

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Andrej Bergant

**Merjenje sentimenta do slovenskih podjetij in povezanosti z gibanjem cen delnic in
borznega indeksa**

Magistrsko delo

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Andrej Bergant

Mentor: izr. prof. dr. Aleš Žiberna

**Merjenje sentimenta do slovenskih podjetij in povezanosti z gibanjem cen delnic in
borznega indeksa**

Magistrsko delo

Ljubljana, 2016

ZAHVALA

Iskreno se zahvaljujem mentorju dr. Alešu Žiberni za potrpežljivost, smernice, dovoljeno ustvarjalno svobodo in za vso njegovo strokovno pomoč pri nastajanju tega dela.

Zahvaljujem se svoji puncu Tini, ki me je podpirala med nastajanjem tega dela.

Zahvaljujem se tudi svoji družini, očetu Andreju in mami Zdenki, ki sta me finančno podpirala skozi čas. Posebna zahvala pa velja tudi teti Olgi za vse spodbudne besede med študijem.

Merjenje sentimenta do slovenskih podjetij in povezanost z gibanjem cen delnic in borznega indeksa

Analiza sentimenta v zadnjih nekaj letih privablja ogromno število raziskovalcev. Masovne količine ustvarjenih podatkov na internetu omogočajo zbiranje, analizo in izgradnjo modelov s pomočjo podatkovnega rudarjenja. V ospredju našega magistrskega dela je razvoj programa za analizo sentimenta slovenskega Twitter omrežja v povezavi s slovenskimi delnicami (indeksom SBITOP). Teoretično pregledamo razvoj masovnih podatkov, izzive in orodja, s katerimi se podatkovni znanstvenik srečuje pri delu z velikimi količinami podatkov, osredotočenim na programski jezik R. Opredelimo znanstvena polja podatkovnega rudarjenja in postavimo teoretični okvir analiz in metod sentimenta. V empiričnem delu predstavimo implementacijo naše informacijske rešitve in metod analize sentimenta. Program je v celoti napisan v jeziku R, s pomočjo Twitterjevega API-ja in napisanih algoritmov za avtomatsko luščenje podatkov s spletne strani SBITOP pa v skoraj realnem času (v razponu 15 min) računa sentiment in filtrira sporočila uporabnikov, ki se navezujejo na posamezno delnico. Končna rešitev in rezultati so prikazani v obliki spletne aplikacije s pomočjo R-jevih knjižnic. Naša sklepna ugotovitev analiz je, da na podlagi rezultatov analiz ne moremo sklepati o obstoju povezanosti med slovenskim sentimentom in ceno slovenskih delnic oz. vrednostjo indeksa. *Kljub temu* pa tudi ne moremo z gotovostjo trditi, da sentiment ne bi bil uporaben za napovedovanje cene delnic, med drugim tudi zato, ker so se v nekaterih analizah korelacije vseeno pokazale.

Ključne besede: analiza sentimenta, masovni podatki, podatkovno rudarjenje, programski jezik R, delnice, SBITOP, Twitter API.

Measuring the sentiment of Slovenian companies and connection with the movement of stock prices and stock market index

The sentiment analysis has attracted huge number of researchers in the last few years. Massive amounts of created data on the internet allows us to gather, analyse and build models using data mining techniques. At the forefront of our master thesis was development of a program for analysis of Slovenian Twitter sentiment and then use this computational infrastructure to benchmark it with Slovenian stocks (or SBITOP index). We theoretically review the development, challenges and tools of big data, which data scientists use when constructing models and building solutions, specifically focused on R language. We introduce our computational model and implementation of our information solution in the empirical work. The program uses Twitter API and our programmed algorithms to automatically web scrap data from SBITOP website in nearly real time (on the 15 min intervals), calculates sentiment and filters Twitter messages, which express sentiment towards the stock. The complete solution is then presented as a web application written with the help of R libraries. On the basis of our analysis of 16 million tweets, we cannot confirm the existence of the correlation between Slovenian sentiment and stock values (or index). *Nevertheless*, we also cannot claim with certainty, that sentiment would not be useful in constructing forecasting models for the stock prices (or index). That is because some analysis showed promising results and some correlations were found.

Keywords: sentiment analysis, big data, data mining, programing language R, stocks, SBITOP, Twitter API.

Glavno kazalo:

1 Uvod	9
2 Teoretični del.....	12
2.1 Masovni podatki in podatkovna znanost	12
2.1.1 »Velikanski kup lego kock«	12
2.1.2 Definicija masovnih podatkov.....	12
2.1.3 Problem definicije masovnih podatkov	13
2.1.4 Definicija 3V	13
2.1.5 Definicija 4V	13
2.1.6 »Moorov zakon« in hitrost ustvarjanja podatkov	14
2.2 Pričakovanja, trendi in izzivi masovnih podatkov.....	15
2.2.1 Gartnerjeva krivulja pričakovanj	16
2.2.2 Google trendi	17
2.2.3 Vloga podatkovnega znanstvenika	18
2.2.4 Vloga primerne infrastrukture	19
2.2.5 Računske gruč in algoritem »MapReduce«	20
2.2.6 Porazdeljeni datotečni sistem, podatkovne baze	21
2.2.7 Programski sistem MapReduce in R	22
2.3 Podatkovno rudarjenje in modeli.....	23
2.3.1 Podatkovno rudarjenje.....	23
2.3.2 Statistično modeliranje	24
2.3.3 Strojno ali statistično učenje.....	25
2.3.4 Računski pristopi k modeliranju.....	25
2.3.5 Pametni model kot produkt interdisciplinarnosti in konvergence znanstvenih polj	26
2.4 Analiza sentimenta	26
2.4.1 Pomen in uporabnost sentimenta.....	27
2.4.2 Definicije sentimenta.....	27
2.4.3 Proces analize sentimenta in različni pristopi.....	28
2.4.4 Faza obdelave podatkov	29
2.4.5 Faza klasifikacije	30
2.4.6 Pristop s pomočjo slovarja.....	30
2.4.7 Problemi z analizo sentimenta in omejitve	32
2.4.8 Pristop s strojnimi učenjem	32
2.5 Socialno omrežje Twitter in sentiment.....	33
2.5.6 Napovedovanje gibanja cene delnic	33
3 Empirični del	36

3.1 Merjenje sentimenta in zajemanje podatkov	36
3.1.1 O izdelavi in implementaciji programa	37
3.1.2 Twitterjev programski vmesnik (API) in R	39
3.1.3 Shema diagrama poteka aplikacije in razlaga MapReduce algoritma	40
3.1.4 Luščenje vsebin in pridobivanje podatkov preko API-jev	40
3.1.5 Geolocirani tviti	41
3.1.6 Twitterjeve »liste« oziroma po meri ustvarjeni sezname uporabnikov	42
3.1.7 Sezname uporabnikov za časovnico	42
3.1.8 MapReduce algoritem in porazdeljeni datotečni sistem	43
3.2 Analiza	51
3.2.1 Opisni podatki	51
3.2.2 Tabela slovenskih uporabnikov	52
3.2.3 Tabele zajetih tvitov na posamezno delnico	53
3.3 Merjenje povezanosti	53
3.3.1 Delnica Telekom Slovenija in sentiment	56
3.3.2 Metoda s slovarjem AFINN	59
3.3.3 Metoda s slovarjem AFINN rangi	61
3.3.4 Slovenski borzni indeks in sentiment v Sloveniji	64
3.3.5 Metoda s slovarjem AFINN	65
3.3.6 Metoda s slovarjem AFINN rangi	70
3.4 Zaključna interpretacija merjenja povezanosti	73
4 Zaključek	76
5 Literatura	78
Priloga	81
A - Grafi	81
B - Seznam Twitter list	84
C - Koda	84

Kazalo enačb:

Enačba 3.1: Drseči skalarni produkt	54
Enačba 3.2: Sprememba tečaja	55
Enačba 3.3: Sprememba sprememb tečaja	55
Enačba 3.4: Sprememba sprememb indeksa	55
Enačba 3.5: Sprememba sentimenta	56

Kazalo slik:

Slika 2.1: Masovni podatki kot velik kup lego kock	12
--	----

Slika 2.2: Definicija 4V masovnih podatkov	13
Slika 2.3: Gartnerjeva krivulja pričakovane adopcije tehnologij	16
Slika 2.4: Trendi iskanja ključnih besed.....	17
Slika 2.5: Skica platform, tehnologij, storitev in podjetij.....	20
Slika 2.6: Skica obdelave podatkov pred fazo klasifikacije	29
Slika 2.7: Poenostavljen proces ocenjevanja sentimenta z AFINN slovarji na primeru Twitterja.....	31
Slika 3.1: Geolocirani viri podatkov	41
Slika 3.2: Shema (poenostavljenega) diagrama poteka aplikacije.....	45
Slika 3.3: Graf frekvence sporočil za delnico Telekom Slovenija	56
Slika 3.4: Časovne korelacije sprememb tečaja in vsota sentimentov z AFINN slovarjem (od 1.1.2014 do 24.2.2016).....	60
Slika 3.5: Graf tečaja, sprememb in sentimenta (5.3.2015 in 13.5.2015)	61
Slika 3.6: Časovne korelacije med spremembo tečaja in vsoto sentimentov z AFINN slovarjem rangov (od 1.1.2014 do 24.2.2016)	61
Slika 3.7: Sprememba sprememb tečaja in sprememba vsote sentimentov AFINN rangi	62
Slika 3.8: Graf spremembe sprememb tečaja in sprememb sentimenta (5.3.2015 in 13.5.2015)	63
Slika 3.9: Graf frekvence sporočil v Sloveniji	64
Slika 3.10: Indeks v točkah, sprememba indeksa in vsota sentimentov AFINN (1.6.2014-23.2.2016)	65
Slika 3.11: Sprememba indeksa in vsote AFINN (1.6.2014-23.2.2016).....	67
Slika 3.12: Vrednost indeksa in vsota sentimentov AFINN (1.6.2014-23.2.2016).....	67
Slika 3.13: Sprememba sprememb indeksa in sprememba sentimenta AFINN (1.10.2015 - 1.12.2015)	68
Slika 3.14: Sprememba sprememb indeksa in sprememba vsote sentimentov AFINN (1.6.2014-23.2.2016).....	68
Slika 3.15: Sprememba sprememb indeksa in sprememba sentimentov AFINN (1.10.2015 - 1.12.2015)	69
Slika 3.16: Spremembe indeksa in vsote sentimentov AFINN rangi (1.6.2014-23.2.2016)	70
Slika 3.17: Vrednost indeksa in vsota sentimentov AFINN rangi (1.6.2014-23.2.2016).....	70
Slika 3.18: Sprememba indeksa in sprememba vsote sentimentov AFINN rangi (1.6.2014-23.2.2016)	71
Slika 3.19: Sprememba sprememb indeksa in sprememba vsote AFINN rangi (1.6.2014-23.2.2016).	71

Kazalo tabel:

Tabela 3.1: Opis vsebine elementov programa - funkcije	46
Tabela 3.2: Opis vsebine elementov programa – podatkovne strukture.....	48
Tabela 3.3: Top 100 uporabniških računov	52
Tabela 3.4: Tabela zajetih tvitov na posamezno delnico	53

1 UVOD

Pomemben del zbiranja, interpretiranja in učenja iz informacij je v družboslovnem raziskovanju vedno bilo poslušanje ali opazovanje človeka in ljudi. Takšno raziskovanje običajno izvajamo s kvalitativnimi metodami, kjer skušamo s pomočjo intervjujev ali fokusnih skupin izluščiti potrebne podatke in informacije, ali pa preko kvantitativnih metod, pri katerih s pomočjo anketne metodologije statistično preverimo hipoteze ali teoretične modele. Še vedno so takšne metode zbiranja podatkov za specifične primere pridobivanja znanj in informacij seveda uporabne in nujne, kljub temu pa je porast internetnih medijev in socialnih omrežij ter s tem generiranih vsebin povzročil, da je ta način postal preširok, časovno potraten in finančno predrag za preučevanje, če bi za takšne potrebe zbiranja podatkov ali analize uporabljali človeka ali več ljudi, ki bi »ročno« in sistematično zbirali podatke in jih nato analizirali s poljubnim statističnim programom. Takšne metode zbiranja, interpretiranja in analize niso primerne hitrosti kopičenja podatkov (Aggarwal in Zhai 2012), ki jih vsakodnevno generirajo ljudje (v nadaljevanju uporabniki) na internetu.

Po nekaterih podatkih (Bradicich in Orci 2013) se količina podatkov na internetu podvoji vsaki dve leti, medtem ko velike mednarodne korporacije napovedujejo 10-kraten porast podatkov do leta 2020 (Williams 2014). S porastom rabe komunikacijskih socialnih omrežij, njihovo dostopnostjo in privlačnostjo prihaja do velikanskega kopičenja podatkov in besedil, ki jih uporabniki puščajo za seboj. V večini primerov so dostopni in odprti. S tem se zbiranje podatkov in analiz vsebin takšnih dokumentov prenaša na avtomatizirane računalniške algoritme, sisteme in orodja, s katerimi je možen vpogled, njihova analiza in interpretacija s pomočjo vizualizacije in povratnih informacij statističnih modelov, ki nam jih takšne računalniško-informacijske rešitve lahko nudijo. Socialno omrežje Twitter je hitro in globalno komunikacijsko omrežje, ki v zadnjih letih pridobiva ogromno pozornosti raziskovalcev (Takahashi in drugi 2011) ter s tem spreminja družbene pogoje in posledično našo objektivno realnost, kot pravijo nekateri raziskovalci.

Ali lahko npr. uporabniki omrežja Twitter vplivajo na vrednosti delnic in indeksov, da le te zanihajo, s tem pa raziskovalci napovemo, ali se splača investirati denar? Prvič v zgodovini smo priča okolju in pogojem, zaradi katerih lahko s pomočjo orodij, interneta ter hitrosti pretoka informacij resno podvomimo o družbenih »dejstvih«, ki so bila še pred 20 leti realnost (*npr. vse javno dostopne informacije so integrirane v menjalnih*

tečajih – teza o učinkovitost odprtih trgov), danes pa so pod težo vedno hitrejšega dogajanja in komunikacij pod resnim vprašanjem. (Dietmar 2014, 1–4)

Izbrana magistrska tema o povezanosti gibanja cene slovenskih delnic in sentimenta, izmerjenega na socialnem omrežju Twitter, se zdi kot idealna in aktualna tema za raziskovanje omenjenih vprašanj za slovenski prostor. A kljub vsem tem vzpodbudnim mednarodnim raziskovalnim rezultatom smo v začetku sklepali, da v Sloveniji metoda sentimenta le ne bo tako zelo uporabna, tako zaradi majhnega števila »vitov« kot tudi zaradi neučinkovitosti slovenske borze. Statistični kazalnik »delež trgovanja do BDP« (World Bank 2012a; World Bank 2012b) je namreč v Sloveniji 3 do 6-odstoten, medtem ko se v zahodnih evropskih deželah (npr. Nemčija) takšen delež giblje tudi med 40 do 80 odstotki. To pa nam nakazuje, da Slovenci še nismo tako aktivni pri trgovanju z delnicami.

Zaradi zavedanja raziskovalnega problema smo napisali algoritem zbiranja Twitterjevih dokumentov v programu R. Tehnično gledano naš algoritem sekvenčno izrablja vse Twitter API potenciale (Twitter Developers 2015) ter najhitrejšo knjižnico »data.table« (Dowle 2015) za zbiranje podatkov in ravnanje z njimi v delovnem pomnilniku. V splošnem algoritem zbira podatke v dveh tabelah, kjer se zbirajo podatki o uporabnikih ter Twitterjevih dokumentih po različnih atributih. Algoritem je popolnoma avtomatiziran, v bazo pa s pomočjo posebno napisane funkcije za prepoznavo "Slovencev" zapisuje le slovenske uporabnike. Naš cilj v magistrskem delu bo raziskati, kako dobro se algoritem merjenja sentimenta odreže pri napovedovanju gibanja cene (slovenskih) delnic. Zanimalo nas bo: *»Ali je izmerjeni sentiment uporaben za napovedovanje cen delnic?«* Zraven pa se poraja tudi problem, kako ustvariti učinkovit algoritem in aplikacijo z grafičnim izpisom ob rabi programa R in njegovih knjižnic. Aplikacija mora namreč prikazovati gibanje cen delnic (ali tečajev) in ustrezen sentiment na spletni strani v skoraj realnem času - v približno 20-minutnem razponu, glede na omejitve Twitter API-ja. Drugo raziskovalno vprašanje se bo torej glasilo: *»Kako lahko na kar najbolj eleganten in enostaven način konstruiramo vizualno reprezentacijo proučevane teme s pomočjo programa R in njegovih knjižnic?«*

Zaradi zahtevnosti dimenzij zastavljenih raziskovalnih vprašanj bomo skozi magistrsko delo v teoretičnem delu pregledali znanstvena področja podatkovne znanosti in masovnih podatkov. V ospredju bo opredelitev in poudarek na razvoju masovnih podatkov in izzivov, ki jih predstavljajo in običajno prinašajo podatkovnemu znanstveniku. Zanimala nas bodo vprašanja kaj, zakaj in *kako* masovni podatki

»nastajajo« in kakšna je vloga primerne infrastrukture, da takšne »gromozanske« količine podatkov lahko modeliramo. Zanimale nas bodo glavne ideje in koncepti, *zakaj* probleme modeliranja podatkov gruče računalnikov med seboj obdelujejo paralelno in pri takšnih operacijah programski algoritmi uporabljajo porazdeljene datotečne sisteme. S pregledom primerne infrastrukture, osredotočene na okolje R in naš problem zasnove avtomatiziranega modeliranja zastavljenega problema, bomo v naslednjem poglavju pregledali znanstveno področje podatkovnega rudarjenja in modeliranja. Zanimalo nas bo, kaj pravzaprav je proces podatkovnega rudarjenja in *kaj* pomeni, ko v splošnem pravimo, da gradimo *model*. V zadnjem poglavju teoretičnega dela bomo raziskali pomen analize sentimenta in ga definirali. Zanimalo nas bo, za *kaj* lahko analize sentimenta uporabljamo ter kako se tehnično oz. statistično lotevamo analize sentimenta. Več o naši konstrukciji programa (zgrajenega za potrebe namiznega računalnika z omejitvijo dveh gigabajtov podatkov), zajemanju tvitov in učinkovitosti merjenja sentimenta bomo povedali v empiričnem delu. Predstavili bomo ustvarjeni program¹ za analizo podatkov iz slovenskega omrežja Twitter ter preverjali primernost zgrajenega računskega modela z neposrednim merjenjem podobnosti krivulj gibanja cene delnic (indeksa). Naknadno pa bomo izračunane povezanosti preverjali tudi praktično preko izpisanih grafov.

¹ Naj kot opombo poudarimo, da glede na to, da celoten program zahteva le en računalnik, ne moremo trditi, da smo ustvarili program za masovne podatke (takšni programi namreč običajno izkoriščajo gruče računalnikov), kljub temu pa zadostuje za modeliranje celotnega slovenskega omrežja Twitter.

2 TEORETIČNI DEL

2.1 Masovni podatki in podatkovna znanost

2.1.1 »Velikanski kup lego kock«

Kaj predstavljajo masovni podatki (*angl. big data*), lahko ponazorimo s sliko **velikanskega kupa lego kock**. Kup lego kock nam še ne predstavlja nekega smisla ali pomena, če ga opazujemo od daleč. Šele ko si zamislimo in predstavljamo, kaj bi lahko s takšnimi kockami lahko ustvarjali in počeli, se nam lahko ponudijo velike si ideje in možnosti (Forbes 2013).

Slika 2.1: Masovni podatki kot velik kup lego kock



Vir: Forbes 2013

2.1.2 Definicija masovnih podatkov

Pojem je sicer relativno »slabo« definiran, saj obstaja ogromno definicij. V splošnem lahko rečemo, da so to *nepredstavljljive količine podatkov*, ki se pretakajo v medmrežju in neposredno ali posredno merijo delovanje in aktivnosti uporabnikov ali dogodkov (objave besedil in slik, filmov, beleženje »piškotkov«, Googlova analitika ter podobno). Vendar pa je takšen splošen opis preveč laičen in ne zajame problema definicije masovnih podatkov.

2.1.3 Problem definicije masovnih podatkov

Izraz masovni podatki predstavlja podatkovne baze, ki so tako obsežne in velike, da jih je nemogoče analizirati, upravljati in obdelovati s konvencionalno programsko in strojno opremo v sprejemljivem in dovolj hitrem času (Snijders in drugi 2012). Verjetno se vsakemu izmed nas poraja vprašanje: »Kolikšna pa je ta številka, da lahko govorimo o masovnih podatkih?« Pravzaprav se velikost »masovnih podatkov« oziroma definicija »velikosti« venomer oddaljuje in beži stran od nas in nikakor ni enoznačna ali primerna oznaka za operiranje s takšnim pojmom. Obstaja torej več definicij pojma, prevladujoči dve pa bomo preverili v naslednjih dveh podpoglavjih.

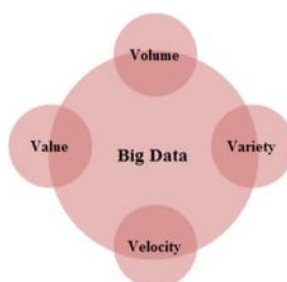
2.1.4 Definicija 3V

Leta 2001 je raziskovalec v Gartnerju (Laney 2001) kot eden izmed prvih definiral problem velikosti masovnih podatkov kot 3-dimenzionalen: **količina, hitrost, raznolikost** (*angl. Volume, Velocity, Variety*). Te dimenzije sovpadajo z razlago večanja prostora oz. količino podatkov, s hitrostjo procesiranja podatkov (podatki noter, procesiranje in ven) in z raznolikostjo podatkovnih struktur in virov. Večina tehnološke stroke še vedno uporablja omenjeno definicijo treh V-jev pri operacionalizaciji pojma masovnih podatkov.

2.1.5 Definicija 4V

Drugi (Gantz in drugi, 2015) dodajajo tej definiciji še en V – vrednost (*angl. value*). Ta definicija (slika 2) je širše sprejeta, saj dodaja definiciji 3V še označevanje pomembnosti in pomena.

Slika 2.2: Definicija 4V masovnih podatkov



Vir: Abaker in drugi 2015

Masovni podatki so torej za nas skupek tehnik in tehnologij, ki zahtevajo nove oblike integracij za odkrivanje velikih skritih vrednosti (iz velikanskih podatkovnih baz, ki so različne, kompleksne in zelo obsežne (Abaker in drugi 2015, poglavje 2).

Avtorji študije povzamejo naslednje ključne točke pri razlagi definicije 4V:

1. **Količina** se nanaša na velikost vseh vrst podatkov, ki so pridobljeni iz različnih virov in se še naprej širijo (prav tam). Koristi zbiranja velikih količin podatkov vključujejo ustvarjanje skritih informacij in vzorcev s pomočjo analize podatkov.

2. **Raznolikost** se nanaša na različne vrste podatkov, zbranih s senzorji pametnih telefonov ali socialnih omrežij (prav tam). Takšni podatkovni tipi vključujejo video, slike, besedila, avdio in podatkovne dnevnike v strukturiranih ali nestrukturiranih formatih. Večina takšnih podatkov, pridobljenih iz mobilnih aplikacij, je v nestrukturirani obliki. Na primer besedilna sporočila spletne igre, blogi in socialni mediji ustvarjajo različne vrste nestrukturiranih podatkov prek mobilnih naprav in senzorjev.

3. **Hitrost** se nanaša na hitrost prenosa podatkov (prav tam). Vsebine podatkov se nenehno spreminjajo. Zaradi absorpcije dopolnilnih zbirk podatkov, že prej arhiviranih podatkov ali podedovanih podatkovnih zbirk, ki se pretakajo iz različnih virov.

4. **Pomen** je najbolj pomemben vidik masovnih podatkov; nanaša se namreč na proces odkrivanja ogromnih, za nas »skritih vrednosti« (pomenljivih, uporabnih) iz obsežnih podatkovnih baz, z različnimi vrstami in hitrostjo ustvarjanja (prav tam).

2.1.6 »Moorov zakon« in hitrost ustvarjanja podatkov

Količina podatkov na *internetu* se neizbežno povečuje. Velike mednarodne korporacije napovedujejo, da se bo količina podatkov na vsaki 2 leti podvojila do leta 2020 (Williams 2014). **Internet stvari** (*angl. Internet of Things*) in masovne količine podatkov nakazujejo trende razvoja in velikanskega kopičenja podatkov za premostitev potreb med platformami stvari, uporabniki in poslovnimi subjekti (Smith 2015). Nekateri strokovnjaki (Bradicich in drugi 2013) ter študije napovedujejo kar 10-kratno rast podatkov med leti 2013 in 2020. Drugi orisujejo sliko s primerjavo količine podatkov, ki jih zmore zajeti tablica

Apple iPad Air. Če te tablice danes zlagamo na kup, tako da bodo ležale ena na drugi, bodo skupaj dosegale dolžino dveh tretjin poti do Lune. Do leta 2020 pa lahko pričakujemo, da se bo ta kup povečal kar za 6,6-krat (Williams 2014).

V ozadju masovnih podatkov (po nekaterih razlagah) *naj bi se skrival Moorov zakon*. Gordon Moore je že leta 1965 opazil (Brdicich in Orci 2013), da se količina tranzistorjev na integriranem čipu podvoji na približno vsaki dve leti. Njegov zakon je bil posledično prenesen tudi na podatkovno področje. Posledica Moorovega zakona je tehnologija, ki jo dnevno uporabljamo, saj je vedno bolj dostopna masam uporabnikom ter s tem »demokratizirana«. Omogoča nam, da lahko relativno velike količine podatkov bistveno hitreje shranjujemo, obdelujemo in analiziramo.

»Če orišemo problem, leta 1995 je bilo izdelanih za 20 peta bajtov podatkovnih nosilcev. Leta 2013 je Google procesiral že za več kot 24 peta bajtov podatkov - vsak dan.« (prav tam) »Povprečno gospodinjstvo danes ustvari dovolj podatkov, da zapolni 65 mobilnih telefonov iPhone (32 Gb) na leto. V letu 2020 se bo število telefonov povečalo na 318.« (Williams 2014) IBM ocenjuje (IBM 2015), da danes ustvarimo za kar 2,5 kvintilijona² (*angl. quintillion*) bajtov podatkov vsak dan. Kot zanimivost, takšna hitrost ustvarjanja podatkov povzroči, da ocenjujemo, da podatki, ustvarjeni na internetu v zadnjih dveh letih, predstavljajo kar 90 % vseh podatkov na internetu (prav tam).

2.2 Pričakovanja, trendi in izzivi masovnih podatkov

² Kvintilijon predstavlja število velikosti 10 na 18-to potenco.

Pregledali bomo trende ter tehnologije, povezane z rabo, storitvami masovnih podatkov in podatkovnega rudarjenja, z raziskavami podjetja Gartner in orodja Google Trends. Omenili pa bomo tudi profil osebe **podatkovni znanstvenik**, ki je povezan z vzponom masovnih podatkov. Nadaljevali bomo s pregledom izzivov **primerne infrastrukture – programske in strojne**, teoretično pa bomo raziskali tehnični problem **porazdeljenega datotečnega sistema** in algoritem **MapReduce**, saj se bomo na te pojme opirali v empiričnem delu pri zasnovi naše aplikacije. Hkrati pa menimo, da so ti pojmi pomembni za razumevanje procesov in računskega modeliranja masovnih podatkov ali podatkov, ki so preveliki za delovni pomnilnik računalnika.

2.2.1 Gartnerjeva krivulja pričakovanj

Gartner je svetovalna agencija na področju novih informacijskih tehnologij. V letu 2014 so bili masovni podatki na Gartnerjevi krivulji pričakovanj, leta 2015 pa so jih umaknili, na njihovo mesto pa postavili nekatere sorodne pojme. Pravzaprav so ti sorodni pojmi kot nekakšni podelementi in sistemi, ki (lahko) upravljajo z relativno velikanskimi količinami podatkov, tako da je ta pojem po naši oceni že nekako integriran v njih, saj nenazadnje podatki predstavljajo vse, kar je ustvarjeno digitalno (na internetu).

Slika 2.3: Gartnerjeva krivulja pričakovane adopcije tehnologij



Vir: Gartner 2015

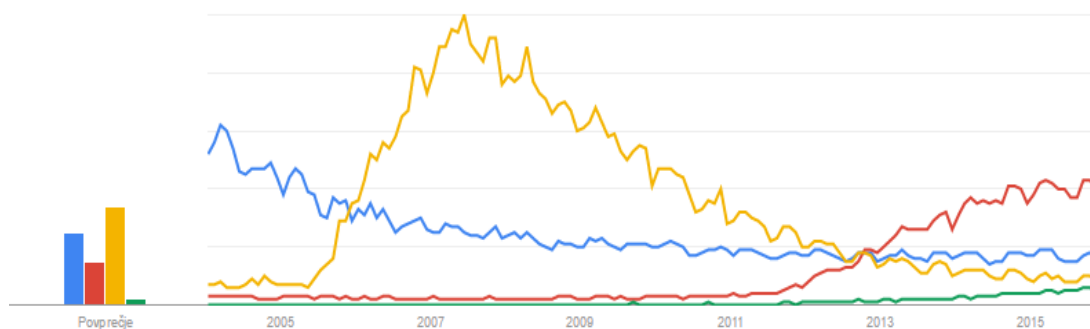
2.2.2 Google trendi

Analiza ključnih besed na Googlu predstavlja uporabno predstavo za raziskave zanimanja uporabnikov, kaj se je najbolj iskalo in bralo skozi čas.

V tem trenutku je ključna beseda masovni podatki zelo aktualen in iskan pojem na Googlovem iskalniku. Na spodnjem grafu je primerjava iskanja po ključnih besedah: data mining, web 2.0, big data in data scientist. Lahko rečemo, da bo popularnost big data še nekaj časa rasla, pričakuje pa se, da bo s časom doživela podoben padec, kot sta ga doživela web 2.0 in data mining. Tudi iskalni pojem podatkovni znanstvenik (Ferle 2013), (*angl. data scientist - profil osebe, ki je kompetentna za obdelavo, analizo in interpretacijo masovnih podatkov*), ki je bil prvič omenjen v letu 2008 in sovpada z analizo masovnih podatkov, začena rasti.

Slika 2.4: Trendi iskanja ključnih besed

Legenda: "data mining", "web 2.0", "big data", "data scientist"



Vir: Google Trends 2015

Glavni izziv, s katerim se soočamo pri analizi velikanskega vira podatkov, je najti načine, da zmoremo ogromne količine vsebin in podatkov tehtno in pametno analizirati, pa tudi učinkovito in dovolj hitro obdelati in procesirati zaradi vedno hitrejša akumulacije toka podatkov. Zaradi takšnih zahtev in potreb potrebujemo tako strojne kot programske primerne rešitve in algoritme, ki nudijo potrebno podlago in kompetentno infrastrukturo. Običajno v takšnem kontekstu govorimo o podatkovni znanosti (*angl. data science*), kjer analizira oziroma rudari (*angl. data mining*) podatkovni znanstvenik.

2.2.3 Vloga podatkovnega znanstvenika

Podatkovno rudarjenje je interdisciplinarno področje, ki je izredno široko ter prepleteno med različnimi strokovnimi znanostmi (Williams 2011). Poleg tega pojma običajno v kombinaciji z masovnimi podatki (Ferle 2013) omenjamo tudi tako imenovano podatkovno znanost. Potencial digitalnega sveta in možnosti analiz masovnih podatkov raste, s tem pa se omenja tudi profil osebe, ki poseduje ustrezna znanja, izkušnje, potrebno izobrazbo in ustvarjalno radovednost (prav tam). Ferle v reviji Monitor pravi, da takšno osebo običajno imenujemo **podatkovni znanstvenik**, saj odkriva skrite vzorce in išče informacije v svetu velikih količin podatkov, izsledke pa prenese v poslovno okolje. Naziv podatkovni znanstvenik se je prvič začel omenjati v podjetju LinkedIn leta 2008. Podatkovnega znanstvenika opredeljuje kot osebo z naslednjimi lastnostmi: »Lastnosti, ki jih mora imeti podatkovni znanstvenik, so na primer neke vrste podatkovni heker, analitik, komunikator in zaupanja vreden svetovalec.« (prav tam).

Te lastnosti so torej nekakšna zmes računalniških znanosti, podatkovnih baz, statistike, metodologij in različnih teorij znanosti (ekonomske, psihološke, biološke ipd.), odvisno od

področja problema, ki ga rešujemo. Seveda pa ne morejo imeti vseh znanj, zato sodelujejo s strokovnjaki iz drugih disciplin.

Običajno poleg kompetentne osebe v vlogi podatkovnega znanstvenika potrebujemo tudi strojno in programsko infrastrukturo. Vlogo teh dveh zahtev bomo opredelili v naslednjem poglavju, ko bomo govorili o vlogi primerne infrastrukture. Osredotočili se bomo na zahteve programskega okolja R in strežniške infrastrukture v oblaku.

2.2.4 Vloga primerne infrastrukture

V tem poglavju se bomo osredotočili na pregled programske infrastrukture in na informacije, ki so za naš problem masovnih podatkov (oziroma podatkov, ki so preveliki za delovni pomnilnik) in raziskovalnih vprašanj uporabne, hkrati pa je z njimi možno operirati in jih povezati s programskim paketom R.

Podjetja so začutila priložnosti za analizo naših podatkovnih sledi, ki jih puščamo na internetu. Nastale so različne programske platforme, storitve in podjetja, povezana z analizo masovnih podatkov (slika 4). Naj omenimo dve najbolj popularni in uporabni odprtokodni tehnologiji za shranjevanje in obdelavo podatkov (rdatamining 2015), ki omogočata povezavo s programom R:

Apache Hadoop - ogrodje (*angl. framework*), ki omogoča porazdeljeno obdelavo velikih podatkovnih zbirk prek gruč računalnikov (prav tam). Ustvarjeno je predvsem za shranjevanje prostorsko zahtevnih podatkov, ki jih procesiramo po skupinah oz. delih (Il-Yeol Song 2015). Paketki, ki omogočajo povezavo z ogrodjem, so Hive, HadoopStreaming, RHIPE, Segue, RHadoop (rmr, rhdfs, rhbase).

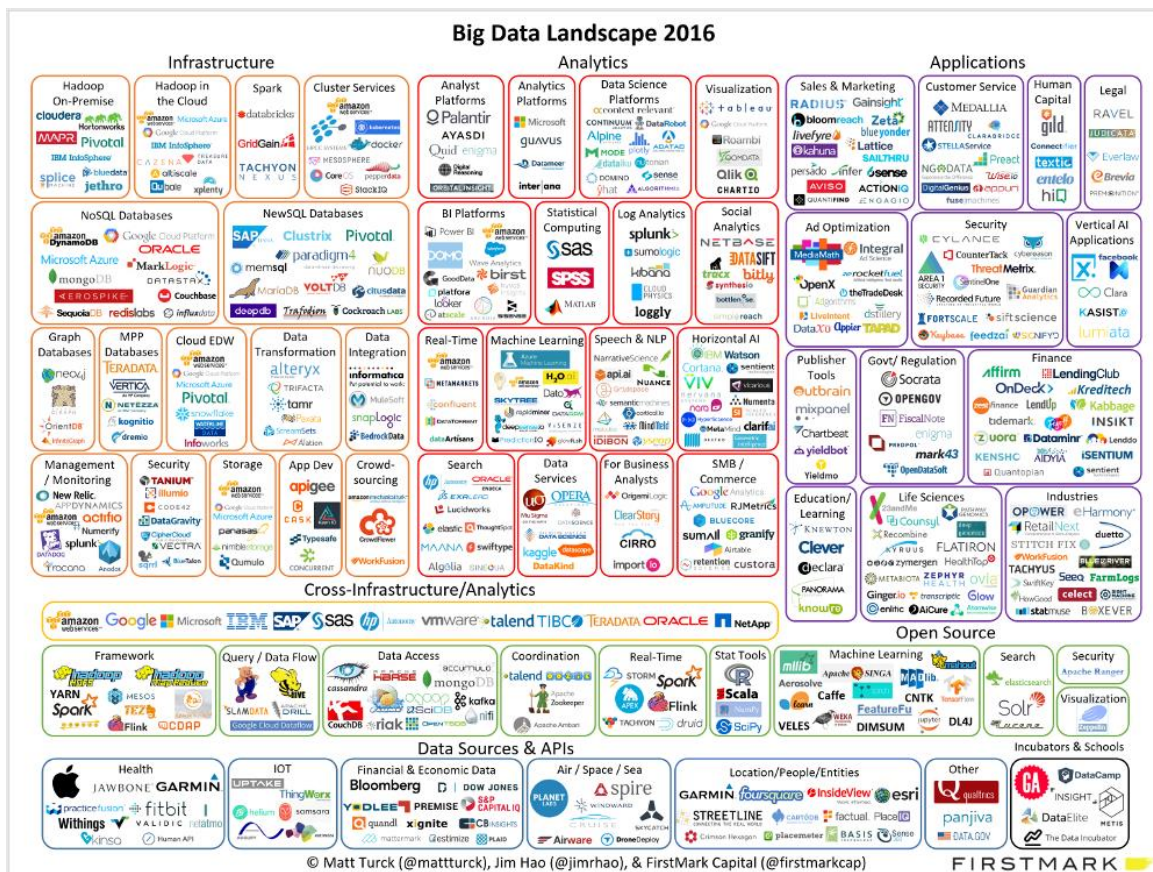
Apache Spark - splošno in hitro ogrodje za procesiranje masovnih podatkovnih zbirk, 100-krat hitrejša od Hadoop-a (rdatamining.com 2015). Ustvarjeno je za operacije v pomnilniku, za zahteve algoritmov pri ponavljajočih se operacijah, običajno za statistično učenje (Il-Yeol Song 2015). Paketek za dostop se imenuje SparkR.

Večina takšnih in podobnih tehnoloških rešitev (na nivoju rešitev platform in ogrodij za masovne podatke) je odprtokodne narave in jih lahko izdelamo ali povežemo s programom R in njegovimi knjižicami. Ostale analitične storitve, ki jih običajno najdemo kot storitve na

internetu za analize in vizualizacije, pa imajo običajno plačljiv poslovni model (večina jih je zgrajenih na odprtokodnih temeljih, kar bomo skušali pokazati v naslednjih poglavjih).

Na naslednji strani prilagamo skico platform, tehnologij, storitev in podjetij, povezanih s masovnimi podatki (slika 5).

Slika 2.5: Skica platform, tehnologij, storitev in podjetij



Vir: Google.com

2.2.5 Računske gruč in algoritem »MapReduce«

Moderne aplikacije podatkovnega rudarjenja zahtevajo hitro upravljanje z velikanskimi količinami podatkov (Leskovec in drugi 2014, 21). Takšne aplikacije uporabljajo programske sisteme, ki običajno učinkovito upravljajo z arhitekturo **paralelizma**. Običajna **strojna** infrastruktura ne temelji na super računalniku, temveč na veliki zbirki običajnih računalnikov (njihova procesorska jedra imenujemo računska vozlišča - *angl. compute nodes*), sestavljenih v tako imenovane računske gruč (angl. *computational clusters*) in povezanih v mrežo. Leskovec pravi, da so aplikacije v preteklosti upravljale paralelno klicanje vozliščem, npr. velikanske znanstvene kalkulacije, ki so bile ustvarjene za posebne

namenske (super) računalnike, ki so delovali paralelno. Imele so ogromno procesorjev in specializirano strojno opremo. Danes pa se takšne rešitve običajno **izvajajo v podatkovnih oblakih**, preko »spletnih rešitev«, ki so stroškovno bistveno cenejše od namenskih strežnikov, npr. Amazon EC2 (Amazon EC2) ali pa DigitalOcean. Najem najmočnejše instance enega vozlišča v oblaku stane manj kot en evro na uro pri ponudniku DigitalOcean, medtem ko je cena najcenejše instance »vozlišča« pet evrov na mesec (DigitalOcean).

Instalacija RStudio v podatkovnem oblaku je enostavna. Bistvo je, da pri ponudniku spletnega oblaka konfiguriramo Unix strežnik, nanj pa namestimo RStudio server, do katerega nato lahko dostopamo preko brskalnika v okolje kodnega IDE urejevalnika RStudio. Odprtokodna skupnost je zadevo še bolj poenostavila preko sistema oz. paketa docker (Docker). Takšna rešitev (Boettiger 2015) poenostavi konfiguracijo, s kreditno kartico pa lahko v nekaj minutah vstopimo v novo spletno instanco R seje.

2.2.6 Porazdeljeni datotečni sistem, podatkovne baze

Večina računskih operacij (Leskovec in drugi 2014, 21) je izvedenih na enem procesorju, kjer glavni pomnilnik, predpomnilnik (*angl. cache*) in lokalni trdi disk predstavljajo **računsko vozlišče** (*angl. compute node*). Sklad programov vsebuje obliko sistema shranjevanja datotek, imenovano porazdeljeni datotečni sistem (*angl. distributed file system* – v nadaljevanju DFS). DFS lahko vsebuje večje enote, kot so konvencionalni diski v operacijskem sistemu, tudi do velikosti 1 TB, običajno pa so porazdeljeni po delih in veliki 64 MB (Leskovec in drugi 2014, 24). Na primer format GFS (Google File System), HDFS (Hadoop Distributed File System) ali pa CloudStore (odprtokodni DFS). Pri takšnem skladiščenju se podatki redko posodablajo ali urejajo. Običajno se na koncu »verige« nove podatke pripne zadnjemu delu. Porazdeljeni datotečni sistemi vsebujejo tudi replikacijo podatkov in metode za odkrivanje napak v bloku digitalnih podatkov (prav tam), za obrambo pred pogostimi napakami pri zapisu ali branju podatkov, ki so distribuirani skozi tisoče nizkocenovnih računskih vozlišč (*angl. compute nodes*).

Pojma relacijska podatkovna zbirka in datotečni sistem sta *običajno* med seboj različna termina v računalniških znanostih (cs.princeton 2009). Kljub temu pa se danes med seboj lahko prepletata in *lahko* vsebujeta nekatere elemente drug drugega (Sas). Kot opombo omenimo, da datotečni sistem običajno temelji na operacijskem sistemu oz. je odvisen od njega in ostalih pravil ter praks. Datotečni sistem vsebuje podatke - običajno nestrukturirane oblike, medtem ko imajo podatkovne zbirke vedno strukturirano obliko. Ti pojmi pa so se ob

pojavo masovnih podatkov in tehnologij pomešali med seboj, nekako konvergirali ali (če se lahko tako izrazimo) celo nadgradili, napredovali in standardizirali.

2.2.7 Programski sistem MapReduce in R

Implementacije algoritmov MapReduce omogočajo zmožnost procesiranja (pre)obsežnih podatkov, ki jih običajno poganjamo na računskih gručah. Teh podatkov ne moremo prilegati v delovni pomnilnik enega računalnika, saj je naš problem računsko oz. časovno preveč potraten (Leskovec, Rajaraman in Ullman 2014, 40). Konceptualno v splošnem takšne metode modeliranja izvajamo na računskih gručah (implementacija zbirk več »računalnikov«), ki jih običajno vodi (en) **nadzornik** (*angl. master*), ostala računska vozlišča pa imenujemo **delavce** (*angl. workers ali slaves*). Delavci običajno delujejo vzporedno, seveda pa je odvisno od implementacije in zahtev posameznega algoritma.

Takšni algoritmi morajo biti napisani tako (Leskovec in drugi 2014, 24), da z njimi ne presežemo omejenega pomnilnika v računskem vozlišču, časovno pa morajo biti takšni delavci med seboj usklajeni, glede na zahteve nadzornika v računski gruči. Kot bomo v empiričnem delu kasneje pokazali »MapReduce« algoritem na primeru **enega sekvenčnega računskega vozlišča**, je tudi tam bistvo takšnega pristopa deli in vladaj (torej brez logike »nadzornika« in paralelnega izvajanja kode in še to na enem procesorskem jedru). Uporabijo se **naslovi ključev**, ti pa »vodijo« do naših **shranjenih vrednosti** (*angl. key-value pair*) porazdeljene datotečne tabele (*angl. filehash*) (Peng 2015,1) na podatkovnem nosilcu. Slednji so nato prebrani in obdelani v **mejah** oz. zmožnostih pomnilnika, s katerim lahko operiramo preko naše strojne oziroma systemske infrastrukture. Kot pravi bivši Googlov inženir Jeffrey Breen glede algoritma MapReduce, je treba najprej napisati funkcijo »Map«, pri kateri definiramo ključe in podatke, v fazi »Reduce« pa zmodeliramo problem, ki je bil npr. prej razporejen po ključih (Breen 2012, 4). Običajno za to poskrbijo namensko ustvarjena ogrodja (glej 2.2.4), možno pa je takšno tehnično implementacijo napisati tudi v R-ju, npr. s knjižnicama snow (Tierney 2015) in Rmpi (Yu 2015), za potrebe vzpostavitve sistema **nadzornik-delavci** v gruči. To so samo nekatere možne rešitve implementacije takšnih algoritmov (gotovo jih je še več).

Za izdelavo **hitrejših** MapReduce algoritmov za porazdeljeno, paralelno (večjedrno) procesiranje v računskih gručah (oz. enojnem računskem vozlišču) je v R-ju namenjena npr. knjižnica foreach (Weston 2015), doParallel in doSNOW (Calaway 2015). Kot zanimivost, Steve Weston (skrbnik odprtokodnega paketa foreach) je eden izmed ustanoviteljev že

»bivšega« podjetja *RevolutionAnalytics*. Ustvaril je R Revolution, danes pa se imenuje Microsoft R (revolutionanalytics 2016). Gre za celovito, zaprtokodno R-jevo rešitev, pri kateri se uporabnikom programskega okolja problemi, kot so večjedrne računske operacije, avtomatično rešeni ali pa poenostavljeni. Pri tem gre za uporabnika plačljiv poslovni model uporabe, kljub temu pa si ga lahko univerze pridobijo zastonj. Podjetje je leta 2015 Microsoft odkupil za neznano vsoto, ideja podjetja pa je povezati akademike in podjetja ter uporabiti infrastrukturo Microsoft R za izgradnjo novih programskih rešitev za podjetja in korporacije.

Če iz različnih razlogov kompleksnejših MapReduce algoritmov v R-ju na računskih gručah ne želimo implementirati, lahko torej uporabimo enega izmed prej omenjenih ogrodi (poglavje 2.2.4), npr. za vzpostavitev Hadoop ali pa Spark gruč. Takšen sistem poskrbi za izvajanje paralelnih, *časovno učinkovitih* distribuiranih datotečnih sistemov podatkov in procesiranja na več 1000 računskih gručah.

Kot smo nekako skušali nakazati, se programska infrastruktura gradi na programski infrastrukturi (orodje zgradi orodje). Bistvo dobrega in učinkovitega podatkovnega znanstvenika je torej nuja, da razume, kako in na kakšen način lahko kar najhitreje implementira rešitve problemov podatkovnega rudarjenja in ustvarja modele iz podatkov.

2.3 Podatkovno rudarjenje in modeli

Podatkovno rudarjenje se kot pojem uveljavlja pri poslu, znanosti in z njimi povezanih informacijsko-računalniških rešitvah. Kaj pravzaprav je podatkovno rudarjenje in kaj pomeni, ko podatkovni znanstvenik ustvarja »modele«?

2.3.1 Podatkovno rudarjenje

V zaledju informacijsko-računalniških rešitev se kot metoda in znanost zbiranja podatkov pojavlja izraz podatkovno rudarjenje. **Podatkovno rudarjenje** je umetnost in znanost **pametne analize podatkov** (Williams 2011, 23). Cilj podatkovnega rudarjenja je poiskati pomenljive in uporabne ugotovitve ter zaključke iz takšnih podatkov. Ena izmed svetovno znanih avtoritet, slovenski profesor Jure Leskovec (Leskovec in drugi 2014, 1), definira podatkovno rudarjenje kot *preprosto iskanje modelov iz podatkov*. Definicija modela pa lahko kljub temu zajame eno od več stvari. Zato bomo v naslednjih poglavjih pogledali glavne usmeritve modeliranja podatkov.

2.3.2 Statistično modeliranje

Leskovec pravi, da so bili statistiki prvi, ki so uporabili izraz podatkovno rudarjenje (Leskovec in drugi 2014, 1) oziroma podatkovno izkopavanje (*angl. data dredging*). Na začetku je podatkovno rudarjenje pridobilo *slabšalen* prizvok, ki se je nanašal na poskus pridobivanja informacij (podatkov), ki niso bili veljavni (podprti). Če se bolj nazorno izrazimo, bi to pomenilo iskanje igle v kupu sena, ki je tam sploh ni. Takšen slabšalen izraz se pri uporabi podatkovnega rudarjenja po navadi nanaša na odkrivanje vzorcev v podatkih, ki so predstavljeni kot statistično značilni (*angl. statistically significant*), brez poprej opredeljenih specifičnih hipotez, ki bi imele (**metodološki** ali **teoretični okvir**) vzrok za iskanje. Npr. najdena je statistično značilna povezanost med številom prodanih sladoledov in številom na novo rojenih delfinov v določenem časovnem obdobju. V takšnem primeru bi skoraj intuitivno lahko rekli, da gre za lažno povezanost – iskanje povezanosti (kaj, kako in kdaj) torej še ne pomeni vzročnosti (zakaj). Posledično bomo z uporabo klasičnih metod statističnega sklepanja tako skoraj zagotovo dobili vsaj »kakšen sklep«, ki bo statistično značilen pri npr. 5 % stopnji tveganja.

Takšnemu načinu »iskanja« so rekli tudi **Bonferronijevo načelo (oz. problem)** (Leskovec in drugi 2014, 1.2.2 pogl.), ki je kasneje postal statistični izrek. Kot pravi Leskovec (Leskovec in drugi 2014, 1.2 pogl.), vidi problem (takšnega) podatkovnega rudarjenja v zaznavanju popolnoma slučajnih dogodkov (vplivov, učinkov, korelacij ipd.), skritih znotraj masovnih količin podatkov.

Vzemimo primer, da imamo na populaciji 20 nepovezanih spremenljivk, v naši analizi pa posedujemo podatke 190 parov spremenljivk. Če vzamemo nek vzorec iz te populacije in za vseh 190 parov testiramo domnevo o nepovezanosti pri 5 % tveganju (za vsak par posebej), lahko pričakujemo, da bomo pri 9-10 parih »ugotovili« statistično značilno povezanost, čeprav vemo, da v populaciji *ni* povezanosti. Izrek v statistiki, poznan kot »**Bonferronijev popravek**« (*angl. Bonferroni correction*) – **Bonferronijev test ali metoda**, pri kateri se opravlja več ponovitev testiranja zavračanja ničelne hipoteze, pa se takšnemu nepravilnemu sklepanju v iskanju **vzroka** (vplivov, učinkov, korelacij ipd.) pojavov iz slučajnih podatkov izogiba. Bonferronijev popravek in podobne tehnike nas torej poskušajo obvarovati pred takšnimi napačnimi ugotovitvami.

Zardi takšnih spoznanj je danes med statistiki podatkovno rudarjenje pridobilo pozitiven prizvok. Kot pravi Leskovec in drugi (Leskovec in drugi 2014, 1), se izraz nanaša na **izgradnjo statističnih modelov**, pri katerih je ključna (oz. se domneva) **verjetnostna**

porazdelitev podatkov. Gre za *postopek*, ki nam je v družboslovju (in naravoslovju) bolje poznan kot **proces inferenčne statistike**. Slovenski sinonim za angleški izraz »inference« je sklepanje, primer takšnega sklepanja pa je npr. Gaussova krivulja, statistikom bolj poznana tudi pod imenom normalna porazdelitev (prav tam).

2.3.3 Strojno ali statistično učenje

Nekateri razumejo podatkovno rudarjenje kot sinonim za strojno učenje (Leskovec in drugi 2014, 20). Leskovec pravi, da se strinja, da nekateri algoritmi podatkovnega rudarjenja uporabljajo strojno učenje.

Strojno učenje uporablja za učenje *podatke*, pri katerih se uporablja enega izmed mnogih algoritmov, npr. Baysove mreže, metodo podpornih vektorjev, nevronske mreže, odločitvena drevesa, skrite markovske verige ipd. Obstajajo situacije, kjer je smiselno uporabiti takšen pristop. Predvsem takrat, kadar ne vemo, kaj iščemo v podatkih (prav tam). Glavni element »pameti«, bolje formulirano »pametnega modela«, je pravzaprav učenje, slednje pa je lahko nadzorovano ali nenadzorovano (Aggarwal in Zhai 2012, 6). Znanost podatkovnega rudarjenja tako lahko poimenujemo tudi kot podstat znanosti o umetni inteligenci. Umetna »pamet« torej skuša preko relevantnih vhodnih podatkov, ki jih pridobi iz podatkovnih baz, računsko (statistično) razumevati in predvidevati svet okoli sebe ter se iz njih učiti. Polje proučevanja so v računalniških znanosti poimenovali strojno učenje (*angl. machine learning*). Strojno učenje je torej zbiranje opazovanih podatkov preko interakcije s svetom in gradnja modelov sveta iz teh podatkov (Williams 2011). **Statistiki** pa so ga poimenovali tudi **statistično učenje** (Gareth in drugi 2014).

Po drugi strani pa strojno učenje posredno vedno ne pripomore k uspehu, saj lahko včasih opišemo cilje podatkovnega rudarjenja bolj direktno, z relativno enostavnejšimi računalniškimi algoritmi.

2.3.4 Računski pristopi k modeliranju

Nedavno so računalniški znanstveniki opredeljevali podatkovno rudarjenje kot algoritmični problem (Leskovec in drugi 2014, 1.1.3 pogl.). V takšnem primeru je model preprosto odgovor na kompleksno poizvedbo iz podatkov. Npr. iz našega prejšnjega primera (poglavja 2.3.2, drugi odstavek) želimo iz danega niza števil izračunati povprečje in standardni odklon.

Leskovec omenja, da sem spadata še podkategoriji tega problema, kot sta:

1. **Jedrnato in približno povzemanje podatkov** – npr. Leskovec (Leskovec in drugi 2014, 1.1.4 pogl.) podaja kot primer ideje Googlovega algoritma »PageRank«, ki je oblika spletnega rudarjenja. Pravi, da algoritem povzame kompleksno strukturo (semantičnega) spleta z enoznačnim številom posamezne spletne strani. Takšno število se (poenostavljeno) imenuje PageRank spletne strani. To *naj* bi bila (prav tam) verjetnost slučajnega sprehajalca, da se ta nahaja na spletni strani, na grafu spletnih strani v nekem času.
2. **Ekstrahiranje najbolj pomembnih značilnosti podatkov in neupoštevanje preostalih** – Leskovec (Leskovec in drugi 2014, 1.1.5 pogl.) poda primer tipičnega modela na osnovi značilnosti, ki pregleda najbolj ekstremne primere fenomena. Ponudimo poenostavljeno razlago, kjer statistična metoda, npr. Baysove mreže (*angl. Bayes nets*), uporabi reprezentacijo kompleksnih povezav med objekti s pomočjo iskanja najmočnejših statističnih odvisnosti med takšnimi objekti in pri tem »uporabi« samo tiste, ki predstavljajo vse statistične povezave. Kot primer Leskovec tu omenja nabor pogostih elementov (*angl. Frequent Itemsets*), kjer model iz podatkov, ki predstavljajo »košarico nakupov«, išče elemente (primere), ki so običajno kupljeni skupaj.

2.3.5 Pametni model kot produkt interdisciplinarnosti in konvergence znanstvenih polj

Zaključki, odkritja ter rezultati (analiz) so torej običajno opisani s tako imenovanimi modeli, kot smo bolj natančno opredeljevali pojem poprej. Pogosto opišemo podatkovno rudarjenje tudi kot proces gradnje modelov. Model zajame in formulira bistvo zajetega znanja. Model nam lahko pomaga pri razumevanju sveta, predvidevanju ter učenju novih in uporabnih informacij (Williams 2011). Tako je med znanstvenimi polji statistike (statistični algoritmi), računalniških algoritmov in podatkovnih baz prišlo do tako imenovane konvergence znanj.

2.4 Analiza sentimenta

Velike anketne raziskave reprezentativnih vzorcev populacije so na splošno drago in časovno zamudno opravilo. Nekateri so zato predlagali posredno merjenje sentimenta preko rezultatov nogometnih tekem ali vremenskih razmer. Kot pravijo avtorji (Johan in drugi

2010), pa je takšno merjenje kazalcev ali indikatorjev omejeno z njihovo točnostjo. V zadnjem desetletju je bil dosežen velik napredek tehnik sentimenta, ki izluščijo kazalce javnega razpoloženja neposredno iz družabnih medijskih vsebin, kot so blogi in obsežne Twitter baze. Kljub temu da je vsako sporočilo s Twitterja omejeno s 140 znaki, pa je skupek milijonov takšnih sporočil lahko natančen prikaz sentimenta (prav tam).

Kaj menijo strokovnjaki, zakaj je dobro uporabljati in meriti sentiment in ali so takšne analize sentimenta *sploh pomembne in uporabne* konvencionalnim marketinškim metodam merjenja sentimenta oz. javnega mnenja?

2.4.1 Pomen in uporabnost sentimenta

Aplikacije tako imenovanega spleta 2.0 (*angl. Web 2.0*), kot so forumi, blogi, mikroblog storitve, kot je Twitter, socialna omrežja, kot je npr. Facebook, ustvarjajo gromozanske količine vsebin potencialno bogatih virov sentimenta skozi širok spekter različnih subjektov, do katerih bi radi merili sentiment. Pojav uporabniško kreiranih vsebin, ki jih uporabniki pogosto polnijo večkrat na dan, spodbujajo javnost in potrošnike k »družabni participaciji«. S tem ustvarjajo potencialno bogat delež tržnih informacij, ki jih lahko uporabljamo pri poslu, politiki in v podobnih trženjskih sferah (Schillewaert *et al.* 2009 v Gunter in drugi 2014). Nekatere takšne storitve in spletne strani so visoko specializirane glede pogojev, tematike in ciljnih skupin (npr. Facebookov sistem uporabnika, strani in skupnosti), medtem ko so se druge razvile v generične komunikacijske platforme na ravni skoraj celotnega interneta (npr. Twitter) (Gunter in drugi 2014).

V takšnih okoliščinah in pogojih je možno ugotavljati in meriti stališča o komercialnih produktih in storitvah, učinkovitosti vladnih in službenih sektorjev, organizacij in politikov ter drugih javnih oseb, zvezd, filmov in televizijskih programov ter ostalih aktualnih družbenih problemih in dogodkih (prav tam). Sledenje javnega mnenja o specifičnih temah omogoča **identifikacijo trženjskih pričakovanj in potreb**, reakcij javnosti do **trženjskih znamk**, **političnih odločitev** in pa tudi reakcij do določenih družbenopolitičnih **dogodkov**.

Strokovnjaki (Gunter in drugi 2014) **priznavajo porast uporabnosti** potencialno bogatega vira informacij sentimenta iz spleta in spletnih podatkovnih virov.

2.4.2 Definicije sentimenta

Naša enostavna razlaga pojma analize sentimenta je **določitev razpoloženja pisca glede na temo ali splošen kontekst dokumenta**. »V 90. letih prejšnjega stoletja so panogo imenovali statistično razčlenjevanje (angl. *statistical parsing*), prve sentimentalne analize večjih količin besedil pa so bile narejene že leta 2002.« (Martinc 2013) Uporabna vrednost sentimenta je tako boljše »poslušanje« ali razumevanje večje količine besedil.

Zhang in Liu (Aggarwal in Zhai 2012, 465) definirata analizo sentimenta oziroma **rudarjenje mnenja** (angl. *opinion mining*) kot računsko študijo človeškega mnenja, ocen, odnosov in čustev do entitet, posameznikov, problemov, dogodkov, tem in njihovih lastnosti. Osnovna naloga sentimentalne analize je torej opredelitev polaritete stanja pisca dokumenta. V splošnem nas zanima, ali je stanje pisca do teme pozitivno, negativno ali nevtrarno.

Analiza sentimenta, znana tudi kot klasifikacija mnenja (angl. *sentiment classification*), se nanaša na uporabo naravnega procesiranja jezika in analizo besedil za identifikacijo in ekstrakcijo subjektivne ali čustvene evalvacije iz določenega vira (Pang in Lee 2004 v Gunter in drugi 2014).

Še posebej si prizadeva proučiti, katere tekstovne lastnosti (leksikalne, skladenjske ali bolj nedavno frekvence besed) prispevajo k čustveni vsebini besedila, kako so lahko takšne lastnosti samodejno zaznane (na nivoju besede, stavka ali na ravni celotnega besedila) in kvantificirane. Soroden izraz je **analiza subjektivnosti**, čeprav ta pojem ni vedno zamenljiv s pojmom analiza sentimenta (Gunter in drugi 2014). Subjektivnost v tem kontekstu je izraz, ki ga je predlagal Banfield leta 1982, nanaša pa se na jezikovno izražanje osebnih stanj (duševnih ali čustvenih stanj, ki so objektivnemu opazovalcu *skrita*). Na ta način je analiza subjektivnosti namenjena identifikaciji različnih atributov osebnih stanj v smislu narave samega subjekta, ki izraža svoje osebno stanje (stališče ali mnenje), vrsto odnosa, ki ga ta oseba izraža, in njeno intenzivnost predmetu (oz. subjektu), na katerega se to stanje nanaša. Takšna študija teoretične tekstovne abstraktifikacije *skritih dimenzij in lokacij čustvenih stanj* se primarno nanaša na valenčno dimenzijo dobro-slabo in dimenzijo jakosti močno-šibko, ki ji po navadi sledi še dimenzija opredeljenega pomena (glej Osgood et al. 1957 v Gunter in drugi 2014).

2.4.3 Proces analize sentimenta in različni pristopi

Obstajata dva splošna pristopa do analize sentimenta: **strojno učenje** (oz. statistično učenje) in **analiza na nivoju slovarja besed**, pri katerih skušamo klasificirati določen del besedila oziroma označiti njegovi razvrstitev v kategorijo. Pri obeh pristopih se običajno

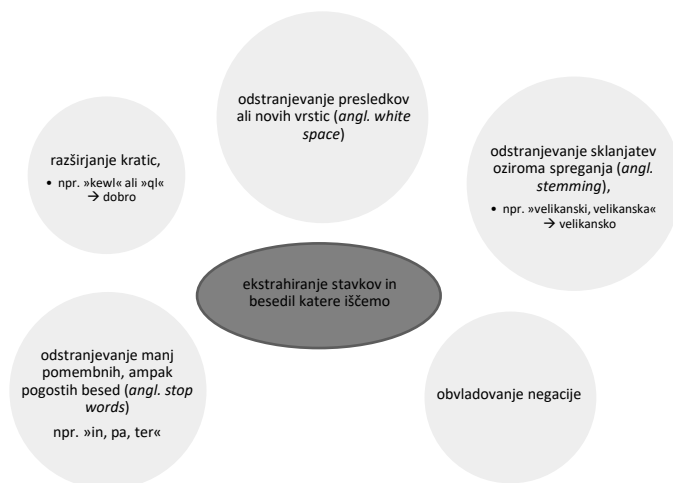
stavčno besedilo oziroma dokumenti prej uredijo, prečistijo in filtrirajo glede na zelene pogoje. Fazi sta poimenovani klasifikacija in obdelava podatkov.

2.4.4 Faza obdelave podatkov

Gre za proces čiščenja in priprave besedila za kasnejšo fazo klasifikacije sentimenta (Haddia in drugi 2013, 2). Spletna besedila običajno vsebujejo ogromno šuma nepomembnih delov, kot so html značke ali skripte. Poleg tega, če je besedilo daljše, ga je treba razčleniti na posamezne stavke (na primeru sporočil iz Twitterja to navadno ni potrebno, saj celotno besedilo zajame 140 znakov). Na nivoju besed jih veliko preprosto nima vpliva na splošno orientacijo pomena pri klasifikaciji. Primeri so url povezave, simboli in podobno (če odvezemo negacijo, sarkazem in cinizem, kar pa se že lahko nanaša tudi na kasnejšo fazo klasifikacije). Če pustimo takšne besede neočiščene, ustvarjamo problem (pre)visoke dimenzionalnosti podatkov, kar običajno privede do tega, da je naša klasifikacija časovno manj učinkovita in prostorsko potratna. Celoten proces običajno vsebuje naslednje korake čiščenja besedila (prav tam):

1. odstranjevanje presledkov ali novih vrstic (*angl. white space*),
2. razširjanje kratic,
 - npr. »kewl« ali »ql« → dobro
3. odstranjevanje sklanjatev oziroma spreganja (*angl. stemming*),
 - npr. »velikanski, velikanska« → velikansko
4. odstranjevanje manj pomembnih, ampak pogostih besed (*angl. stop words*),
 - »in, pa, ter«
5. obvladovanje negacije,
6. ekstrahiranje stavkov in besedil, ki jih iščemo

Slika 2.6: Skica obdelave podatkov pred fazo klasifikacije



Vsi ti koraki, razen zadnjega so del transformacije, medtem ko se zadnji imenuje filtracija (prav tam).

Dodajmo, da je v primeru *našega slovenskega algoritma* omembe vredno tudi to, da omenjenim postopkom dodamo še zamenjavo šumnikov s sičniki, predvsem zaradi nedosledne rabe uporabnikov, ki pišejo besedila. V fazi obdelave besedil za potrebe redukcije podatkov naš algoritem pobriše še vse številke. Poleg tega algoritem ne obvladuje sklanjanja ali spreganja (*angl. stemming*), saj v R-ju za zdaj še ni implementacije takšnega algoritma za slovenski jezik. V ta namen je zato v fazi klasifikacije (glej naslednje poglavje) izdelan »razširjen« slovar besed AFINN (Martinc 2013), v katerem se pojavljajo skoraj vse možne variacije besed na osnovi spreganja in sklanjanja. Našima slovarjema AFINN in AFINN rangi pa smo pripeli tudi angleški verziji. Obvladovanje negacije stavkov, tako imenovanega cinizma ali zanikanja naš algoritem v fazi obdelave ne obvladuje. Možno obvladovanje negacije bi se dalo izvesti v primeru kasnejše faze klasifikacije s pristopom strojnega učenja. Več o izdelavi in implementaciji algoritma v poglavju 3.1.

2.4.5 Faza klasifikacije

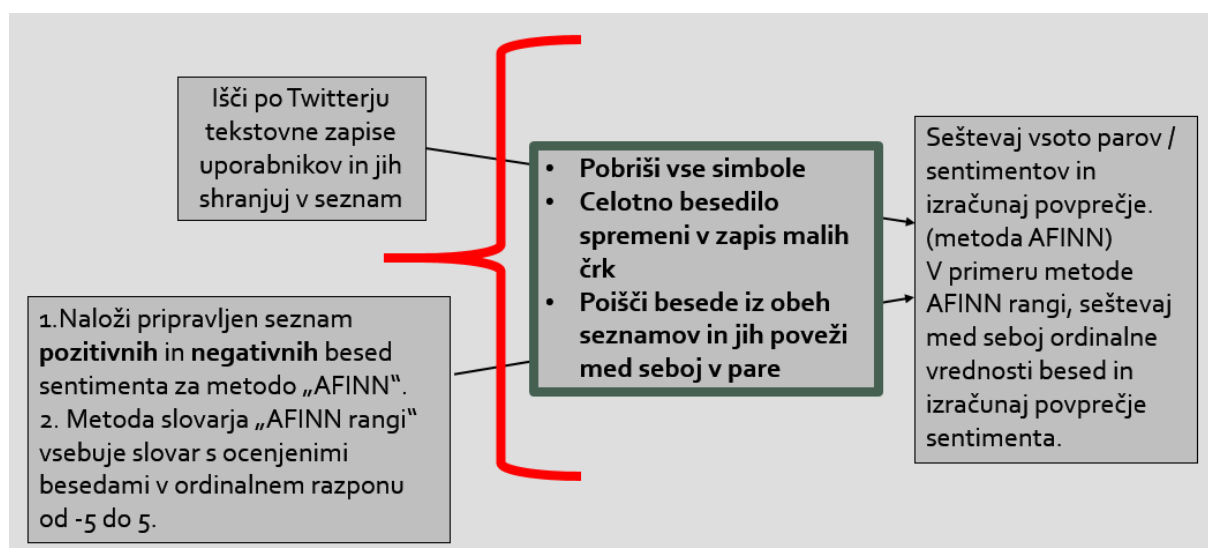
V fazi klasifikacije se pri raziskovanju računskega sentimenta uporabljata dve metodi: pristop s slovarjem in pristop s strojnimi učenjem.

2.4.6 Pristop s pomočjo slovarja

Analiza na nivoju slovarja (Nielsen 2011b) je metoda z ustvarjenima vektorjema pozitivnih in negativnih besed. Algoritem ocenjevanja sentimenta je podrobneje orisan v

shemi spodaj (slika 7). Pri tem algoritmu gre pravzaprav za merjenje polaritete sentimenta na ravni besed oziroma za iskanje besed iz slovarjev AFINN in AFINN rangi v preučevanih dokumentih, ki jim želimo izračunati sentiment (Nielsen 2011a, 1). Če algoritem najde ujemajočo se besedo iz pozitivnega ali negativnega slovarja besed, se stavku, v katerem se nahaja ta beseda, prišteje kumulativna vrednost +1 ali -1, odvisno seveda, iz katerega slovarja iščemo. Vsota (običajno) ali povprečje pa predstavlja skupno vrednost stavka ali besedila.

Slika 2.7: Poenostavljen proces ocenjevanja sentimenta z AFINN slovarji na primeru Twitterja



Ustvarjenih je bilo že precej slovarjev (Hu in Liu 2004; Nielsen 2011a; SentiWordNet 2013; MPQA 2014; Harvard University 2015; Potts 2015). Eden prvih je zgoraj omenjeni slovar AFINN, za katerega obstaja tudi slovenska verzija AFINN slovarja z rangi (Martinc 2013). Slovar z rangi poleg besede vsebuje še ocenjeno ordinalno vrednost v razponu med -5 in 5. Pri metodi sentimenta s takšnim slovarjem seštejemo pripadajoče vrednosti in izračunamo željeno vsoto (ali povprečje) v določenem časovnem razponu (npr. vsota v dnevu).

Drugi slovarji se razlikujejo po številu besed, ostali spet po smiselnosti in pomenu. Obstajajo pa tudi slovarji za štetje čustvenih besed v besedilih (jeza, žalost, veselje ipd.), pri katerih nato z metodo sentimenta izračunamo posamezna čustvena stanja besedil (Saif 2015).

2.4.7 Problemi z analizo sentimenta in omejitve

Kolega Martinec pravi: *»Klasična težava analize sentimenta s slovarjem za iskanje polaritete o objektu, ki mu želimo meriti polarnost (določeni temi), je, da v širšem tekstu lahko obstaja več objektov/subjektov, do katerih se izraža mnenje. Prav tako so v tekstu lahko informacije, ki se ne nanašajo na opazovani objekt/subjekt, a kljub temu prispevajo h končni oceni sentimenta.«* (Martinc 2013) Večji problem npr. na Twitterju lahko predstavljajo umetna mnenja oziroma t. i. »opinion spam« medijskih agencij, kjer plačane osebe ali organizacije namerno popačijo ali tvorijo vsebino medijskega prostora oz. spletne kampanje.

Kot dodatno omenimo še probleme sarkazma, ko slovnična struktura vsebuje samo pozitivne besede v stavku (npr. »zelo dobro si skuhal kosilo«, opomba: v resnici je bilo kosilo slabo). *»Nadalje težave predstavlja tudi zanikanje. Zaradi negacije v povedih bi morala biti realna ocena sentimenta manjša ali celo obratna. A ker gledamo samo posamezne besede, te negacije ne opazimo in ocena polaritete sentimenta je tako napačna.«* (Martinc 2013)

Področje analize podatkov sentimenta je hitro razvijajoče se polje. Pojavljajo se algoritmi s strojnim učenjem, ki se s pomočjo globokega učenja (*angl. deep learning*) zmorejo naučiti zaznavo sentimenta znotraj stavčnega konteksta. Takšen primer je npr. leta 2013 »pompozno« predstavila univerza Stanford s pomočjo algoritma nevronske mreže (Abate 2013). V splošnem se je takšen algoritem zmožen »sam« naučiti tipičnih stavčnih struktur in reševati probleme zanikanja. Kljub temu obstaja več statističnih algoritmov oz. pristopov s strojnim učenjem.

2.4.8 Pristop s strojnim učenjem

Pristop strojnega učenja se opira na zaznamovanje človeškega kodiranja besedil, npr. kot pozitivna, negativna, nevtralna (tako imenovano kategoriziranje stavkov, s katerim ustvarimo vektor, ki ga deklariramo za odvisno spremenljivko v modelu). 70 odstotkov podatkov se kasneje uporabi za fazo, ko »treniramo« algoritem oz. skušamo prilegati podatke modelu. Statistični algoritem nato pregleda podobne značilnosti novega besedila, da lahko napove njegovo polariteto (Pak in Paroubek 2010 v Gunter in drugi 2014). Pristop uporabi korpus ali drug podatkovni tip, ki vsebuje kodirane sezname besed za polarnost in nato identificira njihovo ponavljanje v povedih, ki so bile že obdelane v prejšnji fazi obdelave podatkov (Patil in drugi 2014). Temu sledi še testni set podatkov in na koncu set za preverjanje veljavnosti, kjer podatke po navadi razdelimo s deleži 70/15/15 (Williams 2011).

Te tehnike se uporabljajo za izgradnjo modela, izboljšanje učinkovitosti (prileganja modela podatkom) s pomočjo podanih parametrov.

Kot ugotavljajo nekateri raziskovalci (Patil in drugi 2014), se med najbolj učinkovitimi algoritmi za analizo sentimenta besedil pojavlja metoda podpornih vektorjev oziroma algoritem SVM (*angl. support vector machines*). Konceptualno je ideja algoritma SVM zasnovana tako, da v osnovi klasificira oz. ločuje točke v visoko razsežnem prostoru, tam pa se išče optimalno n-dimenzionalno hiper ravnino (*angl. hyperplane*), ki ločuje oz. maksimira prostor med takšnimi točkami (Patil in drugi 2014, 2).

2.5 Socialno omrežje Twitter in sentiment

Z analizo sentimenta na Twitterju lahko določimo podobo posamezne znamke ali pa celo primerjamo med seboj ugled primerljivih, tekmujočih znamk v kategoriji. Omrežje Twitter je občutljivo na ekonomske, politične in kulturne dogodke, ki jih je po poročanju raziskovalcev (Dietmar 2014) mogoče učinkovito meriti. Ko operiramo s takšnim kontekstom, pravi Dietmar, se pod vprašanje postavlja veljavnost hipoteze o učinkovitosti odprtih trgov (Investopedia 2015), delnic in mednarodnih menjalnih tečajev (Dietmar 2014). Ta hipoteza domneva, da so v vrednosti tečajev že vpete vse javno dostopne informacije. Lahko Twitter uporabniki, ki prostovoljno »debatirajo« o družbeno pomembnih temah, vplivajo na družbene dogodke? Ali morda bolje zastavljeno, koliko so njihova mnenja dejanski odraz »stanja v družbi«? Raziskovalci so že pokazali da lahko sentiment na Twitterju uporabimo za napovedovanje gibanja cene delnic (Johan in drugi 2010), napovedovanje ocene IMDB filmov (Oghina in drugi 2012), napovedovanje cen menjalnih tečajev (Dietmar 2014) ali pa celo za informiranje o širjenju bolezni prašičje gripe (Signorini in drugi 2015).

Socialno omrežje Twitter je hitro in globalno komunikacijsko omrežje, ki v zadnjih letih pridobiva ogromno pozornosti raziskovalcev (Takahashi in drugi 2011) ter s tem po poročanju raziskovalcev spreminja družbene pogoje in posledično našo objektivno realnost, hkrati pa nudi nove poslovne priložnosti.

2.5.6 Napovedovanje gibanja cene delnic

Napovedovanje gibanja cene delnic je v zadnjih letih pritegnilo veliko pozornosti tako akademskih kot tudi poslovnih krogov (Takahashi in drugi 2011). Ampak ali lahko trg delnic dejansko učinkovito napovemo?

Zgodnje raziskave napovedovanja gibanja cene delnic (H. Cootner in P. 1964) so temeljile na teoriji slučajne hoje (*angl. random walk theory*) in teoriji učinkovitosti trgov (*angl. efficient market hypothesis*) (Fama in E.F. 1965).

Po teoriji EMH je gibanje cene delnic v večji meri odvisno od informacij iz novic kot pa od trenutnih in preteklih vrednosti delnic. Ker so novice **nepredvidljive**, bodo vrednosti delnic, ki kotirajo na borzi, sledile vzorcu **slučajne hoje**, ki pa ne more biti predvidena z več kot 50-odstotno natančnostjo (Johan in drugi 2010).

Drugič, nedavne raziskave (prav tam) kažejo, da so novice lahko nepredvidljive, vendar pa je treba vzeti v zakup **zelo zgodnje kazalnike**, ki jih lahko izvlečemo iz spleta družbenih medijev (blogi, twitter itd.). Z njimi lahko napovedujemo spremembe gospodarskih in poslovnih kazalcev. Raziskovalci pravijo, da omrežje Twitter **je** občutljivo na ekonomske, politične in kulturne dogodke, ki jih je (Dietmar 2014) moč učinkovito meriti. Hipoteza o učinkovitosti trgov (Dietmar 2014; Investopedia 2015), delnic in mednarodnih menjalnih tečajev domneva, da so v vrednosti tečajev že vpete vse javno dostopne informacije. Po nekaterih študijah uporabniki Twitterja **lahko** vplivajo na družbene dogodke (Johan in drugi 2010; Oghina idr. 2012; Dietmar 2014).

Zato kljub vsemu lahko trdimo (teorija o učinkovitosti trgov), da novice zagotovo vplivajo na gibanje cene delnic. Iz psiholoških raziskav vemo, da čustva poleg informacij igrajo pomembno vlogo pri odločevanju (Damasio in A.R. 1994). Zato na naše odločitve znatno vplivajo čustva in naše razpoloženje. Upravičeno torej lahko sklepamo, da *javno razpoloženje in čustva* lahko vplivajo na vrednosti borznih tečajev in delnic enako kot novice. Javno mnenje oziroma razpoloženje (oziroma sentiment v nadaljevanju), ki ga bomo v tem raziskovalnem projektu zajemali preko omrežja Twitter, pri tem igra pomembno vlogo. To tezo podpira raziskava iz leta 2010 (Gilbert in drugi 2010), kjer so izluščili kazalnik javne anksioznosti. Iz objav revije LiveJournal so raziskovali, ali variacije lahko napovejo vrednosti S&P500 podjetij. Obstaja še nekaj raziskav, ki smo jih že omenili (poglavje 2.2.4), zagotovo pa je najbolj poznana prva spektakularna raziskava iz tujine »Twitter napoveduje trg delnic« (Johan in drugi 2010). V študiji so uporabljali orodje, ki analizira vsebine tekstovnih dokumentov sporočil uporabnikov, poslanih v določenem dnevu. Orodje meri pozitivna in negativna sporočila, poimenovali pa so ga OpinionFinder (prav tam). Drugo orodje, ki so ga uporabili, pa se imenuje GPOMS. Orodje analizira vsebine sporočil ter jim določi čustvena

stanja. Ustvari se 6-dimenzionalna dnevna časovna serija sentimenta, ki pripomore k natančnejšemu vpogledu sprememb skozi variabilnost različnih čustvenih dimenzij. Ocenjene sentimente in čustvene dimenzije so nato povezali z borznim indeksom Dow Jones Industrial Average. Ocenili so visoko zmožnost napovedovanja sprememb (variabilnosti uradnega tečaja) delnic skozi čas (Johan in drugi 2010).

Kot smo omenili v uvodu, je naš cilj proučevati, kako sentiment vpliva na gibanje tečajev delnic v Sloveniji. Zanimalo nas bo, ali je izračunani sentiment v časovnih točkah enega dne povezan z vrednostmi tečajev delnic (indeksa). Za primerjavo podobnosti takšnih dveh krivulj bomo potrebovali zanesljive, prilagodljive in zelo **zgodnje kazalnike ocene sentimenta**. Kot pravijo prej omenjene študije, to v praksi pomeni, da bomo morali v relativno kratkem ali **hitrem** časovnem obsegu zaznati povezanost spremembe v tečaju z vrednostjo (spremembo oziroma *diferenciranimi časovnimi vrstami*) sentimenta. Sklepali bomo, da sentiment povezuje vrednost delnice na podlagi časovnih korelacij v obdobju do maksimalno 5 delovnih dni. Če bi torej v našem modelu na primer dobili statistično značilno korelacijo med povezanostjo (vplivom) sentimenta na tečaj delnice, ki bi bila daljša od našega maksimalnega časovnega obdobja, bi enostavno morali sklepati, da sentiment ni povezan oz. ne vpliva na vrednost (spremembo) tečaja delnice. Predvsem zaradi teorije, kot smo v teoretičnem delu že razpravljali, saj potrebujemo za to zgodnje kazalnike. Več o tehnikah izpeljav, zakaj bomo to sploh počeli in o metodah analize bomo še razpravljali v nadaljevanju empiričnega dela.

3 EMPIRIČNI DEL

3.1 Merjenje sentimenta in zajemanje podatkov

Preden se bomo lotili analize podatkovnega rudarjenja in sentimenta, se bomo oprli na problem rudarjenja podatkov in *ideje* o izdelavi programa za »masovne podatke« na namiznem računalniku. Naš namen je bil izdelati in optimizirati hiter R algoritem za zajemanje, zbiranje in računanje sentimenta, da bi napovedali oz. bolje ugotovili povezanost gibanja cene delnic na slovenski borzi SBITOP in slovenskega borznega indeksa. Zato bomo v naslednjem podpoglavju najprej kritično evalvirali naš pristop in ga opisali.

Pri merjenju sentimenta smo se odločili za pristop s slovarjem besed. Izbrali smo metodo AFINN in AFINN rangi (glej poglavji 2.4.6 in 2.4.8). Za pristop s strojnim učenjem bi morali ročno zaznamovati polarnost posameznih besedil uporabnikov (vsaj 10 do 50 tisoč), kar pa bi vzelo preveč časa za potrebe tega magistrskega dela. Druga možnost bi bila, da bi skušali avtomatično ekstrahirati besedila in njihove označbe s pomočjo algoritma za luščenje, žal pa se tudi v takšnem primeru poraja problem pomanjkanja podatkov v slovenskem prostoru. Npr. spletni portali, kjer je podana ocena produkta (od 1 do 5), zraven pa komentar uporabnika, kjer bi izluščili stavek, v katerem bi iskali omembe subjekta/objekta do katerega merimo sentiment.

Širše gledano, nam je celotna izdelava programa, ki smo ga dopolnjevali in izpopolnjevali po potrebah, vzela približno 6 do 9 mesecev dela. Kako smo se lotili problema zasnove aplikacije oziroma izbire programske infrastrukture?

3.1.1 O izdelavi in implementaciji programa

Podatkovno rudarjenje lahko izvajamo preko raznih t. i. pajkov ali orodij za luščenje vsebin (*angl. web scrapping*). Obstajajo računalniški algoritmi, ki prečešajo omrežje in zbirajo relevantno besedilo. Druga možnost pa je z namenskimi API-ji (Profitt 2013), ki skrbijo za povezave med platformami in podatkovnimi viri. Zato se v osnovi naš problem opira na **tok** dveh podatkovnih virov. Eden je zajem podatkov preko Twitterja, drugi pa zajem podatkov slovenskih delnic z borze. Medtem ko zajem podatkov preko Twitterja že ima izdelan API, preko katerega lahko programsko zahtevamo sprejemanje s pomočjo ukazov knjižnice TwitteR (Gentry 2015), je drugi problem bolj specifičen, saj slovenska borza nima takšnega API-ja. Zato smo ustvarili algoritem s pomočjo knjižnice Rvest (Wickham 2015), ki omogoča luščenje podatkov.

Ker je zajem podatkov in avtomatizacija analiz na dolgi rok časovno in računsko potraten način, smo se na začetku osredotočili na problem hitrosti, saj smo želeli dobiti ocene analiz v skoraj realnem času. Zato smo se osredotočili na strategijo izdelave programa od zgoraj navzdol (*angl. top-down*), kodirno tabelo, kompatibilno z virom podatkov, in nato enako pri branju. Moramo sicer omeniti, da (vsaj trenutna) količina podatkov na slovenskem prostoru Twitter ne zahteva takega pristopa, kljub temu pa smo se za to odločili, ker smo imeli cilj spoznavanja teh tehnologij, hkrati pa smo želeli napisati algoritem, ki bi bil zmožen zajemati podatke npr. tudi v drugih državah.

Raziskali smo R knjižnice za visoko zmogljivo računanje (*angl. High performance computing*) (Cran HPC 2015) in ostale programske možnosti R ekosistema. Na koncu smo se odločili, da bomo celoten program napisali s knjižnico `data.table` (Dowle 2015), saj po ocenah avtorjev knjižnice in po prebiranju komentarjev uporabnikov portala `stackoverflow.com` omogoča najhitrejšje pristope za obdelavo, agregacijo podatkov v delovnem pomnilniku in je tudi 100-krat hitrejša od standardne R podatkovne strukture `data.frame`. Hitrejša naj bi bila celo od Python knjižnice Pandas (Dowle 2015). Bistvena lastnost pristopa s takšno podatkovno strukturo je ustvarjanje kopij z referenco in avtomatično ustvarjanje indeksa v tabelah po ključih, s katerimi nato lahko hitreje uporabljamo elemente za zbiranje unikatnih vrstic v tabelah. S pomočjo ključev pa lahko tudi hitreje urejamo stolpce ali vrstice in iščemo

elemente z avtomatskim redix ali binarnim algoritmom (prav tam). Ostale alternativne izbire so predstavljale podatkovne strukture v obliki zapisa korpusa, `data.frame` ali »listov«, za katere je bilo že v času odločanja o izbiri programske infrastrukture kar veliko izdelanih odprtokodnih rešitev (npr. na GitHubu), ampak se zaradi izziva nismo odločili zanje. V času pisanja algoritma se je pojavila še ena zanimiva knjižnica *quanteda*, ki deluje podobno kot korpus, s tem da je zgrajena na hitrejši `data.table` podatkovni strukturi.

Kasneje se je izkazalo, da je bila pomanjkljivost izbire našega pristopa dejstvo, da se je bilo treba na novo naučiti sintakso R programiranja z `data.table` podatkovnimi strukturami ter arhitekturo in logiko hitrejšega obdelovanja podatkov v pomnilniku. Dodaten problem je predstavljala tudi zelo pomembna pomanjkljivost, da v času izbire podatkovnih struktur, s katerimi bi implementirali algoritem, nismo poznali načina, kako obiti problem *ekstrakcije podvojenih vrstic pri pisanju/branju* s trdega diska. S tem problemom so se takrat »spopadali« tudi ostali uporabniki (glede na raziskane komentarje na internetu), saj podatkovna struktura v nekaterih primerih ni hotela v celoti rekodirati kodirne tabele znakov (npr. ukaz »`enc2native()`«). Ta problem smo reševali skoraj 3 tedne, ko smo končno ugotovili, da imata program R in operacijski sistem, na katerem poganjamo algoritmične operacije, težave pri kodiranju znakov, ko takšno tekstovno datoteko zapišemo in potem ponovno preberemo, saj nam takšna sistemska konfiguracija enostavno ni omogočala rekodirati zapisa znakov. Problem duplikatov smo nato rešili tako, da smo namesto zapisa kodirnih znakov (to so zapisi znakov v besedilnih datotekah, npr. `.txt`, `.csv`, in kodirnih formatih, npr. UTF-8, ASCII, Unicode ipd.) celotno datoteko shranili v formatu stisnjenega **binarnega zapisa** »`RDS`«. To je veljajo za verzijo 1.9.4 knjižnice `data.table`, v verziji 1.9.6 pa so pri prebiranju podatkov razvijalci omogočili opsijsko možnost »vsiljene« kodirne tabele.

Slabost pri rabi takšne podatkovne strukture je tudi to (kar smo spoznali šele kasneje), da se bistvena razlika v hitrosti pri takšnem pristopu začne poznati šele pri večjih količinah števila vrstic v objektu, to je (običajno) nad milijon vrstic (Dowle 2015). Takšen pristop pa v končni fazi zasede »še nekaj več pomnilnika« v računalniku, predvsem zaradi dodatno implementiranih algoritmov za iskanje elementov po ključih. Problem nastane pravzaprav pri omejitvi našega namiznega računalnika. Če bi imeli bistveno večje ali celo *hitrejš*e količine pomnilnika v računalniku, bi takšen pristop dejansko postal še učinkovitejši in hitrejši, tako pa predvidevamo, da je razlika v hitrosti manjša, kot bi lahko bila. Kljub temu je algoritem *teoretično* hitrejši (v razponu od mikro do nekaj sekund), hkrati pa je pripravljen za večje količine podatkov in deluje stabilno za analizo sentimenta v realnem času.

R-jeva omejitev pomnilnika nam je v poznejšem razvoju programa delala preglavice, zato smo iskali različne načine, kako rešiti problem. Na koncu smo s pomočjo knjižnice filehash (Peng 2015b) ustvarili preprost sekvenčni algoritem z RDS porazdelitvenim datotečnim sistemom shranjevanja podatkovnih zbirk podatkov. Napisan je z namenom redukcije podatkov (čiščenje besedila, ocena sentimenta ter avtomatične ekstrakcije besedil, ki se nanašajo na omembe slovenskih podjetij), poleg tega pa je možno vse takšne dele nato kasneje ponovno združiti oziroma sestaviti. Npr. za funkcijo, ki preišče celotno prečiščeno bazo besedila za poljubnim »regex« iskalnim vzorcem, poišče vektor tweet ID-jev z originalnim besedilnim atributom in jih nato združi s tabelo z že izračunanim sentimentom. (za natančnejšo razlago glej od poglavja od 3.1.3 naprej)

Bistveno je, da je celotna infrastruktura programa ustvarjena za namizni računalnik z dvema gigabajtoma pomnilnika. Velikosti RDS delov so omejene na 20 MB, kar v praksi pomeni 300.000 vrstic in okoli 100 Mb zasedenega prostora v pomnilniku računalnika. Celoten sekvenčni programski proces pri polni obremenitvi pa zasede največ ~ 700Mb. V nadaljevanju bomo razložil koncept API-ja in pregledali tehnične lastnosti Twitterjevega API-ja, kasneje pa tehnično razložili specifične procese zajemanja podatkov in shematsko orisali diagram procesa aplikacije.

3.1.2 Twitterjev programski vmesnik (API) in R

Programski vmesnik, krajše API (*angl. Application Programming Interface*), skrbi za povezavo med različnimi platformami. V splošnem API omogoča enostavne ukaze, s katerimi lahko zahtevamo podatke iz podatkovnih baz ali njihovo procesiranje v »oblaku« omreženih računskih gruč. V kategoriji računalniškega programiranja programski vmesnik (API) predstavlja sklop rutin, protokolov in orodij za gradnjo programskih aplikacij (Roos 2015). API izraža programsko komponento v smislu njene dejavnosti, vhodov, izhodov in z njimi povezanih zvrsti.

Twitter API 1.1 ima po referenci spletne strani 140dev.com (140dev) dva (pod)API-ja. Prvi skrbi za vpogled v preteklost (imenuje se »Search API«), drugi pa za sedanost (imenuje se »Streaming API«). Tisti v preteklosti lahko dostopa do razpoložljivih podatkov do enega tedna v zgodovino, drugi pa je namenjen pridobivanju trenutnih, aktualnih tweetov v realnem času. Podatkovna struktura je v obeh primerih enaka, v vsakem primeru API vrne podatke v obliki JSON podatkovne strukture. Razlika med obema – kakor navaja avtor – pa je v iskalnih nizih, saj omogoča prvi bolj natančen zajem podatkov s svojimi poizvedbami (*angl. queries*),

drugi pa omogoča le iskani niz in opsijsko geolokacijo žiga. Količinski zajem je večji pri API-ju v sedanjiku, kljub temu pa ostajajo špekulacije, koliko tweetov je možno v splošnem sploh zajemati, saj tega podatka iz tehnične dokumentacije ni mogoče razvideti (Twitter Developers).

Omenimo še, da sta v programskem jeziku R na voljo dve knjižnici. StreamR je paketek za sprejemanje podatkov v »sedanjiku« (Barberá 2015), TwitteR pa zadošča za potrebe rudarjenja v »pretekliku« (Gentry 2015). TwitteR vsebuje ukaz *getCurrentRateInfo()*, s katerim, lahko izvemo, koliko podatkov lahko v danem času sprejmemo. Odvisno od količine klicev API-ju, se prvotna vrednost ponastavi v približno 15 minutah.

3.1.3 Shema diagrama poteka aplikacije in razlaga MapReduce algoritma

Programski proces je razložen s postopkovnim diagramom v shemi diagrama poteka aplikacije (glej sliko 8), kjer je shematsko orisana implementacija celotnega programa. Predvsem zaradi omejitve prostora bomo razložili le procese zajemanja in obdelave podatkov.

3.1.4 Luščenje vsebin in pridobivanje podatkov preko API-jev

Algoritem na vsakih 24 ur avtomatsko generira in pridobi podatke s tabel delnic na spletnih straneh slovenske borze. S pomočjo označenih CSS značk HTML dokumenta izlušči podatke, ki se nahajajo v tabelah. Podatke dodatno kodira v format UTF-8, jih uredi in shrani. Podatki se nato zapišejo v podatkovno strukturo *data.table*, unikatne vrstice pa se združi z glavno tabelo, kjer se jih v *.csv* formatu zapiše na trdi disk.

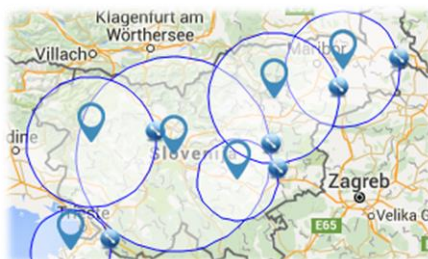
Celotna aplikacija sekvenčno zajema podatke s Twitterjevega API-ja iz treh virov. Gre za sporočila, ki prihajajo z geolociranim žigom, z list in od uporabnikov. Ti trije načini zbiranja podatkov na 15 minut producirajo od približno 6.000 do 18.000 tvitov, ti pa se shranijo v RDS datoteko, ki jo bomo poimenovali **pripona**. Pripona se imenuje zato, ker jo program po pretečenem času »pripne« v porazdeljeni datotečni sistem (glej poglavje 3.1.5) oziroma porazdeljeno datotečno tabelo. Vsi takšni zajeti tviti seveda **niso** unikatni, dejansko število unikatov pa zavisi od različnih dejavnikov, ki so pogojeni s posamičnim zajemom implementiranih funkcij, ter končnega primerjanja pripone s porazdeljeno datotečno tabelo. Takšni podatki se torej shranjujejo v pripono s pomočjo posebej napisanih funkcij, ki jih bomo na kratko razložili v nadaljevanju.

3.1.5 Geolocirani tviti

Geolocirani tviti prihajajo iz šestih slovenskih regij (glej sliko 8 na naslednji strani). Največ podatkov v povprečju dobimo iz osrednjeslovenske regije, sledita pa štajerska in primorska regija. Tu se ustvari namreč največ podvojenih podatkov, saj se regije med seboj prekrivajo, poleg tega pa algoritem vsakih 15 minut nadaljuje od zadnjega tvita v tabeli, kjer nam kljub temu vrne še visok delež prejšnjih tvitov.

Zajeti podatki, ki jih tako pridobimo, vsebujejo geolociran žig, žal pa kljub temu nekateri izmed njih takšnega žiga nimajo. Zato algoritem izbere samo tiste, ki žig imajo. Nato med njimi izbere slovenske uporabnike – tiste, ki »pišejo« slovensko, - ti pa se shranijo v tabelo slovenskih uporabnikov Twitterja. Pri tej metodi se iz slovarja daljših slovenskih pridevnikov išče ujemajoče se besede sporočil uporabnikov.

Slika 3.1: Geolocirani viri podatkov



Vir: freemaptools.com

Pri tej metodi iskanja slovenskih uporabnikov podrobneje omenimo problem, s katerim smo se srečali in ga potem kasneje rešili. Na začetku smo napisali algoritem, ki je potreboval pri velikosti objekta ~1500 tvitov približno 8 ur, da je v njem uspel poiskati ujemajoče se slovenske pridevnike uporabnikov. Pri tej rešitvi smo uporabljali »for, while in if« stavke. Ker takšen pristop ni zadostoval pogojem hitrosti, smo skušali algoritem pospešiti. Uspelo nam je čas 8 ur zreducirati na le nekaj sekund. Problem smo rešili s pomočjo razbijanja vektorjev in tabel ter s pomočjo regexa, kjer smo iskali ujemajoče se preslikave takšnih vektorjev. Tu nam je še posebej pomagala podatkovna struktura `data.table`, ki je ustvarjala kopije podatkov z referenco, poleg tega pa je optimizirana za operiranje s podatkovnimi tipi »znakov« (*angl. characters*) in nizi (*angl. string*). Postopkovni primer naše rešitve:

1. UVOZI TABELO UPORABNIKOV IN SPOROČIL

- **VEKTOR SPOROČIL RAZBIJ NA POSAMIČNE BESEDE**

- VSAKA VRSTICA V TABELI IMA SEDAJ VEKTOR POSAMIČNO LOČENIH BESED IN UPORABNIKOV**
- TRANSPONIRAJ TABELO NA VEKTOR BESED IN VEKTOR KOPIJ UPORABNIKOV S REFERENCO**
- IŠČI UJEMAJOČE SE BESEDE IZ PREBRANEGA SLOVARJA IN POSAMEZNE VRSTICE VEKTORJA BESED Z REGEXOM**

2. ZAPIŠI UJEMAJOČO VRSTICO UPORABNIKA, KJER OBSTAJA KARTEZIČNI PRODUKT UNIJE VEKTORJA SLOVARJA IN POSAMEZNE VRSTICE VEKTORJA BESED

Algoritem torej prebere besedila uporabnikov in s pomočjo ujemajočih se besed iz zgrajenega slovarja (v prilogi) poišče slovenske uporabnike. Hitrost je pridobljena s pomočjo podatkovne strukture `data.table`, kopij podatkov stolpca uporabnika s pomočjo referenc, regex iskanja ter funkcij družine `apply` (glej primer teh dveh algoritmov v prilogi).

Nadalje algoritem s pomočjo atributa »ID uporabnika« s funkcijo zapisa unikatnih uporabnikov zapiše/prepiše aktualne (zadnje) attribute uporabnika. Dodatno pa tem procesu sledi še funkcija, kjer se ročno briše uporabnike, pri katerih obstaja napaka pri delegaciji (npr. Hrvati in Srbi včasih uporabijo podobne si besede kot Slovenci).

3.1.6 Twitterjeve »liste« oziroma po meri ustvarjeni seznamami uporabnikov

Twitterjeve »liste« so ročni oziroma po meri ustvarjeni **seznamami uporabnikov**, ki jim sledi posamezen uporabnik na Twitterju. Sestavili smo tabelo uporabnikov, katerih seznane lahko najdete v prilogi. Najbolj zanimiv je gotovo seznam Jokerjevega mn3njalnika (mn3njalnik 2015), ki ima najobsežnejšo ročno zbirko »vseh« slovenskih uporabnikov. Preostali seznamami vsebujejo slovenske medije, novinarje, agencije in politike. Seznamami producirajo dnevna sporočila uporabnikov. Ta funkcija posredno shranjuje »Slovence« v tabelo slovenskih uporabnikov, saj jih prej primerja s tabelo nezaželenih uporabnikov.

3.1.7 Seznamami uporabnikov za časovnico

Zadnji vir vsebuje seznam uporabnikov za časovnico, pri katerih zajemamo podatke (do 3200 tvitov v preteklost) s Twitterjeva javnega uporabniškega zidu. V seznam uporabnikov smo dodali 20.000 slovenskih uporabnikov Jokerjevega mn3njalnika (mn3njalnik 2015), poleg tega pa se v to funkcijo hkrati zbirajo in dodajo **na novo najdeni** »Slovenci« iz funkcije geolociranih sporočil, ki jih išče. Razlika med listami in sezname uporabnikov je v tem, da omogočajo sezname pridobitev večjih količin sporočil uporabnikov iz **daljše preteklosti** (npr. več let v preteklost).

3.1.8 MapReduce algoritem in porazdeljeni datotečni sistem

Porazdeljeni datotečni sistem (DFS) se ustvarja po principu enostavnega sekvenčnega »MapReduce« algoritma (obširnejša razlaga je v teoretičnem delu). Datotečni deli so ustvarjeni s pomočjo »filehash« kazalcev (tudi ključev, vendar pa se enak termin uporablja tudi pri data.table podatkovni strukturi in med seboj ni zamenljiv), ki kažejo na vrednosti tabele in njenih atributov (*angl. key-value pairs*). Vrednosti tabele so vrstice z atributi tvit sporočil, kjer **velja pravilo**, da mora biti celotna »vrednost« takšne tabele dolga 300.000 vrstic. Algoritem na približno 15 min vzame tabelo poprej zajetih tweetov (**pripone**) in jih primerja z verigo vseh datotečnih delov podatkovne zbirke. To enostavno povedano pomeni, da prebere del tabele (ali 300.000 vrstic), ki je zapisana v datoteki. Vsaka vrstica pregleda unikatni ID ključ v »verižno« porazdeljeni tabeli »tweets«. Če se najde podvojena vrstica, se odstrani iz pripone. Ko postopkovni algoritem pride do konca datotečne verige, unikatne vrstice pripone združi z zadnjim delom porazdeljene datotečne tabele. Če zadnji del verige preseže 300.000 unikatnih vrstic, se ustvari nov del tabele (in tako naprej). Postopkovnemu procesu lahko do tu rečemo »**Map**«, kjer smo s pomočjo kazalcev po delih prebrali tabelo tvitov in na koncu pripeli preostanek vrstične pripone. Vsak »tweet ID« pripone se torej posebej primerja s celotnim datotečnim sistemom. Kot opombo naj dodamo, da smo preverjanje unikatnih vrstic *celotnega* DFS naredili zato, ker funkcija seznamov uporabnikov zajema podatke iz daljše preteklosti (npr. sedem let v preteklost). Če bi funkcijo zajemanja podatkov iz dolge preteklosti npr. preprosto onemogočili, bi algoritem dodatno pospešili tako, da se preverja npr. samo zadnjih nekaj blokov DFS, priponi pa bi dodali pravilo časovnega preverjanja do maksimalno dovoljenega časa v preteklost.

Temu sledi prečiščen vektor besedil uporabnikov in dodaten ID vektor uporabnika (razlaga v poglavju 2.4.4), ki ga algoritem ustvari iz zadnjega dela tabele tvitov. Novo tabelo zapiše v DFS tabelo »clean_tweets«. Končno s takšno tabelo izračuna sentiment s pomočjo

dveh slovarjev besed AFINN in AFINN rangi (razlaga v poglavjih 2.4.6 in 2.4.8), to pa zapiše v tretjo DFS tabelo »sentiment_tweets« z vektorjema ID-ja uporabnika, časovnim žigom in izračunanim sentimentoma za obe metodi.

Algoritem nato izvrši iskanje s pomočjo »regex« sintakse. Regex je nastavljen tako, da iz dveh datotek prebere napisane iskalne pogoje. Ena datoteka služi iskanim pogojem (pozitivne ključne besede), druga pa pogojem, kjer se izvede iskalno filtriranje stavkov, ki jih ne želimo zajeti z iskalno poizvedbo. Zaradi smiselnosti se tu iščejo omembe podjetij, do katerih uporabniki izražajo sentiment (in ne samo omembe delnic). Takšnih sporočil je namreč več kot sporočil o vrednosti tečajev delnic. Hkrati pa pri takšnih sporočilih algoritem *izloča* tiste, ki izražajo sentiment do športnih ekip, dogodkov in ostalih stavčnih struktur, pri katerih se ne izraža sentiment do podjetij neposredno. Npr. stavek »Tako lep razgled je s Triglava« ne izraža sentimenta do Zavarovalnice Triglav. Pravilno ujemajoče se omembe delnic algoritem nato ločeno zapiše v dve datoteki, pri katerih zajame potrebne attribute za izračunanje sentimenta in uporabnikov ter jih zapiše v datoteke glede na ime posamezne delnice. Vsem preostalim korakom do tu rečemo »**Reduce**«. Kot že samo ime namiguje, je bistvo vseh teh faz redukcija podatkov na manjše in časovno bolj obvladljive dele. Če bi se problema lotili drugače, torej da bi namesto ustvarjanja treh porazdeljenih tabel enostavno stolpce atributov dodali glavni tabeli, bi s tem ustvarili večje količine podatkov, ki pa so iz vidika časovne optimizacije nepotrebni. Npr. napisana je funkcija za dodatno regex iskanje besed, ki bi jih morda naknadno potrebovali. Npr. za primer merjenja javnega mnenja do politikov ali strank. V tem primeru se podtabelo »clean_tweets« uporabi za iskanje besed v prečiščenem vektorju besedil. Takšna tabela nam tako nudi učinkovito ogrodje, po katerem zahtevamo vektor IDjev, hkrati pa se znebimo koraka ponovnega čiščenja besedil. Ko nam takšno iskanje besedil vrne vektor ID rezultatov, se jih enostavno združi s tretjo porazdeljeno tabelo »sentiment_tweets«. Bistvo tega je časovna optimizacija, saj nam za takšen primer ni treba ponovno izračunavati sentimenta. Prav tako je velikost zadnje datoteke DFS ocenjenega sentimenta za več kot 4-krat manjša od glavne tabele, medtem ko je tabela prečiščenih besedil za 2krat manjša od glavne tabele. S takšnimi velikostmi pa je potem možno učinkoviteje in računsko hitreje upravljati pri masovnih količinah podatkov.

Na koncu opisanega MapReduce algoritma se s pomočjo shiny knjižnice in ustvarjene namenske spletne aplikacije izvede še analiza in prikaz grafov ter tabel v obliki dinamičnih HTML dokumentov. Celotna shema diagrama procesov je orisana na naslednji strani. Kot opombo naj dodamo, da lahko do vseh ustvarjenih tabel (programsko, preko R kode)

dostopamo z atributom ID-ja tvita ali pa z atributom ID-ja uporabnika skozi našo infrastrukturo zasnovano aplikacije.

Na naslednjih straneh je priložena (poenostavljena) shematska slika diagrama poteka aplikacije, sledita pa še dve tabeli, v katerih so opisane funkcije in podatkovne strukture programa.

Slika 3.2: Shema (poenostavljenega) diagrama poteka aplikacije

	slovenskih regijah s pomočjo zadnjega shranjenega IDja tvita (do 1500). Prebere/shrani zadnji tvit ID v tabelo DT_lastTwit_id.	getDT_tweetsTemp(), getDT_users(),
getVipListTimeline()	Izvede iskanje tvitov s pomočjo po meri ustvarjenih seznamov uporabnikov.	VipListCounter – celo število (<i>angl. integer</i>), VipUserList - tabela
getUsersTimeline()	S pomočjo ustvarjenih uporabniških ID-jev zajema tvite uporabnikov iz daljše preteklosti (do 3200)	DT_userTimelineIDs - tabela
makeUsertimelineIDs()	Ustvari posodobljen seznam uporabnikov za pridobivanje tvitov iz daljše preteklosti. Seznam se ustvari na vsake 3 mesece oz. takrat, ko getUsersTimeline() prebere celotno tabelo.	DT_UsertimelineIDs - tabela getUsersTimeline(), getDT_users(), getDT_VipUsers()
GenerateSBITOP()	Na vsakih 24 ur pridobi podatke za vseh 13 delnic s spletne strani www.ljse.si	Podatke shrani v tabele pod imenom: SBITOP_<ime_delnice>
append_unique_db()	Pripne pripono (DT_tweetsDaily) v porazdeljeni datotečni sistem in izvede procese <i>čiščenja</i> , <i>filtriranja</i> in <i>računanja sentimenta</i> .	getDT_tweetsTemp(), DT_tweetsDaily - tabela
manualUserFilter()	Odstrani uporabnike, ki niso Slovenci (»ročna« filtracija).	userBanData – tabela, extractGeoValidNames()
extractGeoValidNames()	Izbere uporabnike, ki imajo geolociran žig, ostale zavrne.	manualUserFilter(), getUsersFilter()
extractVerifiedNames()	Poišče Slovence – tiste, ki	extractGeoValidNames()

	pišejo slovensko.	
getUsersFilter()	Glavna funkcija za iskanje Slovencev.	extractGeoValidNames(), extractVerifiedNames()
getDT_tweetsTemp()	Pretvori JSON podatkovno strukturo v data.table. Unikatne podatke shrani v tabelo DT_tweetsDaily.	getGeoTweets(), getVipListTimeline(), getUsersTimeline()
getDT_users()	Iz geolociranih in iz tvitov uporabnikov pridobi uporabniške ID-je in iz njih ustvari unikatne vrstice DT_users. V primeru najdenega duplikata zapiše zadnje vrednosti pod atributi. (glej v tabeli 2: DT_users)	getUsersFilter(), manualUserFilter()
getDT_VipUsers()	Iz po meri ustvarjenih tvitov pridobi uporabniške ID-je in iz njih ustvari unikatne vrstice DT_users. V primeru najdenega duplikata zapiše zadnje vrednosti pod atributi. (glej v tabeli 2: DT_users)	manualUserFilter()

Tabela 3.2: Opis vsebine elementov programa – podatkovne strukture

Tabela (ali datoteka)	Opis	Podatkovna struktura; vektorji
DT_lastTwit_id	Tabela za pomoč funkciji getGeoTweets(), pri kateri program dobi zadnji tvit ID, od katerega zajemamo naprej.	data.table ; "region", "id", "created" – vsi tipa znakov (<i>angl. character</i>)
VipListCounter	Spremenljivka tipa »integer«, v kateri shranjujemo zadnjo vrstico	celo število (<i>angl. integer</i>)

	nahajanja pri prebiranju podatkov iz tabele VipUserList.	
VipUserList	Tabela po meri ustvarjenih seznamov uporabnikov (v prilogi).	data.table ; V1 ~ uporabniško ime , V2 ~ ime ustvarjene liste uporabnika - vsi tipa znakov
DT_users	Tabela slovenskih uporabnikov.	data.table ; "description", "statusesCount", "followersCount", "favoritesCount", "friendsCount", "url", "name", "created", "sysDateUserCreated" ³ , "protected", "verified", "screenName", "location" "lang", "id" ⁴ , "listedCount", "vip", "profileImageUrl" – vsi tipa znakov
DT_tweetsDaily	Tabela, v katero se shranjujejo tviti za kasnejše pripenjanje podatkov v porazdeljeni datotečni sistem.	data.table ; "text", "tweet_url", "favoriteCount", "replyToSN", "created", "replyToSID", "id" ⁴ , "replyToUID",

³ sysDateUserCreated označuje datum, ko je bila ustvarjena vrstica uporabnika. Program preko te vrednosti posodablja podatke v celicah z novjšimi.

⁴ »id« v obeh podatkovnih strukturah ni zamenljiv. DT_users »id« pripada imenu uporabnika, medtem ko DT_tweetsDaily »id« pripada specifičnemu tuitu.

		"screenName", "retweetCount", "longitude", "latitude" – vsi tipa znakov
DT_tweetsTemp	Tabela za začasno hranjenje, obdelavo in transformiranje podatkov iz JSON formata v data.table standard.	data.table, enaka struktura vektorjev kot prejšnja vrstica
DT_userTimelnIDs	Tabela, v kateri je seznam ID-jev za getUserTimelines(). S tabelo operira funkcija makeUsertimelineIDs().	data.table; id, screenName, sysDateUsrTimelnCreated ⁵ - vsi tipa znakov
userBanData	Tabela ročno zaznamovanih uporabnikov, ki se jih uporabi za filtriranje tabele DT_users.	Shranjena je v datoteki userBanList.csv, v kateri so zapisani IDji ločeni z vejico.
dictionary_AfinnRanks-sl-V2.rda	Datoteka, v kateri se nahaja sestavljen slovenski in angleški slovar AFINN in AFINN rangi.	Besede ; Vrednost Vrednost je ordinalna ocena posamične besede od -5 do 5. V primeru AFINN metode algoritem izbere besede glede na predznak in jih nato razvrsti v pozitivno ali negativno skupino.
slovar.txt	Datoteka, v kateri se nahajajo regex vzorci (pogoji) za iskanje	regex zapisi v formatu znakov

⁵ Označuje čas, ko je tvit »id« uporabnika bil zajet

	Slovencev.	
searchTerm_<ime_delnice> searchNegTerm_<ime_delnice>	Skupno 26 datotek iskalnih pogojev, zapisanih v regex sintaksi. Prva datoteka služi iskanju zahtevanih terminov o delnici. Druga datoteka služi iskanju terminov in stavčnih struktur, ki niso povezani s delnico.	regex zapisi v formatu znakov

3.2 Analiza

3.2.1 Opisni podatki

Opisne podatke smo v analizo statistik povzeli iz posameznih tabel:

- tabela slovenskih uporabnikov Twitterja (poglavje 3.2.2),
- 13 tabel slovenskih SBITOP podjetij s pripadajočimi atributi podatkov (3.2.3),
- glavna tabela tvitov porazdeljenega datotečnega sistema oz. porazdeljena datotečna tabela.

Vsi ti podatkih so zajeti od 25.4.2015 do **24.2.2016**. Med zajemanjem smo imeli občasne prekinitve toka podatkov. Včasih zaradi zunanjih dejavnikov (izpad elektrike, interneta), včasih pa zato, ker smo izboljševali algoritem zajema podatkov (npr. nadgradnja algoritma za zajemanje večjih količin podatkov ali pa zaradi pojava programskih hroščev). Kljub temu smo po naši oceni pridobili dovolj velik vzorec podatkov, teh bo v tej analizi 16.200.000 unikatnih vrstic sporočil, zajetih v slovenskem prostoru, slednji pa zasedajo približno 5,5 gigabajtov prostora na disku, ko podatke odkompresiramo. V njih je zajetih približno 20.000 slovenskih uporabnikov (mn3njalknik 2015), ki so kadar koli pošiljali tvite (do 3200 v daljšo preteklost), geolocirana sporočila in sporočila iz po meri ustvarjenih seznamov. Deležev iz posameznih metod zajema zaradi zasnove programa v tem trenutku ne moremo določiti.

3.2.2 Tabela slovenskih uporabnikov

Skozi časovno obdobje zajema podatkov smo pridobili 8.820 slovenskih uporabniških računov. Naj ponovno poudarimo, da so to uporabniki, pridobljeni iz avtomatsko prečiščenih geolociranih virov ter Twitterjevih list (brez seznamov uporabnikov za časovnico). Od tega smo ročno zaznamovali 204 uporabniške račune, za katere je jasno, da niso Slovenci. Za sestavo posameznih atributov oziroma v jeziku R vektorjev tabele, glej poglavje 3.1.8, tabelo 2 , vrstico DT_users.

V tabeli 3 je objavljenih top 100 uporabniških imen po številu sledilcev (padajoče po vektorju followersCount).

Tabela 3.3: Top 100 uporabniških računov

Pozicija: ime uporabnika		
1:Anze Kopitar	20:Bitstamp	40:Slaven A. Martinović
2:Goran Dragić	21:Janez Potočnik	41:Slavoj Žizek
3:Gregor Prajs	22:Časnik Delo	42:Jonas Žnidaršič
4:Tina Maze	23:Boštjan Nachbar	43:Boštjan Gorenc
5:Tanya Plibersek	24:Borut Pahor	44:Slovenia Tourism
6:UMEK	25:Nataša Pirc Musar	45:Maja Kocijančič
7:Aljaz Skorjanec	26:Dnevnik	46:dr. Miro Cerar
8:24ur.com	27:Outfit7	47:Andraz Tori
9:Sasha Vujacic	28:Zan Ilic	48:Finance.si
10:simon legen	29:DJ Sylvain	49:Tanja Fajon
11:Playboy Slovenia	30:PSE d.o.o.	50:Žurnal24
12:Jaka Lakovic	31:NK Maribor	51:Oddaja 24UR POP TV
13:Val 202	32:Masa	52:Jure Leskovec
14:MMC RTV Slovenija	33:Volodja Balzalorsky	53:Violeta Bulc
15:janibrajkovic	34:Katarina Cas	54:adoreanahi.
16:Zoran Dragic	35:Luka Doncic	55:Katarina Kresal
17:RTV Slovenija	36:Beno Udrih	56:Marko Crnkovič
18:Janez Janša	37:Mladina	57:Team Slovenia
19:Vlada R. Slovenije	38:Ziam	58:STA novice
	39:Slavoj Žižek	59:Akrapovič

60:Slavko Bobovnik	74:@Furion	88:Si.mobil
61:Peter Prevc	75:Planet Siol.net	89:Murekar Pavla
62:Večer	76:Edis T.	90:Monster Energy SI
63:US Embassy Ljubljana	77:Visit Ljubljana	91:Taksist Ibro
64:Miha Šalehar	78:Jaka Blazic	92:Tia hi JUSTIN hi
65:IEBergant.me	79:Marjan Grman	93:policija_si
66:roni kordis	80:Godler	94:SDS
67:SloveniaArabic	81:Studio City	95:Planet TV
68:Svet	82:Telekom Slovenije	96:teenage dirtbag
69:Nika Kljun	83:Fieldoo	97:Tadej Toš
70:Nogometna zveza Slo	84:Rene Krhin	98:<U+3164>
71:Blaž	85:emo j3\$u\$	99:Primož Brezec
72:Žiga Turk	86:2 punk rock 4 u	100:Damien Hirst
73:Flaviar	87:Ivan Pepelnjak	

3.2.3 Tabele zajetih tvitov na posamezno delnico

Zaradi učinkovitosti merjenja algoritma smo v nadaljevanju analize izmed vseh tabel vzeli podatke delnice Telekoma Slovenije, ki vsebuje 16.099 sporočil.

Preostale frekvence sporočil so posredovane v tabeli 4 (spodaj), a se v nadaljevanju na njih ne bomo ozirali zaradi premajhne količine podatkov ter poenostavljene analize.

Tabela 3.4: Tabela zajetih tvitov na posamezno delnico

Cinkarna Celje - CICG : 183	Gorenje - GRVG : 1344
Intereuropa - IEKG : 214	KD Group - KDHR : 38
Krka - KRKG : 1531	Luka Koper - LKPG : 1366
Mercator - MELR : 8158	Petrol - PETG : 3296
Pivovarna Laško - PILR : 465	SavaRe - POSR : 50
Sava - SAVA : 83	Telekom Slovenija - TLSG : 16099
Zavarovalnica Triglav - ZVTG : 1640	

3.3 Merjenje povezanosti

V naši analizi je postavljeno raziskovalno vprašanje; »Ali je sentiment povezan z vrednostjo tečaja delnice (oz. borznim indeksom)?«. S pregledom sekundarnih virov v teoretičnem delu smo preverjali načine, metode (algoritme) in zaključke podobnih raziskav do sedaj. Kako pa se odreže naš algoritem, bomo v nadaljevanju preverjali za delniški tečaj Telekoma Slovenije v povezavi s sentimentom do podjetja ter s slovenskim borznim indeksom SBITOP v povezavi s celotnim zajetim slovenskim javnim sentimentom omrežja

Twitter. Kljub temu, da domnevamo, da se naš »vzorec« zelo verjetno približa celostni slovenski Twitter populaciji, (za zdaj) ne moremo oceniti dejanske primerjave s slovensko Twitter populacijo. Zagotovo pa tu gre za nam največji poznani proučevani vzorec sporočil v Sloveniji do sedaj.

Za **analizo merjena podobnosti dveh diskretnih časovnih krivulj** bomo v nadaljevanju uporabili funkcijo časovnih korelacij (*angl. cross-correlation*), ki primerja vrednosti vektorjev v zamikih.

Matematična metoda je drugače poznana tudi kot drseči skalarni produkt, v R-ju pa je to funkcija `ccf()` paketa `stats` (Gilbert 2016). R-jeva funkcija izračuna časovne korelacije v zamikih po enotah, v našem primeru po zamikih enega dne. Vrednost časovnega zamika »k«, ki ga funkcija `ccf(x, y)` vrne, oceni moč korelacije med $x[t + k]$ in $y[t]$ (prav tam). Moč korelacije predstavlja standardizirano verjetnostno vrednost (p) med 1 in -1. Funkcija izriše graf; vrednosti v posameznih časovnih točkah na grafu pa predstavljajo pozitivno ali negativno korelacijo. Moč korelacije je definirana na Y osi takšnega grafa. Če gre posamezna vrednost **vizualno** skozi modro črto, rečemo, da je povezanost statistično značilna. Funkcija izračuna privzeto stopnjo gotovosti pri 95 %.⁶ Interval zaupanja izračunava na podlagi $0 \pm 1.96 / \sqrt{N}$, kjer N predstavlja število opazovanih časovnih enot.

V praksi pomeni, da iščemo, za koliko časovnih enot n moramo vektor podatkov g premakniti po osi x , da bomo našli najbolj podobno (največjo korelacijo) vektorju podatkov f (Halls-More 2015). Matematično pa torej pomeni, da formula (glej enačbo 3.1) v evklidskem vektorskem prostoru primerja vrednosti skalarnih produktov vektorja f v času 0 z vektorjem g v času minus n do plus n . Na koncu matematičnih operacij f in g izberemo maksimalen vektorski produkt, pri katerem rečemo, da sta si pri takšnem zamiku n , s standardizirano vrednostjo p , takšna dva vektorja med seboj najbolj podobna.

Enačba 3.1: Drseči skalarni produkt

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f * [m] g[m + n].$$

Za merjenje povezanosti (v nadaljevanju naše analize) bomo razložili *izpeljave diferenciranih časovnih vrst* za:

⁶ Če pogledamo izvorno kodo knjižnice `stats`, opazimo, da funkcija `ccf()` pošlje rezultate funkciji `acf.plot()`. Tam pa se med klici skriva »nevidna« (`ukaz invisible()`) funkcija knjižnice `stats`, ki izračuna takšen privzeti interval tveganja (privzeta stopnja intervala tveganja je torej 95 % - razvidno iz izvorne kode, argument »ci«).

* Trditev lahko preverimo z R-jevim ukazom: `getAnywhere(plot.acf)`

- spremembo tečaja/indeksa,
- spremembo sprememb tečaja/indeksa,
- spremembo sentimenta.

Takšne izpeljave nam bodo služile za **boljše** zaznavanje merjenja povezanosti med tečajem (indeksom) in sentimentom. Enostavna razlaga je pravzaprav transformacija diskretnih točk na grafu. Takšna transformacija pa omogoča, da v posameznih časovnih enotah dobimo jasnejšo sliko gibanja vrednosti (vrednosti naraščajo/padajo).

V ekonometriji in pri analizah finančnih modelov (Iordanova 2007) se pri uporabi nestacionarnih časovnih vrst podatkov (kot so to v našem primeru podatki vrednosti tečaja ali sentimenta, ti pa imajo običajno za seboj viden trend) izkaže, da ti po navadi ustvarijo problem nezanesljivosti (pri napovedovanju ali merjenju korelacij). Rezultati analiz pa so posledično lahko zavajajoči, ko ima »krivulja« diskretnih točk zaznan določen trend (npr. v nekem obdobju narašča). Rešitev problema je torej transformacija (nestacionarnih) podatkov časovnih vrst v stacionarne (prav tam). To naredimo zato, da se izognemo umetni korelaciji (*angl. spurious correlation*), ki lahko nastane, ko ima »krivulja« (diskretnih točk) zaznan trend. Podrobneje si torej oglejmo izpeljave diferenciranih časovnih enot.

Sprememba tečaja (ali indeksa) je preprosta razlika vrednosti uradnega tečaja ob zaključku (*angl. »Close«*) in ob začetku dneva.

Enačba 3.2: Sprememba tečaja

$$(Close - Open)_{Današnji\ dan}$$

Sprememba sprememb tečaja predstavlja razliko med spremembo uradnega tečaja ob zaključku in začetku dneva (*angl. »Open«*). Nato se enako izračuna razlika iz prejšnjega dne, na koncu pa razlike med seboj ponovno odštejemo. Sledi torej:

Enačba 3.3: Sprememba sprememb tečaja

$$(Close - Open)_{Današnji\ dan} - (Close - Open)_{Včerajšnji\ dan}$$

Sprememba sprememb indeksa bo v tem primeru podobna kot pri tečaju, s to razliko da bomo od vrednosti tečaja danes odšteli vrednost včeraj in nato od vrednosti tečaja včeraj odšteli vrednost pred dvema dnevoma. Na koncu pa bomo razlike med seboj ponovno odšteli. Sledi torej:

Enačba 3.4: Sprememba sprememb indeksa

$$(\text{indeks}_{\text{Današnji dan}} - \text{indeks}_{\text{Včerajšnji dan}}) - (\text{indeks}_{\text{Včerajšnji dan}} - \text{indeks}_{\text{Predvčerajšnji dan}})$$

Sprememba sentimenta, je preprosta razlika med vsoto sentimentov (po metodi AFINN ali AFINN rangi) današnjega in prejšnjega dne.

Enačba 3.5: Sprememba sentimenta

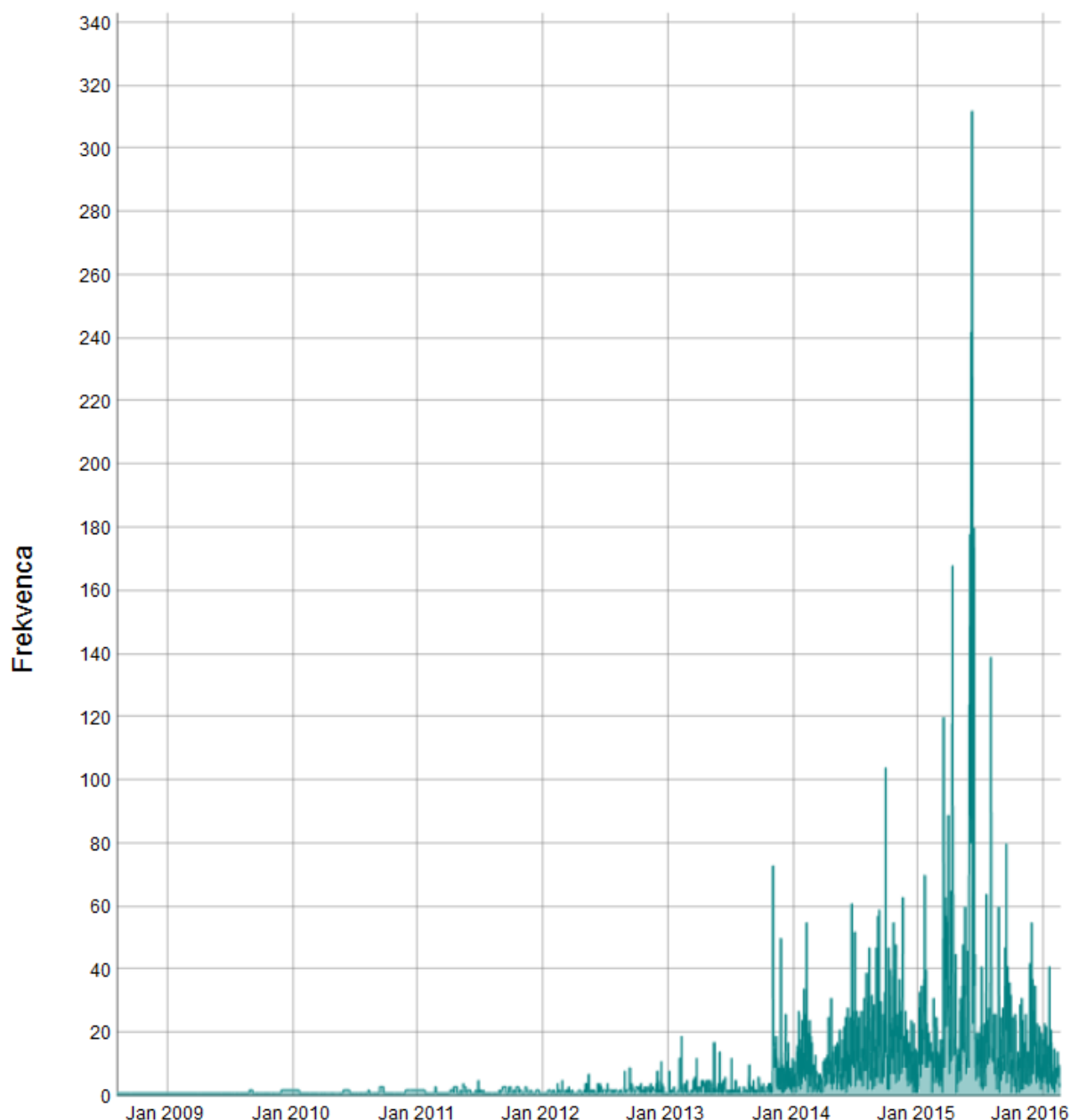
$$Vsota\ AFINN_{\text{Današnji dan}} - Vsota\ AFINN_{\text{Včerajšnji dan}}$$

V nadaljevanju bomo v analizi za izpise grafov uporabili knjižnico dygraphs. (DygraphsR). Gre za zelo uporaben način vizualizacije podatkov, ki vsebuje interaktiven in dinamičen način prikaza podatkov v HTML dokumentih, kjer se s pomočjo JavaScript-a in CSS-ja prikazuje željene podatke pod določenimi pogoji (približevanje/oddaljevanje, premikanje po X osi ipd.). Kot opombo dodajmo, da za zdaj knjižnica še ne vsebuje slovenskega prevoda, ima pa tudi težave s kodiranjem slovenskih znakov, zato bodo nekatere oznake brez šumnikov ali pa v angleščini.

3.3.1 Delnica Telekom Slovenija in sentiment

Poglejmo graf frekvenc tvitov (Slika 10) in podrobneje preverimo opisne statistike.

Slika 3.3: Graf frekvence sporočil za delnico Telekom Slovenija



Skupno število vseh sporočil v obdobju od 8. 8. 2008 do 24. 2. 2016 je 16.099. Od teh jih je bilo največ 312 dne 10. 6. 2015, kar predstavlja največjo frekvenco poslanih sporočil v enem dnevu. Povprečna vrednost vseh sporočil znaša 12,28 na dan (izračunano za vse dni, ko so poslana sporočila). Če vključimo tudi dni, ko ni bilo sporočil, je povprečna vrednost 5,8. Od 2008 do 2011 je ogromno manjkajočih enot. Glede na frekvenčno porazdelitev sporočil (glej slika 10) in smiselnosti analize se bomo v nadaljevanju osredotočili na časovni interval od 1. 1. 2014 do 24. 2. 2016, saj je razvidno (slika 10), da je padec frekvenc od leta 2014 v preteklost že kar drastičen.

To potrjujejo frekvence preteklih sporočil in opisna statistika za frekvence tvitov, starejše od leta 2014. Delež manjkajočih enot se tam začne tudi vidno povečevati (slika 10). Razlika med tema dvema intervalom je 1.564 tvitov. V obdobju do leta 2014 je povprečno število tvitov 0,8 na dan, vključujoč manjkajoče dni. Manjkajočih dni je kar 1.431 od 1.972. To trditev dodatno potrjuje tudi ročni pregled tabele, kjer se opazi, da se število sprememb vrednosti v celicah začne »spreminjati« za vsak dan posebej šele po letu 2014. Za potrebe lažjega vpogleda smo napisali skripto, ki dodatno preverja in izračunava opisne statistike frekvenc tvitov ter izriše histograme frekvenc tvitov za posamezne časovne intervale. Frekvenca vseh tvitov v preučevanem intervalu od 2014 naprej je 14.535. Povprečno število tvitov v tem obdobju je 18,9 na dan oziroma 18,5, če dodamo v izračun še manjkajoče dneve brez sporočil. Takšnih manjkajočih dni je le 13.

Za razlago posameznih točkovnih krivulj za potrebe pojasnitve grafov na slikah v nadaljevanju prikazujemo vrednosti tečajev (indeksa), sentimentov in izpeljav diferenciranih časovnih enot z angleškimi okrajšavami. Npr. »Close« pomeni uradni tečaj ob zaključku dneva, sentiment z algoritmom AFINN »AFINN_daySum«. Izpeljave diferenciranih časovnih enot pa se začnejo z »diff«. Na slikah prva Y os predstavlja mersko lestvico za vrednost tečaja v točkah, druga Y os pa predstavlja izračunani sentiment. Enote so vedno računan na osnovi dnevnih podatkov (en dan, ena enota).

S pomočjo funkcije `ccf()` v R-ju smo izračunali koeficiente povezanosti med različnimi kombinacijami vektorjev (in izpeljav vektorjev podatkov) tečaja (indeksa) in sentimenta po metodah AFINN in AFINN rangi:

- vsota sentimentov v dnevu,
- vrednost tečaja/indeksa,
- sprememba tečaja/indeksa,
- sprememba sprememb tečaja,
- sprememba sentimenta.

Poleg tega smo naknadno primerjali grafe v istih časovnih intervalih za obe metodi slovarjev v različnih pogojih obeh krivulj in za grafični primer podobnosti vizualno iskali korelacije obeh krivulj za boljšo predstavo bralcu. Ker razumemo, da je takšen način iskanja podoben iskanju igle v kupu sena, takšno iskanje pa je običajno podprto s predsodki (morda celo iluzijami), se bomo osredotočili na pregled s poudarkom ogleda mer podobnosti in interpretacij. Pri tem nas bodo zanimali predvsem pozitivni koeficienti korelacije. V primeru negativnega koeficienta korelacije se namreč sklepa, da je povezanost gibanja (ujemanja)

krivulj obratna. Takšna povezanost pa je z vidika praktičnosti in predvsem vsebine popolnoma nelogična.

Vpeljali bomo tudi pojem »nesmiselna korelacija«, s katerim bomo v nadaljevanju analize včasih operirali. To je korelacija, ki ni **pomensko** smiselna. Če podamo primer gibanja delnic, bi morali pričakovati, da se pozitivna krivulja sentimenta (sentiment v območju pozitivne osi Y) istočasno ujema z naraščanjem delnice/tečaja (in obratno). Ker pa se krivulja sentimenta giba znotraj negativnega območja osi Y, a kljub temu v danem časovnem intervalu opazovanja raste, nastane **pojav** navidezne povezanosti, ta povezanost pa vsebinsko ni smiselna, kar je razvidno preko praktičnega preverjanja grafov. Če poenostavimo bralcu razlago z vprašanjem: »Zakaj bi negativen sentiment o delnici vplival na rast njene vrednosti?«

V nadaljevanju bomo preverjali ***le tiste primere***, ki so najbolj pomembni za interpretacijo. Glede merskih lestvic, ki niso standardizirane, se ne bomo obremenjevali, saj bomo enostavno preverjali, ali krivulje diskretnih podatkov naraščajo/padajo in ali so smiselne. Med takšnimi primeri analize povezanosti pa so bile kombinacije naslednjih vektorjev:

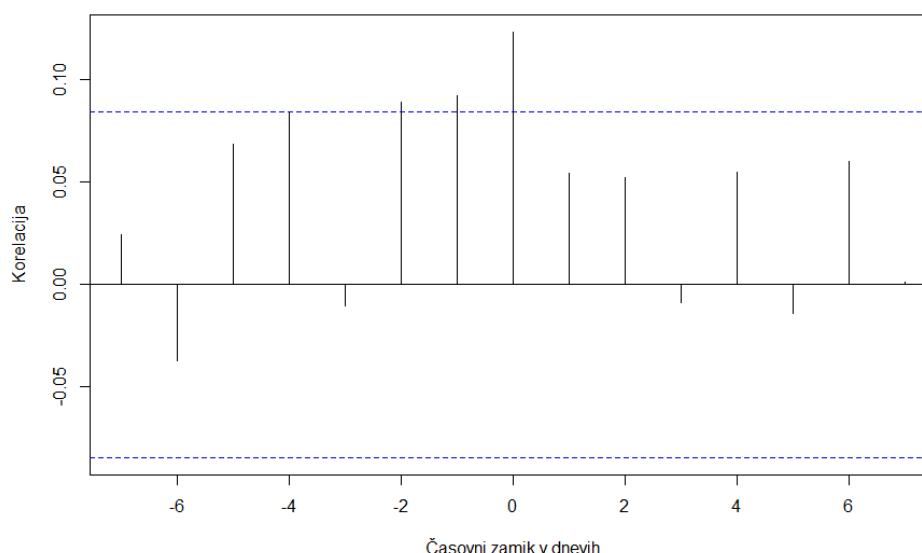
1. sprememba tečaja in sprememba vsote sentimentov AFINN,
2. sprememba sprememb tečaja in sprememba vsote sentimentov AFINN,
3. vrednost tečaja in sprememba vsote sentimentov AFINN,
4. vrednost tečaja in vsote sentimentov AFINN,
5. sprememba tečaja in vsote sentimentov AFINN,
6. sprememba tečaja in sprememba vsote sentimentov AFINN rangi,
7. sprememba sprememb tečaja in sprememba vsote sentimentov AFINN rangi,
8. vrednost tečaja in sprememba vsote sentimentov AFINN rangi,
9. vrednost tečaja in vsote sentimentov AFINN rangi,
10. sprememba tečaja in vsote sentimentov AFINN rangi.

3.3.2 Metoda s slovarjem AFINN

Iz grafa na sliki 3.4 je razvidno, da je med spremembo tečaja in sentimentom algoritem statistično značilen v časovnih točkah -2, -1 in 0 (prečka modro črto). Najmočnejša pozitivna korelacija ($p \sim 0.123$) se nahaja v točki nič (realni čas), kar pomeni, da je zaznana hkratna povezanost delnice in sentimenta. Kljub temu so zaznane še nekatere »šibkejšee«, a statistično značilne korelacije v zamikih -1 in -2 dni, iz katerih se lahko sklepa povezanost

delnice s sentimentom. Pri zamiku -4 se nam tudi nakazuje možna korelacija, vendar pa ni statistično značilna. Iz tega lahko sklepamo, da v splošnem obstaja povezanost med sentimentom in delnico pri merjenju sprememb tečaja in vsote sentimentov z AFINN metodo, še posebej zato, ker obe časovni seriji delujeta precej stacionarno (razvidno iz grafa v prilogi).

Slika 3.4: Časovne korelacije sprememb tečaja in vsota sentimentov z AFINN slovarjem (od 1.1.2014 do 24.2.2016)

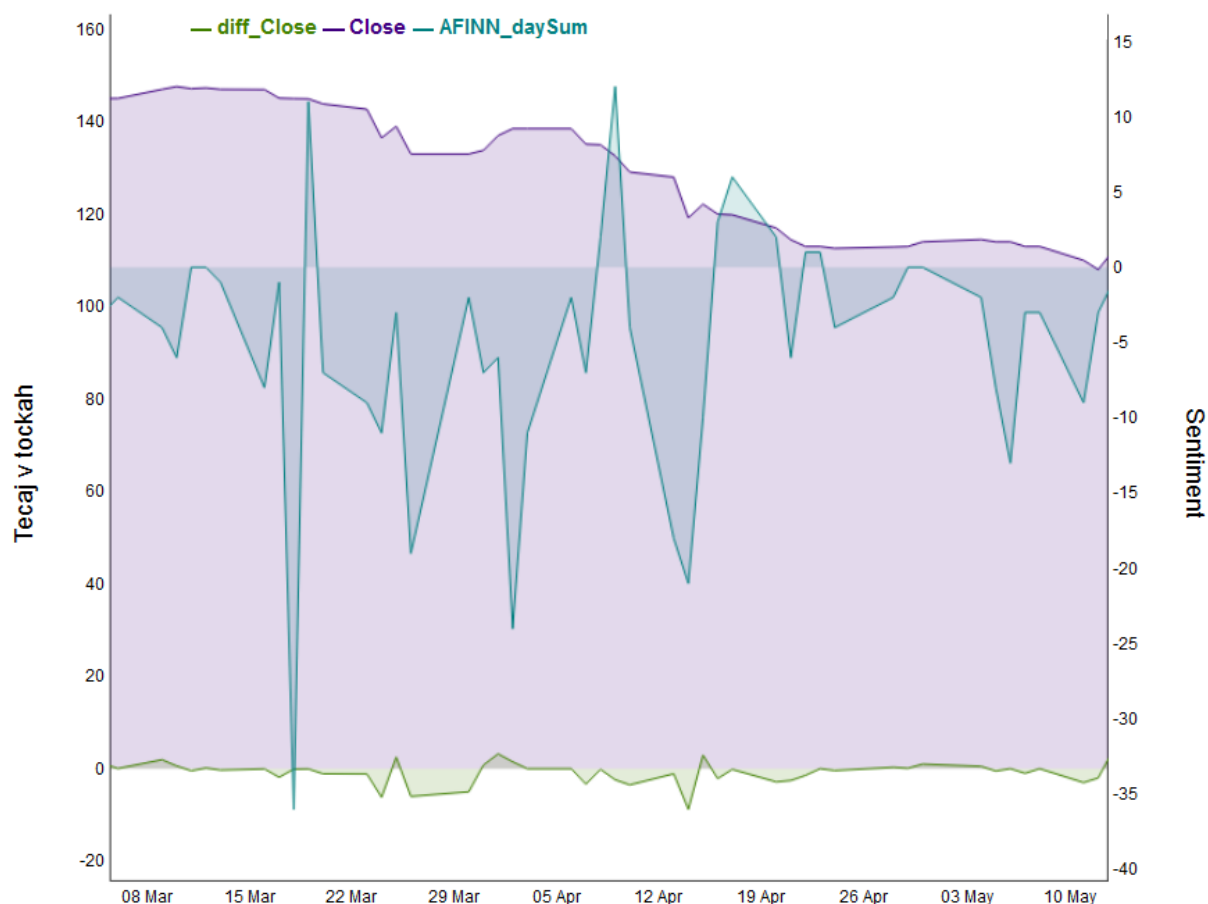


Ker je zajeti časovni interval analize grafično preobširen, krivulje v njem pa preveč zgoščene, si pobližje pogledjmo nekatere prakse merjenja učinkovitosti podobnosti obeh krivulj ter s tem skušajmo še vizualno (praktično) preveriti izračunane koeficiente korelacij v zamikih in preveriti opisne statistike frekvenc tvitov.

Za primer vzemimo obdobje med 5. 3. 2015 in 13. 5. 2015, kjer vrednost tečaja pada, zaznan pa je tudi izrazito negativen sentiment (obdobje v času politične afere »veberkom« z začetkom 18. in 19. marca). V tem času je razviden šok v vsotah sentimenta, ko se je začela afera. Opazimo, da je včasih zelo težko govoriti o *pomenski* povezanosti sentimenta s spremembo tečaja v realnem času, saj je negativen sentiment rahlo »kaotičen« in večino časa negativen, a se gibanje obeh krivulj vrši istočasno ali pa delnica vrši vpliv na sentiment. Tu gre po našem mnenju predvsem za vsebinsko *nesmiselno* korelacijo.

To je tudi obdobje večje frekvence sporočil. Povprečno število tvitov v tem obdobju je približno 30 z maksimalno frekvenco sporočil 168 na dan 14. 4. 2015. Vsote izračunanega sentimenta po dnevih so tudi visoke (glej sliko 3.5). Sporočila so po naši oceni zaradi kompleksnosti političnih ideologij uporabnikov verjetno izrazito zahtevnejša, algoritem pa po našem sklepanju ne razume najboljše *pomena* takšnih sporočil. Za enak primer bo v nadaljevanju zelo zanimivo pogledati primerjavo metode AFINN rangi (glej poglavje 3.3.3).

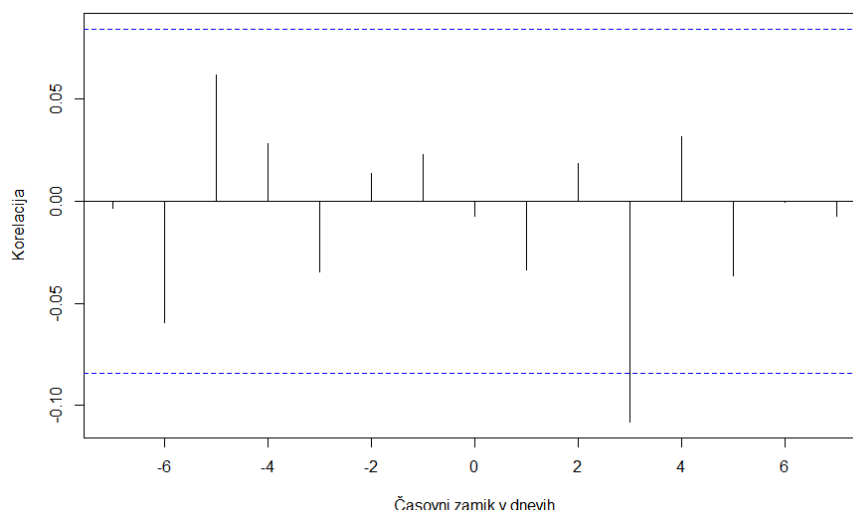
Slika 3.5: Graf tečaja, sprememb in sentimenta (5.3.2015 in 13.5.2015)



3.3.3 Metoda s slovarjem AFINN rangi

Iz slike 3.6 na grafu lahko opazimo, da je drugi algoritem izrazito statistično značilen v časovni točki +3 ($p \sim -0,108$). To pomeni, da obstaja *obratna* korelacija med sentimentom in vrednostjo tečaja. Iz tega lahko sklepamo, da sentiment je v splošnem povezan s spremembo vrednosti delnice v treh dneh, vendar pa temu primeru zaradi vsebinske nelogičnosti ne bomo posvečali pozornosti.

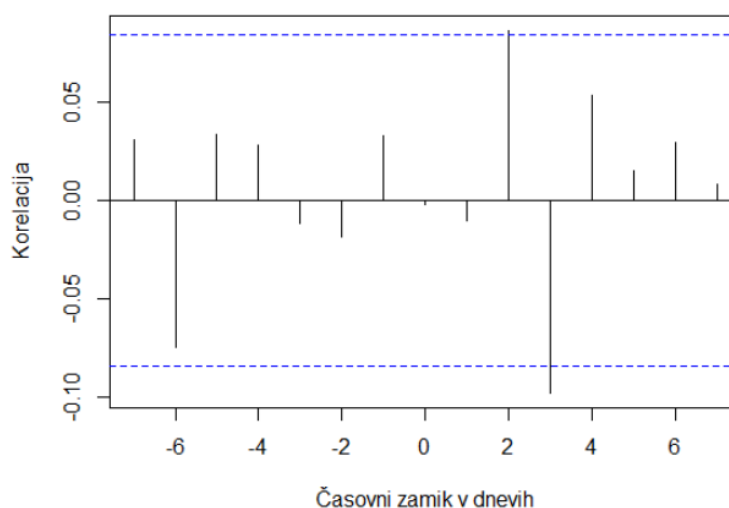
Slika 3.6: Časovne korelacije med spremembo tečaja in vsoto sentimentov z AFINN slovarjem rangov (od 1.1.2014 do 24.2.2016)



Za drugi primer izračuna povezanosti si bomo ogledali dve stacionarni (diferencirani) časovni seriji, izpeljavi spremembe sprememb tečaja in spremembe sentimenta (slika 14). Razvidni sta dve statistično značilni povezanosti v dneh +2 in +3. Med seboj sta si kontrastni, +2 je **vsebinska korelacija**, ko se z večanjem ene spremenljivke večja tudi druga, medtem ko je v točki +3 ta obratna in vsebinsko nesmiselna. Največja korelacija je v zamiku +3 dni 0.098, v zamiku dveh dni pa je nekoliko šibkejša, a še vedno statistično značilna ~ 0.090 .

Kot smo omenili v prejšnjem poglavju, je to obdobje večje frekvence sporočil, algoritem AFINN rangi pa se tu bolje odreže od preprostega štetja pozitivnih in negativnih besed, če sklepamo na podlagi izračunanih koeficientov (sliki 3.6 in 3.7).

Slika 3.7: Sprememba sprememb tečaja in sprememba vsote sentimentov AFINN rangi



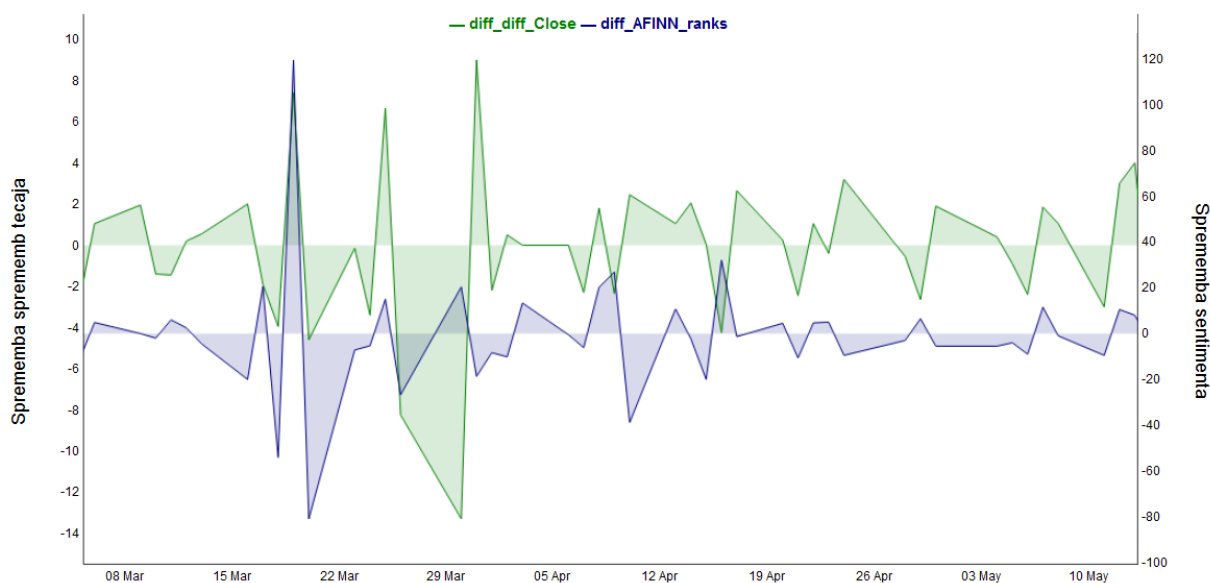
Morda se ordinalna kategorizacija besed v slovarju AFINN rangi zaradi večje frekvence sporočil in kompleksnosti znotraj konteksta pomenskih situacij stavčnih struktur (kot smo to omenili že v prvem primeru, glej komentar pri sliki 3.3) bolje odreže pri takšnih

pogojih. Predvsem zaradi tega, ker vse besede nimajo enake teže kot v primeru s slovarjem AFINN. Za vsak primer bo torej zanimivo praktično preveriti gibanje obeh krivulj.

Preko grafa (slika 3.8) opazimo, da zelo podroben pregled obeh krivulj ne potrди zaupanja v izračunane koeficiente. Od 5. 3. do 8. 3. morda lahko opazimo dan ali dva, ko se korelacija vrši v istem ali naslednjem dnevu, vendar pa je obdobje do 18. 3. kaotično in tam se ne opazi nekega vzorca korelacij. Od 18. 3. do 3. 4. se opazi korelacija v realnem času ali pa je razvidno, da od 1. 4. do 3. 4. sentiment napove za en v naprej. Od 4. 4. do 16. 4. je ponovno kaotično in ni razvideti konsistentnosti. Zdi se celo, da sentiment zaostaja. Od 16. 4. do 24. 4. se zdi, da sentiment napove, predvsem pa je razvidna korelacija v realnem času. Do 5. 4. ponovno dobimo vzorec, ko ni moč razvideti korelacije v realnem času, glede napovedovanja pa je ta časovni razpon bolj kot ne verjetno slučajen. V obdobju od 4. 5. in do konca opazovanega grafa je moč ponovno zaznati korelacijo v realnem času.

V splošnem se izkaže, da se obravnavan primer analize v tem obdobju odreže boljše glede na izračunane koeficiente, ko pogledamo od blizu, pa kljub temu ni natančen.

Slika 3.8: Graf spremembe sprememb tečaja in sprememb sentimenta (5.3.2015 in 13.5.2015)



Iz grafov in analiz povezanosti torej sklepamo, da lahko kdaj obstaja povezanost med sentimentom in delnico, vendar pa je težje jasno izraziti, kdo in predvsem *k*daj vpliva drug na drugega, še posebej v kontekstu ožjih časovnih intervalov – ko sliko »približamo« (pogledamo od blizu).

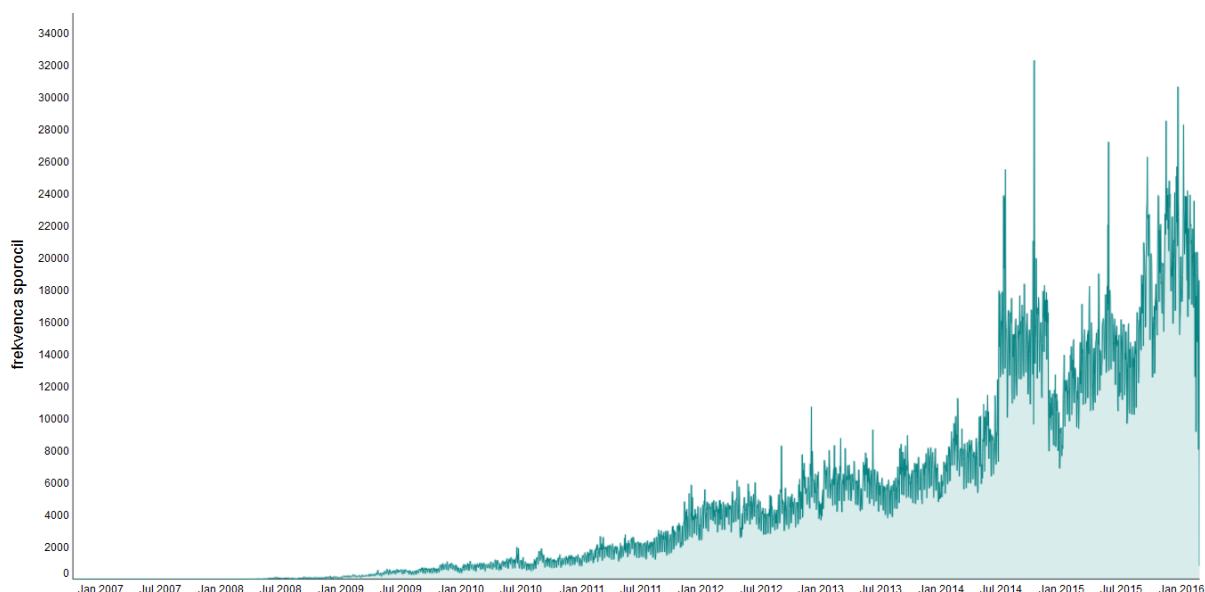
Moramo priznati, da je takšno praktično preverjanje grafov zelo problematično, že iz vidika pristranskosti – »vidimo, kar želimo videti«, še posebej, če nam metoda pokaže statistično značilno povezanost, hkrati pa zanemarimo, da obstajajo verjetnosti, ki so sicer

manj močne v drugih dneh z ocenjenim koeficientom korelacije. Kot dodatno obrazložitev vzemimo primer iz slike 3.7, kjer smo opazili, da imamo dve statistično značilni povezanosti. Ena je v zamiku dveh, druga pa treh dni. Kot smo omenili, sta si med seboj kontrastni, poleg teh pa lahko vidimo, da v zamiku -6 dni obstaja še ena šibkejša, ki pa ni statistično značilna, ampak lahko kljub temu nakazuje določeno (manjšo) verjetnost povezanosti. Metoda merjenja povezanosti nam torej pove verjetnost zamika povezanosti, vendar pa moramo pri tem biti pozorni na vse ocenjene parametre korelacije (in ne le enega).

3.3.4 Slovenski borzni indeks in sentiment v Sloveniji

Skupno število vseh sporočil v obdobju od 5. 10. 2006 do 24. 2. 2016 je 16.200.000. Od teh jih je bilo največ 32.314 dne 10. 9. 2014, kar predstavlja največjo frekvenco poslanih sporočil v enem dnevu. Povprečna vrednost vseh sporočil znaša 4.911 na dan (izračunano za vse dni, ko so bila poslana sporočila, dnevi, ko sporočil ni bilo, pri izračunu niso upoštevani). Frekvenčna porazdelitev sporočil je razvidna na sliki 3.9.

Slika 3.9: Graf frekvence sporočil v Sloveniji



Analiza frekvenčne porazdelitve sporočil je v tem primeru lahko bolj problematična kot pri analizi delnice Telekomu. Razvidna je nekonsistentna izmenjava sporočil. Kot možen razlog bi lahko navedli tehnično naravo, saj se je naš datum zajema začel 24. 4. 2015. Drugi razlog (domneva) za »rast« sporočil pa je lahko organske narave, saj se število sporočil v omrežju Twitter povečuje zaradi porasta in aktivnosti uporabnikov omrežja Twitter v Sloveniji. To sta dva med seboj kontrastna razloga, kljub temu pa se pojavlja vprašanje, zakaj

je v drugi polovici leta 2014 tako znaten poskok sporočil, nato pa je razviden relativno visok padec. Zaradi teh razlogov smo se odločili, da bomo v nadaljevanju analizirali podatke od druge polovice leta 2014, z začetkom 1. julija.

Ponovno smo preverjali izračune korelacij na primeru 10 kombinaciji vektorjev. Ti so bili:

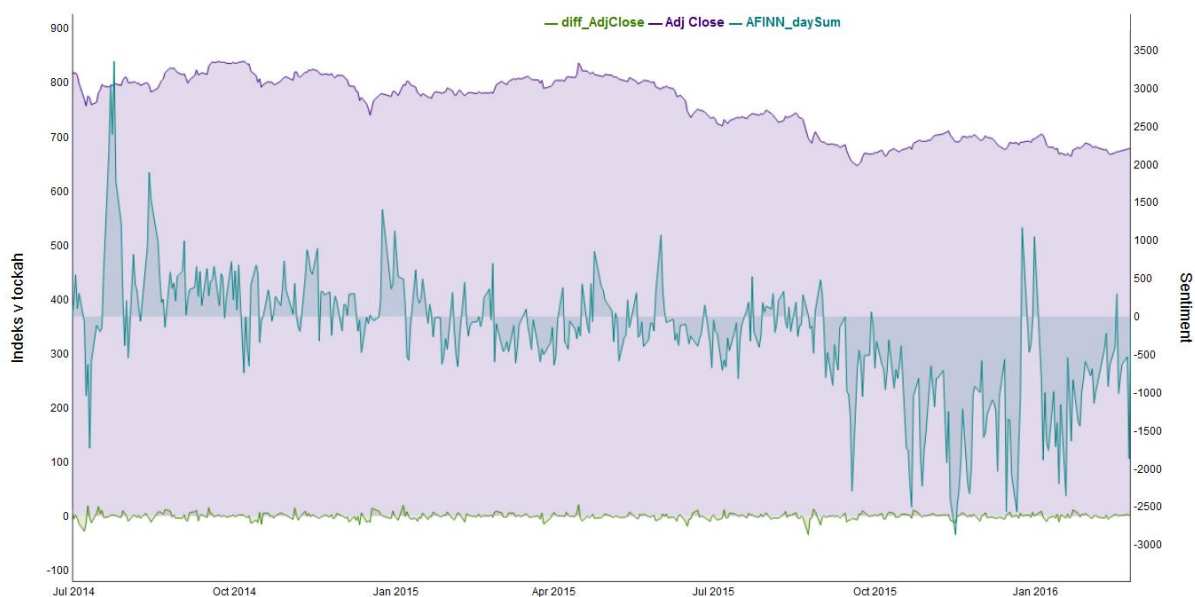
1. sprememba indeksa in sprememba vsote sentimentov AFINN,
2. sprememba sprememb indeksa in sprememba vsote sentimentov AFINN,
3. vrednost indeksa in sprememba vsote sentimentov AFINN,
4. vrednost indeksa in vsote sentimentov AFINN,
5. sprememba indeksa in vsote sentimentov AFINN,
6. sprememba indeksa in sprememba vsote sentimentov AFINN rangi,
7. sprememba sprememb indeksa in sprememba vsote sentimentov AFINN rangi,
8. vrednost indeksa in sprememba vsote sentimentov AFINN rangi,
9. vrednost indeksa in vsote sentimentov AFINN rangi,
10. sprememba indeksa in vsote sentimentov AFINN rangi.

Kot opombo dodajmo, da je bil primer analize (1) **skoraj** statistično značilen v točki -2 (v prilogi). Ostali primeri (razen tistih v nadaljevanju) pa niso dali posebnih značilnosti povezanosti, zaradi katerih bi jim tu posvečali posebno pozornost. V nadaljevanju bomo preverjali in interpretirali 7 metod od 10, ki so se izkazale za statistično značilne, v poglavju 3.4 pa podali sklepno interpretacijo naše analize in uporabljenih metod.

3.3.5 Metoda s slovarjem AFINN

Na sliki 3.10 je prikazan graf slovenskega borznega indeksa v točkah, spremembe in vsote sentimentov. Kot smo v prejšnjem poglavju utemeljili, bomo našo analizo opravili od 1. 6. 2014 naprej, v prilogi pa je prikazan tudi graf v celotnem časovnem obdobju.

Slika 3.10: Indeks v točkah, sprememba indeksa in vsota sentimentov AFINN (1.6.2014-23.2.2016)



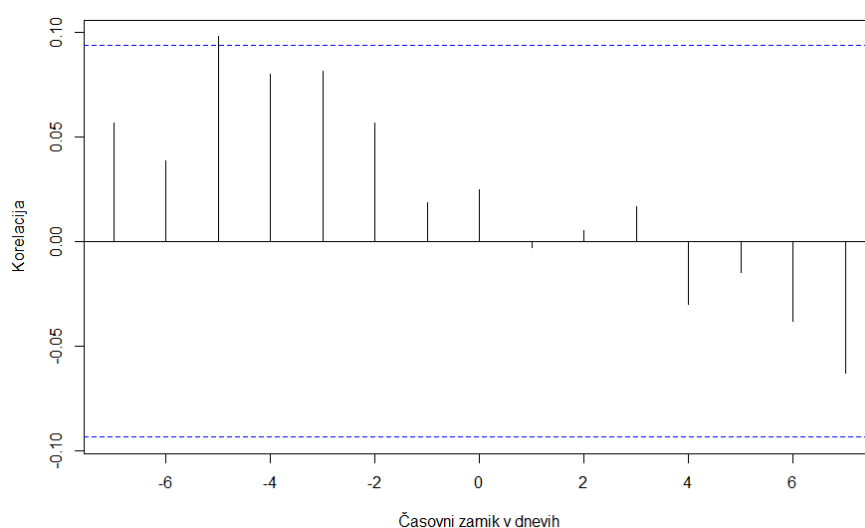
Od julija 2014 do aprila 2015 se na prvi pogled kaže zelo kaotično ujemanje med vrednostjo indeksa in vsotami sentimentov po dnevih (tu ne opazujemo sprememb sentimenta), gre za neko slučajno hojo pri obeh serijah, pri katerih pa se nakazuje trend obeh spremenljivk. Od maja 2015 do oktobra 2015 na prvi pogled opazimo relativno boljšo vizualno korelacijo (boljša - obe krivulji imata med seboj še bolj podoben trend) med sentimentom in borznim indeksom. Kljub temu je v nadaljevanju iz slike 3.10 razvidna *nesmiselna* vsebinska korelacija. To je vidno predvsem v obdobju kasneje, od oktobra do decembra 2015, ko pomenska interpretacija ni mogoča. Zaznan je tudi izrazito negativen sentiment, indeks pa tu narašča ali stagnira. Slika nam torej kaže med seboj nasprotujoče si vzorce gibanja. Obe časovni seriji pa lahko označimo za *nestacionarni*.

Za boljšo interpretacijo analize med vrednostjo indeksa in sentimentom torej raje pogledjmo analizo verjetnostnih časovnih korelacij vrednosti indeksa in sentimenta (slika 3.12), kjer opazimo močno statistično značilno povezanost. Iz slike zelo težko rečemo, kdaj obstaja povezanost, saj so zaznane povezanosti v vseh dneh (-7 do +7). Najmočnejša je seveda v realnem času, kar nam je hiter pregled grafa (slika 3.10) potrdil že prej, ko smo ponekod zaznali skupen trend. Kljub temu pa bomo morali reči, da tu obstaja **umetna korelacija**, predvsem zaradi dejstev, ki smo jih našli v prejšnjem odstavku, saj ta nastopi, če dve količini imata trend, same spremembe pa se ne ujemajo (kar bomo dokazali še v nadaljevanju, saj iz slike 3.10 v tako širokem intervalu tega ni moč oceniti).

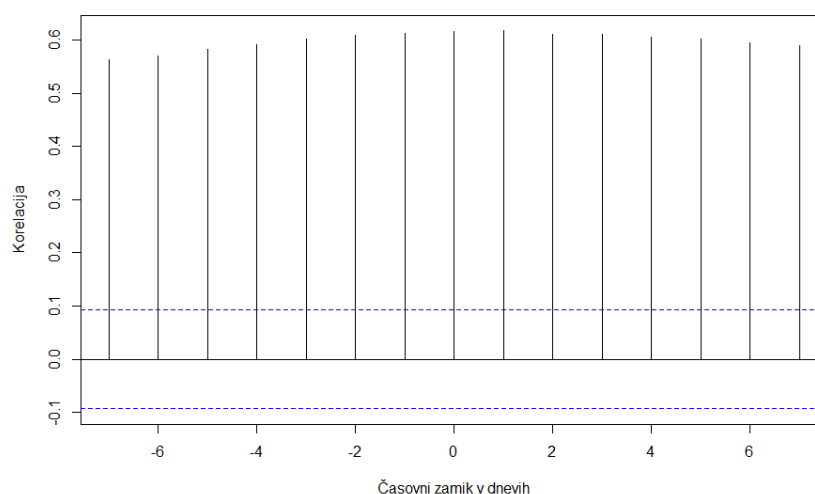
Analiza grafa verjetnosti časovnih korelacij med *spremenbo indeksa* in *vsoto sentimentov* (slika 3.11) kaže na pozitivno statistično značilno povezanost v točki -5. V točkah od -1 do -4 se stopnja korelacije nekako vidno povečuje. Graf verjetnosti časovnih korelacij

med spremembo indeksa in vsote sentimentov lahko vsebinsko logično interpretiramo, da vrednost borznega indeksa vpliva na izražen sentiment. Povezanost je statistično značilna s 95-odstotno verjetnostjo gotovosti, pri čemer se ocenjeni parameter korelacije 0.098 nahaja v intervalu zaupanja pri ± 0.094 . Kljub temu pa smo poprej že ugotavljali, da ima vsota sentimentov zaznan trend v tem časovnem intervalu, zato lahko sklepamo o efektu umetne korelacije med spremenljivkama. Zato si v nadaljevanju podrobneje pogledjmo na sliki 3.14 dve stacionarni časovni izpeljavi za ožje časovno obdobje (od 1. 10. 2015 do 1. 12. 2015) in na sliki 3.13 hkrati preverimo analizo časovnih korelacij po dnevih.

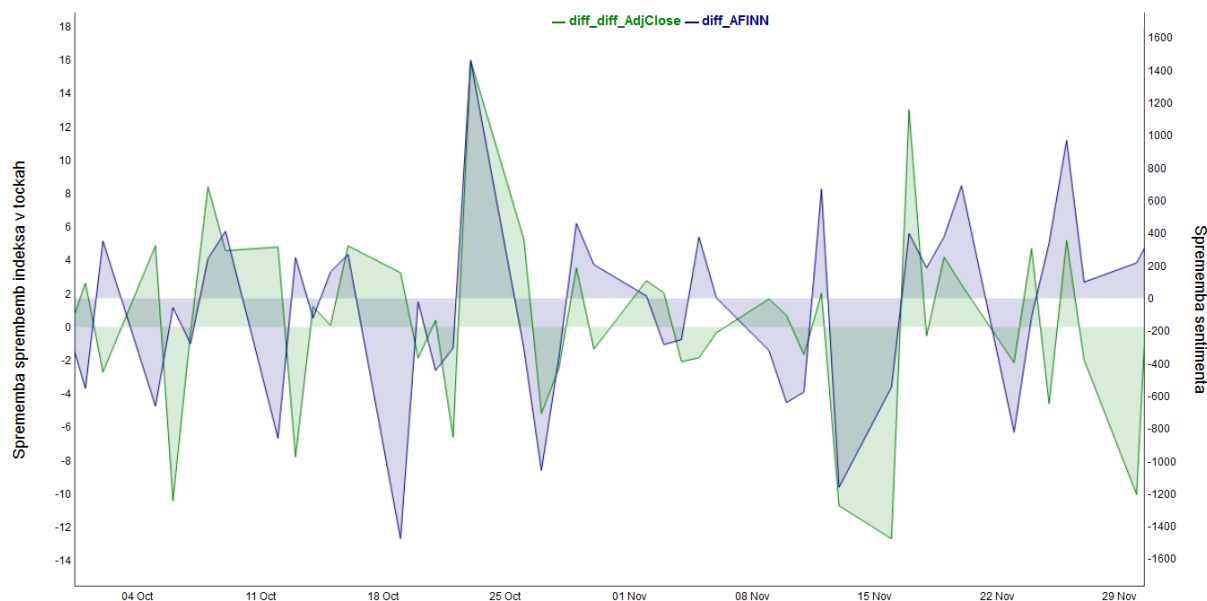
Slika 3.11: Sprememba indeksa in vsote AFINN (1.6.2014-23.2.2016)



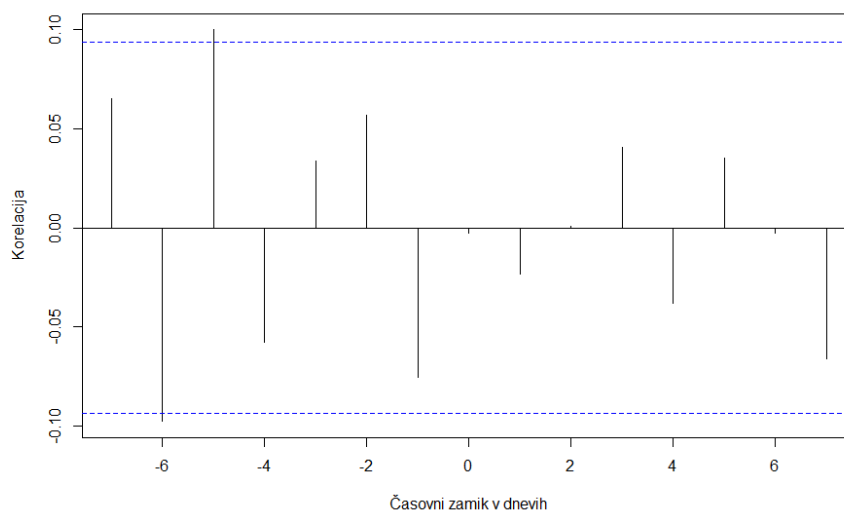
Slika 3.12: Vrednost indeksa in vsota sentimentov AFINN (1.6.2014-23.2.2016)



Slika 3.13: Sprememba sprememb indeksa in sprememba sentimenta AFINN (1.10.2015 - 1.12.2015)



Slika 3.14: Sprememba sprememb indeksa in sprememba vsote sentimentov AFINN (1.6.2014-23.2.2016)

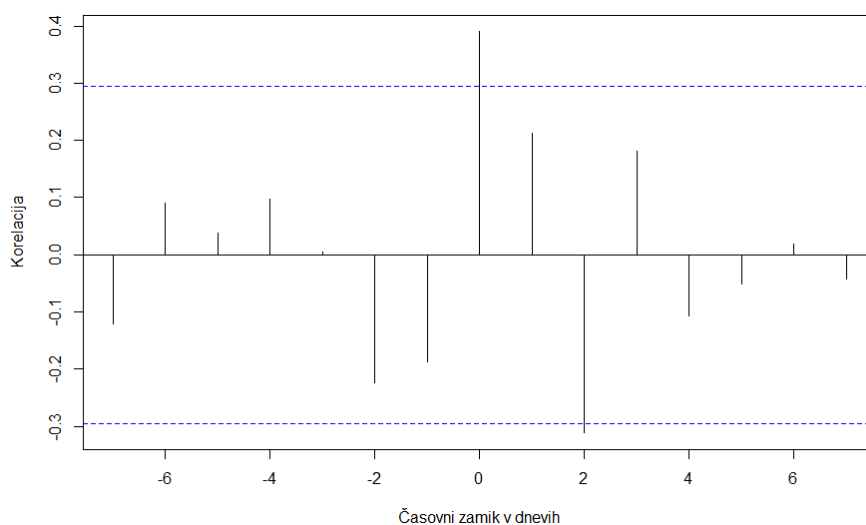


Največja pozitivna korelacija s standardizirano vrednostjo korelacijskega koeficienta ~ 0.1 se nahaja v zamiku -5 dni. Gre za pozitivno povezanost (z večanjem vrednosti ene spremenljivke se večja tudi vrednost druge), ta pa je statistično značilna s 95-odstotno verjetnostjo gotovosti, pri čemer se ocenjeni parameter korelacije ~ 0.1 nahaja v intervalu zaupanja pri ± 0.094 . Graf verjetnosti časovnih korelacij med spremembami sprememb indeksa in spremembami vsote sentimentov tako lahko vsebinsko logično interpretiramo, da vrednost borznega indeksa vpliva na izražen sentiment v približno 5 dneh. Druga največja korelacija se nahaja v zamiku -6 dni, vendar je vsebinsko nelogična, ker se na podlagi negativnega koeficienta sklepa obratna povezanost med spremenljivkama. Obratna korelacija pomeni, da se z večanjem vrednosti prve spremenljivke manjša vrednost druge. V tem

primeru bi torej kazalo na nelogično vsebinsko povezanost, kljub temu pa nam ta informacija v nadaljevanju lahko že nakazuje problem, saj sta si obe povezanosti, ki sta si med seboj kontrastni, močno skupaj v razmiku enega dne.

Praktičen pregled slike 3.14 nam nakazuje, da se v tem časovnem intervalu ne moremo zanašati na ocenjene koeficiente in korelacije pri izračunanih zamikih, kar je nekako smiselno, saj je izračunani interval veliko daljši. Ponovimo preverjanje računanja časovnih korelacij, tokrat le za ožji opazovani interval (slika 3.15).

Slika 8 Slika 3.15: Sprememba sprememb indeksa in sprememba sentimentov AFINN (1.10.2015 - 1.12.2015)



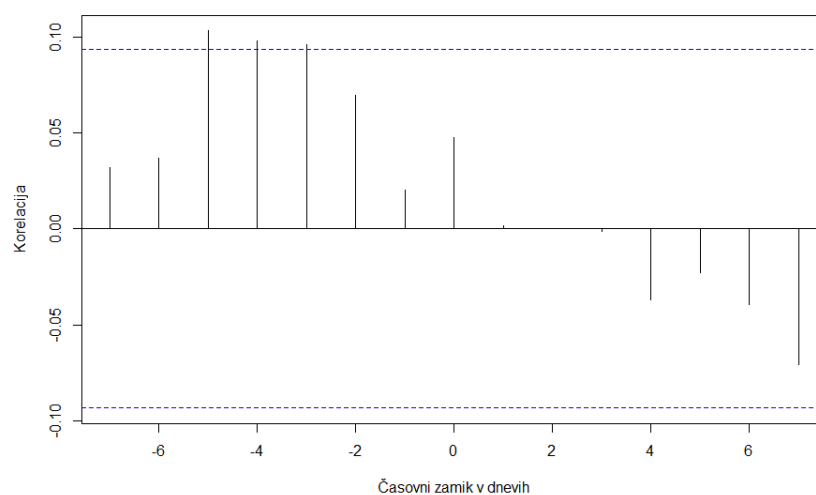
Največja pozitivna korelacija na sliki 3.14 s standardizirano vrednostjo korelacijskega koeficienta ~ 0.39 se nahaja v realnem času, ta pa je statistično značilna s 95-odstotno verjetnostjo gotovosti, pri čemer se ocenjeni parameter korelacije 0.39 nahaja v intervalu zaupanja pri ± 0.295 . Druga korelacija se nahaja v zamiku +2 dni; ta je obratna in malo šibkejša, kar kaže na nelogično vsebinsko povezanost. V obeh proučevanih primerih časovnih intervalov je razvidna nekonsistentnost in nasprotujoči se rezultati.

Če sedaj končno praktično pregledamo graf na sliki 3.14, lahko opazimo bolj verodostojno ocenjene koeficiente. Opazijo se namreč (ponekod) ujemanja v realnem času, kljub temu pa prav tako opazimo slučajne pojave »neujemanja« oziroma smiselno statistično domnevo drugega izračunanega koeficienta o obratni povezanosti spremenljivk v zamiku dveh dni, kjer sentiment vpliva na gibanje indeksa.

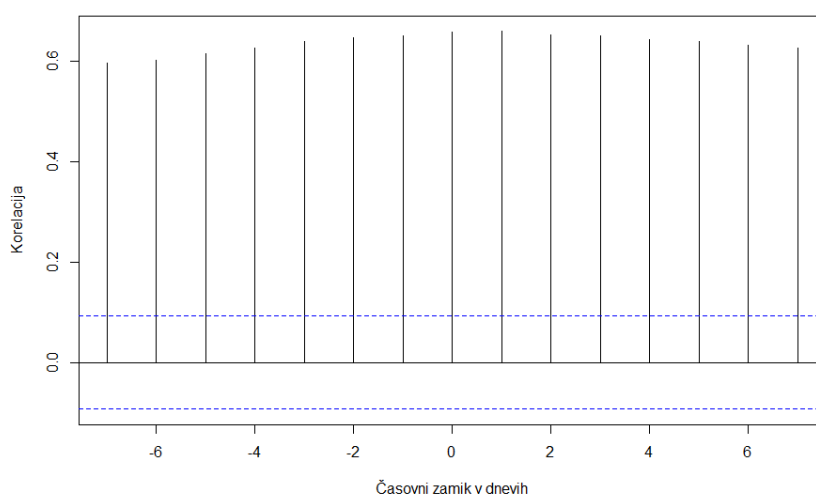
3.3.6 Metoda s slovarjem AFINN rangi

Metoda s slovarjem AFINN rangi vrne zelo podobne rezultate (primerjava slik 3.11, 3.12 in 3.16, 3.17), zaradi česar lahko trdimo, da obstaja tako vsebinsko *nesmiselna* korelacija med povezanostjo spremembe indeksa in sentimenta kot tudi umetna korelacija med vrednostjo indeksa in sentimentom, kot smo ugotavljali, interpretirali in utemeljevali bolj obširno v poglavju prej. Dodaten graf, v katerem lahko vidimo gibanje časovnih vrst po metodi AFINN rangi, se nahaja v prilogi.

Slika 3.16: Spremembe indeksa in vsote sentimentov AFINN rangi (1.6.2014-23.2.2016)

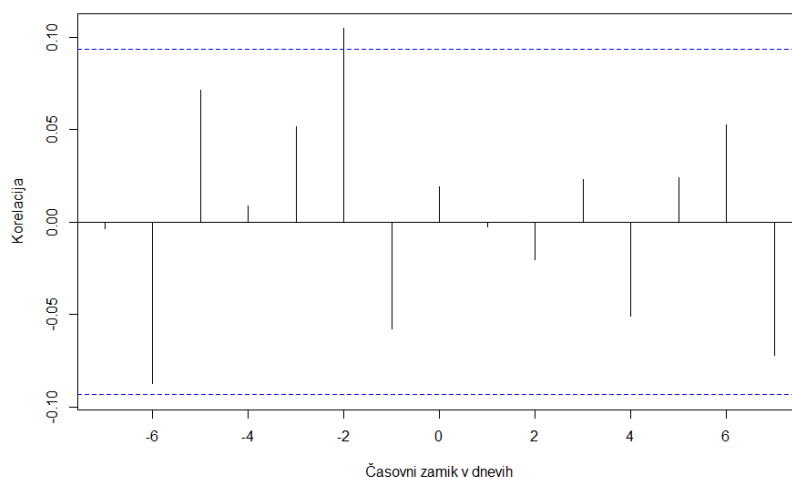


Slika 3.17: Vrednost indeksa in vsota sentimentov AFINN rangi (1.6.2014-23.2.2016)

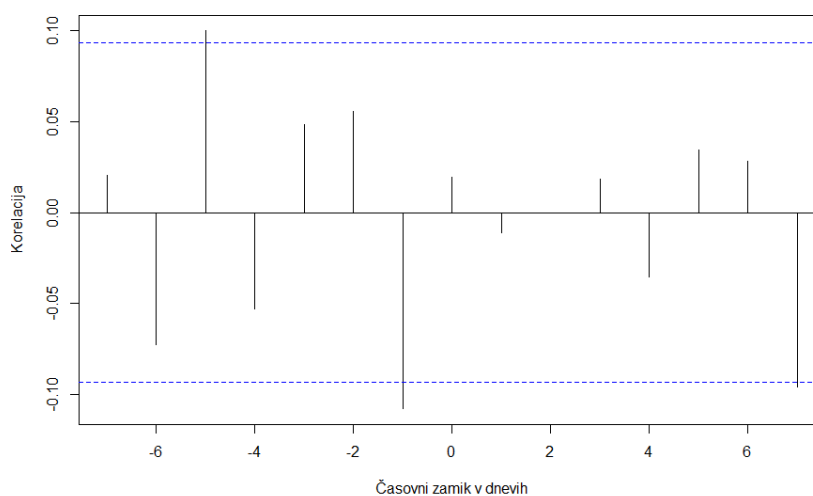


Pobliže si oglejmo slike 3.18 in 3.19, ki sicer vrmeta nekoliko drugačne rezultate. Obe analizi sta primer stacionarnih časovnih serij, zaradi česar smo se izognili efektu umetne korelacije.

Slika 3.18: Sprememba indeksa in sprememba vsote sentimentov AFINN rangi (1.6.2014-23.2.2016)



Slika 3.19: Sprememba sprememb indeksa in sprememba vsote AFINN rangi (1.6.2014-23.2.2016)



Največja pozitivna korelacija iz slike 3.18 je ~ 0.10 pri zamiku v točki -2. Povezanost je statistično značilna s 95-odstotno verjetnostjo gotovosti, pri čemer se ocenjeni parameter korelacije ~ 0.10 nahaja v intervalu zaupanja pri ± 0.094 . Zaradi tega lahko sklepamo, da je zaznan vpliv (zamik povezanosti) indeksa na sentiment v 2 dneh. Kljub temu se ponovno pojavljajo verjetnosti v zamikih različnih dni, ki pa sicer niso statistično značilne (so šibkejše), največja je pri zamiku v točki -6, zaradi česar ne bomo ponovno praktično preverjali grafov, saj nam bo rezultat izpisov in gibanja krivulj poznan.

Na sliki 3.19 so razvidne 3 statistično značilne korelacije, ki so si med seboj popolnoma nasprotujoče. Zaradi vsebinske nelogičnosti (najmočnejša korelacija je v točki -1 in nakazuje obratno korelacijo) pa se jim dodatno ne bomo več posvečali.

Iz vseh teh grafov lahko sklepamo, da prav tako obstaja vsebinsko *nesmiselna, slučajna korelacija, ki ji je težje pripisati konsistentne rezultate*. Kot dodaten problem lahko tu izpostavimo preveliko število preverjanja oziroma izračunov z našimi izpeljavami. Zgodi se namreč, da se ravno zaradi tega dodatnega izračunavanja kdaj čisto slučajno pokažejo korelacije (še posebej, če spreminjamo še časovne intervale za analizo podatkov, kot smo to nakazali v poglavju 3.3.5, ko smo spremenili opazovani časovni interval in ponovno izračunali koeficiente).

Glede na dobljene rezultate analiz z obema metodama (pri katerih smo sicer dobili večje število korelacij z metodo AFINN rangi) ne moremo sklepati, katera metoda se je v splošnem odrezala bolje. Razlogov za to je seveda lahko več, a bomo o njih in drugih možnih izboljšavah našega algoritma govorili še v zaključni interpretaciji. Kljub temu se **lahko** najdejo dobre prakse, vendar pa so te bolj šibke in nezanesljive.

3.4 Zaključna interpretacija merjenja povezanosti

Naj dodamo, da se v analizi časovnih serij dostikrat pojavljajo umetne korelacije, predvsem pri daljših časovnih intervalih, kar se nakazuje kot morebiten problem. Ko smo skozi prejšnja poglavja preverjali metode analiz in časovne intervale, smo prišli do različnih interpretacij. Izvedli smo precej analiz, ki pa niso dale konsistentnih rezultatov, pogosto so si bili celo nasprotujoči. Razen nekaterih umetnih korelacij, so bile tudi tiste korelacije, ki so bile zaznane, v glavnem precej negativne ali šibke – še posebej, ko smo skušali praktično preverjati naše grafe.

Zaradi teh dejstev moramo na naše prvotno raziskovalno vprašanje, ki se glasi: »*Ali je izmerjeni sentiment uporaben za napovedovanje cen delnic?*« odgovoriti, da na podlagi rezultatov ne moremo sklepati o obstoju povezanosti med sentimentom in ceno delnic oz. vrednostjo indeksa. Kljub temu pa tudi ne moremo z gotovostjo trditi, da sentiment ne bi bil uporaben za napovedovanje cene delnic, med drugim tudi zato, ker so se v nekaterih analizah korelacije vseeno pokazale.

Analiza časovnih korelacij *na daljši časovni rok* je pokazala, da obstajajo statistično značilne korelacije med posamičnimi vektorji podatkov, kjer sentiment povezuje vrednost tečaja (indeksa). Te povezave pa so relativno šibke, zlasti zaradi tega, ker se lahko najdejo na več mestih, ko zamaknemo vektorja med seboj po dnevih ali pa jih primerjamo v različnih časovnih intervalih. Pri obeh metodah s slovarjem torej lahko obstajajo takšne statistično značilne povezanosti. Vendar pa je bistveno težje in jasno izraziti, kaj vpliva (povezuje) drug na drugega, v kontekstu različnih ali pa ožjih časovnih intervalov – ko sliko »približamo« (pogledamo od blizu).

Izmerjene korelacije sodijo ob bok primerljivih svetovnih študij (Johan in drugi 2010; Tetsuro in drugi 2011; Chung in Liu 2011; Dietmar 2014). Lahko bi tudi dodatno trdili po analogiji študije (Chung in Liu 2011, 4), da naš algoritem (še) ne obvladuje pomena sentimenta, zato se večkrat zgodi, da je izračunan sentiment resnično pomensko napačen takrat, ko delnica pada, sentiment pa je negativen, kot takrat, ko je sentiment pozitiven, delnica pa raste. Večkrat smo dobili občutek, da je izmerjen sentiment, kadar je ta pozitiven, delnica pa raste, tudi *pomensko* pravilnejši (vendar pa tega dokumentirano nismo naredili). Ta občutek je seveda lahko varljiv, vendar pa trdimo, da so stavčne strukture zanikanja, sarkazma, cinizma zahtevnejše oziroma bolj problematične in **kompleksnejše** s štetjem slovarjev besed. Že zaradi rezultatov omenjenih avtorjev Chung in Liu, ki sta pokazala takšno asociacijo (Chung in Liu 2011, 4), kjer so s slovarjem besed primerjali delnice, te pa so bile

razvrščene v skupini zmagovalcev in poražencev. Ko so primerjali časovne korelacije obeh skupin, so opazili, da pri zmagovalcih obstajajo statistično značilne korelacije, medtem ko jih pri poražencih ni bilo. Kot drugo pa to lahko trdimo iz popolnoma pragmatične narave, saj kadar smo optimistični, smo običajno pozitivni in veseli, naši miselni procesi pa v takšnem stanju *niso* kompleksni. Pokazali smo – kar nam daje nekaj dodatnega optimizma – da je metoda AFINN rangov (lahko) bolj učinkovita v takšnem primeru. To so sicer večkrat pokazali izračunani koeficienti, ko smo med seboj kombinirali različne vektorje, vendar pa to ni nujno, saj se je večkrat pokazala tudi vsebinsko nelogična, ne smiselna ali pa umetna korelacija. Naše sklepanje ali domneva temelji na predpostavki, da vse besede nimajo enake teže, pri tem pa se vsote z metodo AFINN rangov včasih lahko odrežejo bolje. Pri analizi hkrati tvegamo napake na nivoju filtracije podatkov. Podatke smo filtrirali na regex vzorcu 5,8 milijona sporočil, proučevani vzorec je bil skoraj trikrat večji, filtracije pa nismo delali naknadno. Vse to kaže, da ima algoritem potencial za razvoj in prostor za izboljšave.

Kakšne bi lahko bile izboljšave? Zgradili bi lahko napovedne modele za časovne vrste, pri katerih bi kot ena izmed spremenljivk nastopala spremenljivka sentimenta. Pri tem bi lahko uporabili poleg že obstoječih spremenljivk še obseg trgovanja (*angl. Volume*) in nekatere izpeljave diferenciranih časovnih vrst ter pri tem temeljito preverjanje regresijskih modelov. Ideja je, da bi program lahko preverjal pogoj frekvence tvitov (morda tudi frekvence pojavljanja kakšne izmed ključnih besed v besedilih), takšne pogoje pa nato ekstrahiralo oz. rekodiralo v dodatno spremenljivko in na njihovi podlagi zaobjel časovne intervale za izgradnjo napovednih modelov s pomočjo sentimenta. Poleg prej omenjenih dodatnih izboljšav vidimo v algoritmičnih operacijah prostor za dodatne pohitritve. Algoritem v tej fazi ni optimalen (kljub temu da je za trenutne potrebe dovolj hiter). Naša kritika se usmerja predvsem na računanje sentimenta in čiščenje besedil, kjer lahko opazimo »ozka grla«. Druga je uporaba paralelizma s pomočjo alokacije večjedrnih paralelnih računskih operacij (npr. paralelno procesiranje blokov podatkov pri porazdeljeni datotečni tabeli). Naš algoritem v tej fazi uporablja Twitterjev search API, torej pridobivanje podatkov v pretekliku (glej poglavje 3.1.2). Če bi temu algoritmu dodali še streaming API, pridobivanje podatkov v sedanjiku, bi lahko dobili še bolj aktualne podatke za potrebe *napovedovanja v realnem času*. Resnične dodatne izboljšave vidimo predvsem s pomočjo strojnega učenja, s katerim bi skušali sentiment izračunati s pomočjo algoritmov za klasifikacijo sporočil (npr. Naive Bayes, nevronske mreže, z metodo podpornih vektorjev ipd.). S tem bi lahko dobili bolj verodostojne ocene sentimenta in zelo verjetno pomensko pravilnejše. Če bi se to vse uresničilo, pa dodatno vidimo izboljšave tudi v tem, da tabele tvitov vsebujejo atributa retweetCount in

favoriteCount, ki bi ju uporabili za uteži, ki bi kazale moč posamičnega sporočila. Na daljši časovni interval bi s takšno metodo izluščili vplivnejše uporabnike slovenskega omrežja.

Za takšne stvari pa je potrebno še nekaj dodatne energije in časa, hkrati pa bi po naši oceni za takšne potrebe modeliranja slovenskega omrežja Twitter zadostoval 10 let star računalnik (npr. matična plošča z LGA 755 vmesnikom in štirijedrnim procesorjem Q9650), z malo večjo količino pomnilnika (od 6 do 8 Gb pomnilnika) in SSD diskom, pri računskih operacijah strojnega učenja pa bi lahko za dodatno hitrost skrbela še moderna grafična kartica na PCI express vodilu.

Takšen algoritem je moč naknadno uporabiti tudi za potrebe modeliranja slovenskega uporabniškega omrežja Twitter, za potrebe merjenja sentimenta do različnih trženjskih znamk, zaupanja vladi, politikom in podobno.

4 ZAKLJUČEK

Naloga nam je predstavljala poseben izziv, saj je pri delu šlo za zelo aktualno področje podatkovnih znanosti. Kako obvladati velikanske količine podatkov iz Twitterja in iz njih poiskati zanimive in uporabne ugotovitve, zaključke pa uporabiti za praktične namene, s pomočjo modernih in interaktivnih vizualnih tehnik predstavitve rezultatov in modeliranja.

Skozi naše magistrsko delo smo se naučili ogromno. Najprej smo raziskovali odprtokodne rešitve, kjer smo ugotovili, da ne obstaja »celovit«¹ prosto dostopen program za analize masovnih podatkov na Twitterju. Prebrali smo nekaj študij sentimenta in računskih analiz naravnega jezika, nato pa se lotili kodiranja. Potrebno je bilo obvladati operiranje s podatkovnimi strukturami, regex iskalnimi vzorci, sintaktično pisanje z R-jevim standardom `data.table` in razumeti `twitteR` knjižnico. Napisali smo algoritem za avtomatično ekstrakcijo podatkov delnic in tečaja s slovenske borze. Preko prakse projektnega dela, teorije, raziskovanja in razvoja programa smo se naučili algoritma MapReduce za potrebe zbiranja in analize masovnih podatkov.

Naše ugotovitve so kratkoročno gledano lep uspeh, vendar pa menimo, da je možno obstoječi algoritem še izboljšati in ga uporabiti tudi na drugih področjih prepoznavne in računske klasifikacije sentimenta besedil.

V teoretičnem delu smo raziskovali trende, izzive in definirali masovne podatke. Zaobjeli smo bistvene dele, za katere menimo, da so nujni pri operiranju z izzivom masovnih podatkov. Utemeljili smo pomen analize sentimenta, ga definirali in postavili v širši teoretični okvir. Preverjali smo tehnične lastnosti analize in posamične izzive obdelave podatkov po korakih ter metode in pristope klasifikacije. Raziskali smo podobne študije napovedovanja s pomočjo omrežja Twitter in se osredotočili predvsem na teoretični vidik gibanja delnic in tečajev.

V empiričnem delu smo razložili delovanje našega programa in zaobjeli razvojni proces izdelave posamičnih algoritmov. V analizi smo računsko odkrili, da je naš algoritem pri 5-odstotnem tveganju (oz. s 95-odstotno gotovostjo) pri nekaterih analizah pokazal statistično značilne korelacije, s tem pa smo ugotovili, da sentiment lahko povezuje gibanje delnice ali tečaja. Kljub temu pa smo s praktičnim pregledom (nekaterih) naših rezultatov prišli do zaključka, da je pri veliko analizah takšna korelacija vsebinsko nelogična, slučajna ali zgolj umetna. Prišli smo tudi do teoretičnega zaključka, da ima naš algoritem s pristopom slovarja besed lahko težave s razumevanjem pomena stavčnih struktur posameznih besedil. Pri tem obstaja še širok spekter možnih izboljšav in praktičnih primerov uporabe, vendar pa

smo se zaradi zahtevnosti čim bolj popolnega zbiranja podatkov samemu izpopolnjevanju analiz v tem magistrskem delu manj posvečali.

5 LITERATURA

- 140dev. *Twitter API Tutorial – Aggregating tweets: Search API vs. Streaming API*. Dostopno preko: <http://140dev.com/twitter-api-programming-tutorials/aggregating-tweets-search-api-vs-streaming-api/>. (27 februar 2016).
- Aggarwal, C., in Cheng Zhai. 2012. *Mining Text Data*. Dostopno preko: <http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-1-4614-3222-7>. (15 januar 2015)
- Amazon EC2. *EC2 Instance Pricing – Amazon Web Services (AWS)*. <https://aws.amazon.com/ec2/pricing/>. (27 februar 2016).
- Andrei Oghina, Mathias Breuss, Manos Tsagkias, in Maarten de Rijke. 2012. *Predicting IMDB Movie Ratings Using Social Media*. Springer-Verlag Berlin Heidelberg.
- Barberá, Pablo. 2015. *Introducing the streamR package*. Dostopno preko: <http://pablobarbera.com/blog/archives/1.html>. (27 februar 2016).
- Batagelj, Vladimir. 2016. *Centralni Limitni Izrek*. Dostopno preko: <http://vlado.fmf.uni-lj.si/educa/dmfa/OZ97/CLT/cenlimiz.htm>. (27 februar 2016).
- Bradicich, Tom in Stephanie, Orci. 2013. *The Moore's Law of Big Data*. Dostopno preko: <http://www.ni.com/newsletter/51649/en/>. (27 februar 2016).
- Brian, Profitt. 2013. *What APIs Are And Why They're Important - ReadWrite*. Dostopno preko: <http://readwrite.com/2013/09/19/api-defined>. (27 februar 2016).
- Carl, Boettiger. 2015. *ropensci/docker*. Dostopno preko: <https://github.com/ropensci/Docker>. (27 februar 2016).
- Chris Snijders, Uwe Matzat, in Ulf-Dietrich Reips. 2012. *"Big Data": Big Gaps of Knowledge in the Field of Internet Science*. *International Journal of Internet Science* 2012 (7 (1)): 1–5.
- Christopher, Potts. 2015. *Sentiment Symposium Tutorial: Lexicons*. Dostopno preko: <http://sentiment.christopherpotts.net/lexicons.html>. (27 februar 2016).
- Cran HPC. 2015. *CRAN Task View: High-Performance and Parallel Computing with R*. Dostopno preko: <https://cran.r-project.org/web/views/HighPerformanceComputing.html>. (27 februar 2016).
- cs.princeton. 2009. *File systems and databases: managing information*. Dostopno preko: <https://www.cs.princeton.edu/courses/archive/fall09/cos109/09filesys.pdf>. (27 februar 2016).
- Damasio in A.R. 1994. *Descartes' error: emotion, reason, and the human brain*. (Putnam), pp. xix, 312.
- Dave Roos. 2015. *What is an API? - HowStuffWorks*. Dostopno preko: <http://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm>. (27 februar 2016).
- Dietmar, Janetzko. 2014. *Predictive modeling in turbulent times – What Twitter reveals about the EUR/USD exchange rate*. *Springer Science+BusinessMedia* 2014, *Netnomics* (2014) 15:69–106
- DigitalOcean. *Pricing | DigitalOcean*. Dostopno preko: <https://www.digitalocean.com/pricing/>. (16 maj 2016).
- Docker. *Docker Installation on Ubuntu*. Dostopno preko: <https://docs.docker.com/engine/installation/ubuntulinux/#install>. (16 maj 2016).
- Doug Laney. 2001. *3D Data Management: Controlling Data Volume, Velocity, Volume*. *Gartner* 2001: 4.
- Dowle, Matt. 2015. *CRAN - Package data.table*. Dostopno preko: februar 27. <http://cran.r-project.org/web/packages/data.table/index.html>. (16 maj 2016).
- DygraphsR. *dygraphs for R*. Dostopno preko: <http://rstudio.github.io/dygraphs/>. (4 marec 2016).
- Emma Haddia, Xiaohui Liua, in Yong Shib. 2013. *The Role of Text Pre-processing in Sentiment Analysis*. *Procedia Computer Science* 2013 (17): 26–32.
- Fama, in E.F. 1965. *The Journal of Business*, št. 38: 34–105.
- Ferle. 2013. *Znanost podatkov | MonitorPro*. Dostopno preko: <http://www.monitorpro.si/147145/praksa/znanost-podatkov/>. (21 januar 2015).

- Finn Arup Nielsen. 2011a. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs DTU Informatics, Technical University of Denmark, Lyngby, Denmark. Dostopno preko: <http://arxiv.org/pdf/1103.2903v1.pdf>. (21 januar 2015).
- . 2011b. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs, Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages 718 in CEUR Workshop Proceedings : 93-98. 2011 May. Dostopno preko: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010. (7 januar 2015).
- Forbes. 2013. *What Analytics And Big Data Can Learn From Lego*. <http://www.forbes.com/sites/sap/2013/04/11/what-analytics-and-big-data-can-learn-from-lego/>. (18 december 2015).
- Gareth, James in Tibshirani, Robert. 2014. *An Introduction to Statistical Learning w Applications in R*. New York Heidelberg Dordrecht London: Springer.
- Gaurangi Patil, Varsha Galande, Vedant Kekani in Kalpana Dange. 2014. Sentiment Analysis Using Support Vector Machine. *International Journal of Innovative Research in Computer and Communication Engineering* 2014 (Vol. 2, Issue 1). http://www.ijrcce.com/upload/2014/january/16K_Sentiment.pdf.
- Gentry, Jeff. 2015. *geoffjentry/twitteR*. Dostopno preko: <https://github.com/geoffjentry/twitteR>. (27 februar 2015).
- Gibilisco, Stan. 2012. What is law of large numbers? - Definition from WhatIs.com. Dostopno preko: <http://whatis.techtarget.com/definition/law-of-large-numbers>. (21 januar 2016).
- Gilbert, 2015. 2016. R: Auto- and Cross- Covariance and -Correlation Function. "Spletni priročnik za R *stat.ethz.ch*." Dostopno preko: <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/acf.html>. (3 marec 2016.)
- Gilbert, E & Karahalios, in K. 2010. Widespread worry and the stock market. Dostopno preko: <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1513>. (21 januar 2016).
- Gunter, Barrie, Nelya Koteyko, in Dimitrinka Atanasova. 2014. A market-relevant and reliable measure of public feeling? *International Journal of Market Research* 56 (2): 231–47.
- Hadley Wickham. 2015. *CRAN - Package rvest*. Dostopno preko: maj 30. <http://cran.r-project.org/web/packages/rvest/index.html>. (30 januar 2015).
- Halls-More. 2015. *Serial Correlation in Time Series Analysis - QuantStart*. Dostopno preko: <https://www.quantstart.com/articles/Serial-Correlation-in-Time-Series-Analysis>. (30 januar 2015).
- Hao Yu. 2015. *Rmpi*. Dostopno preko: <https://cran.r-project.org/web/packages/Rmpi/Rmpi.pdf>. (30 januar 2015).
- Harvard University. 2015. *Inquirer Categories and Use of Inquirer Dictionaries*. Dostopno preko: <http://www.wjh.harvard.edu/~inquirer/homecat.htm>. (3 januar 2016)
- H.Cootner, in P. 1964. *The random character of stock market prices*.
- Hu in Liu. 2004. *Opinion Mining, Sentiment Analysis, Opinion Extraction*. Dostopno preko: <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#datasets>. (30 januar 2015).
- IBM. 2015. *What is big data? Spletna stran - IBM*. Dostopno preko: <https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>. (3 januar 2016).
- Ibrahim Abaker, Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan. 2015. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems* 2015: 18.
- Il-Yeol Song. 2015. BIG DATA TECHNOLOGIES, USE CASES, AND RESEARCH ISSUES. College of Computing & Informatics, Drexel University. Dostopno preko: http://www.acm.org/conferences/sac/sac2015/SAC%202015_Keynote_BDT_IY_Song-2.pdf. (30 januar 2016).
- Investopedia. 2015. *Definition: Efficient market hypothesis*. Dostopno preko: <http://www.investopedia.com/terms/e/efficientmarkethypothesis.asp>. (30 januar 2015).
- Iordanova, Tzveta. 2007. Introduction To Stationary And Non-Stationary Processes. Dostopno preko: <http://www.investopedia.com/articles/trading/07/stationary.asp>. (30 januar 2015).
- Jeffrey Breen. 2012. Running R on Hadoop - CHUG - 20120815. Dostopno preko: <http://www.slideshare.net/ChicagoHUG/getting-started-with-r-hadoop-chug->

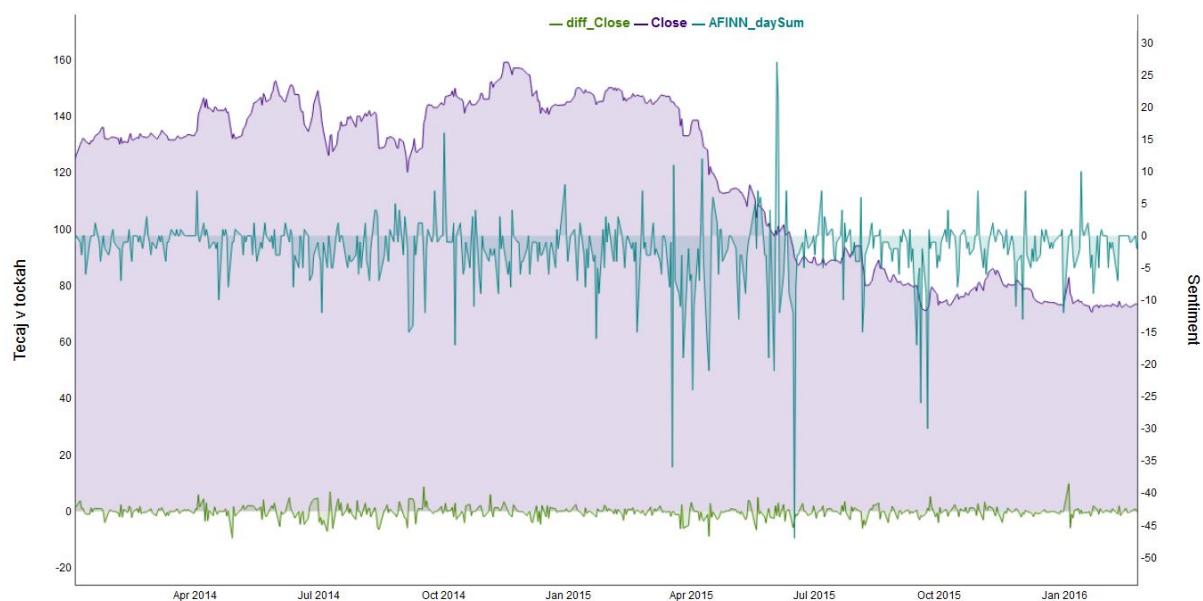
- 20120815?ref=http://www.r-bloggers.com/getting-started-with-r-and-hadoop/?relatedposts_hit=1&relatedposts_origin=49429&relatedposts_position=1. (30 januar 2015).
- Johan, Bollen, Mao Huina, in Zeng Xiao-Jun. 2010. Twitter mood predicts the stock market : (1–8). Dostopno preko: <http://arxiv.org/pdf/1010.3003.pdf%3Fiframe%3Dtrue%26width%3D90%2525%26height%3D90%2525&sa=X&scisig=AAGBfm2dF-QKobX1fGBnO9rUIPMzX3IhmQ&oi=scholar&ei=d8HAUqCWN02p7AaP3oD4BQ&ved=0CCwQgAMoADAA>. (30 januar 2015).
- Jure Leskovec, Anand Rajaraman, in Jeffrey David Ullman. 2014. *Mining of Massive Datasets*. 2nd Edition. Kalifornija: Stanford University. Dostopno preko: <http://www.mmms.org/#ver21>. (7 januar 2016).
- Luke, Tierney. 2015. snow. Dostopno preko: <https://cran.r-project.org/web/packages/snow/snow.pdf> (7 maj 2015).
- Martinc, Rok. 2013. *Merjenje sentimenta na družabnem omrežju twiterju*. Magistrsko delo, Ljubljana: Fakulteta za družbene vede.
- Matt, Dowle. 2015. *Benchmarks : Grouping · Rdatatable/data.table Wiki*. Dostopno preko: <https://github.com/Rdatatable/data.table/wiki/Benchmarks-%3A-Grouping>. (7 januar 2016).
- mn3njalknik. 2015. *Slovenski Mnenjalknik (@mn3njalknik) | Twitter*. Twitter račun. Dostopno preko: <https://twitter.com/mn3njalknik/lists>. (7 maj 2016).
- MPQA. 2014. *MPQA Resources*. Dostopno preko: <http://mpqa.cs.pitt.edu/>. (23 marec 2016).
- rdatamining. *Big Data Platforms - RDataMining.com: R and Data Mining*. Dostopno preko: <http://www.rdatamining.com/big-data/big-data-platforms>. (7 januar 2016).
- revolutionanalytics. 2016. „Revolution R renamed Microsoft R, available free to developers and students“. Dostopno preko: <http://blog.revolutionanalytics.com/2016/01/microsoft-r-open.html>. (13 februar 2016).
- Rich, Calaway. 2015. Package doSNOW. Dostopno preko: <https://cran.r-project.org/web/packages/doSNOW/doSNOW.pdf>. (7 januar 2016).
- Roger, D. Peng. 2015a. Interacting with Data using the filehash Package for R. *Department of Biostatistics Johns Hopkins Bloomberg School of Public Health* 2015. Dostopno preko: <https://cran.r-project.org/web/packages/filehash/vignettes/filehash.pdf>. (7 januar 2016).
- . 2015b. Filehash. Dostopno preko: <https://cran.r-project.org/web/packages/filehash/filehash.pdf>. (7 januar 2016).
- Saif. 2015. NRC Emotion Lexicon. Dostopno preko: <http://www.saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>. (7 januar 2016).
- Sang Chung in Sandy Liu. 2011. Predicting Stock Market Fluctuations from Twitter. Dostopno preko: https://www.stat.berkeley.edu/~aldous/157/Old_Projects/Sang_Chung_Sandy_Liu.pdf. (7 januar 2016).
- Sas. Hadoop What is it and why does it matter? Dostopno preko: http://www.sas.com/en_us/insights/big-data/hadoop.html. (7 januar 2016).
- Scott, L. Smith. 2015. The Internet of Things. *Gartner*. Dostopno preko: <http://www.gartner.com/technology/research/internet-of-things/>. (7 januar 2016).
- SentiWordNet. 2013. SentiWordNet. Dostopno preko: <http://sentiwordnet.isti.cnr.it/>. (7 januar 2016).
- Siegrist, Kyle. 2015. The Central Limit Theorem. Dostopno preko: www.math.uah.edu/stat/. <http://www.math.uah.edu/stat/>. (7 januar 2016).
- Signorini, A., Segre, A.M., Polgreen, in P.M. 2015. The use of twitter to track levels of disease activity and public concern in the us during the influenza A H1N1 pandemic. *PloS one*, PloS one, 6(5), e19467, 2011. Dostopno preko: web.iitd.ac.in/~sumeet/sig11.pdf. (26 maj 2015)
- Stahl, Saul. 2006. The evolution of the normal distribution. *Mathematics magazine* 79 (2): 96–113.
- Steve Weston. 2015. Using The foreach Package. Dostopno preko: <https://cran.r-project.org/web/packages/foreach/vignettes/foreach.pdf>. (21 januar 2016)
- Tetsuro Takahashi, Shuya Abe, in Nobuyuki Igata. 2011. Can Twitter Be an Alternative of Real-World Sensors? *Fujitsu Laboratories, Ltd.*, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki 211-8588, , št. 1-1.

- Tom Abate. 2013. Stanford algorithm analyzes sentence sentiment, advances machine learning. *Stanford Engineering*. Dostopno preko: <http://engineering.stanford.edu/news/stanford-algorithm-analyzes-sentence-sentiment-advances-machine-learning>. (21 februar 2015)
- Twitter Developers. API Rate Limits | Twitter Developers. Dostopno preko: <https://dev.twitter.com/rest/public/rate-limiting>. (27 februar 2015)
- Williams, David 2014. World's data doubling every two years.... Blog. *Techday*. Dostopno preko: <http://techday.com/it-brief/news/emc-worlds-data-doubling-every-two-years/182718/>. (8 junij 2015)
- Williams, Graham. 2011. *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*. Let. 2011. Springer New York Dordrecht Heidelberg London.
- World Bank. 2012a. Stock Market Total Value Traded to GDP for Germany. *FRED, Federal Reserve Bank of St. Louis*. Dostopno preko: <https://research.stlouisfed.org/fred2/series/DDDM02DEA156NWDB/>. (3 marec 2016)
- . 2012b. Stock Market Total Value Traded to GDP for Slovenia. *FRED, Federal Reserve Bank of St. Louis*. <https://research.stlouisfed.org/fred2/series/DDDM02SIA156NWDB/>. (3 marec 2016)

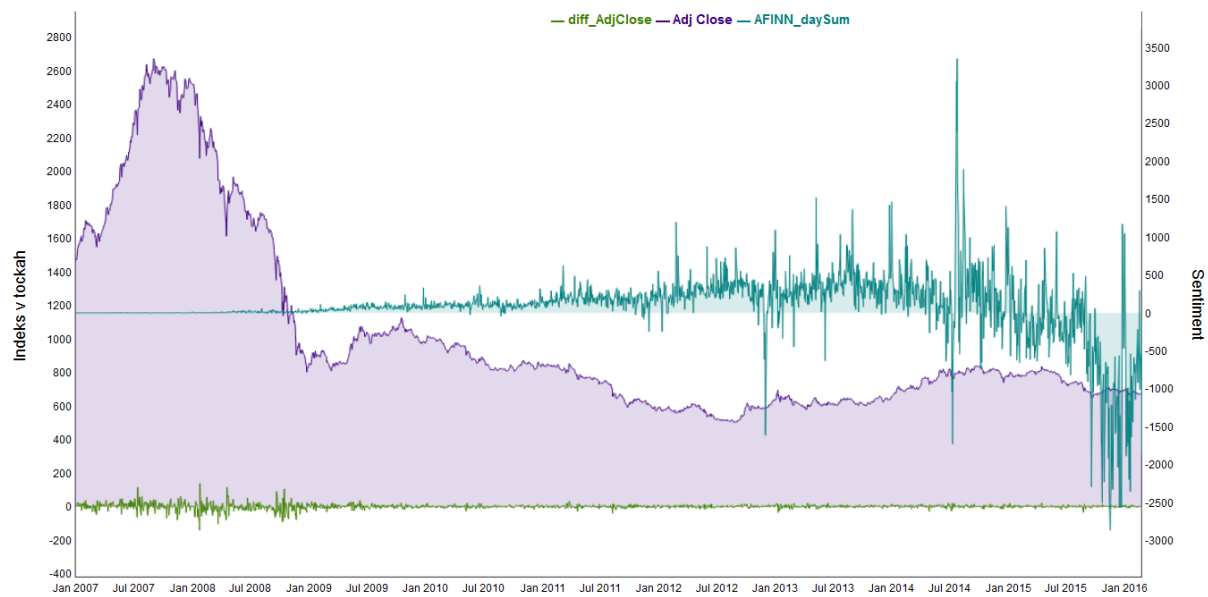
PRILOGA

A - Grafi

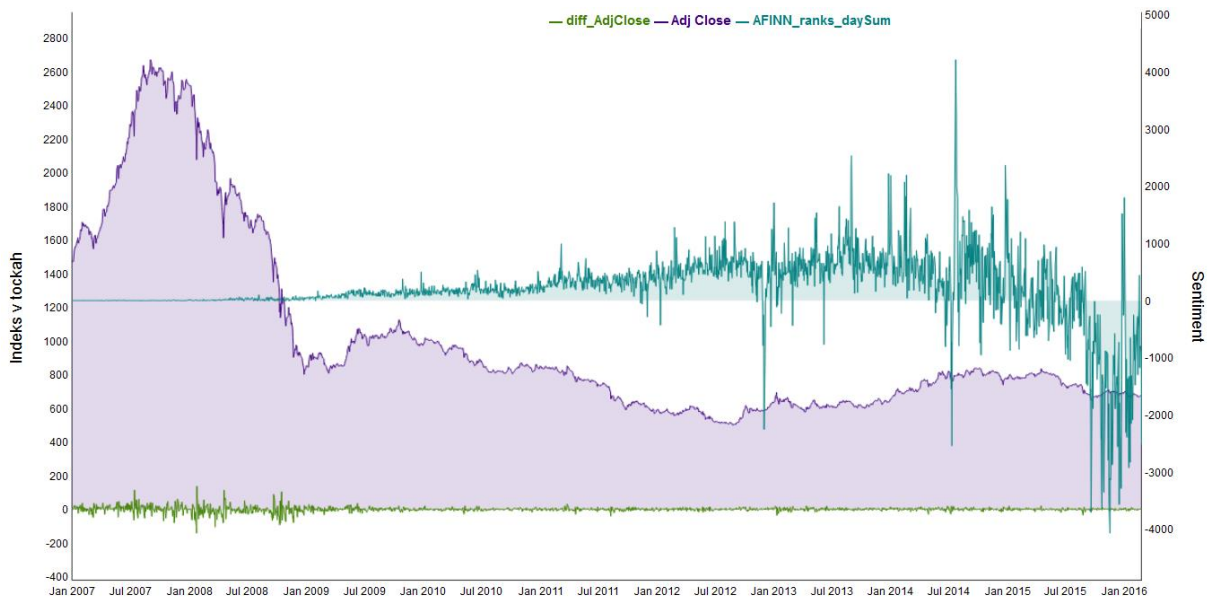
Vrednost tečaja, sprememba tečaja in vsota sentimentov AFINN za delnico Telekom Slovenija (1.1.2014 - 23.2.2016)



Indeks v točkah, sprememba indeksa in vsota sentimentov AFINN



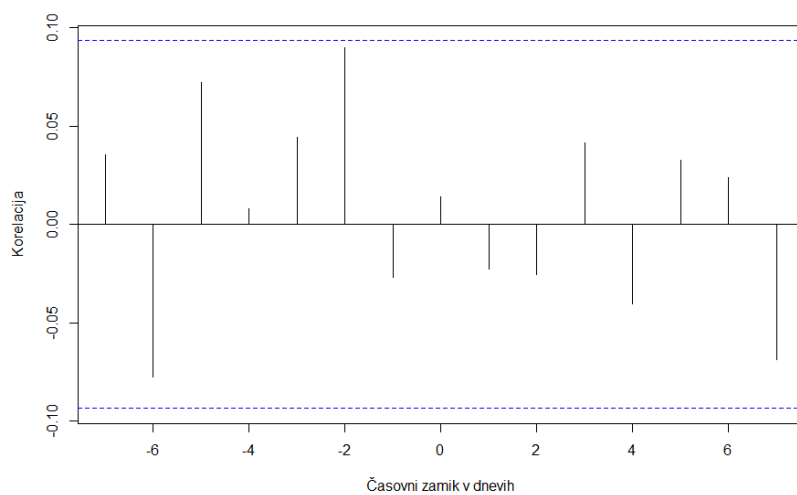
Indeks v točkah, sprememba indeksa in vsota sentimentov AFINN rangi



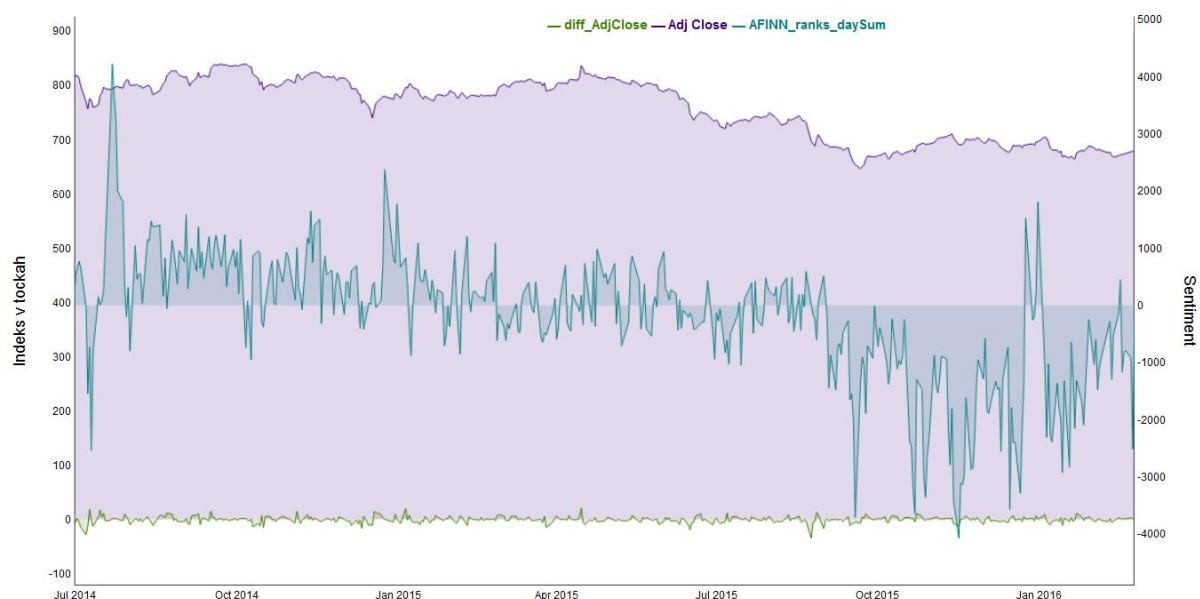
Sprememba sprememb indeksa in sprememba sentimenta AFINN



Sprememba indeksa in sprememba vsote sentimentov AFINN



Vrednost indeksa, sprememba indeksa in vsota sentimenta AFINN rangi (1.6.2014 - 24.2.2016)



B - Seznam Twitter list

Opomba: Na prvem mestu je uporabniški profil, na drugem pa ime liste; npr. – uporabniški profil = »mn3njalnkn«, lista = »slovenija-twitter-1«

"mn3njalnkn", "slovenija-twitter-1"	"mn3njalnkn", "slovenija-twitter-2"	"mn3njalnkn", "slovenija-twitter-3"	"mn3njalnkn", "slovenija-twitter-4"	"24ur", "slovenska-politika"
"24ur", "novinarji"	"24ur_com", "novice"	"StatSlovenia", "SloMedij i"	"reportedly", "slovenia"	"lippylipe", "politika"
"strankar", "slovenija"	"matejivanus", "slovensk a-pomlad"	"rkopac", "Črna-luknja"	"milanznidarsic", "politik a-slo"	"svet24_si", "svet24"
"EvropaMoja", "slo-stranke"	"EvropaMoja", "slo-mediji"	"savicdomen", "slo-twitter-mediji"	"savicdomen", "slo-twitter-politiki"	"bmertelj", "slovenia"
"Val202", "stranke"	"Val202", "politiki"	"Val202", "valovci"	"Svet_KanalA", "svet-novinarji"	"Dnevnik_si", "slovenij a"
"Dnevnik_si", "gospodarstv o"	"kraljb", "politikus"	"GregorHorvatic", "politika-slo"	"politikarica", "polit-stranke"	

C - Koda

Dva primera funkcij iskanja slovenskih uporabnikov. Prvi primer je približno 3600 krat hitrejši od drugega.

Faktor je izračunan na primeru približno 1500ih sporočil, kjer druga funkcija porabi ~ 8h * 60min * 60 sek = 28800sek proti 8sek *prve funkcije*

```
extractVerifiedNames = function(data_temp) {
  word.list <- (str_split(data_temp[,text], "\s+"))
  m <- as.data.table(matrix(as.character(NULL), nrow = (length(word.list)),
    ncol = 1))
  if (!(length(word.list) == 0)) {
    for (i in 1:length(word.list)) {
      d <- (toString(word.list[[i]]))
```

```

set(m, i = i, j = 1L, value = d)
}
}

data_temp[,text := m]

data_temp2 <- data.table(text = as.character(NULL), screenName = as.character(NULL))

for (i in 1:nrow(data_temp)) {
  tmp <- as.data.table(unlist(strsplit(toString(data_temp[i,c(text,screenName)]),
  ", ")))
  tmp[, screenName := tmp[.N]]
  data_temp2 <- unique(rbindlist(list(tmp[!.N,],data_temp2)))
}

data_temp <- data_temp2

rm(data_temp2);gc()

data_temp_filtered <- t(data_temp[, lapply(V1, function(x) grepl(
  slovenian.words, x, perl = T, fixed = F))])
data_temp <- data_temp[rowSums(data_temp_filtered) > 0]

cat("\n","Found", length(unique(data_temp[,screenName])), "Slovenian users: ")
slovenianUsers <- unique(data_temp[,screenName])
cat("\n",slovenianUsers)
return(slovenianUsers)
}

```

Primer časovno potratne funkcije

```

extractVerifiedNames = function(data_temp) {
  countSloUsers<-1
  slovenianUsers<-character()
  slovenian.words<-fread("sloBesede.csv",header = F)
  word.list = str_split(data_temp[,text], "\\s+")
  for(row_number in 1:length(word.list)){
    rowStringList<-unlist(word.list[row_number])
    for(pos_number in 1:length(rowStringList)){
      for(i in 1:nrow(slovenian.words)){
        test<-match(rowStringList[pos_number],slovenian.words[i,V1])
        while(!is.na(test)){
          slovenianUsers[countSloUsers]<-data_temp[row_number,screenName]
          countSloUsers<-countSloUsers+1

```

```
break
}
}
}
}
slovenianUsers<-unique(slovenianUsers, by=names(slovenianUsers))
return(slovenianUsers)
}
```