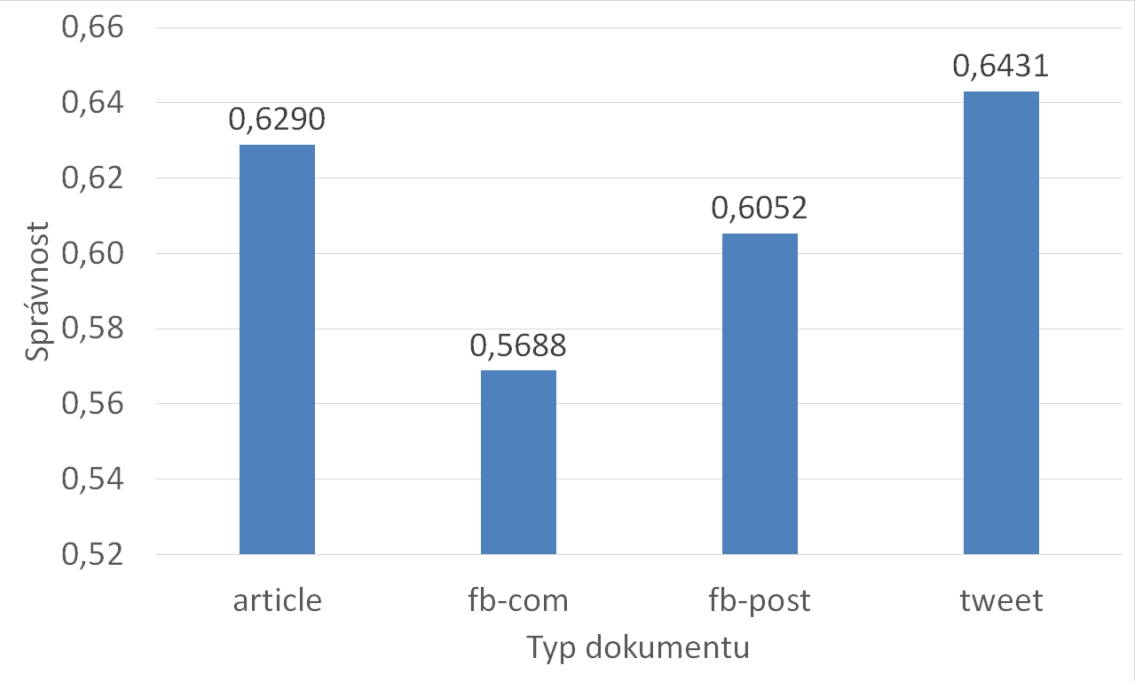


Obr. 40: Správnost pro Adjclose a zpoždění 1

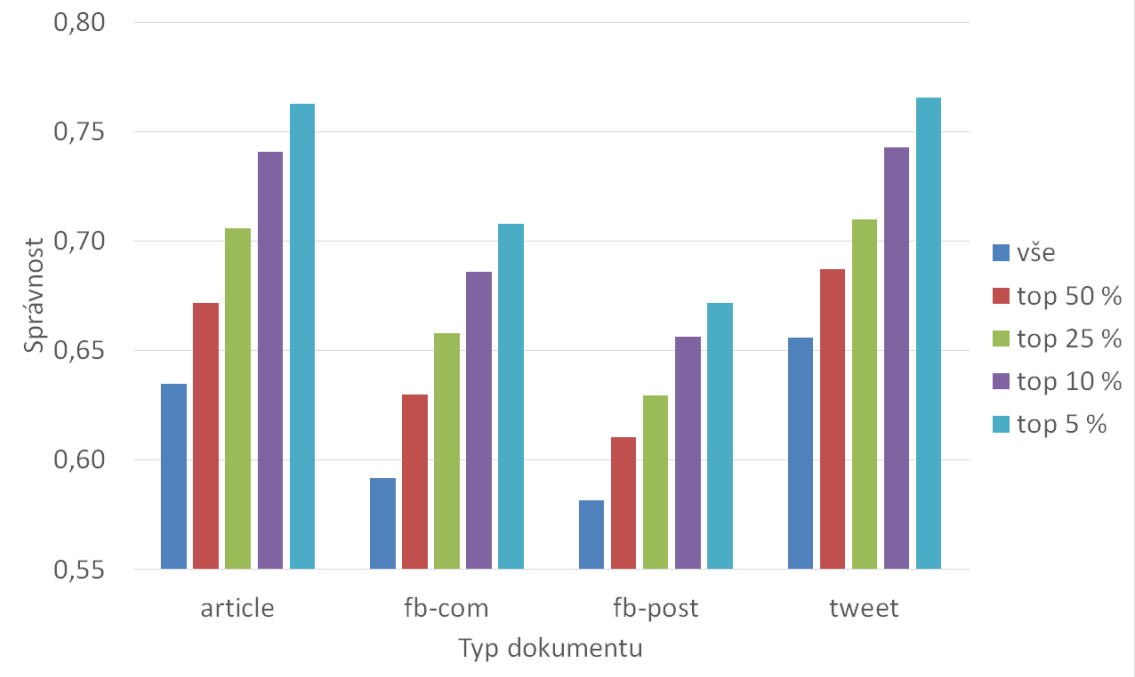
Obrázek 40 ukazuje, že s roustoucí hranicí intervalu se správnost zvyšuje, což je v souladu s předešlými výsledky. Výjimkou jsou Yahoo články, kde je nejlepší správnosti dosahována pro 1% hranici.

Pokud se zaměříme na poměrně výrazný, ale stále běžný cenový pohyb 2 % (nastává asi ve 20 % obchodních dní, viz sekce 4.5.3) je správnost 0,61 pro Yahoo články, 0,55 pro FB komentáře, 0,59 pro FB příspěvky a 0,64 pro tweety. Kromě FB komentářů jsou hodnoty asi o 10 % vyšší než pravděpodobnost pro náhodný výběr ze dvou možností (která je 50 %), což se dá považovat za ještě uspokojivý výsledek.

Pokud srovnáme aritmetický průměr hodnot pro všechny hranice intervalu (viz obrázek 41), lze konstatovat, že nejlepší výsledek podávají tweety, následované Yahoo články a FB posty, a FB komentáře jsou poslední.



Obr. 41: Průměrná správnost pro Adjclose a zpoždění 1



Obr. 42: Skupinová správnost typů dokumentů (pro více než 500 dokumentů)

4.4 Analýza č. 2 – Feature selection

V další fázi byl proveden postup popsaný v sekci 3.5.3 – výběr atributů (*Feature selection*) metodou CHI k získání informačně významných slov (z hlediska příslušnosti dokumentu k dané třídě). Bylo použito 36 zdrojových souborů, poskytujících pro jednotlivé typy dokumentů nejlepší správnost (alespoň 0,69). Jejich seznam se nalézá v tabulce 58 v příloze H. Konkrétně se jednalo o 17 souborů pro Yahoo články, 7 pro FB komentáře, 4 pro FB příspěvky a 8 pro Twitter statusy.

Výsledný soubor obsahuje 1 001 slov, kde 795 pochází z třídy 1 (pozitivní +) a 206 slov z třídy 2 (negativní −). Tabulka 36 zobrazuje 56 nejvýznamnějších slov (seřazených po sloupcích sestupně dle hodnoty CHI). Poslední řádek každého sloupce zobrazuje interval, ve kterém se nachází hodnoty metriky CHI (nejvyšší hodnotu má slovo v prvním řádku, nejnižší hodnotu slovo v posledním řádku).

Lze vidět, že ve sloupci 1 se jedná převážně o názvy firem, ale dále se objevují i běžná slova jako „attorney“ nebo „investigating“. Slov tvořených dvěma písmeny je v celém seznamu pouze 22. Kompletní seznam se nachází v el. příloze A.

Tab. 36: Informačně nejvýznamnější slova (analýza 2)

1	2	3	4
bank (−)	rt (+)	stake (+)	checked (−)
america (−)	peltz (+)	google (+)	higher (+)
every (−)	oct (+)	fcpa (+)	february (+)
microsoft (+)	idf (−)	porn (+)	rose (+)
fargo (−)	stadium (−)	upgradeyourworld (−)	llp (+)
wells (−)	ge (+)	mplusplaces (−)	investigating (+)
ces (−)	march (+)	panthers (−)	esq (+)
wellsfargo (−)	surface (+)	jumped (+)	shareholders (+)
generalelectric (+)	vmworld (−)	mwc (+)	share (+)
bofa (−)	trian (+)	attorney (+)	adapter (−)
of (−)	jan (−)	kla (+)	surged (+)
electric (+)	nelson (+)	tencor (+)	airgas (+)
wef (−)	apple (−)	industrial (+)	surfacebook (+)
general (+)	seemasapralaw (+)	gemm (+)	ac (−)
[644; 6116]	[405; 615]	[338; 393]	[296; 336]

4.5 Analýza č. 3 – slovníková metoda

V rámci třetí analýzy je potřeba dle postupu uvedeného v sekci 3.5.5 vytvořit a použít nové slovníky sentimentu pro vyhodnocení souvislosti mezi sentimentem dokumentů a pohybem cen akcií.

4.5.1 Vytvořené slovníky sentimentu

Byly vytvořeny tři nové slovníky sentimentu:

1. `custom_dict_orig`: Kombinovaný slovník z McDonald, Henry, Hajek, VADER. Duplikátní slova byla odstraněna dle uvedeného pořadí (zůstala ta z McDonald).
2. `custom_dict_fs_added`: Do slovníku výše přidána slova z analýzy 2.
3. `custom_dict_only_fs`: Slovník, obsahující pouze slova z analýzy 2.

Tabulka 37 zobrazuje počet slov jednotlivých zdrojových slovníků v kombinovaném slovníku 1. Lze vidět, že asi 12 % slov z finančních slovníků se překrývá se slovníkem VADER. U těchto slov byly váhy určeny následovně: Pokud bylo v obou slovnících slovo pozitivní/negativní, byla použita váha z VADER. Pokud se polarity lišily, byla použita ta z McDonald (-1 pro negativní a $+1$ pro pozitivní). Ze slovníku Hajek byla použita pouze jednotlivá slova. Lze vidět, že ve všech slovnících se překrývalo asi 10 % slov. Celkově slovník 1 obsahuje 9 412 různých slov.

Tab. 37: Kombinovaný slovník 1 – přehled počtů slov

Slovník	Po sloučení	Před sloučením	Zůstatek [%]
McDonald	2 709	2 709	100,00
Henry	83	190	43,68
Hajek	26	34	76,47
VADER	6 594	7 517	87,72
celkem	9 412	10 450	90,07

Slovník 2 obsahuje navíc slova získaná analýzou 2 (viz sekce 4.4). Třída 1 byla převedena na polaritu $+1$ a třída 2 na -1. Pouze 110 (asi 1 %) slov je přítomných ve slovníku 1, přičemž byla ponechána slova z analýzy 2. Celkově slovník 2 obsahuje 10 303 slov. Slovník 3 je tvořen pouze slovy z analýzy 2 (1 001 slov).

4.5.2 Postup provedení analýzy

Cílem analýzy bylo zjistit, zda má převládající sentiment souvislost s tím, zda cena akcie stoupne, klesne či zůstane konstantní. Celkový sentiment daného dne byl zjištěn tak, že byl spočítán počet pozitivních, neutrálních a negativních dokumentů

(publikovaných daného dne), a jaký z nich byl nejvyšší, takový sentiment pro daný den a pro daný typ dokumentu převládal.

Výsledky byly získány pro každý typ dokumentu a zpoždění 1, 2, 3 dny. Bylo také vyzkoušeno zpoždění −1 den, ale to zde nebude prezentováno, jelikož není relevantní resp. neposkytlo zajímavé výsledky. Použity byly všechny tři vytvořené slovníky. Byla zvolena cenová proměnná Adjclose a interval pro konstantní pohyh ceny akcie $(-2; +2)$. Tyto hodnoty ukázaly v analýze 1 dobrou kombinaci relativně vysokého množství tržních pohybů mimo interval a dostatečné správnosti klasifikace. Jelikož analyzovat a vyhodnotit všechny možné kombinace (jako tomu bylo u analýzy 1) by zabralo mnoho výpočetního (jedna analýza trvala 4 hodiny) a tvůrčího času, bylo rozhodnuto provést experimenty jen pro výše uvedené hodnoty parametrů.

Výsledné CSV soubory obsahují výsledky z časového intervalu 2. 8. 2015 až 2. 4. 2016, což je celkem 244 dnů. Pro tyto dny byly spočítány metriky – to znamená, že počet instancí v matici záměn byl pro každou firmu 244.

Experimenty byly provedeny pro všech 784 firem, nicméně pro mnoho firem je pro jednotlivé dny k dispozici málo dokumentů. Firmy byly zpracovávány sestupně dle celkového počtu dokumentů v databázi. Pro každou firmu existuje 16 řádků (4 hodnoty zpoždění × 4 typy dokumentů). Tím pádem lze snadno vybrat např. 30 % firem s nejvyšším počtem dokumentů a tyto dále zkoumat. CSV soubor má 12 544 řádků (bez hlavičky), tudíž 30% hranice končí na řádku 3 620, což představuje 226 firem (z celkových 784).

Pro každý typ dokumentu nás bude zajímat, jaké zpoždění poskytuje nejlepší správnost. Správnost bude brána jako průměr správností všech firem ze zkoumaného rozmezí. V CSV souboru jsou také přítomny metriky *Precision* a *Recall* – ty je ale nutné zkoumat zvlášť pro každou třídu (jelikož máme tři třídy), tudíž zde nebudou prezentovány.

4.5.3 Vygenerované soubory

Jak je uvedeno v sekci 4.2.3, skript vygeruje vždy tři typy souborů: Soubory s hlavními a podrobnými metrikami a také soubor s denním přehledem, kde je pro každou firmu a den uveden (mj.) počet pozitivních, negativních a neutrálních zpráv pro každý typ dokumentu.

Tabulka 38 zobrazuje procentuální podíl pro jednotlivé pohyby cen akcií a počty dnů zpoždění. Pro zpoždění 1 lze vidět, že většina (80 %) pohybů byla konstantních a že v 10 % případů došlo ke snížení resp. zvýšení ceny alespoň o 2 %.

Tabulky 39, 40 a 41 zobrazují pro každý použitý slovník procentuální podíl pro jednotlivé celkové sentimenty a typy dokumentů. Ve všech tabulkách lze zpozorovat, že většina (80–85 %) dokumentů byla klasifikována jako neutrální, malá část jako pozitivní (12–20 %) a nejmenší část jako negativní (0,3–3 %). Toto ovšem neplatí pro Twitter, v jehož případě lze vidět, že drtivá většina (97–98 %) tweetů je označena jako neutrální, a zbylá 2–3 % tvoří pozitivní tweety, přičemž negativní tweety se téměř nevyskytují. Na toto je nutné pamatovat při interpretaci výsledků.

Tab. 38: Analýza 3 – podíly pro pohyb cen akcií a zpoždění

cenový pohyb	1	2	3
růst	9,47	15,77	19,78
pokles	10,75	17,25	20,92
konstantní	79,78	66,98	59,30

Tab. 39: Analýza 3 – orig – podíly pro celkový sentiment a typy dokumentů

sentiment	fb_post	fb_comment	yahoo	twitter
pozitivní	13,58	12,37	17,59	1,97
negativní	2,22	2,75	2,20	0,08
neutrální	84,20	84,88	80,22	97,95

Tab. 40: Analýza 3 – FS added – podíly pro celkový sentiment a typy dokumentů

sentiment	fb_post	fb_comment	yahoo	twitter
pozitivní	18,88	18,83	21,65	2,52
negativní	0,89	1,12	0,04	0,14
neutrální	80,23	80,05	78,31	97,34

Tab. 41: Analýza 3 – only FS – podíly pro celkový sentiment a typy dokumentů

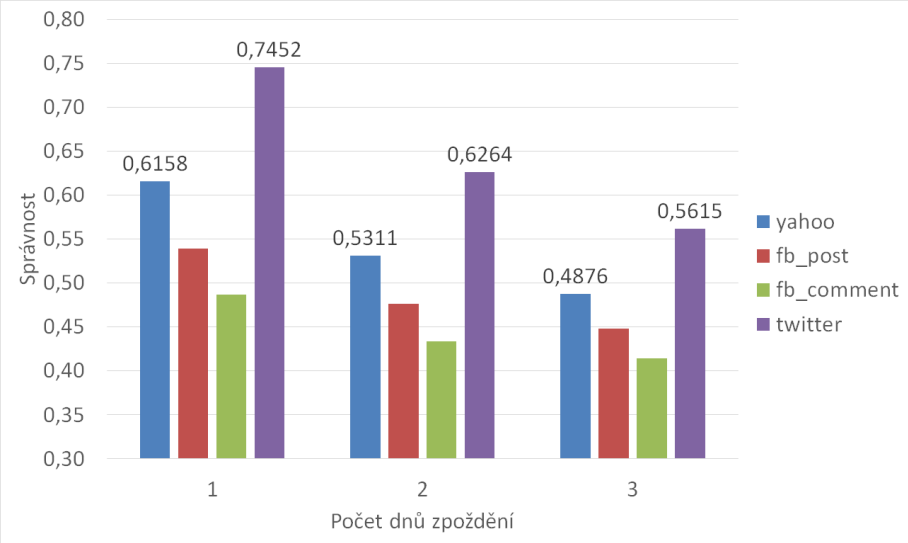
sentiment	fb_post	fb_comment	yahoo	twitter
pozitivní	18,39	13,97	21,70	2,50
negativní	0,39	0,30	0,01	0,16
neutrální	81,22	85,73	78,28	97,34

4.5.4 Výsledky analýzy

Před vyhodnocením výsledků je nutné si připomenout, že Twitter statusy jsou z 97 % klasifikovány jako neutrální a že 80 % (při zpoždění 1) cenových pohybů je konstantních. Tím pádem existuje vysoká pravděpodobnost ($0,97 \cdot 0,80 = 0,776$), že celkový sentiment daného dne bude určen jako neutrální, což bude zároveň odpovídat cenovému pohybu. Je to vidět na výsledných grafech, kde má Twitter vždy nejvyšší správnost. Pro ostatní typy dokumentů je tato pravděpodobnost menší (např. $0,8 \cdot 0,8 = 0,64$ pro Yahoo), takže se budeme zabývat především jimi.

Dále se zaměříme pouze na zpoždění jeden den, jelikož jen to má přímou souvislost se sentimentem předchozího dne. Navíc z grafů plyne, že nejvyšší správnost je vždy pro zpoždění 1 a že pro 2 a 3 správnost klesá. Nicméně pro úplnost jsou v grafech jak tweety, tak zpoždění 2 a 3 uvedeny.

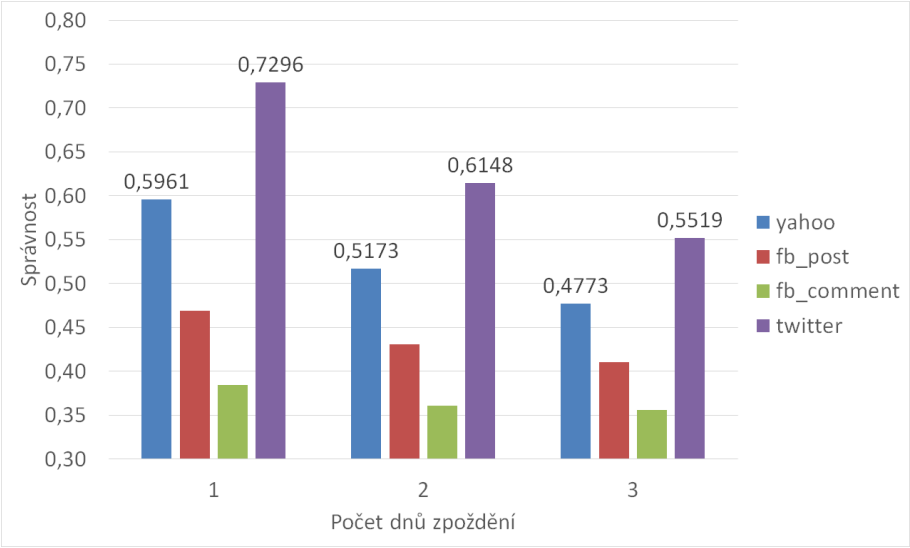
Obrázek 43 ukazuje správnost pro slovník 1. Lze vidět, že nejlepší správnost má Yahoo (0,62). To se dá považovat za uspokojivý výsledek a lze tak říct, že sentiment obsažený v článcích nějak souvisí s pohybem ceny akcie (resp. má na něj vliv). Facebook dokumenty poskytují nízkou správnost, přičemž komentáře ji mají nižší než příspěvky.



Obr. 43: Průměrná správnost pro slovník 1

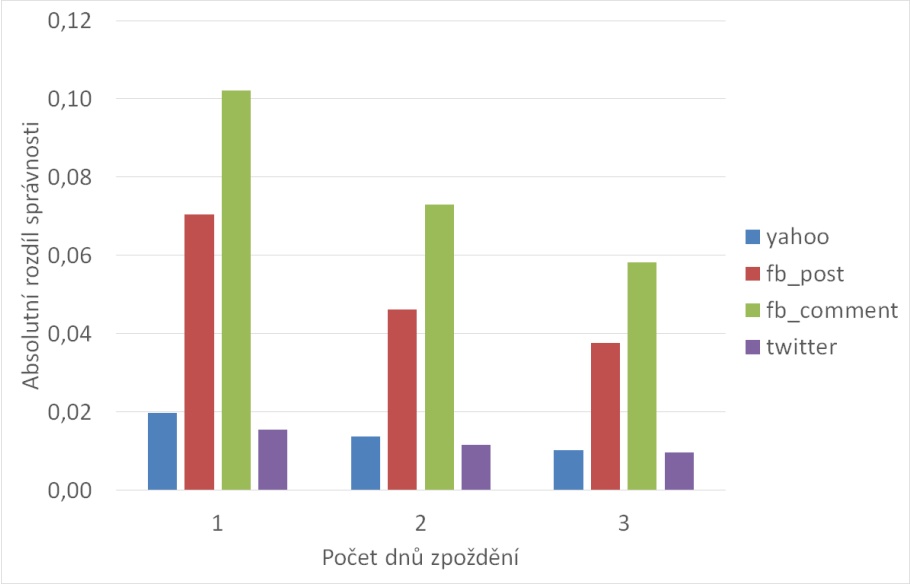
Obrázek 44 ukazuje správnost pro slovník 2 (kombinovaný slovník obohacený o slova z analýzy 2). Pořadí typů dokumentů dle správnosti je stejné jako u slovníku 1. Nicméně je zajímavé, že Facebook dokumenty a zejména komentáře mají nižší správnost, než při použití slovníku 1.

Obrázek 45 tyto dva grafy porovnává. Hodnoty na ose y jsou rozdíly hodnot správnosti slovníku 1 a 2 ($y_i = y_{i1} - y_{i2}$). Jelikož jsou všechny hodnoty kladné, tak pro každý typ dokumentu platí, že slovník 2 dává horší správnost než slovník 1. Stejně tak platí, že rozdíly se zmenšují s počtem dnů zpoždění. Lze vidět, že pro



Obr. 44: Průměrná správnost pro slovník 2

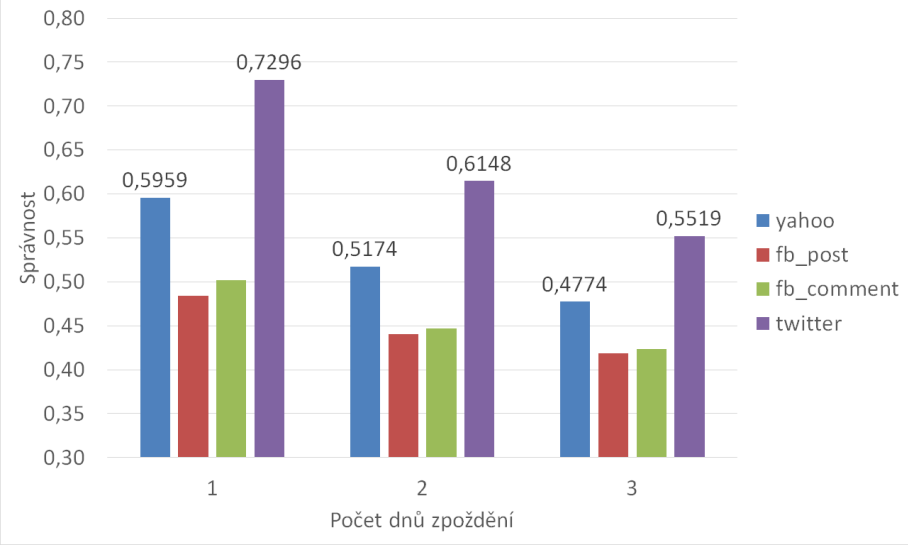
Twitter a Yahoo jsou rozdíly 1–2 %. Situace je odlišná pro Facebook příspěvky, kde slovník 2 poskytuje asi o 7 % horší správnost než slovník 1. Pro Facebook komentáře je správnost horší o 10 %.



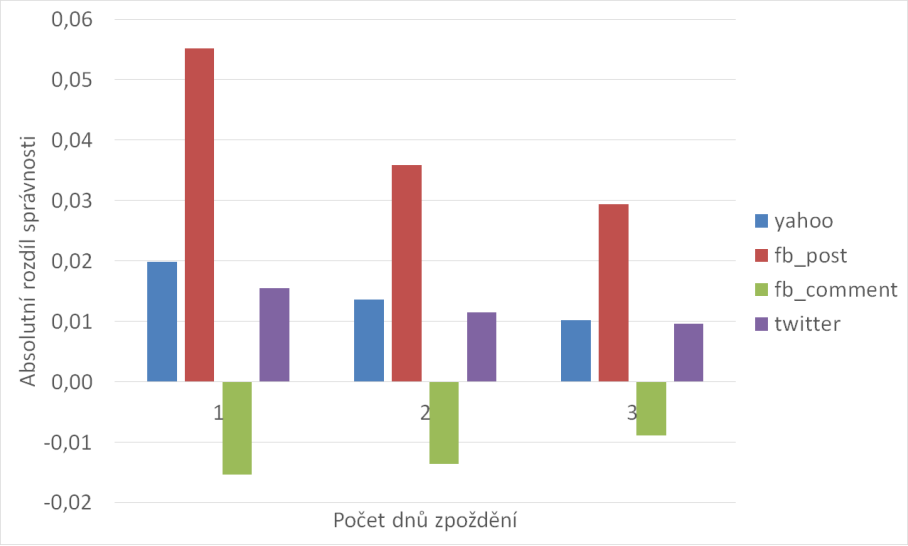
Obr. 45: Rozdíly správností mezi slovníky 1 a 2

Důvodem pro výše popsané rozdíly je zřejmě to, že slova vygenerovaná pomocí *feature selection* nenesou informaci o sentimentu dokumentu. Jsou to totiž často názvy firem či produktů resp. různé pojmy, které v daný zkoumaný čas sice souvisely s pohybem ceny akcie, ale nejsou obecně použitelné pro všechny firmy.

Obrázek 46 ukazuje správnost pro slovník 3 (pouze slova z analýzy 2). Pořadí typů dokumentů dle správnosti je stejné jako u slovníku 1. Tedy až na FB komentáře, které mají mírně lepší správnost než FB příspěvky. Pokud správnost porovnáme se slovníkem 1 (viz obrázek 47), tak lze vidět, že rozdíly jsou 1–6 % ve prospěch slovníku 1. Zajímavé je, že pro Facebook komentáře poskytuje slovník 3 lepší správnost (o 1,5 %). Vzhledem k tomu, že slovník 3 obsahuje jen 10 % slov oproti slovníku 1 a byl vytvořen automaticky, jsou výsledky velmi dobré.

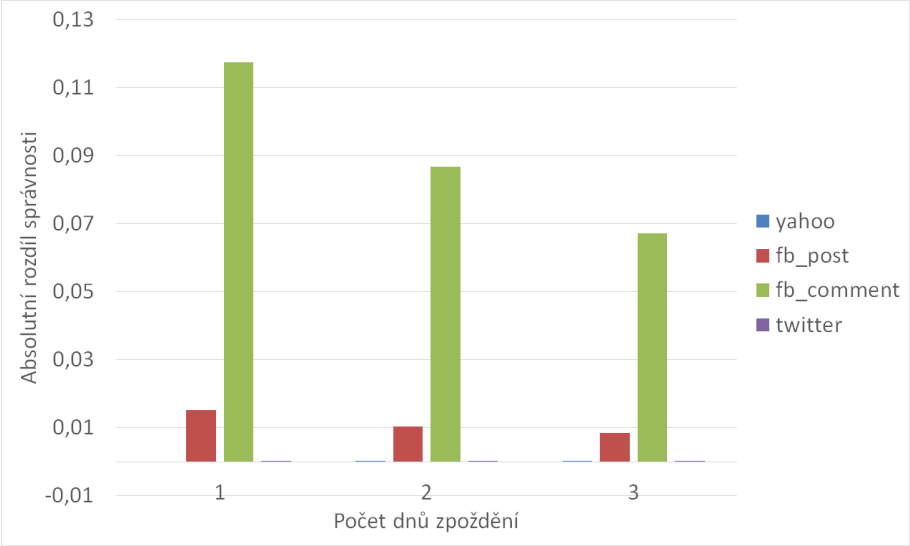


Obr. 46: Průměrná správnost pro slovník 3



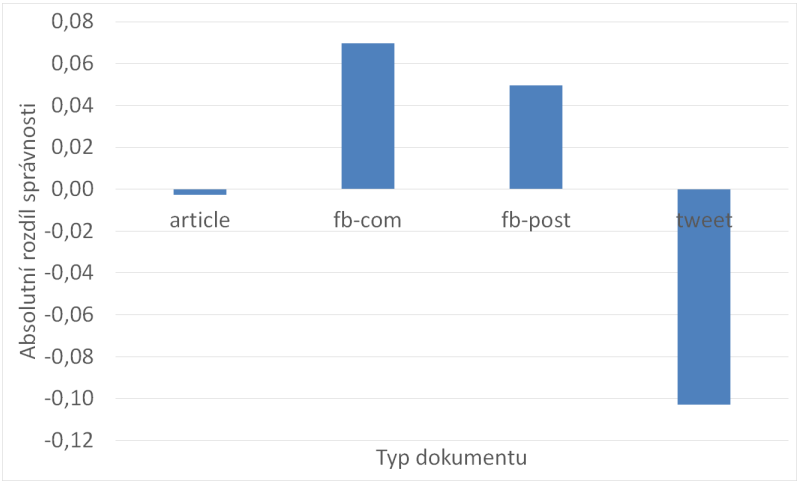
Obr. 47: Rozdíly správností mezi slovníky 1 a 3

Obrázek 48 zobrazuje rozdíly správnosti mezi slovníky 3 a 2. Lze vidět, že pro Yahoo jsou správnosti obou slovníků v podstatě totožné. Pro Facebook příspěvky je slovník 3 mírně lepší (1 %), ale pro Facebook komentáře je lepší výrazně (téměř 12 %). Pro Twitter je slovník 3 nepatrně lepší.



Obr. 48: Rozdíly správností mezi slovníky 3 a 2

Pokud výsledky (pro zpoždění jeden den) porovnáme s analýzou 1 (viz sekce 4.3.8), tak získáme obrázek 49. Ten značí, že pro Yahoo články poskytují obě metody v podstatě stejnou správnost, zatímco pro FB dokumenty je o 7 resp. 5 % lepší metoda strojového učení. Pro Twitter je výrazně lepší slovníková metoda, ale důvodem je již zmíněná drtivá převaha neutrálně určených tweetů.



Obr. 49: Rozdíly správností mezi analýzou 1 a 3

5 Diskuze

V této kapitole jsou zhodnoceny dosažené výsledky (implementované metody a provedené analýzy). Výsledky jsou stručně shrnuty a případně interpretovány, dále je poukázáno na problémy, které bylo v práci nutné překonat a jsou uvedeny návrhy na rozšíření práce. Nakonec je nastíněno využití výsledků v praxi.

5.1 Zhodnocení a interpretace výsledků analýz

5.1.1 Analýza č. 1

Analýza 1 se zabývala tím, jak souvisí obsah dokumentu s pohybem ceny akcie firmy, o které dokument pojednává. Pro klasifikaci byly zvoleny dvě třídy, které popisují, zda cena akcie stoupla nebo klesla. Dokumenty, pro které relativní změna ceny akcie související firmy spadala do konstantního intervalu, byly vyřazeny. Výsledky byly zpracovány pro každý typ dokumentu zvlášť (sekce 4.3.2 až 4.3.5) i pro všechny dokumenty celkově (sekce 4.3.6). Pro každý zdrojový soubor byla vybrána jen ta nejvyšší dosažená správnost (daná použitým typem vektoru a algoritmem).

Správnost klasifikace obecně

Na základě dosažených výsledků lze konstatovat, že pokud jsou cenové pohyby dostatečně výrazné a jdoucí proti aktuálnímu trendu, existuje jasná souvislost. Ukázat to lze na situaci pro cenovou proměnnou EWMA, jeden den zpoždění a hranici 3 až 5 %, kdy je správnost klasifikace nad 80 % (viz tabulka 34). Toto neplatí pro FB příspěvky, které dosahují správnosti nad 80 % jen v jednom případě. Nicméně je nutné podotknout, že takto prudká změna trendu nastává jen výjimečně – důkazem je, že pro 10 analyzovaných firem z Twitteru nenastala ani jednou. V každém případně je nutné vzít v potaz, že počet testovaných dokumentů je zde velmi malý, tudíž výsledky nejsou úplně reprezentativní a je možné, že se algoritmy rozhodují např. podle názvu firmy než podle ostatních „normálních“ slov.

Pokud vezmeme v úvahu podmínku, že celkový počet zkoumaných dokumentů musí být ≥ 500 (data použitá v detailní analýze), dostaneme realističtější výsledky (viz obrázek 42). Zde se ukazuje, že pro horních 10 % souborů (seřazených dle správnosti) je správnost pro Yahoo články 0,74, pro tweety také 0,74, pro FB komentáře 0,69 a pro FB příspěvky 0,66. Tyto výsledky (především pro Yahoo a Twitter) ukazují, že souvislost existuje. Pokud se podíváme na situaci pro horních 50 %, lze vidět, že Yahoo a Twitter si stále udržují dobrou hodnotu správnosti (0,67 resp. 0,69), ale FB komentáře resp. FB příspěvky ji mají nižší (0,63 resp. 0,61).

Průměrná maximální správnost pro všechny zdrojové soubory (viz obrázek 27) potvrzuje výše uvedené pořadí typů dokumentů dle správnosti, přičemž správnosti jsou v podstatě stejné (až na Yahoo a Twitter, pro které jsou hodnoty prohozené) jako ty uvedené výše pro horních 50 % souborů.

Pro určité hodnoty parametrů se tedy správnost klasifikace pohybuje v rozmezí 70–80 % (případně i více). Bylo tak prokázáno, že obsah dokumentu (publikovaného v daný den o dané firmě) v určitých tržních situacích jasně souvisí s tím, zda cena akcie (reprezentovaná především klouzavým průměrem) klesne nebo stoupne.

Hodnoty parametrů

Byly zkoumány tři parametry: cenová proměnná (Adcjlose, SMA, EWMA), počet dnů zpoždění (1, 2, 3) a hranice konstantního intervalu (1 . . . 5). Na základě hodnot těchto parametrů byly generovány zdrojové soubory – pokud pohyb akcie související firmy splňoval dané hodnoty, byly dokumenty přidány do souboru, jinak nebyly. Parametry v podstatě značí, jak velká resp. prudká a neočekávaná (vzhledem k aktuálnímu trendu) byla změna ceny akcie. Výsledky potvrzují, že vysoké hodnoty správnosti byly dosaženy právě pro velké změny, což je logické.

Z celkové analýzy (sekce 4.3.6) plyne, že nejlepší průměrnou maximální správnost poskytuje cenová proměnná EWMA (0,72), jeden den zpoždění (0,67) a 5% hranice intervalu (0,67). Proměnná SMA (0,63) je lepší než Adjclose pouze o 0,02.

Mezi zdrojovými soubory, které mají průměrně pro všechny typy dokumentů nejlepší správnost (viz tabulka 34), jasně dominuje proměnná EWMA, přičemž SMA se poprvé objevuje až na 12. místě (0,66) a Adjclose až na 24. místě (0,63).

Dále byla provedena analýza pro jednotlivé typy dokumentů (viz sekce 4.3.7). Její závěry potvrzují, že pro všechny typy dokumentů je nejlepší cenová proměnná EWMA, jeden den zpoždění a že správnost roste s rostoucí hranicí konstantního intervalu. Výjimkou jsou tweety, pro které je nejlepší zpoždění dva dny a 2% hranice.

Závěry z detailní analýzy všech souborů (viz obrázek 34) v podstatě odpovídají výše uvedeným informacím. Ohledně cenových proměnných lze konstatovat, že Adjclose se v horních 25 % nevyskytuje ani jednou a v horních 50 % má 17% podíl, zatímco SMA má podíl 25 resp. 31 %.

Nakonec byla provedena analýza pro cenovou proměnnou Adjclose a zpoždění jeden den, což je nejlépe představitelná situace. V sekci 4.3.8 vyšlo najevo, že správnost roste s hranicí konstantního intervalu (až na Yahoo, kde je nejlepší pro 1 %), přičemž hodnota pro 5 % je okolo 0,64. To značí, že pouhé jednodenní výkyvy zřejmě nemají hlubší souvislost s publikovanými texty. Průměrná hodnota pro všechny hranice je pro Yahoo články 0,63 a pro tweety 0,64, zatímco pro FB komentáře je 0,57 a pro FB příspěvky 0,60. To jen potvrzuje skutečnost, že s pohyby akcií souvisejí především Yahoo články a tweety a že FB dokumenty tak jasnou souvislost nemají.

Celkově se dá konstatovat, že nejlepších výsledků dosahuje cenová proměnná EWMA, u které ani příliš nezáleží na zpoždění a hranici intervalu. I pro jiné kombinace než hranici 3–5 % a zpoždění 1 totiž poskytuje dobrou správnost v rozmezí 0,7–0,8[27]. Důvodem pro to je zřejmě skutečnost, že i malá změna v celkovém 20den-

[27] Tedy zejména pro Yahoo a Twitter. Výsledky pro FB komentáře jsou ještě relativně dobré, ale pro FB příspěvky příliš přesvědčivé nejsou.

ním exponenciálním trendu znamená, že se děje něco mimořádného, a tato situace pokračuje i v dalších dnech.

Použité algoritmy a typy vektorů

V práci byly také zkoumány výsledky klasifikace pro různé typy vektorů a použité algoritmy. Bylo provedeno mnoho experimentů, což se dá využít pro porovnání toho, jak vhodné jsou jednotlivé vektory a algoritmy pro klasifikaci textových dat. Průměrně nejvyšší správnost pro všechny zdrojové soubory (viz obrázek 32) poskytuje algoritmus LinearSVC (0, 70) a nejmenší NB-multi (0, 62). Vhodná volba algoritmu tedy může znamenat nárůst správnosti až o 8 %, i když reálně jsou rozdíly maximálně v jednotkách procent. Nejlepší je vektor tf-idf-no, avšak rozdíly jsou nepatrné.

Zajímavější jsou výsledky rozdělené dle typu dokumentů (viz konec sekce 4.3.7), které zkoumají průměrnou správnost všech provedených experimententů. Zde se ukazuje, že nejhorší je vždy algoritmus CART a každý typ dokumentu má svůj nejlepší algoritmus (LinearSVC pro Yahoo, NB-berno pro FB-post, NB-multi pro FB-com, RandForest pro Twitter). Výsledky pro vektory jsou si velmi podobné (rozdíl ±1 %). Každopádně nejlepší je vždy tf-idf-no, přičemž výjimkou jsou opět tweety, pro které je nejlepší tp-no-no. Detailní analýza (viz obrázek 34) odhalila, že pro všechny soubory dohromady byl nejčastěji nejlepší vektor tf-idf-cos a algoritmus NB-multi (resp. LinearSVC pro horních 50 % souborů).

Návrhy na alternativní provedení analýzy 1

Nakonec uvedeme návrhy na další možné varianty provedení této analýzy:

- Klasifikace pro tři třídy (dokumenty s neutrálním pohybem by se nevyřazovaly); jinak definované třídy; použití shlukování pro nalezení tříd či významných skupin dokumentů; použití asociačních pravidel.
- Více hodnot pro tři hlavní parametry. Cenová proměnná – další typy klouzavého průměru (kumulativní, aritmetický vážený, vážený objemem obchodů). Počet dnů zpoždění – také 4 a 5 dnů. Hodnot hranice intervalu může být daleko více: např. 1–10 s krokem 0,5. Ovšem je nutné si uvědomit, že každá přidaná možnost způsobí zvýšení počtu experimentů, které je nutné provést.
- Zkoumat nevyvážená data – v případě vyvážení dat (metodou *undersampling*) obsahuje výsledný datový soubor méně dokumentů, než kolik jich reálně splňuje stanovené parametry. Je možné, že použití více dokumentů by lépe odhalilo skryté vazby v datech, ale také by zároveň mělo vliv na věrohodnost výsledků.
- Kromě unigramů použít také bigramy, trigramy, případně jejich kombinace. Nicméně tento krok by výrazně znásobil počet nutných experimentů. Navíc generování vektorových souborů by trvalo dlouho a mohly by být příliš veliké pro provedení klasifikace v rozumném čase.

- Přiřadit dokumentům různou důležitost (váhu) – dle počtu liků, retweetů a sdílení článků – a zahrnout ji do zkoumání. Jednou možností je vynásobit prvky vektoru určitou hodnotou (zde ale není jasné, jaké by to mělo důsledky na schopnost klasifikace). Další možností je zvolit nějaký jiný (zřejmě ekonometrický) model pro predikci. Jeden takový jednoduchý model uvádí Petrovský (2015).
- Pro vyhodnocení výsledků musela být použita nějaká metoda – byl použit tabulkový procesor Microsoft Excel. Zajímavé by bylo použít, stejně jako v Arias et al. (2013), rozhodovací strom, pomocí kterého by bylo možné automaticky zvolit konfigurace parametrů, vedoucí k nejlepším výsledkům.

5.1.2 Analýza č. 2

V rámci analýzy 2 byla pomocí metody CHI (*Feature selection*) vybrána informačně významná slova (z TEXT souborů, exportovaných pro zadané parametry). Seznam obsahuje celkem 1 001 slov. Na předních místech (viz tabulka 36) se nacházejí především názvy (resp. části názvů) firem, ale od pozice 10 jsou v seznamu již přítomna běžná slova jako „investigating“ nebo „unstoppable“.

K získání slov byl použit program *stopwords.pl*, který načte zadané dokumenty, získá všechna slova a zadanou metodou spočítá pro každé slovo hodnotu metriky. Kritickou částí je správné nalezení slov, které ale evidentně proběhlo bez problémů. Zvolena byla ověřená metoda CHI (je otázkou, jaké výsledky by poskytly jiné metody). Program byl upraven tak, aby poskytoval výstup v požadovaném formátu.

Seznam by bylo možné upravit. Názvy firem jsou pro použití seznamu pro jiné (neanalyzované) firmy zbytečné. Pokud by byly odstraněny, tak je ale otázkou, jaké by byly výsledky analýzy 3.

5.1.3 Analýza č. 3

Analýza 3 zahrnovala jak vytvoření nových slovníků sentimentu, tak provedení samotné analýzy sentimentu. Co se týče prvního bodu, byly vytvořeny tři slovníky. První byl vytvořen kombinací existujících slovníků, druhý přidáním slov z analýzy 2 a poslední pouze slovy z analýzy 2. Pro slova z analýzy 2 jsme předpokládali, že třída 1 (pohyb nahoru) znamená pozitivní a třída 2 (pohyb dolů) negativní sentiment.

V analýze bylo zjišťováno, jak počet pozitivních/neutrálních/negativních (maximum určuje celkový sentiment daného typu dokumentu) dokumentů pro daný den souvisí s tím, zda cena akcie (vzhledem k 1, 2, 3 dalším dnům) klesla, stoupla nebo zůstala konstantní. Pro vygenerování cenových změn byla zvolena cenová proměnná Adjclose a 2% hranice konstantního intervalu.

Výsledná správnost se velmi liší dle typu dokumentů (následují údaje pro nejlepší slovník 1 a zpoždění 1 den). Pro Yahoo články je 62 %, zatímco pro FB příspěvky a komentáře je pod 55 %. Znamená to, že na základě článků lze ve více než 60 % případů určit, zda vzhledem ke konci následujícího dne cena akcie klesla, stoupla, nebo zůstala konstantní v intervalu $(-2; 2)$. To se dá považovat za dobrý výsledek.

Pro tweety je správnost 0,75 – je ale nutné vzít v potaz, že 97 % všech tweetů je klasifikováno jako neutrálních a 80 % změn cen je konstantních, takže tato správnost nemá příliš vysokou vypovídající hodnotu.

Bylo také provedeno srovnání analýzy 1 a analýzy 3 (viz obrázek 49), ze kterého vyplývá, že pro Yahoo články je správnost téměř stejná, zatímco pro FB dokumenty je lepší analýza 1. Ta je pro Twitter horší o 10 %, ale jak již bylo řečeno, tento údaj není příliš relevantní. Srovnání sice bylo provedeno pro stejné parametry, ale počet tříd byl rozdílný (2 vs. 3). To ale nemění nic na zajímavém faktu, že pro Yahoo články vyšla pomocí obou metod stejná správnost.

Pozoruhodné bylo, že výsledky slovníku 3 (tvořený pouze slovy z analýzy 2) byly pro Yahoo a Twitter jen mírně horší (o 2 % resp. 1,5 %) než slovníku 1 (originální kombinovaný slovník) – viz obrázek 47. Pro FB komentáře je správnost slovníku 3 dokonce o 1,5 % vyšší, pro FB příspěvky je nicméně výrazně (o 5,5 %) horší. Výhodou slovníku 3 je, že byl vygenerován automaticky. To znamená, že je možné vytvořit slovník sentimentu na základě metody, popsané v analýze 2. A to zřejmě i pro jinou doménu než pro zprávy z akciových trhů resp. ze sociálních sítí.

Seznam slov z analýzy 2 nebyl ručně zkontrolován, takže je možné, že výsledné polarity neodpovídaly běžnému významu daných slov. Bylo vygenerováno pouze 1 000 slov s nejlepší hodnotou metriky. Je možné, že více slov by přineslo lepší výsledek, ale na druhou stranu je logické, že níže umístěná slova nebudou tak dobře určovat hranici mezi třídami (hodnoty metriky jsou si totiž velmi podobné).

Výsledky analýzy závisí na hodnotách použitých parametrů: interval pro neutrální sentiment, hranice konstantního intervalu a použitá cenová proměnná. Experimenty byly provedeny pouze pro jednu sadu hodnot: $(-0,05; +0,05)$, $\pm 2\,\%$, Adjclose. Vyzkoušení všech možných kombinací by zřejmě přineslo lepší výsledky, nicméně by si to vyžádalo velmi dlouhý čas. Dalším faktorem je způsob určování celkového sentimentu pro daný den. Případně lze uvažovat o použití jiných slovníků.

5.2 Získaná data a modul pro získávání dat

Prvním krokem práce bylo získat požadovaná data. K tomu musel být vytvořen modul pro získávání dat. Je implementován v jazyce Python a umožňuje automaticky získávat data z definovaných zdrojů. Databáze použitá pro analýzu má velikost asi 4 GiB, přičemž na serveru Sosna stahování dat stále běží.

Úkol získávání dat z veřejných internetových zdrojů se může zdát jednoduchý, ale jak bylo zjištěno, úplně tomu tak není. I když je k dispozici veřejné API, je nutné vyřešit určité problémy (viz sekce 4.1.3). Pokud žádné API dostupné není, je nutné získat požadované informace parsováním HTML kódu dané webové stránky. K tomuto účelu naštěstí existuje mnoho programových knihoven. Nicméně struktura stránek je obvykle chaotická a může se dokonce během času měnit. To znamená, že k dosažení požadované funčknosti je potřeba provést řadu pokusů (viz sekce 4.1.4).

Tuto část práce lze samozřejmě vylepšit. Archiv Yahoo Finance, sahající až do roku 2013, obsahuje daleko více článků, než jich je v databázi. Tyto by bylo možné

stáhnout a použít pro analýzu. Zadruhé – ukládají se pouze články, publikované na portálu Yahoo Finance, jelikož tyto mají stejnou strukturu a lze z nich získat text jedinou metodou. Avšak existuje mnoho známých serverů (např. Forbes.com), které mají vlastní strukturu. Pokud by byly pro tyto servery vytvořeny další metody pro parsování, počet uložených článků by se dále zvýšil.

Ohledně Facebooku se dá uvažovat o čtení textů ze specifických skupin, které se zabývají obchodováním na akciových trzích. Ty je ovšem nutné najít, získat do nich přístup a je otázkou, zda bude povoleno čtení jejich obsahu skrz API. O autorech Facebook komentářů nejsou ukládány žádné informace kromě jejich jména a ID. Důvodem je, že API neposkytuje k podrobným údajům přístup. Musely by tedy být získávány (po přihlášení jako uživatel) pomocí web-scrappingu.

Twitter dává skrz své Streaming API přístup ke všem tweetům, tudíž počet ukládaných tweetů resp. pokládaných dotazů by mohl být zvýšen. Nicméně pro tento účel by musela být navržena jiná architektura a způsob ukládání takového objemu dat. V neposlední řadě by mohly být přidány další zdroje, uvedené v sekci 2.3.3 (reddit, diskuzní fóra), což by si ale vyžádalo náročnou manuální práci.

5.3 Moduly pro zpracování a analýzu dat

V rámci práce byly také implementovány moduly pro zpracování a export dat (DataProcessor), konverzi (převod textů na vektory) a klasifikaci dat (AnalPipeline) a pro analýzu sentimentu pomocí slovníku (DataAnalyzer).

Export dat přinášel problémy především v případě tweetů, kterých je v databázi velké množství, nejsou seřazené pro každou firmu chronologicky a získávaly se pro každý den samostatně. Použitím paměti cache (v MySQL) se tento problém naštěstí podařilo překonat.

Dá se uvažovat o jiných způsobech předzpracování textů. Např. z tweetů by mohla být odstraněna uživatelská jména, názvy firem či řetězec „rt“ (značí retweety). Počet nahrazovaných emotikon by mohl být rozšířen. Nabízí se také možnost spojit jednotlivé dokumenty (např. Facebook komentáře k jednomu příspěvku) do větších celků a tyto poté analyzovat. V neposlední řadě by mohlo být zajímavé odstranit stopslova, ať už získaná automaticky nebo z obecného seznamu.

Je nutné poznamenat, že převod textů na vektory pomocí programu VecText trval nečekaně dlouho. Například soubory pro Twitter se zpracovávaly téměř 14 hodin. Je zajímavé, že soubor STAT byl vytvořen poměrně rychle, ale samotný zápis do souboru DAT trval dlouho. Možná by tedy stálo za zvážení zápis vektorů urychlit. Byly zvoleny určité parametry pro tvorbu vektorů (min. délka a výskyt slov, typ normalizace), které bychom mohli měnit a zkoumat vliv těchto změn na výsledky klasifikace. Nicméně toto nebylo (ani vzdáleně) tématem práce.

Klasifikace pomocí scikit-learn probíhala nad očekávání rychle a pohodlně. Oproti programu Weka, který byl použit pro prvotní experimenty, se jedná o propastný rozdíl. Nicméně algoritmus SVM (RBF a polynomiální kernel) běžel velmi dlouho a podával špatné výsledky. Otázkou je, co by se změnilo použitím programu

SVM^{light}. Samozřejmě by mohlo být zajímavé vyzkoušet i další klasifikační algoritmy (jako např. Hoeffdingův strom), ale je nutné si uvědomit, že hlavní je dobře si připravit data, přičemž konkrétní metoda následné analýzy není tak důležitá.

Modul pro určování sentimentu pomocí slovníku využívá algoritmus VADER, který je určen (vyladěn) pro analýzu sentimentu na úrovni vět. Jelikož jsou (kromě Yahoo článků) zkoumané dokumenty (až na výjimečné případy) krátké (max. několik vět), není to zásadní problém. Pro Yahoo články byla použita normalizace hodnoty sentimentu počtem vět v článku. Je otázkou, jak by se výsledky změnily použitím jiného způsobu normalizace – např. vážený průměr (počtem slov) nebo na základě distribuce (rozložení) hodnot sentimentu mezi všemi dokumenty.

Všechny moduly fungují správně a dobře posloužily svému účelu. Nicméně jako u každého SW projektu, bylo by záhodno jejich zdrojový kód refaktorovat, případně přidat další funkce. Také by si zasloužily lepší dokumentaci. Ale jelikož hlavní veřejné metody zdokumentovány jsou a Python je přehledný programovací jazyk, nejedná se o zásadní problém.

5.4 Využití modulů a poznatků v praxi

Na závěr se zamyslíme nad tím, jak by bylo možné využít implementované moduly a získané poznatky (výsledky analýz) v praxi.

5.4.1 Moduly

Modul *DataGetter* má poměrně univerzální použití. Pouhým zadáním názvu firmy, jména Facebook stránky nebo Twitter účtu do tabulky `company` lze začít stahovat data pro danou firmu. Firma toho může využít několika způsoby.

Zaprvé tak může provádět zálohu svých účtů na sociálních sítích, aby zde publikované statusy byly, i v případě neočekávané havárie či problémů poskytovatele, přístupné pro zpětnou analýzu. Stejně tak může automaticky získávat články, které jsou o ní publikovány na Yahoo Finance. Pro zálohu on-line dat existují samozřejmě i komerční služby. Pro kontinuální zálohování to je např. `http://www.datto.com/backupify`, pro jednorázové např. `http://services.socialsafe.net/`. Výhodou první služby je, že podporuje více různých zdrojů dat, poskytuje certifikace bezpečnosti a firma se nemusí starat o další aplikaci a databázi. Její nevýhodou je, že data leží na serverech poskytovatele a je zde potenciální riziko, že k datům nebude mít firma přístup. Cena služby bude zřejmě závislá na velikosti firmy a počtu uložených dat a dá se očekávat, že nebude nízká.

Zadruhé lze stažená data napojit na interní systémy a např. nad nimi vytvořit nějakou formu datového skladu, pomocí kterého je bude možné analyzovat z mnoha pohledů. Ukládány jsou i údaje o historickém vývoji počtu liků pro FB příspěvky a komentáře, čehož se dá využít pro sledování dynamiky a typických vzorů chování uživatelů na sociálních sítích.

Zatřetí lze stahovat také data konkurenčních firem a provádět tak analýzu konkurenčního prostředí – jak často konkurenti publikují statusy, kolik mají liků, komentářů, kolik se o nich publikuje článků apod.

V rámci těchto analýz je možné použít různé techniky NLP a text miningu pro zjištění zajímavých informací: např. jaká slova jsou často používaná, jaké hlavní druhy textů jsou přítomné, jak jsou zákazníci spokojeni s produkty dané firmy apod.

Účelem modulu *AnalPipeline* je převést zdrojové TEXT soubory do vektorové podoby a provést nad nimi klasifikaci. Je jasné, že lze takto převádět jakákoliv textová data, stejně jako klasifikovat jakákoliv číselná data (musí být uložena ve formátu SVMlight). Program VecText má mnoho různých parametrů, které je možné v cyklu měnit a zkoušet, jaké hodnoty přinášejí nejlepší výsledky.

Co se týče klasifikace, tak výhodou modulu je, že lze snadno zvolit použité algoritmy, experimenty probíhají automaticky a výsledky se zapisují v přehledné formě do CSV souboru, který lze následně manuálně či strojově zpracovat. Knihovna scikit-learn umožňuje (pokud má k dispozici dostatečně velkou operační paměť) i analýzu značných objemů dat (Big Data). Pro běžné potřeby tak není nutné používat žádné distribuované řešení, ale stačí jeden server s dostatečně výkonným procesorem a velkou pamětí RAM. Pokud je RAM nedostatečná, lze využít algoritmy pro inkrementální učení, které scikit-learn také obsahuje.

Modul *DataProcessor* slouží pro specifickou přípravu dat pro klasifikaci, tudíž jeho použití je poměrně omezené. To stejné platí o modulu *DataAnalyzer*, jehož účelem je analyzovat sentiment jednotlivých dokumentů. Využití těchto modulů souvisí s tím, jak přínosné jsou dosažené výsledky analýz (viz následující sekce). Pokud jsou výsledky kvalitní, je možné uvedené moduly využívat v praxi.

5.4.2 Výsledky analýz

Výsledky *analýzy 1* ukazují, že pokud je cenový pohyb výrazný, existuje značná souvislost mezi obsahem dokumentů a směrem pohybu ceny akcie. Můžeme se tak podívat na zdrojové textové soubory a zjistit, jaké zprávy a události zapříčinily tento pohyb. Toho mohou využít manažeři firem, aby zjistili, co je důvodem aktuálního pohybu ceny, zjistili jaké názory mají vliv na tento pohyb a tím pádem lépe pochopili chování investorů. To stejné může dělat např. i centrální nebo běžná banka s cílem analyzovat nálady investorů a vysvětlovat pohyby cen akcií na celém trhu.

Bylo by lákavé využít analýzu 1 pro předpovídání toho, zda cena akcie klesne či stoupne a dle toho akcii prodat či koupit. Je pravdou, že na základě vytvořeného modelu je možné klasifikovat nová, aktuální data a dle toho předpovídat, zda cena akcie stoupne či klesne. To ale platí jen pokud nastanou specifické podmínky dané použitými parametry – např. pokud SMA klesne o 3 %. Problém je ten, že v daném dni nevíme, jak se cena zítra (či za dva nebo tři dny) změní a tudíž nevíme, jaký konkrétní model zvolit.

Otázkou je, jak tuto situaci vyřešit. Prvním krokem je výběr typu dokumentů, druhým je volba hodnot použitých parametrů. Následuje příklad postupu. Vezmeme

základní cenovou proměnnou Adjclose. Z grafu 41 vyčteme, že nejlepší průměrné správnosti dosahují Yahoo články (0,63) a tweety (0,64). Získáme vytvořené modely pro 1–5% hranici intervalu, vložíme do nich nová testovací data a výsledný počet kladných resp. záporných předpovědí určí nejpravděpodobnější výsledný směr pohybu ceny akcie. Nicméně je logické, že tento postup nebude vždy fungovat a bude dosahovat dlouhodobé úspěšnosti maximálně v hodnotě výše zmíněných správností.

Výstupem *analýzy 2* je seznam 1 000 slov s nejvyšší informační hodnotou pro určení třídy dokumentu. Na jeho základě lze zjistit, jaká slova souvisí s tím, zda cena akcie stoupla či klesla. Daná firma si také může modulem DataProcessor vygenerovat zdrojové soubory, obsahující pouze dokumenty, týkající se jí, a z těchto souborů vygenerovat seznam. Jeho prozkoumáním např. zjistí, že název určitého produktu způsobuje růst ceny, jméno konkurenční firmy znamená pokles ceny apod.

Výsledky *analýzy 3* ukazují, že existuje určitá souvislost mezi sentimentem dokumentů a tím, zda cena akcie zůstane konstantní, stoupne či klesne. Nicméně objektivně řečeno, tato souvislost není nikterak vysoká – pro Yahoo články byla správnost 0,62. Využití modulu DataAnalyzer může spočívat v tom, že firma jím na konci každého dne vyhodnotí, jak byla daný den vnímána u veřejnosti. Následně pak může zkoumat, jak se toto projevilo na ceně akcií.

V rámci práce byl vytvořen kombinovaný slovník sentimentu a upraven algoritmus VADER. Ten tak lze použít pro určování sentimentu dokumentů, týkajících se firem na akciových trzích (případně i pro jiné oblasti), přičemž podporovány jsou jak krátké dokumenty ze sociálních sítí, tak dlouhé novinové články.

Samozřejmě k analýze sentimentu na sociálních sítích existuje řada komerčních nástrojů (např. Twinword, AlchemyAPI, Semantria), které poskytují mnoho funkcí. Ale jejich nevýhodou je, že jsou placené a umístěné na vzdáleném serveru. Navíc není jasné, na jakém principu fungují a zda jsou vhodné pro specifickou doménu zpráv z akciových trhů.

Předpovídání ceny akcie na základě sentimentu naráží na stejný problém volby parametrů jako u analýzy 1. Ale výhodou je, že pozitivní sentiment zřejmě znamená růst ceny a negativní její pokles. Zbývá nám tedy zvolit vhodné hodnoty pro typ cenové proměnné a hranici konstantního intervalu (můžeme použít např. testované hodnoty Adjclose a $\pm 2\,\%$). Poté je teoreticky možné uvedený postup využít pro předpovídání pohybu ceny akcie na další den. Jednoduše to uděláme tak, že před koncem obchodního dne spočítáme celkový sentiment a pokud je pozitivní, nakoupíme akcie a pokud negativní, tak je prodáme (*short sell*) nebo neuděláme nic.

Nicméně správnost pro výše uvedené hodnoty parametrů není příliš vysoká (pro Yahoo články 0,62), takže je otázkou, zda by byla dostatečná k zajištění dlouhodobé ziskovosti. Dále je nutné vzít v potaz, že pro většinu dní je cenový pohyb konstantní a stejně tak je sentiment většiny dokumentů neutrální, což nedává informaci o tom, zda prodat či koupit akcii. Navíc k reálnému obchodování tímto způsobem by bylo nutné provést rozsáhlé testování na dosud neanalyzovaných firmách a to v průběhu delšího časového období.

6 Závěr

Cílem práce bylo provést sběr, zpracování a analýzu textových dat pro oblast finančních trhů, se zaměřením na sentiment textů a jeho vliv na akciový trh. Cíl práce resp. všechny dílčí cíle byly splněny.

Kapitola 2 tvoří přehledovou část práce. Jsou zde popsány finanční a akciové trhy, zdroje a způsoby získávání dat z internetu a přístupy k uchovávání dat. Je podrobně rozebráno dolování znalostí z textových dat a analýza sentimentu.

Kapitola 3 popisuje, jaký byl postup práce resp. co a jak bylo nutné udělat: zvolit sledované firmy a datové zdroje, navrhnout modul pro získávání dat a databázi pro ukládání dat a nakonec data analyzovat. Každý tento krok je podrobně popsán a je uvedeno, proč byl zvolen daný postup.

Kapitola 4 prezentuje dosažené výsledky. Sekce 4.1 se týká získaných dat a zajišťuje splnění prvního dílčího cíle. Nejdříve je představen modul pro získávání dat, který umožňuje automaticky stahovat data ze zadaných zdrojů a ukládat je do databáze. Následně jsou popsána data, získaná v období 1. 8. 2015 – 4. 4. 2016: databáze má velikost asi 4 GiB, přičemž obsahuje asi 82 tisíc Yahoo článků, 135 tisíc Facebook příspěvků, přes 2 miliony Facebook komentářů a téměř 4 miliony Twitter statusů. Obsah této sekce byl (spolu s dalšími informacemi) autorem práce sepsán do článku (Petrovský, 2015) a úspěšně prezentován na konferenci PEFnet 2015.

Sekce 4.2 popisuje další implementované moduly. Jedná se o modul pro zpracování a export dat (DataProcessor), modul pro konverzi (převod textů na vektory) a klasifikaci dat (AnalPipeline) a modul pro analýzu sentimentu pomocí slovníků (DataAnalyzer). Vytvořením těchto modulů byl splněn druhý dílčí cíl.

Další sekce čtvrté kapitoly prezentují výsledky analýz, provedených pomocí výše uvedených modulů. Analýza 1 měla za cíl zjistit, jak souvisí obsah dokumentu se směrem pohybu ceny akcie firmy, které se dokument týká. Bylo zjištěno, že pokud je cenový pohyb (oproti aktuálnímu trendu) dostatečně výrazný, existuje poměrně jasná souvislost – správnost klasifikace se pohybovala na úrovni 70–80 %. Cílem analýzy 2 bylo na základě exportovaných TEXT souborů určit informačně významná slova. Byla použita metoda CHI a bylo získáno 1 000 slov s nejvyšší hodnotou.

Analýza 3 zkoumala souvislost mezi sentimentem dokumentů a směrem pohybů cen akcií. Pro určení sentimentu byla použita slovníková metoda, která měla na vstupu tři různé slovníky. Bylo zjištěno, že správnost se pohybuje okolo 60 %. Provedením a vyhodnocením těchto analýz byl splněn poslední, třetí dílčí cíl.

Kapitola 5 obsahuje zhodnocení a interpretaci dosažených výsledků, popisuje problémy, které bylo v práci nutné překonat, a jsou zde uvedeny návrhy na rozšíření práce. Nakonec je diskutována využitelnost výsledků v praxi.

7 Literatura

ALEXA INTERNET. *About us — Alexa Internet* [online]. 2016 [cit. 2016-03-10]. Dostupné z: `http://www.alexa.com/about`.

APACHE SOFTWARE FOUNDATION. *Apache Hive TM* [online]. 2014 [cit. 2016-03-17]. Dostupné z: `https://hive.apache.org/`.

ARIAS, M., ARRATIA, A. a XURIGUERA, R. Forecasting with Twitter Data. *ACM Transactions on Intelligent Systems and Technology (TIST).* 2013, vol. 5, no. 1, s. 8.

ARLOW, J. a NEUSTADT, I. *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design.* 2nd Edition. Westford, USA: Addison-Wesley Professional, 2005. ISBN 978-0321321275.

BARRIE, J. *Google+ Active Users — Business Insider* [online]. 2015 [cit. 2016-03-11]. Dostupné z: `http://uk.businessinsider.com/google-active-users-2015-1`.

BOLLEN, J., MAO, H. a ZENG, X. Twitter mood predicts the stock market. *Journal of Computational Science.* 2011, vol. 2, no. 1, s. 1–8.

CALTECH. *Historical Stock Data — Caltech Quantitative Finance Group* [online]. 2015 [cit. 2016-03-16]. Dostupné z: `http://quant.caltech.edu/historical-stock-data.html`.

CARBONNELLE, P. *PYPL PopularitY of Programming Language index* [online]. 2015 [cit. 2016-04-09]. Dostupné z: `http://pypl.github.io/PYPL.html`.

COELHO, L. P. a RICHERT, W. *Building Machine Learning Systems with Python.* Second Edition. Birmingham: Packt Publishing, 2015. ISBN 978-1-78439-277-2.

CONNOLLY, T. M. a BEGG, C. E. *Database Systems: A Practical Approach to Design, Implementation and Management.* 4th Edition. Harlow: Addison Wesley, 2005. ISBN 0321210255.

DAŘENA, F. *VecText manual — František Dařena, MENDELU* [online]. 2016 [cit. 2016-05-17]. Dostupné z: `https://akela.mendelu.cz/~darena/VecText/VecText-manual.pdf`.

EDLICH, S. *NoSQL databases* [online]. 2009 [cit. 2016-03-17]. Dostupné z: `http://nosql-database.org/`.

EISENSTEIN, E. L. *The Printing Press as an Agent of Change.* Cambridge, UK: Cambridge University Press, 1979. ISBN 0521220440.

ENGE, E. *Hard Numbers for Public Posting Activity on Google Plus — Stone Temple Consulting* [online]. 2015 [cit. 2016-03-11]. Dostupné z:

https://www.stonetemple.com/real-numbers-for-the-activity-on-google-plus/.

FACEBOOK. *Graph API Reference — Facebook Developers* [online]. 2016 [cit. 2016-04-08]. Dostupné z: https://developers.facebook.com/docs/graph-api/reference/.

FELDMAN, R. a SANGER, J. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press, 2007. ISBN 978-0-521-83657-9.

FTSE RUSSELL. *FTSEurofirst* [online]. 2016 [cit. 2016-03-04]. Dostupné z: http://www.ftse.com/products/indices/Eurofirst.

GLADIŠ, D. *Naučte se investovat*. 2. rozšířené vydání. Praha: Grada, 2005. ISBN 80-247-1205-9.

GO, A., BHAYANI, R. a HUANG, L. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*. 2009, vol. 1, s. 12.

GOLDBERG, A. *Automatic Summarization — CS838-1 Advanced NLP* [online]. 2007 [cit. 2016-03-23]. Dostupné z: http://pages.cs.wisc.edu/~jerryzhu/cs838/summarization.pdf.

GOOGLE. *Google+ API — Google+ Platform for Web, Google Developers* [online]. 2016 [cit. 2016-03-11]. Dostupné z: https://developers.google.com/+/web/api/rest/.

GREENE, S. *WebScaleSQL: A collaboration to build upon the MySQL upstream — Engineering Blog, Facebook Code* [online]. 2014 [cit. 2016-03-18]. Dostupné z: https://code.facebook.com/posts/1474977139392436/webscalesql-a-collaboration-to-build-upon-the-mysql-upstream/.

HAN, J., KAMBER, M. a PEI, J. *Data Mining: Concepts and Techniques*. 3rd ed. Waltham, MA: Morgan Kaufmann, 2012. ISBN 978-0-12-381479-1.

HIPP. *SQLite Home Page* [online]. 2016 [cit. 2016-03-17]. Dostupné z: http://www.sqlite.org/.

HOTHO, A., NURNBERGER, A. a PAAß, G. A Brief Survey of Text Mining. *Ldv Forum*. 2005, vol. 20, no. 1.

HUTTO, C. J. a GILBERT, E. *cjhutto/vaderSentiment: VADER Sentiment Analysis — Github.com* [online]. 2014 [cit. 2016-03-30]. Dostupné z: https://github.com/cjhutto/vaderSentiment.

HUTTO, C. J. a GILBERT, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: *Eighth International AAAI Conference on Weblogs and Social Media*. 2014.

INVESTOPEDIA. *Tick Definition — Investopedia* [online]. 2016 [cit. 2016-03-16]. Dostupné z: http://www.investopedia.com/terms/t/tick.asp.

JOACHIMS, T. Making large-Scale SVM Learning Practical. In: *Advances in Kernel Methods - Support Vector Learning*. Ed. by SCHÖLKOPF, B., BURGES, C. a SMOLA, A. Cambridge, MA: MIT Press, 1999, chap. 11, s. 169–184.

KAPLANSKI, G. a LEVY, H. Sentiment and stock prices: The case of aviation disasters. *Journal of Financial Economics*. 2010, vol. 95, no. 2, s. 174–201.

KAUSHIK, C. a MISHRA, A. A scalable, lexicon based technique for sentiment analysis. *International Journal in Foundations of Computer Science & Technology (IJFCST)*. 2014, vol. 4, no. 5.

KRUPNÍK, J. *Automatizace generování stopslov*. Brno, 2014. Diplomová práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta.

KUMAR, R. *High Disk write and space taken by PostgreSQL — PostgreSQL performance, Nabble* [online]. 2012 [cit. 2016-03-18]. Dostupné z: http://postgresql.nabble.com/High-Disk-write-and-space-taken-by-PostgreSQL-td5720029.html.

LINKEDIN. *Home — LinkedIn Developer Network* [online]. 2016 [cit. 2016-03-11]. Dostupné z: https://developer.linkedin.com/.

LIU, B. *Sentiment Analysis and Opinion Mining*. San Rafael: Morgan & Claypool Publishers, 2012. Synthesis lectures on human language technologies. ISBN 978-1-60845-884-4.

LORICA, B. *Six reasons why I recommend scikit-learn — O'Reilly Radar* [online]. 2013 [cit. 2016-04-29]. Dostupné z: http://radar.oreilly.com/2013/12/six-reasons-why-i-recommend-scikit-learn.html.

LOUGHRAN, T. a MCDONALD, B. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*. 2011, vol. 66, no. 1, s. 35–65.

LUKEBUEHLER. *source of historical stock data — Stack Overflow* [online]. 2013 [cit. 2016-03-16]. Dostupné z: http://stackoverflow.com/a/17263126.

MAAS, A. L., DALY, R. E., PHAM, P. T., HUANG, D., et al. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, vol. 1, s. 142–150.

MANNING, C. D. a SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, USA: The MIT Press, 1999. ISBN 9780262133609.

MARIADB FOUNDATION. *About MariaDB — MariaDB.org* [online]. 2016 [cit. 2016-03-17]. Dostupné z: `https://mariadb.org/about/`.

MIKULÁŠ, R. *Školní slovník cizích slov*. Bratislava: Príroda, s. r. o., 2007. ISBN 978-80-07-01491-6.

MITCHELL, C. *How To Use A Moving Average To Buy Stocks — Investopedia* [online]. 2016 [cit. 2016-05-17]. Dostupné z: `http://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp`.

MYŠKOVÁ, R. a HÁJEK, P. Novel Multi-word Lists for Investors' Decision Making. In: *Text, Speech, and Dialogue: Proceedings of 18th International Conference, TSD 2015, Pilsen, Czech Republic, September 14–17, 2015*. 2015, s. 131–139. Lecture Notes in Artificial Intelligence.

NARAYANAN, V., ARORA, I. a BHATIA, A. Fast and accurate sentiment classification using an enhanced Naive Bayes model. In: *Intelligent Data Engineering and Automated Learning–IDEAL 2013*. Springer, 2013, s. 194–201.

NARR, S., HULFENHAUS, M. a ALBAYRAK, S. Language-independent twitter sentiment analysis. *Knowledge Discovery and Machine Learning (KDML), LWA*. 2012, s. 12–14.

NIST. *EWMA Control Charts — e-Handbook of Statistical Methods* [online]. 2012 [cit. 2016-05-17]. Dostupné z: `http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm`.

NOVAK, P. K., SMAILOVIĆ, J., SLUBAN, B. a MOZETIČ, I. Sentiment of emojis. *PLoS ONE*. 2015, vol. 10, no. 12, s. e0144296.

ORACLE. *MySQL Connectors — Oracle* [online]. 2016 [cit. 2016-04-08]. Dostupné z: `http://dev.mysql.com/downloads/connector/`.

OTIPKA, P. a ŠMAJSTRLA, V. *Statistický soubor s jedním argumentem — Pravděpodobnost a statistika, studijní opora VŠB* [online]. 2013 [cit. 2016-05-13]. Dostupné z: `http://homen.vsb.cz/~oti73/cdpast1/KAP07/KAP07.htm`.

PAK, A., PAROUBEK, P. a FRAISSE, A. Normalization of TermWeighting Scheme for Sentiment Analysis. In: *Human language technology challenges for computer science and linguistics: 5th Language and Technology Conference, LTC 2011, Poznan, Poland, November 25–27, 2011, Revised Selected Papers*. 2014, s. 116–128. ISBN 978-3-319-08957-7.

PANG, B. a LEE, L. A sentimental education: Sentiment analysis using subjectivity. In: *Proceedings of ACL*. 2004, s. 271–278.

PANG, B., LEE, L. a VAITHYANATHAN, S. Thumbs up?: sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. 2002, vol. 10, s. 79–86.

PETROVSKÝ, J. Acquiring and Analyzing Text Data for Stock Market Modelling. In: *PEFnet 2015: abstracts : European scientific conference of doctoral students : Brno, November 19, 2015*. 2015, s. 54.

PRELERT. *MySQL versus PostgreSQL — Prelert Blog* [online]. 2014 [cit. 2016-03-18]. Dostupné z: `http://info.prelert.com/blog/mysql-versus-postgresql`.

PROVALIS RESEARCH. *Sentiment Dictionaries for WordStat Content Analysis Software* [online]. 2016 [cit. 2016-03-30]. Dostupné z: `http://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/sentiment-dictionaries/`.

RÁBOVÁ, I. *Podnikové informační systémy a technologie jejich vývoje*. Brno: Tribun EU, 2008. ISBN 978-80-7399-599-7.

REDDIT. *about reddit* [online]. 2016 [cit. 2016-03-16]. Dostupné z: `https://www.reddit.com/about/`.

REJNUŠ, O. *Finanční trhy*. 4., aktualizované a rozšířené vydání. Brno: Grada Publishing, 2014. ISBN 978-80-247-3671-6.

ROE, S. F. *Understanding the Bias-Variance Tradeoff* [online]. 2012 [cit. 2016-05-09]. Dostupné z: `http://scott.fortmann-roe.com/docs/BiasVariance.html`.

RUSSELL, M. A. *Mining the Social Web*. Second Edition. Sebastopol, CA: O'Reilly Media, 2013. ISBN 978-1-449-36761-9.

SCIKIT-LEARN. *Supervised learning — scikit-learn 0.17.1 documentation* [online]. 2013 [cit. 2016-04-29]. Dostupné z: `http://scikit-learn.org/stable/supervised_learning.html`.

SHILLER, R. J. From efficient markets theory to behavioral finance. *The Journal of Economic Perspectives*. 2003, vol. 17, no. 1, s. 83–104.

SILBERSCHATZ, A., KORTH, H. a SUDARSHAN, S. *Database System Concepts*. 6th Edition. Harlow: McGraw-Hill Education, 2010. ISBN 978-0073523323.

SOCHER, R., PERELYGIN, A., WU, J. Y., CHUANG, J., et al. Recursive deep models for semantic compositionality over a sentiment treebank. In:

Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013, vol. 1631, s. 1642.

SOLID IT. *Method of calculating the scores of the DB-Engines Ranking* [online]. 2016 [cit. 2016-03-17]. Dostupné z: http://db-engines.com/en/ranking_definition.

SOLID IT. *Popularity ranking of relational DBMS — DB-Engines Ranking* [online]. 2016 [cit. 2016-03-17]. Dostupné z: http://db-engines.com/en/ranking/relational+dbms.

STACK EXCHANGE. *Is there a good forum where I can discuss individual US stocks? — Personal Finance & Money Stack Exchange* [online]. 2015 [cit. 2016-03-16]. Dostupné z: http://money.stackexchange.com/questions/4447/.

STATISTA. *Leading global social networks 2016* [online]. 2016 [cit. 2016-03-11]. Dostupné z: http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

SUTTON, R. S. a BARTO, A. G. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998. ISBN 9780262193986.

TABOADA, M., BROOKE, J., TOFILOSKI, M., VOLL, K., et al. Lexicon-based methods for sentiment analysis. *Computational linguistics*. 2011, vol. 37, no. 2, s. 267–307.

TUMBLR. *Tumblr API* [online]. 2016 [cit. 2016-03-11]. Dostupné z: https://www.tumblr.com/docs/en/api/v2.

TWITTER. *Documentation — Twitter Developers* [online]. 2016 [cit. 2016-04-08]. Dostupné z: https://dev.twitter.com/overview/documentation.

TWITTER. *REST APIs — Twitter Developers* [online]. 2016 [cit. 2016-04-08]. Dostupné z: https://dev.twitter.com/rest/public.

USZKOREIT, H. *CL Intro Text* [online]. 2000 [cit. 2016-03-23]. Dostupné z: http://www.coli.uni-saarland.de/~hansu/what_is_cl.html.

WEISS, S. M., INDURKHYA, N. a ZHANG, T. *Fundamentals of Predictive Text Mining*. London: Springer, 2010. ISBN 978-1-84996-225-4.

WIKIPEDIA. *Bloomberg Terminal — Wikipedia, The Free Encyclopedia* [online]. 2016 [cit. 2016-03-16]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Bloomberg_Terminal&oldid=706117471.

WIKIPEDIA. *Moving average — Wikipedia, The Free Encyclopedia*. 2016. Dostupné také z: https://en.wikipedia.org/w/index.php?title=Moving_average&oldid=710326158.

WIKIVS. *MySQL vs PostgreSQL — WikiVS* [online]. 2016 [cit. 2016-03-18]. Dostupné z: `https://www.wikivs.com/wiki/MySQL_vs_PostgreSQL`.

WITTEN, I. H., FRANK, E. a HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. Burlington, MA: Morgan Kaufmann, 2011. ISBN 978-0-12-374856-0.

YAHOO! *IXIC Components, NASDAQ Composite — Yahoo! Finance* [online]. 2009 [cit. 2016-03-04]. Dostupné z: `http://finance.yahoo.com/q/cp?s=%5EIXIC+Components`.

ŽIŽKA, J. *Prezentace do předmětu Informační systémy pro rozhodování*. Brno: Mendelova univerzita v Brně, 2009.

Přílohy

A Elektronická příloha

Přílohou této práce je DVD s elektronickým obsahem.

- `/`
  - `thesis/` ..... **Text diplomové práce**
    - `dp-print.pdf` ..... Vytisknutá diplomová práce
    - `dp-color_links.pdf` ..... Práce obsahující barevné odkazy
    - `dp-full_research.pdf` ..... Původní verze práce s kompletní rešerší
    - `dp-latex/` ..... Zdrojové soubory práce pro systém LaTeX
  - `results/` ..... **Výsledky analýz**
    - `anal_1/` ..... Analýza 1 (klasifikace)
    - `anal_2/` ..... Analýza 2 (Feature selection)
    - `anal_3/` ..... Analýza 3 (určování sentimentu)
  - `FinanceAnalyzer/` ..... **Program FinanceAnalyzer**
    - `AnalPipeline/` ..... Modul pro převod na vektory a klasifikaci
    - `DataAnalyzer/` ..... Modul pro určování sentimentu pomocí slovníku
    - `DataGetter/` ..... Modul pro získávání dat
    - `DataProcessor/` ..... Modul pro (před)zpracování dat
    - `db_files/` ..... Soubory související s vytvořenou databází
  - `various/` ..... **Různé soubory**
    - `companies.xlsx` ..... Seznam zkoumaných firem
    - `stopwords-krupnik/` ..... Upravený program stopwords.pl

Poznámka: Z důvodu velikosti a možných problémů s autorským právem nejsou v příloze přítomny vygenerované (zdrojové) TEXT soubory ani zdrojová databáze.

B Rešerše – doplňkové informace

Tab. 42: Porovnání MySQL a PostgreSQL

Vlastnost	MySQL	PostgreSQL
Implementační jazyk	C, C++	C
Podpora SQL	SQL:1999 + rozšíření	SQL:2011 + rozšíření
Programování na serveru	jazyk založený na SQL/PSM	PL/pgSQL, Perl, Python, Tcl, Javascript
Table partitioning[28]	ano	pouze přes dědičnost tabulek a trigery
Podpora XML/JSON	ano/ano	ano/ano – lepší
Automatická komprese textu	ne	ano – TOAST
Full-text vyhledávání	základní	pokročilé
Hodnoty primárních klíčů	automaticky	pomocí sekvencí
GIS datové typy	dle OpenGIS	plugin PostGIS
GUI klient pro správu DB	MySQL Workbench	pgAdmin
Více typů úložišť	ano	ne
Čtení dat z ostatních systémů	ne	ano
Nativní asynchronní klient API	ne[29]	ano
Transakce	ano (kromě DDL)	ano
Typy indexů	b-tree, hash, fulltext	b-tree, r-tree, hash, expression, partial, reverse
Max. velikost tabulky	64 TB	32 TB
Příkazy Insert Ignore / Replace	ano	ne[30]

[28]Horizontální škálování – umožňuje ukládat vybrané (dle určitého kritéria) řádky z tabulky do více různých souborů.

[29]Speciální verze MySQL zvaná WebScaleSQL tuto funkcionalitu podporuje (Greene, 2014).

[30]Lze simulovat pomocí procedur, ale lze tak vložit naráz pouze jednu hodnotu (WikiVS, 2016).

Tab. 43: Slovníky sentimentu – odkazy

Název	URL
AFINN	http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010
Bing Liu opinion lexicon	http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
Novel multi-word lists (Myskova, Hajek)	Contact authors of (Myšková a Hájek, 2015).
Henry word list	http://papers.ssrn.com/sol3/papers.cfm?abstract_id=933100
LabMT	http://www.plosone.org/article/fetchSingleRepresentation.action?uri=info:doi/10.1371/journal.pone.0026752.s001
Lexicoder Sentiment Dictionary	http://www.lexicoder.com/download.html
Loughran and McDonald Financial sentiment dictionary	http://www3.nd.edu/~mcdonald/Word_Lists_files/LoughranMcDonald_MasterDictionary_2014.xlsx
MICRO-WNOP corpus	http://www.unipv.it/wnop/micrownop.tgz
OpinionFinder's Subjectivity Lexicon	http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
SenticNet 1.0	http://sentic.net/senticnet-1.0.zip
SentiWordNet 3.0	http://sentiwordnet.isti.cnr.it/downloadFile.php
VADER	https://github.com/cjhutto/vaderSentiment
Warriner affective ratings	http://crr.ugent.be/papers/Ratings_Warriner_et_al.csv
WordStat sentiment dictionary 1.2	http://provalisresearch.com/Download/WSD.zip

Tab. 44: Zdroje dat o akciích a jejich charakteristiky

Název	Cena	Rozlišení	Vyřazené akcie	Historie	Změna symbolů	Trhy	Indexy
Yahoo! Finance	zdarma	denní	ne	max	ne	celý svět	ano
QuantQuote Free	zdarma	denní	ne	1998	ano	S&P 500	ne
CSI data	$285/rok pro U.S. akcie a indexy, $600 pro celý svět	denní	ano (za příplatek)	10/30 let	ano	celý svět	ano
QuantQuote	$895 za S&P 500, $9000 za všechny symboly	min, sec, tick	ano	1998	ano	USA	ne
TickData	?	min, tick	?	?	ano	celý svět	?
NYSE TAQ	$3 000/měsíc	tick	ano	1993	ne	USA	ne
Compustat	tisíce dolarů	denní	ano	1960	?	?	
CRSP	?	denní	ano	100 let	?	USA	ano
Xignite	?	denní	1993 USA, 2000 ostatní	?	celý svět	ano	
Nanex	$4 000/rok	tick	ano	2004	ano	USA	ano

Tab. 45: Přehled výsledků *supervised* algoritmů pro oblast určování sentimentu

Algoritmus	Korpus	Správnost [%]	Publikace
ME	tweety (Sentiment140)	80,50	(Go et al., 2009)
ME	Recenze filmů (LP02)	80,40	(Pang et al., 2002)
NB	tweety (Sentiment140)	81,30	(Go et al., 2009)
NB	Large Movie Review Dataset	82,80	(Narayanan et al., 2013)
NB	tweety (Dai-LABOR)	81,30	(Narr et al., 2012)
NB	Recenze filmů (LP02)	78,70	(Pang et al., 2002)
NB	Sentiment Treebank	82,60	(Socher et al., 2013)
NB	Recenze filmů (LP04)	82,80	(Pang a Lee, 2004)
RNTN	Sentiment Treebank	87,60	(Socher et al., 2013)
SVM	tweety (Sentiment140)	82,20	(Go et al., 2009)
SVM	Large Movie Review Dataset	87,80	(Maas et al., 2011)
SVM	Recenze filmů (LP02)	82,90	(Pang et al., 2002)
SVM	Sentiment Treebank	84,60	(Socher et al., 2013)
SVM	Recenze filmů (LP04)	87,15	(Pang a Lee, 2004)

C Popis klasifikačních algoritmů

Popis algoritmů použitých v analýze 1 vychází především ze scikit-learn (2013).

Naive Bayes

Naivní Bayes je rychlý a jednoduchý model, který aplikuje Bayesovský teorém s tím, že předpokládá nezávislost mezi všemi dvojicemi atributů. Třída $\hat{y}$ je přiřazena dokumentu d, který je tvořen vektorem atributů $(x_1, \ldots, x_n)$, následovně:

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y) \; .$$

Jednotlivé verze tohoto klasifikátoru se liší především v tom, jakou předpokládají, že má člen $P(x_i \mid y)$ distribuci.

Multinomial NB přijímá na vstupu TF vektor, který pro každý atribut obsahuje počet jeho výskytů ve zkoumaném dokumentu. Bernoulli NB přijímá na vstupu TP vektor, ve kterém je pro každý atribut označeno, zda je nebo není přítomen ve zkoumaném dokumentu. Rozdílem je, že pravidlo pro Bernoulli NB explicitně penalizuje nepřítomnost atributu i, který je indikátorem pro třídu y, zatímco Multinomial NB tento atribut jen ignoruje (scikit-learn, 2013).

Logistic Regression

Tento model, nazývaný také Maximum Entropy, je založen na myšlence, že by měl být preferován ten nejvíce rovnoměrný model, který splňuje zadanou podmínku. Model narozdíl od Naivního Bayese nepředpokládá nezávislost mezi atributy. Je reprezentován následovně (Go et al., 2009):

$$P_{ME}(c \mid d, \lambda) = \frac{\exp[\Sigma_i \lambda f_i(c, d)]}{\Sigma_{\acute{c}} \exp[\Sigma_i \lambda f_i(c, d)]}$$

V tomto vzorci je c třída, d dokument a λ váhový vektor. Ten pro každý atribut určuje, jak důležitým indikátorem je pro danou třídu. Vektor je nalezen numerickou optimalizací všech λ tak, aby byla maximalizována podmíněná pravděpodobnost.

Rozhodovací strom CART

Cílem metody pro tvorbu rozhodovacích stromů je vytvořit model, který bude predikovat třídu pomocí jednoduchých rozhodovacích pravidel získaných z trénovacích dat. Scikit-learn používá algoritmus CART (Classification and Regression Trees), který je velmi podobný algoritmu C4.5, ale liší se tím, že podporuje numerické cílové proměnné (regresi) a nevytváří sady pravidel. CART buduje binární stromy pomocí atributu a prahu, které v každém uzlu dosahují největšího informačního zisku (scikit-learn, 2013).

Random Forest Classifier

Tento algoritmus patří do skupiny tzv. souborných metod (*ensemble methods*), jejichž cílem je zkombinovat predikce několika základních klasifikátorů ke zvýšení zobecnitelnosti či robustnosti. Rozlišujeme dva typy těchto metod:

- *Averaging* (průměrovací) – Principem je vytvořit nezávisle několik modelů a poté zprůměrovat jejich predikce, čímž obvykle vznikne lepší model.
- *Boosting* (vylepšovací) – Modely jsou budovány sekvenčně, přičemž každý další model se snaží snížit *bias* dosavadního kombinovaného modelu.

RandomForest patří mezi průměrovací metody. Každý strom v souboru je vytvořen ze vzorku náhodně získaného z trénovací množiny. Navíc při rozdělování uzlu (během budování stromu) není vybírán ten nejlepší předěl ze všech atributů, ale pouze z náhodné podmnožiny atributů. Kvůli této náhodnosti se *bias* celého lesu mírně zvýší, ale díky průměrování se sníží jeho (*variance*), což má obvykle za následek vytvoření celkově lepšího modelu (scikit-learn, 2013).

Chyby při predikci mají dva hlavní důvody: *bias* (zaujatost) a *variance* (odlišnost). Schopnosti modelu minimalizovat *bias* a *variance* jsou protichůdné. Chyba kvůli *bias* je rozdíl mezi predikcí našeho modelu a správnou hodnotou. Chyba kvůli *variance* je variabilita predikce modelu pro daný vstup. V případě souborných metod zkoumáme průměr těchto rozdílů, což nám odpoví na otázku, jak kvalitní je vytvořený kombinovaný model (Roe, 2012).

SVM (Support Vector Machines)

Metody podpůrných vektorů jsou použitelné pro klasifikaci, regresi a detekci odlehlých případů. Scikit-learn podporuje prostřednictvím třídy `SVC` klasifikaci s lineárním, polynomiálním, RBF a sigmoidálním kernelem. Pro lineární kernel existuje speciální implementace ve třídě `LinearSVC`, která je více flexibilní ohledně volby penalizace a ztrátových funkcí a měla by lépe zvládat velké množství instancí.

SVM konstruuje nadrovinu (nebo množinu nadrovin) ve vysoko dimenzionálním prostoru, která odděluje data (představovaná body v prostoru) do dvou tříd. Dobré oddělení je dosaženo nadrovinou, která má největší vzdálenost k nejbližším trénovacím datům jakékoliv třídy. Jelikož, obecně řečeno, čím větší mezera, tím nižší je generalizační chyba klasifikátoru (scikit-learn, 2013).

D Moduly – zdrojové kódy

Kód 1: Doplnění cen akcie pro nepracovní dny (DataGetter)

```
def _refill_yahoo_data(self, company_id, orig_yahoo_data, start_date,
    end_date):
        """
        For every missing day (i) from start to end date, insert
            artificial value: d_i = (d_i-1 + d_i+1)/2
        If yahoo date do not go to end_date, end function prematurely.
        """
        # Prepare variables
        current_date = start_date
        plus_day = datetime.timedelta(days=1)
        y_last_day_i = 0
        all_days = []
        # Get last date from Yahoo.
        newest_yahoo_date = datetime.datetime.strptime(orig_yahoo_data
            [0].split(',')[0], '%Y-%m-%d').date()
        # Reverse data from oldest to newest.
        yahoo_data = list(reversed(orig_yahoo_data))
        # Loop all days from given interval.
        while current_date <= end_date:
            # Check if current date is not higher than the newest yahoo
                date.
            if current_date > newest_yahoo_date:
                break
            # Find the date in Yahoo data.
            for pr_i, price_str in enumerate(yahoo_data[y_last_day_i:],
                y_last_day_i):
                price_list = price_str.split(',')
                y_date = datetime.datetime.strptime(price_list[0], '%Y
                    -%m-%d').date()
                # Skip too early dates.
                if y_date < start_date:

def _calculate_new_var_value(self, yahoo_data, last_day_i, var_pos_1,
    var_pos_2):
        date_list_1 = yahoo_data[last_day_i].split(',')
        date_list_2 = yahoo_data[last_day_i + 1].split(',')
        new_va1ue = (float(date_list_1[var_pos_1]) + float(date_list_2[
            var_pos_2])) / 2.0
        return new_va1ue
```

Kód 2: Zpracování textu před exportem (DataProcessor)

```
def _process_facebook_text(self, text):
        # Remove hash tag symbols.
        text = text.replace('#', '')
        # Remove at symbols.
        text = text.replace('@', '')
        # Replace URL links.
        text = re.sub(r'https?://\S+', 'XURL', text)
        # Replace emoticons with descriptions.
        text = re.sub(r':\)|:—\)|:D|=\)', ' XyzPosEmoticon ', text)
        text = re.sub(r':\(|:—\(', ' XyzNegEmoticon ', text)
        # Remove whitespace.
        text = ' '.join(text.strip().split())
        # Lowercase the text.
        text = text.lower()
        # Result
        return text

def _process_article_text(self, text):
        # Remove URL links.
        #text = re.sub(r'(https?://\S+)|(www\.\w+\.\S+)', 'URL', text)
        text = re.sub(r'https?://\S+', 'XURL', text)
        # Remove paragraph tags.
        text = re.sub(r'<p>|</p>', '', text)
        # Lowercase the text.
        text = text.lower()
        # Result
        return text
```

Kód 3: Získání tweetů pro daný den (DataProcessor)

```
def get_daily_tweets_for_company(self, company_id, for_date,
    docs_query_limit=1000):
        """
        Get tweets created on given day. Exclude duplicates using GROUP
            BY(text).
        """
        cursor = self.dbcon.cursor(dictionary=True)
        query = "SELECT SQL_CACHE created_at, text, retweet_count FROM
            tw_status " \
                "WHERE DATE(created_at) = \%s AND company_id = \%s " \
                "GROUP BY(text) ORDER BY retweet_count DESC LIMIT \%s"
        cursor.execute(query, [for_date, company_id, docs_query_limit])
        return cursor
```

Kód 4: Výpočet SMA (DataProcessor)

```
def calculate_sma_for_company(self, company_id, period_length):
        """
        Calculate and save to DB simple moving average of adjclose
            company price.
        """
        # Get daily closing prices.
        min_date = datetime.date(1900, 1, 1)
        close_prices = self.db_model.get_stock_prices(company_id,
            min_date, 'adjclose')
        # For first N-1 days just insert the close price.
        values = []
        for s_date, s_price in close_prices[0:period_length - 1]:
            values.append([company_id, s_date, s_price])
        # For remaining days calculate the MA: (sum of current +
            previous n - 1 days) / period length
        for p_i, (s_date, s_price) in enumerate(close_prices[(
            period_length - 1):], start=period_length - 1):
            # Save values to list.
            previous_days_sum = sum([x[1] for x in close_prices[(p_i -
                period_length + 1):p_i]])
            total_sum = previous_days_sum + s_price
            mov_avg = total_sum / float(period_length)
            values.append([company_id, s_date, mov_avg])
        # Save values to DB.
        self.db_model.update_stock_prices_for_company('sma', values)
```

Kód 5: Vyvážení dokumentů ve vektorovém souboru (AnalPipeline)

```
def _balance_given_file(input_filepath, output_filepath,
    min_class_count):
        """
        Create a new file, where each class will have the same number
            of items.
        """
        # Open files.
        input_file = open(input_filepath, 'r')
        output_file = open(output_filepath, 'w')
        # Define class count variables.
        c_1 = 0
        c_2 = 0
        # Loop through all documents (lines).
        for line in input_file:
            if line[0] == '1':
                c_1 += 1
                if c_1 <= min_class_count:
                    output_file.write(line)
            if line[0] == '2':
                c_2 += 1
                if c_2 <= min_class_count:
                    output_file.write(line)
        # OK
        return True


def _get_class_counts_from_stat_file(stat_filepath):
    file_stat = open(stat_filepath)
    f_lines = file_stat.readlines()
    try:
        c_1 = int(f_lines[25].split(' ')[0])
        c_2 = int(f_lines[31].split(' ')[0])
    except IndexError:
        print('>>Only one class in file, skipping it.')
        return False
    # OK
    return [c_1, c_2]
```

Kód 6: Provedení klasifikace pomocí scikit-learn (AnalPipeline)

```
def classify_data(clf_obj, X_train, X_test, y_train, y_test,
    save_model_filepath=False):
    # Train model.
    start_time = time.time()
    clf_obj.fit(X_train, y_train)
    train_runtime = round(time.time() - start_time, 6)
    # If desired, save model to disk.
    if save_model_filepath:
        joblib.dump(clf_obj, save_model_filepath)
    # Test model.
    start_time = time.time()
    y_predicted = clf_obj.predict(X_test)
    test_runtime = round(time.time() - start_time, 6)
    # Evaluate model.
    accuracy = metrics.accuracy_score(y_test, y_predicted)
    precision = metrics.precision_score(y_test, y_predicted, average='
        weighted', pos_label=None)
    recall = metrics.recall_score(y_test, y_predicted, average='
        weighted', pos_label=None)
    f1_score = metrics.f1_score(y_test, y_predicted, average='weighted'
        , pos_label=None)
    # Get number of tested samples for class 1 and 2.
    conf_matrix = metrics.confusion_matrix(y_test, y_predicted, labels
        =[1, 2])
    cl_1_count = conf_matrix[0][0] + conf_matrix[0][1]
    cl_2_count = conf_matrix[1][0] + conf_matrix[1][1]
    # Return data.
    tested_counts = [cl_1_count, cl_2_count]
    results = [accuracy, precision, recall, f1_score, train_runtime,
        test_runtime]
    eval_data = [y_test, y_predicted]
    return results, eval_data, tested_counts
```

E Doby běhu skriptů

Doby běhu skriptů byly získány zpětně na základě času poslední modifikace prvního a posledního souboru v daném výstupním adresáři (až na klasifikaci, pro kterou byl startovní čás získán z logu). Proto je možné, že nejsou úplně přesné.

Co se týče použitého hardwaru, tak modul AnalPipeline (a samozřejmě také DataGetter) byl spouštěn na virtuální serveru ÚIT MENDELU, který měl přiděleno 32 GiB operační paměti a 4 virtuální CPU. Modul DataProcessor byl spouštěn na notebooku s 8 GiB operační paměti, procesorem Intel Core i5-3340M (frekvence 2,7 Ghz, 2 jádra, 4 vlákna) a 256 GB SSD diskem.

Tab. 46: Doba běhu skriptů – získávání dat (DataGetter)

skript	`fb_update`	`fb_new`	`yahoo_update`	`yahoo_new`	`save_prices`	`tw`
doba běhu [h]	0:20	1:00	1:30	1:30[31]	0:10	0:01[32]

Tab. 47: Doba běhu skriptů – zpracování a export dat (DataProcessor)

typ dokumentů	Yahoo články	FB komentáře	FB příspěvky	Twitter
doba běhu [h]	0:47	7:26	0:45	0:38[33]

Tab. 48: Doba běhu skriptů – převod na vektory (AnalPipeline)

typ dokumentů	Yahoo články	FB komentáře	FB příspěvky	Twitter
doba běhu [h]	18:30	21:25	7:34	13:42

Tab. 49: Doba běhu skriptů – klasifikace (AnalPipeline)

typ dokumentů	Yahoo články	FB komentáře	FB příspěvky	Twitter
doba běhu [h]	1:39	1:50	0:26	1:01

[31] Údaj pro prvotní stahování – pro získání pouze nejnovějších článků je doba 0:45.
[32] Pro 20 firem – pro všechny firmy je doba asi 2 hodiny.
[33] Před spuštěním bylo nutné naplnit MySQL cache, což trvalo asi 10 hodin (pro 10 firem).

F Detailní výsledky analýzy 1

Tab. 50: Yahoo – hlavní metriky (detailní analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5479	0,8142	0,6346	0,6212	0,6138
Precision	0,5479	0,8227	0,6378	0,6244	0,6161
Recall	0,5479	0,8142	0,6346	0,6212	0,6138
F1 skóre	0,5333	0,8127	0,6324	0,6192	0,6116
Počet dokumentů	522	45 342	12 003	7 368	–
Počet atributů	2 811	32 827	14 508	12 915	–
Čas trénování [s]	0,0020	210,82	7,4881	0,3317	–

Tab. 51: FB-com – hlavní metriky (detailní analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5229	0,7352	0,5917	0,5832	0,5652
Precision	0,5228	0,7716	0,5942	0,5849	0,5663
Recall	0,5229	0,7352	0,5917	0,5832	0,5652
F1 skóre	0,5136	0,7352	0,5901	0,5816	0,5636
Počet dokumentů	1 508	94 870	49 183	41 302	–
Počet atributů	816	19 618	11 567	12 008	–
Čas trénování [s]	0,0014	74,77	8,4922	0,2881	–

Tab. 52: FB-post – hlavní metriky (detailní analýza 1)

metrika	min	max	průměr	medián	vážený průměr
Správnost	0,5020	0,6940	0,5814	0,5787	0,5721
Precision	0,5021	0,7527	0,5878	0,5808	0,5775
Recall	0,5020	0,6940	0,5814	0,5787	0,5721
F1 skóre	0,5020	0,6919	0,5773	0,5750	0,5676
Počet dokumentů	904	69 038	16 079	7 572	–
Počet atributů	789	23 650	7 582	5 354	–
Čas trénování [s]	0,0012	57,5003	1,8835	0,0538	–

Cenová proměnná															
all	count	percent		top50%	count	percent	top25%	count	percent	top10%			top5%		
adjclose	270	36,58537		adjclose	60	16,26016	adjclose	0	0	adjclose	0	0	adjclose	0	0
ewma	198	26,82927		ewma	191	51,76152	ewma	135	72,97297	ewma	69	93,24324	ewma	37	100
sma	270	36,58537		sma	118	31,97832	sma	50	27,02703	sma	5	6,756757	sma	0	0
	738	100			369	100		185	100		74	100		37	100
Počet dnů zpoždění															
all				top50%			top25%			top10%			top5%		
1	216	29,26829		1	111	30,0813	1	62	33,51351	1	25	33,78378	1	12	32,43243
2	252	34,14634		2	124	33,60434	2	68	36,75676	2	28	37,83784	2	16	43,24324
3	270	36,58537		3	134	36,31436	3	55	29,72973	3	21	28,37838	3	9	24,32432
	738	100			369	100		185	100		74	100		37	100
Hranice konstantního intervalu															
all				top50%			top25%			top10%			top5%		
1	162	21,95122		1	80	21,68022	1	25	13,51351	1	8	10,81081	1	2	5,405405
2	162	21,95122		2	83	22,49322	2	54	29,18919	2	26	35,13514	2	13	35,13514
3	144	19,5122		3	70	18,97019	3	37	20	3	11	14,86486	3	3	8,108108
4	144	19,5122		4	74	20,0542	4	41	22,16216	4	20	27,02703	4	12	32,43243
5	126	17,07317		5	62	16,80217	5	28	15,13514	5	9	12,16216	5	7	18,91892
	738	100			369	100		185	100		74	100		37	100
Typ vektoru															
all				top50%			top25%			top10%			top5%		
tp-no-no	246	33,33333		tp-no-no	120	32,52033	tp-no-no	63	34,05405	tp-no-no	22	29,72973	tp-no-no	11	29,72973
tf-idf-no	246	33,33333		tf-idf-no	130	35,23035	tf-idf-no	65	35,13514	tf-idf-no	30	40,54054	tf-idf-no	14	37,83784
tf-idf-cos	246	33,33333		tf-idf-cos	119	32,24932	tf-idf-cos	57	30,81081	tf-idf-cos	22	29,72973	tf-idf-cos	12	32,43243
	738	100			369	100		185	100		74	100		37	100
Algoritmus															
all				top50%			top25%			top10%			top5%		
NB-multi	123	16,66667		NB-multi	71	19,24119	NB-multi	45	24,32432	NB-multi	17	22,97297	NB-multi	8	21,62162
NB-berno	123	16,66667		NB-berno	39	10,56911	NB-berno	27	14,59459	NB-berno	5	6,756757	NB-berno	3	8,108108
MaxEnt	123	16,66667		MaxEnt	89	24,11924	MaxEnt	47	25,40541	MaxEnt	23	31,08108	MaxEnt	12	32,43243
CART	123	16,66667		CART	36	9,756098	CART	11	5,945946	CART	2	2,702703	CART	1	2,702703
RandFores	123	16,66667		RandFores	42	11,38211	RandFores	9	4,864865	RandFores	2	2,702703	RandFores	2	5,405405
LinearSVC	123	16,66667		LinearSVC	92	24,93225	LinearSVC	46	24,86486	LinearSVC	25	33,78378	LinearSVC	11	29,72973
	738	100			369	100		185	100		74	100		37	100

Obr. 50: Detailní analýza parametrů pro Yahoo články

Cenová proměnná																
all	count	percent		top50%	count	percent		top25%	count	percent		top10%			top5%		
adjclose	270	36,58537		adjclose	12	3,252033		adjclose	0	0		adjclose	0	0	adjclose	0	0
ewma	198	26,82927		ewma	187	50,67751		ewma	137	74,05405		ewma	69	93,24324	ewma	37	100
sma	270	36,58537		sma	170	46,07046		sma	48	25,94595		sma	5	6,756757	sma	0	0
	738	100			369	100			185	100			74	100		37	100
Počet dnů zpoždění																	
all				top50%				top25%				top10%			top5%		
1	216	29,26829		1	106	28,72629		1	47	25,40541		1	34	45,94595	1	20	54,05405
2	252	34,14634		2	135	36,58537		2	72	38,91892		2	32	43,24324	2	15	40,54054
3	270	36,58537		3	128	34,68835		3	66	35,67568		3	8	10,81081	3	2	5,405405
	738	100			369	100			185	100			74	100		37	100
Hranice konstantního intervalu																	
all				top50%				top25%				top10%			top5%		
1	162	21,95122		1	56	15,17615		1	19	10,27027		1	12	16,21622	1	3	8,108108
2	162	21,95122		2	83	22,49322		2	61	32,97297		2	36	48,64865	2	17	45,94595
3	144	19,5122		3	81	21,95122		3	31	16,75676		3	1	1,351351	3	0	0
4	144	19,5122		4	83	22,49322		4	47	25,40541		4	22	29,72973	4	15	40,54054
5	126	17,07317		5	66	17,88618		5	27	14,59459		5	3	4,054054	5	2	5,405405
	738	100			369	100			185	100			74	100		37	100
Typ vektoru																	
all				top50%				top25%				top10%			top5%		
tp-no-no	246	33,33333		tp-no-no	126	34,14634		tp-no-no	63	34,05405		tp-no-no	26	35,13514	tp-no-no	14	37,83784
tf-idf-no	246	33,33333		tf-idf-no	123	33,33333		tf-idf-no	62	33,51351		tf-idf-no	23	31,08108	tf-idf-no	12	32,43243
tf-idf-cos	246	33,33333		tf-idf-cos	120	32,52033		tf-idf-cos	60	32,43243		tf-idf-cos	25	33,78378	tf-idf-cos	11	29,72973
	738	100			369	100			185	100			74	100		37	100
Algoritmus																	
all				top50%				top25%				top10%			top5%		
NB-multi	123	16,66667		NB-multi	72	19,5122		NB-multi	39	21,08108		NB-multi	17	22,97297	NB-multi	9	24,32432
NB-berno	123	16,66667		NB-berno	69	18,69919		NB-berno	33	17,83784		NB-berno	12	16,21622	NB-berno	6	16,21622
MaxEnt	123	16,66667		MaxEnt	67	18,15718		MaxEnt	43	23,24324		MaxEnt	18	24,32432	MaxEnt	7	18,91892
CART	123	16,66667		CART	46	12,46612		CART	9	4,864865		CART	6	8,108108	CART	3	8,108108
RandFores	123	16,66667		RandFores	50	13,55014		RandFores	22	11,89189		RandFores	6	8,108108	RandFores	5	13,51351
LinearSVC	123	16,66667		LinearSVC	65	17,61518		LinearSVC	39	21,08108		LinearSVC	15	20,27027	LinearSVC	7	18,91892
	738	100			369	100			185	100			74	100		37	100

Obr. 51: Detailní analýza parametrů pro Facebook komentáře

Cenová proměnná																
all	count	percent		top50%	count	percent		top25%	count	percent		top10%				top5%		
adjclose	270	34,88372		adjclose	195	50,3876		adjclose	93	47,93814		adjclose	9	11,68831		adjclose	0	0
ewma	234	30,23256		ewma	150	38,75969		ewma	91	46,90722		ewma	68	88,31169		ewma	39	100
sma	270	34,88372		sma	42	10,85271		sma	10	5,154639		sma	0	0		sma	0	0
	774	100			387	100			194	100			77	100			39	100
Počet dnů zpoždění																		
all				top50%				top25%				top10%				top5%		
1	234	30,23256		1	148	38,24289		1	76	39,17526		1	44	57,14286		1	22	56,41026
2	270	34,88372		2	124	32,04134		2	65	33,50515		2	19	24,67532		2	11	28,20513
3	270	34,88372		3	115	29,71576		3	53	27,31959		3	14	18,18182		3	6	15,38462
	774	100			387	100			194	100			77	100			39	100
Hranice konstantního intervalu																		
all				top50%				top25%				top10%				top5%		
1	162	20,93023		1	68	17,57106		1	28	14,43299		1	6	7,792208		1	3	7,692308
2	162	20,93023		2	79	20,41344		2	39	20,10309		2	13	16,88312		2	6	15,38462
3	162	20,93023		3	70	18,08786		3	31	15,97938		3	18	23,37662		3	13	33,33333
4	144	18,60465		4	73	18,86305		4	26	13,40206		4	1	1,298701		4	0	0
5	144	18,60465		5	97	25,0646		5	70	36,08247		5	39	50,64935		5	17	43,58974
	774	100			387	100			194	100			77	100			39	100
Typ vektoru																		
all				top50%				top25%				top10%				top5%		
tp-no-no	258	33,33333		tp-no-no	129	33,33333		tp-no-no	61	31,4433		tp-no-no	25	32,46753		tp-no-no	14	35,89744
tf-idf-no	258	33,33333		tf-idf-no	130	33,59173		tf-idf-no	66	34,02062		tf-idf-no	28	36,36364		tf-idf-no	13	33,33333
tf-idf-cos	258	33,33333		tf-idf-cos	128	33,07494		tf-idf-cos	67	34,53608		tf-idf-cos	24	31,16883		tf-idf-cos	12	30,76923
	774	100			387	100			194	100			77	100			39	100
Algoritmus																		
all				top50%				top25%				top10%				top5%		
NB-multi	129	16,66667		NB-multi	88	22,73902		NB-multi	53	27,31959		NB-multi	20	25,97403		NB-multi	11	28,20513
NB-berno	129	16,66667		NB-berno	87	22,48062		NB-berno	54	27,83505		NB-berno	15	19,48052		NB-berno	15	38,46154
MaxEnt	129	16,66667		MaxEnt	79	20,41344		MaxEnt	33	17,01031		MaxEnt	12	15,58442		MaxEnt	3	7,692308
CART	129	16,66667		CART	20	5,167959		CART	13	6,701031		CART	7	9,090909		CART	3	7,692308
RandFores	129	16,66667		RandFores	46	11,8863		RandFores	17	8,762887		RandFores	10	12,98701		RandFores	3	7,692308
LinearSVC	129	16,66667		LinearSVC	67	17,31266		LinearSVC	24	12,37113		LinearSVC	13	16,88312		LinearSVC	4	10,25641
	774	100			387	100			194	100			77	100			39	100

Obr. 52: Detailní analýza parametrů pro Facebook příspěvky

Cenová proměnná															
all	count	percent		top50%	count	percent		top25%	count	percent		top10%			top5%		
adjclose	270	36,58537		adjclose	64	17,34417		adjclose	1	0,540541		adjclose	0	0	adjclose	0	0
ewma	198	26,82927		ewma	149	40,3794		ewma	102	55,13514		ewma	53	71,62162	ewma	31	83,78378
sma	270	36,58537		sma	156	42,27642		sma	82	44,32432		sma	21	28,37838	sma	6	16,21622
	738	100			369	100			185	100			74	100		37	100
Počet dnů zpoždění																	
all				top50%				top25%				top10%			top5%		
1	216	29,26829		1	83	22,49322		1	34	18,37838		1	6	8,108108	1	0	0
2	252	34,14634		2	144	39,02439		2	87	47,02703		2	49	66,21622	2	19	51,35135
3	270	36,58537		3	142	38,48238		3	64	34,59459		3	19	25,67568	3	18	48,64865
	738	100			369	100			185	100			74	100		37	100
Hranice konstantního intervalu																	
all				top50%				top25%				top10%			top5%		
1	162	21,95122		1	71	19,24119		1	59	31,89189		1	25	33,78378	1	13	35,13514
2	162	21,95122		2	101	27,37127		2	55	29,72973		2	28	37,83784	2	18	48,64865
3	144	19,5122		3	84	22,76423		3	28	15,13514		3	17	22,97297	3	6	16,21622
4	144	19,5122		4	63	17,07317		4	41	22,16216		4	4	5,405405	4	0	0
5	126	17,07317		5	50	13,55014		5	2	1,081081		5	0	0	5	0	0
	738	100			369	100			185	100			74	100		37	100
Typ vektoru																	
all				top50%				top25%				top10%			top5%		
tp-no-no	246	33,33333		tp-no-no	130	35,23035		tp-no-no	64	34,59459		tp-no-no	27	36,48649	tp-no-no	13	35,13514
tf-idf-no	246	33,33333		tf-idf-no	132	35,77236		tf-idf-no	65	35,13514		tf-idf-no	26	35,13514	tf-idf-no	13	35,13514
tf-idf-cos	246	33,33333		tf-idf-cos	107	28,99729		tf-idf-cos	56	30,27027		tf-idf-cos	21	28,37838	tf-idf-cos	11	29,72973
	738	100			369	100			185	100			74	100		37	100
Algoritmus																	
all				top50%				top25%				top10%			top5%		
NB-multi	123	16,66667		NB-multi	60	16,26016		NB-multi	24	12,97297		NB-multi	11	14,86486	NB-multi	5	13,51351
NB-berno	123	16,66667		NB-berno	63	17,07317		NB-berno	24	12,97297		NB-berno	12	16,21622	NB-berno	3	8,108108
MaxEnt	123	16,66667		MaxEnt	62	16,80217		MaxEnt	34	18,37838		MaxEnt	14	18,91892	MaxEnt	7	18,91892
CART	123	16,66667		CART	44	11,92412		CART	21	11,35135		CART	8	10,81081	CART	6	16,21622
RandFores	123	16,66667		RandFores	71	19,24119		RandFores	41	22,16216		RandFores	14	18,91892	RandFores	9	24,32432
LinearSVC	123	16,66667		LinearSVC	69	18,69919		LinearSVC	41	22,16216		LinearSVC	15	20,27027	LinearSVC	7	18,91892
	738	100			369	100			185	100			74	100		37	100

Obr. 53: Detailní analýza parametrů pro Twitter statusy

G Analýza 1 – zdrojové soubory

Tab. 53: Analýza 1 – nejlepší výsledky pro všechny soubory

pořadí	typ souboru	article	fb-post	fb-com	tweet	průměr
1	ewma_1_5	1,0000	0,7895	0,9091		0,8995
2	ewma_1_4	0,9643	0,8488	0,8421		0,8851
3	ewma_1_3	0,8261	0,6940	0,8070		0,7757
4	ewma_2_5	0,8132	0,6913	0,7861		0,7635
5	ewma_1_2	0,8122	0,6753	0,7352	0,6863	0,7273
6	ewma_2_4	0,8142	0,6390	0,7311	0,6957	0,7200
7	ewma_3_2	0,7428	0,6156	0,6562	0,7995	0,7035
8	ewma_3_5	0,7774	0,6782	0,6962	0,6529	0,7012
9	ewma_1_1	0,7468	0,6553	0,6836	0,7148	0,7001
10	ewma_2_2	0,7497	0,5879	0,6722	0,7257	0,6839
11	ewma_2_1	0,7140	0,6026	0,6350	0,7640	0,6789
12	sma_1_2	0,7083	0,6056	0,6639	0,6853	0,6658
13	ewma_2_3	0,7454	0,5843	0,6563	0,6668	0,6632
14	ewma_3_3	0,7428	0,5795	0,6503	0,6781	0,6627
15	ewma_3_4	0,7158	0,6067	0,6662	0,6484	0,6593
16	sma_2_3	0,6800	0,5622	0,6386	0,7479	0,6572
17	sma_2_4	0,6827	0,5672	0,6557	0,7190	0,6562
18	ewma_3_1	0,6857	0,5840	0,5907	0,7098	0,6425
19	sma_1_3	0,7026	0,6056	0,6280	0,6229	0,6398
20	sma_3_4	0,6545	0,5718	0,6279	0,7040	0,6396
21	sma_3_5	0,6747	0,5613	0,6425	0,6783	0,6392
22	sma_1_4	0,7238	0,5994	0,6085	0,6208	0,6381
23	sma_2_5	0,6980	0,5470	0,6229	0,6664	0,6336
24	adjclose_1_5	0,6302	0,6453	0,5914	0,6657	0,6331
25	adjclose_3_5	0,6409	0,6196	0,5841	0,6774	0,6305
26	adjclose_2_5	0,6424	0,6332	0,5750	0,6680	0,6296
27	sma_1_1	0,6530	0,5812	0,5919	0,6876	0,6284
28	adjclose_3_4	0,6485	0,6119	0,5775	0,6630	0,6252
29	adjclose_2_4	0,6393	0,6233	0,5718	0,6578	0,6231
30	sma_2_2	0,6474	0,5589	0,5924	0,6890	0,6219
31	sma_1_5	0,6761	0,6116	0,5877	0,6122	0,6219
32	sma_3_3	0,6445	0,5554	0,5947	0,6897	0,6211
33	adjclose_3_3	0,6425	0,6120	0,5599	0,6631	0,6194
34	adjclose_2_3	0,6334	0,6113	0,5640	0,6614	0,6175
35	adjclose_1_4	0,6293	0,6058	0,5801	0,6529	0,6170
36	adjclose_1_3	0,6209	0,6014	0,5751	0,6630	0,6151
37	adjclose_3_2	0,6423	0,6168	0,5444	0,6419	0,6113
38	adjclose_2_2	0,6275	0,6158	0,5421	0,6405	0,6065
39	sma_3_2	0,6239	0,5505	0,5664	0,6774	0,6045
40	adjclose_1_2	0,6131	0,5886	0,5562	0,6423	0,6000
41	adjclose_3_1	0,6389	0,6089	0,5425	0,6063	0,5991
42	adjclose_2_1	0,6307	0,6236	0,5415	0,5898	0,5964
43	sma_3_1	0,6214	0,5464	0,5486	0,6636	0,5950
44	sma_2_1	0,6183	0,5468	0,5516	0,6608	0,5944
45	adjclose_1_1	0,6513	0,5852	0,5414	0,5916	0,5924

Tab. 54: Yahoo články – zdrojové soubory (analýza 1)

Pořadí	Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
1	ewma_1_5	1,0000	tf-idf-cos	LinearSVC	46	226
2	ewma_1_4	0,9643	tf-idf-no	LinearSVC	80	566
3	ewma_1_3	0,8261	tf-idf-no	LinearSVC	196	1 559
4	ewma_2_4	0,8142	tf-idf-no	NB-multi	522	2 811
5	ewma_2_5	0,8132	tf-idf-cos	LinearSVC	258	1 948
6	ewma_1_2	0,8122	tp-no-no	NB-multi	654	3 063
7	ewma_3_5	0,7774	tp-no-no	MaxEnt	808	3 605
8	ewma_2_2	0,7497	tf-idf-no	MaxEnt	2 772	7 868
9	ewma_1_1	0,7468	tf-idf-cos	LinearSVC	2 706	8 288
10	ewma_2_3	0,7454	tf-idf-cos	LinearSVC	1 234	4 539
11	ewma_3_2	0,7428	tf-idf-cos	LinearSVC	6 476	12 915
12	ewma_3_3	0,7428	tf-idf-no	MaxEnt	2 686	7 450
13	sma_1_4	0,7238	tf-idf-cos	NB-multi	900	4 111
14	ewma_3_4	0,7158	tf-idf-no	MaxEnt	1 568	5 216
15	ewma_2_1	0,7140	tf-idf-no	MaxEnt	12 728	17 925
16	sma_1_2	0,7083	tf-idf-cos	LinearSVC	3 496	9 037
17	sma_1_3	0,7026	tp-no-no	MaxEnt	1 564	5 561
18	sma_2_5	0,6980	tf-idf-cos	LinearSVC	2 126	6 513
19	ewma_3_1	0,6857	tf-idf-no	MaxEnt	25 014	24 053
20	sma_2_4	0,6827	tf-idf-cos	LinearSVC	3 394	8 659
21	sma_2_3	0,6800	tf-idf-cos	LinearSVC	6 212	11 967
22	sma_1_5	0,6761	tp-no-no	MaxEnt	606	3 198
23	sma_3_5	0,6747	tf-idf-cos	LinearSVC	4 952	10 504
24	sma_3_4	0,6545	tf-idf-cos	LinearSVC	7 822	13 399
25	sma_1_1	0,6530	tf-idf-cos	LinearSVC	14 154	19 007
26	adjclose_1_1	0,6513	tp-no-no	MaxEnt	35 630	29 168
27	adjclose_3_4	0,6485	tf-idf-no	MaxEnt	11 166	16 174
28	sma_2_2	0,6474	tf-idf-no	MaxEnt	13 776	18 433
29	sma_3_3	0,6445	tf-idf-no	MaxEnt	13 444	17 933
30	adjclose_3_3	0,6425	tf-idf-cos	LinearSVC	17 372	20 581
31	adjclose_2_5	0,6424	tf-idf-no	NB-multi	4 944	10 551
32	adjclose_3_2	0,6423	tf-idf-no	MaxEnt	27 090	26 330
33	adjclose_3_5	0,6409	tf-idf-no	MaxEnt	7 368	12 812
34	adjclose_2_4	0,6393	tf-idf-no	MaxEnt	7 968	13 389
35	adjclose_3_1	0,6389	tf-idf-no	MaxEnt	41 384	32 827
36	adjclose_2_3	0,6334	tf-idf-cos	LinearSVC	13 066	17 523
37	adjclose_2_1	0,6307	tf-idf-cos	LinearSVC	36 922	30 886
38	adjclose_1_5	0,6302	tp-no-no	NB-multi	2 456	6 969
39	adjclose_1_4	0,6293	tf-idf-cos	LinearSVC	4 352	9 427
40	adjclose_2_2	0,6275	tf-idf-no	MaxEnt	21 950	23 380
41	sma_3_2	0,6239	tf-idf-no	MaxEnt	25 296	24 333
42	sma_3_1	0,6214	tf-idf-no	MaxEnt	45 342	32 749
43	adjclose_1_3	0,6209	tf-idf-no	MaxEnt	8 364	13 357
44	sma_2_1	0,6183	tp-no-no	MaxEnt	35 328	28 888
45	adjclose_1_2	0,6131	tf-idf-no	MaxEnt	16 510	19 424

Tab. 55: Facebook komentáře – zdrojové soubory (analýza 1)

Pořadí	Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
1	ewma_1_5	0,9091	tf-idf-cos	NB-multi	30	16
2	ewma_1_4	0,8421	tf-idf-cos	MaxEnt	54	101
3	ewma_1_3	0,8070	tf-idf-cos	LinearSVC	162	437
4	ewma_2_5	0,7861	tf-idf-cos	LinearSVC	572	517
5	ewma_1_2	0,7352	tf-idf-cos	NB-multi	1 876	1 003
6	ewma_2_4	0,7311	tf-idf-cos	NB-multi	1 508	816
7	ewma_3_5	0,6962	tf-idf-no	NB-multi	1 740	1 027
8	ewma_1_1	0,6836	tp-no-no	MaxEnt	13 480	5 404
9	ewma_2_2	0,6722	tp-no-no	NB-multi	11 806	4 841
10	ewma_3_4	0,6662	tf-idf-cos	NB-multi	4 228	1 847
11	sma_1_2	0,6639	tf-idf-cos	LinearSVC	18 124	7 169
12	ewma_2_3	0,6563	tp-no-no	NB-multi	3 472	1 650
13	ewma_3_2	0,6562	tf-idf-cos	LinearSVC	35 804	11 810
14	sma_2_4	0,6557	tp-no-no	MaxEnt	15 758	6 176
15	ewma_3_3	0,6503	tf-idf-cos	NB-multi	9 672	3 998
16	sma_3_5	0,6425	tp-no-no	MaxEnt	23 144	7 901
17	sma_2_3	0,6386	tf-idf-cos	LinearSVC	32 850	10 767
18	ewma_2_1	0,6350	tf-idf-cos	MaxEnt	71 038	18 351
19	sma_1_3	0,6280	tf-idf-no	NB-multi	7 880	3 637
20	sma_3_4	0,6279	tp-no-no	MaxEnt	40 852	12 008
21	sma_2_5	0,6229	tf-idf-cos	NB-multi	9 114	3 632
22	sma_1_4	0,6085	tp-no-no	MaxEnt	5 108	2 323
23	sma_3_3	0,5947	tf-idf-cos	LinearSVC	80 326	19 069
24	sma_2_2	0,5924	tp-no-no	MaxEnt	81 424	18 942
25	sma_1_1	0,5919	tf-idf-no	NB-berno	82 002	18 978
26	adjclose_1_5	0,5914	tf-idf-cos	NB-multi	14 338	5 145
27	ewma_3_1	0,5907	tf-idf-cos	NB-multi	80 812	18 067
28	sma_1_5	0,5877	tf-idf-no	NB-multi	2 654	1 243
29	adjclose_3_5	0,5841	tp-no-no	MaxEnt	62 480	14 756
30	adjclose_1_4	0,5801	tf-idf-cos	NB-multi	27 690	8 054
31	adjclose_3_4	0,5775	tf-idf-no	MaxEnt	90 120	19 342
32	adjclose_1_3	0,5751	tf-idf-cos	NB-multi	60 612	14 591
33	adjclose_2_5	0,5750	tf-idf-cos	NB-multi	41 302	11 191
34	adjclose_2_4	0,5718	tf-idf-cos	NB-multi	69 986	16 214
35	sma_3_2	0,5664	tp-no-no	MaxEnt	87 724	18 597
36	adjclose_2_3	0,5640	tp-no-no	NB-multi	92 212	19 618
37	adjclose_3_3	0,5599	tp-no-no	NB-multi	91 140	18 697
38	adjclose_1_2	0,5562	tf-idf-no	NB-multi	94 014	19 203
39	sma_2_1	0,5516	tp-no-no	NB-multi	92 142	18 086
40	sma_3_1	0,5486	tf-idf-cos	NB-berno	94 640	17 946
41	adjclose_3_2	0,5444	tp-no-no	NB-multi	94 720	18 653
42	adjclose_3_1	0,5425	tf-idf-cos	NB-multi	92 750	18 086
43	adjclose_2_2	0,5421	tf-idf-cos	LinearSVC	93 746	19 070
44	adjclose_2_1	0,5415	tf-idf-cos	NB-multi	91 306	18 287
45	adjclose_1_1	0,5414	tf-idf-cos	NB-multi	91 572	18 500

Tab. 56: Facebook příspěvky – zdrojové soubory (analýza 1)

Pořadí	Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
1	ewma_1_4	0,8488	tf-idf-cos	NB-berno	244	289
2	ewma_1_5	0,7895	tf-idf-no	MaxEnt	162	224
3	ewma_1_3	0,6940	tf-idf-no	MaxEnt	904	789
4	ewma_2_5	0,6913	tf-idf-cos	NB-multi	1 044	875
5	ewma_3_5	0,6782	tf-idf-cos	NB-berno	1 242	1 052
6	ewma_1_2	0,6753	tf-idf-no	NB-multi	1 654	1 188
7	ewma_1_1	0,6553	tf-idf-cos	NB-berno	3 430	2 840
8	adjclose_1_5	0,6453	tp-no-no	NB-multi	2 448	2 541
9	ewma_2_4	0,6390	tf-idf-cos	LinearSVC	1 272	1 048
10	adjclose_2_5	0,6332	tp-no-no	NB-multi	6 432	5 363
11	adjclose_2_1	0,6236	tf-idf-no	NB-multi	61 324	22 934
12	adjclose_2_4	0,6233	tf-idf-no	NB-multi	10 708	7 779
13	adjclose_3_5	0,6196	tf-idf-cos	NB-berno	9 230	6 897
14	adjclose_3_2	0,6168	tf-idf-cos	NB-berno	40 904	17 819
15	adjclose_2_2	0,6158	tf-idf-no	NB-multi	34 668	16 397
16	ewma_3_2	0,6156	tf-idf-cos	NB-multi	7 572	4 762
17	adjclose_3_3	0,6120	tp-no-no	NB-multi	24 418	13 049
18	adjclose_3_4	0,6119	tp-no-no	NB-multi	14 708	9 355
19	sma_1_5	0,6116	tf-idf-no	LinearSVC	1 728	1 369
20	adjclose_2_3	0,6113	tp-no-no	NB-multi	19 116	11 329
21	adjclose_3_1	0,6089	tp-no-no	NB-multi	69 038	23 650
22	ewma_3_4	0,6067	tf-idf-cos	NB-multi	2 034	1 524
23	adjclose_1_4	0,6058	tf-idf-cos	NB-berno	4 456	4 141
24	sma_1_3	0,6056	tf-idf-cos	NB-berno	2 352	1 943
25	sma_1_2	0,6056	tf-idf-cos	NB-multi	4 098	3 235
26	ewma_2_1	0,6026	tf-idf-no	NB-multi	15 730	8 379
27	adjclose_1_3	0,6014	tf-idf-no	NB-multi	9 638	6 836
28	sma_1_4	0,5994	tf-idf-cos	NB-multi	1 966	1 565
29	adjclose_1_2	0,5886	tf-idf-no	NB-multi	21 520	11 651
30	ewma_2_2	0,5879	tf-idf-no	RandForest	3 638	2 597
31	adjclose_1_1	0,5852	tf-idf-no	NB-multi	48 590	19 127
32	ewma_2_3	0,5843	tf-idf-no	MaxEnt	2 034	1 410
33	ewma_3_1	0,5840	tf-idf-cos	NB-multi	31 386	12 927
34	sma_1_1	0,5812	tf-idf-cos	NB-multi	17 730	9 679
35	ewma_3_3	0,5795	tf-idf-no	MaxEnt	3 430	2 301
36	sma_3_4	0,5718	tp-no-no	NB-multi	8 640	5 354
37	sma_2_4	0,5672	tp-no-no	NB-multi	3 886	2 794
38	sma_2_3	0,5622	tf-idf-cos	MaxEnt	6 916	4 660
39	sma_3_5	0,5613	tf-idf-cos	NB-multi	5 588	3 746
40	sma_2_2	0,5589	tf-idf-no	NB-multi	16 900	8 845
41	sma_3_3	0,5554	tf-idf-cos	NB-multi	15 628	8 284
42	sma_3_2	0,5505	tf-idf-cos	NB-multi	32 694	13 289
43	sma_2_5	0,5470	tf-idf-no	NB-multi	2 792	2 071
44	sma_2_1	0,5468	tf-idf-cos	NB-multi	49 826	17 396
45	sma_3_1	0,5464	tf-idf-cos	NB-berno	68 088	21 232

Tab. 57: Twitter statusy – zdrojové soubory (analýza 1)

Pořadí	Typ souboru	Správnost	Typ vektoru	Algoritmus	PD	PA
1	ewma_3_2	0,7995	tf-idf-no	RandForest	22 398	8 185
2	ewma_2_1	0,7640	tp-no-no	RandForest	45 596	12 589
3	sma_2_3	0,7479	tp-no-no	RandForest	20 798	7 348
4	ewma_2_2	0,7257	tp-no-no	LinearSVC	10 800	5 186
5	sma_2_4	0,7190	tp-no-no	RandForest	14 400	5 347
6	ewma_1_1	0,7148	tp-no-no	RandForest	11 600	5 897
7	ewma_3_1	0,7098	tp-no-no	RandForest	91 072	17 021
8	sma_3_4	0,7040	tp-no-no	RandForest	28 398	8 775
9	ewma_2_4	0,6957	tf-idf-cos	LinearSVC	1 998	1 566
10	sma_3_3	0,6897	tp-no-no	RandForest	50 798	11 898
11	sma_2_2	0,6890	tp-no-no	MaxEnt	51 598	11 957
12	sma_1_1	0,6876	tp-no-no	MaxEnt	53 598	12 501
13	ewma_1_2	0,6863	tf-idf-no	LinearSVC	3 196	2 580
14	sma_1_2	0,6853	tp-no-no	CART	14 800	5 475
15	sma_3_5	0,6783	tp-no-no	RandForest	20 800	6 820
16	ewma_3_3	0,6781	tp-no-no	LinearSVC	11 598	4 633
17	sma_3_2	0,6774	tf-idf-no	RandForest	93 196	17 114
18	adjclose_3_5	0,6774	tp-no-no	RandForest	34 000	8 904
19	adjclose_2_5	0,6680	tp-no-no	RandForest	24 000	7 173
20	ewma_2_3	0,6668	tp-no-no	LinearSVC	6 800	3 633
21	sma_2_5	0,6664	tf-idf-no	LinearSVC	10 800	3 990
22	adjclose_1_5	0,6657	tf-idf-cos	NB-berno	9 188	3 482
23	sma_3_1	0,6636	tf-idf-no	RandForest	89 766	16 984
24	adjclose_3_3	0,6631	tf-idf-no	RandForest	85 274	15 591
25	adjclose_3_4	0,6630	tp-no-no	RandForest	53 298	11 402
26	adjclose_1_3	0,6630	tf-idf-no	RandForest	28 228	7 771
27	adjclose_2_3	0,6614	tf-idf-no	RandForest	61 890	12 551
28	sma_2_1	0,6608	tf-idf-no	RandForest	98 492	16 790
29	adjclose_2_4	0,6578	tp-no-no	RandForest	36 500	9 092
30	adjclose_1_4	0,6529	tp-no-no	MaxEnt	15 188	4 855
31	ewma_3_5	0,6529	tf-idf-no	LinearSVC	5 998	2 589
32	ewma_3_4	0,6484	tp-no-no	NB-multi	8 798	3 772
33	adjclose_1_2	0,6423	tp-no-no	RandForest	66 804	13 614
34	adjclose_3_2	0,6419	tp-no-no	RandForest	95 012	16 857
35	adjclose_2_2	0,6405	tp-no-no	RandForest	97 624	16 948
36	sma_1_3	0,6229	tf-idf-no	RandForest	9 600	3 615
37	sma_1_4	0,6208	tp-no-no	RandForest	9 600	3 615
38	sma_1_5	0,6122	tf-idf-no	MaxEnt	9 600	3 615
39	adjclose_3_1	0,6063	tf-idf-no	RandForest	93 660	16 749
40	adjclose_1_1	0,5916	tp-no-no	RandForest	98 586	16 984
41	adjclose_2_1	0,5898	tf-idf-cos	NB-berno	97 390	16 918

H Analýza 2 – zdrojové soubory

Tab. 58: Zdrojové soubory pro Feature selection (analýza 2)

Číslo	Typ dokumentu	Typ souboru	Správnost	PD	PA
1	article	ewma_1_5	1,0000	46	226
2	article	ewma_1_4	0,9643	80	566
3	article	ewma_1_3	0,8261	196	1 559
4	article	ewma_2_4	0,8142	522	2 811
5	article	ewma_2_5	0,8132	258	1 948
6	article	ewma_1_2	0,8122	654	3 063
7	article	ewma_3_5	0,7774	808	3 605
8	article	ewma_2_2	0,7497	2 772	7 868
9	article	ewma_1_1	0,7468	2 706	8 288
10	article	ewma_2_3	0,7454	1 234	4 539
11	article	ewma_3_2	0,7428	6 476	12 915
12	article	ewma_3_3	0,7428	2 686	7 450
13	article	sma_1_4	0,7238	900	4 111
14	article	ewma_3_4	0,7158	1 568	5 216
15	article	ewma_2_1	0,7140	12 728	17 925
16	article	sma_1_2	0,7083	3 496	9 037
17	article	sma_1_3	0,7026	1 564	5 561
18	fb-com	ewma_1_5	0,9091	30	16
19	fb-com	ewma_1_4	0,8421	54	101
20	fb-com	ewma_1_3	0,8070	162	437
21	fb-com	ewma_2_5	0,7861	572	517
22	fb-com	ewma_1_2	0,7352	1 876	1 003
23	fb-com	ewma_2_4	0,7311	1 508	816
24	fb-com	ewma_3_5	0,6962	1 740	1 027
25	fb-post	ewma_1_4	0,8488	244	289
26	fb-post	ewma_1_5	0,7895	162	224
27	fb-post	ewma_1_3	0,6940	904	789
28	fb-post	ewma_2_5	0,6913	1 044	875
29	tweet	ewma_3_2	0,7995	22 398	8 185
30	tweet	ewma_2_1	0,7640	45 596	12 589
31	tweet	sma_2_3	0,7479	20 798	7 348
32	tweet	ewma_2_2	0,7257	10 800	5 186
33	tweet	sma_2_4	0,7190	14 400	5 347
34	tweet	ewma_1_1	0,7148	11 600	5 897
35	tweet	ewma_3_1	0,7098	91 072	17 021
36	tweet	sma_3_4	0,7040	28 398	8 775