UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

ZAKLJUČNA NALOGA

UPORABA METOD STROJNEGA UČENJA ZA UGOTAVLJANJE SPOLA UPORABNIKOV TWITTERJA NA PODLAGI VSEBINE NJIHOVIH TVITOV

JAN ŠKORJANC

2017
ZAKLJUČNA NALOGA
ŠKORJANC

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Uporaba metod strojnega učenja za ugotavljanje spola uporabnikov Twitterja na podlagi vsebine njihovih tvitov

(The use of machine learning methods for determining the gender of Twitter users based on the contents of their tweets)

Ime in priimek: Jan Škorjanc
Študijski program: Računalništvo in informatika
Mentor: doc. dr. Branko Kavšek
Somentor: doc. dr. Jernej Vičič

Koper, avgust 2017

Ključna dokumentacijska informacija

Ime in PRIIMEK: Jan ŠKORJANC

Naslov zaključne naloge: Uporaba metod strojnega učenja za ugotavljanje spola uporabnikov Twitterja na podlagi vsebine njihovih tvitov

Kraj: Koper

Leto: 2017

Število listov: 60	Število slik: 28	Število tabel: 4
Število prilog: 1	Št. strani prilog: 1	Število referenc: 69

Mentor: doc. dr. Branko Kavšek

Somentor: doc. dr. Jernej Vičič

Ključne besede: tviti, podatkovno rudarjenje, strojno učenje, Weka, klasifikacija

Izvleček:

V zaključni nalogi je predstavljen problem ugotavljanja spola uporabnikov Twitterja na podlagi vsebine njihovih tvitov. Glavni namen je iz besedila tvita in ostalih podatkov, ki pripadajo določenemu profilu, napovedati ali gre za moškega, žensko ali za profil, ki ga vodi več ljudi. Tak primer so podjetja, blagovne znamke in mediji. Problema smo se lotili s postopkom podatkovnega rudarjenja. Nad podatki je bilo pognanih več algoritmov strojnega učenja, med katerimi je bil v splošnem najbolj učinkovit RandomForest, ki je v 28 sekundah pravilno napovedal 63,42% primerov. Pri analizi najpogostejših besed je bil najkoristnejši atribut opisa profila, v katerem se pojavljajo besede, ki izstopajo glede na posamezen spol. Pri moških so to »man«, »sports« in »fan«, pri osebah ženskega spola pa »lover«, »world« in »ig«. Besede »news«, »follow«, »twitter«, in »updates« najbolj zaznamujejo profile, ki jih vodi več ljudi. Analiza ostalih atributov je pokazala tudi, da dajo ženske največji poudarek na izgled svojega profila.

Key words documentation

Name and SURNAME: Jan ŠKORJANC

Title of the final project paper: The use of machine learning methods for the determining the gender of Twitter users based on the contents of their tweets

Place: Koper

Year: 2017

Number of pages: 60 Number of figures: 28 Number of tables: 4

Number of appendix: 1 Number of appendix pages: 1 Number of references: 69

Mentor: Assist. Prof. Branko Kavšek, PhD

Co-Mentor: Assist. Prof. Jernej Vičič, PhD

Keywords: tweets, data mining, machine learning, Weka, classification

Abstract:

The final project paper presents the problem of determining the gender of Twitter users based on the contents of their tweets. The main goal is to predict whether tweet's text and other Twitter data belong to a male, female or a brand. We used the data mining process for this purpose. Among many machine learning algorithms tested on the data the most efficient proved to be RandomForest which correctly classified 63,42% of instances in 28 seconds. The analysis of most used words shows that the most useful attribute is profile description attribute which contains words that strongly belong to only one group of Twitter users. Male users mostly use words like »man«, »sports« and »fan«, females use »lover«, »world« and »ig«, while brands use »news«, »follow«, »twitter« and »updates«. Further analysis shows that female users care the most about how their Twitter profile looks like.

ZAHVALA

Zahvaljujem se mentorju doc. dr. Branku Kavšku in somentorju doc. dr. Jerneju Vičiču za svetovanje in pomoč pri izdelavi zaključne naloge. Prav tako se zahvaljujem celotni družini za podporo med študijem.

KAZALO VSEBINE

KAZALO PREGLEDNIC

KAZALO SLIK IN GRAFIKONOV

KAZALO PRILOG

SEZNAM KRATIC

3D	tri dimenzionalno
IBM	znano računalniško podjetje (angl. International Business Machines Corporation)
SMS	sistem kratkih sporočil (angl. Short Message Service)
URL	enotni naslov vira (angl. Uniform Resource Locator)
API	programski vmesnik (angl. Application Programming Interface)
HTTP	protokol za izmenjavo hiperteksta na spletu (angl. HyperText Transfer Protocol)
GNU	operacijski sistem in zbirka programske opreme (angl. GNU's Not Unix)
CLI	vmesnik z ukazno vrstico (angl. Command-line Interface)
SIGKDD	skupnost na področju podatkovnega rudarjenja in podatkovne znanosti (angl. Special Interest Group on Knowledge Discovery and Data Mining)
CSV	format za besedilno datoteko, ki vsebuje vrednosti ločene z vejico (angl. Comma Separated Values)
LibSMV	knjižnica za metodo podpornih vektorjev (angl. Library for Support Vector Machines)
ARFF	format za besedilno datoteko, ki vsebuje opis atributov in množico instanc (angl. Attribute-Relation File Format)
ASCII	ameriški standard za zapis in izmenjavo znakov (angl. American Standard Code for Information Interchange)
UTC	univerzalni koordinirani čas (angl. Coordinated Universal Time)
TF-IDF	statistični podatek, ki razkrije pomembnost posamezne besede v zbirki besedil (angl. Term Frequency-Inverse Document Frequency)
ROC	statistični podatek za analizo klasifikatorja (angl. Receiver Operating Characteristic)

1 UVOD

V zadnjih dvajsetih letih je na raznih področjih količina podatkov skokovito narasla. Po poročanju International Data Corporation (IDC) je v letu 2011 velikost vseh podatkov na svetu znašala 1,8 ZB ($1{,}8 \times 10^{21}$ B), kar pomeni, da se je povečala za približno devetkrat v petih letih. Če se bo trend nadaljeval, se pričakuje, da se bo količina podatkov povečala za dvakrat na vsaki dve leti [5].

To je spodbudilo podjetja, da bi iz podatkov pridobili določene informacije. Začelo se je predvsem zavedati, da lahko velike količine podatkov vsebujejo vzorce, ki jih na prvi pogled ni mogoče razbrati. Na začetku so bili računalniki počasni in je bilo pridobivanje ter obdelava podatkov počasnejša. Z večanjem moči in nižanjem cene računalnikov, pa je podatkovno rudarjenje postalo vse bolj uporabno. Sedaj se ga uporablja tako za znanstvene, kot za komercialne namene [46].

Poleg tega si danes nihče več ne predstavlja življenja brez socialnih omrežij. V letu 2016 je v Sloveniji kar 75% oseb starih 16–74 let redno uporabljalo internet, 38% pa vsaj enkrat uporabljalo socialna omrežja [43]. Uporaba metod podatkovnega rudarjenja za analizo socialnih omrežij je dandanes ena najbolj aktualnih raziskovalnih področij računalništva in podatkovnih znanosti. Zato bo tudi glavni cilj zaključne naloge analiza socialnega omrežja Twitter.

V prvem delu zaključne naloge so predstavljene osnovne značilnosti strojnega učenja in podatkovnega rudarjenja – temu je posvečeno drugo poglavje. V tretjem poglavju je podan opis uporabljene programske opreme. To sta Twitter, socialno omrežje, s katerega so bili pridobljeni podatki in Weka, orodje, s katerim smo jih klasificirali. Uporabljeni podatki so v četrtem poglavju tudi opisani, kjer je glavni poudarek na njihovi strukturi in predobdelavi. Sledi peto poglavje, ki vsebuje osnovno razumevanje delovanja uporabljenih algoritmov. Za klasifikacijo so bili uporabljeni algoritmi, ki nam kot rezultat vrnejo pravila (PART), drevo (J48) ali pa zgolj podatek o pravilno klasificiranih primerih (Naivni Bayes). V šestem poglavju so predstavljeni rezultati in primerjava uporabljenih algoritmov, kar je eden izmed ciljev zaključne naloge. Zaključna naloga se konča s sedmim poglavjem, ki poda zaključke in predstavi možnosti za nadaljnje delo.

2 STROJNO UČENJE IN PODATKOVNO RUDARJENJE

Za analizo podatkov v zaključni nalogi so bili uporabljeni principi strojnega učenja in podatkovnega rudarjenja.

2.1 Strojno učenje

Strojno učenje je po definiciji Arthurja Samuela zmožnost učenja računalnika, ne da bi ga eksplicitno programirali [34]. Ker je ta definicija nastala leta 1959, jo je Tom M. Mitchell posodobil in podal bolj natančno definicijo tega pojma, ki pravi, da se računalniški program uči na podlagi izkušenj le, če s pridobivanjem le-teh njegova učinkovitost narašča [33].

V splošnem se strojno učenje uporablja z namenom, da se določena stvar izpopolni na podlagi izkušenj iz preteklosti, kjer je pomembno, da postopek poteka samodejno, brez interakcije uporabnika. Lahko bi se reklo, da gre za programiranje na podlagi primerov, saj uporabnik prepusti računalniku, da reši določen problem, namesto, da bi ga sam rešil s pomočjo programiranja. Pomembno je tudi, da je algoritem, s katerim računalnik operira učinkovit, kar pomeni, da ne zasede veliko prostora in za izvedbo ne porabi veliko časa [41]. Leta 2016 je besedna zveza strojno učenje dosegla vrh Gartnerjevega »cikla navdušenja«, kar oznanja, da je bila med najpopularnejšimi pojmi na področju tehnologije [15]. Prepleta se z več področji, kot so prepoznavanje vzorcev, umetna inteligenca, računalniški vid, statistika, pa tudi matematika in fizika. Poznamo mnogo primerov, kjer je strojno učenje uporabno. Med najpogostejše sodijo:

- prepoznavanje na roke napisanih črk,
- zaznavanje obraza,
- zaznavanje nezaželenih elektronskih sporočil,
- določanje teme novic,
- razumevanje govora,
- postavljanje diagnoze bolnikom,
- ugotavljanje navad strank,
- zaznavanje spletnih goljufij,
- vremenska napoved [41].

Algoritmi strojnega učenja so del večjega procesa, imenovanega podatkovno rudarjenje, ki je bil uporabljen tudi v zaključni nalogi.

2.2 Podatkovno rudarjenje

Za razliko od strojnega učenja, je **podatkovno rudarjenje** bolj praktična veda. To je proces zbiranja in analize podatkov, iz katerih izluščimo koristne informacije. Ker je teh podatkov veliko, je iz njih brez pomoči računalnika težko pridobiti smiselne informacije. Predstavljeni so v obliki velike množice atributov, kjer je eden izmed njih razredni atribut. Po podatkih lahko »rudarimo« na več načinov, kar je natančneje predstavljeno v razdelku 2.2.2. Temeljni cilj pri vseh načinih je iz podatkovne zbirke pridobiti določeno znanje, ki je med drugim pridobljeno tudi z algoritmi strojnega učenja. Podano je lahko v obliki odločitvenih pravil, dreves ipd. [68].

2.2.1 Primeri podatkovnega rudarjenja

Najpogostejši razlog za podatkovno rudarjenje je zaslužek, vendar je lahko uporabljeno tudi zgolj za osvojitev novega znanja in razumevanje določenega problema. V realnem svetu se postopek izvaja na velikih datotekah z nekaj tisoč ali celo nekaj sto tisoč vrsticami, saj nam večje število podatkov nudi boljši približek dejanskemu stanju [68].

V igrah je podatkovno rudarjenje uporabljeno predvsem za iskanje zmagovalnega postopka igre pri miselnih igrah, kot je šah. Podjetjem pomaga pri iskanju vzrokov za dobiček ali izgubo, predvidevanju novih trendov in pridobivanju novih strank. V medicini podatkovno rudarjenje pomaga pri postavitvi diagnoz pacientom glede na sindrome, ki jih kažejo in napovedovanju verjetnosti bolezni določenega pacienta. Na glasbenem področju je uporabljeno predvsem za klasifikacijo glasbe glede na zvrst in iskanje podobnih vzorcev v pesmih, kar lahko nakazuje na plagiatorstvo.

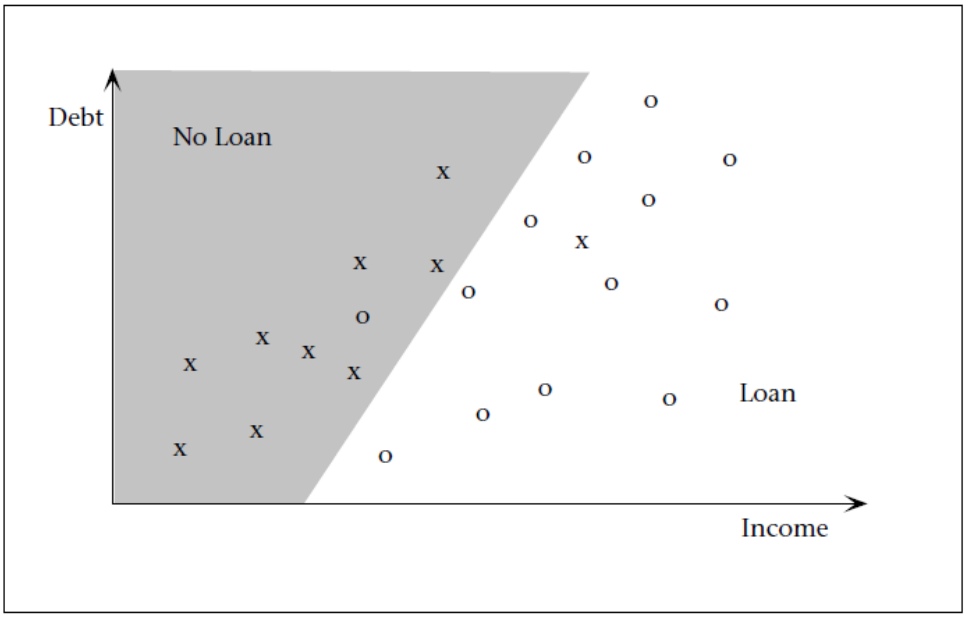
To je zgolj nekaj primerov, kjer je podatkovno rudarjenje uporabljeno, vendar je danes prisotno še na mnogo drugih področjih [64].

2.2.2 Metode podatkovnega rudarjenja

Največja cilja podatkovnega rudarjenja sta napovedovanje in opis podatkov. Napovedovanje vključuje uporabo določenih vrednosti spremenljivk, s katerimi si pomagamo določiti nove, za nas zanimive vrednosti. Pri opisu podatkov pa je pomembno, da najdemo vzorce in pravila, ki jih lahko potem človek interpretira.
Meje med opisom podatkov in napovedovanjem niso natančno določene, saj lahko iz podatkov, ki so opisani s pravili, izpeljemo tudi napovedi in obratno. Kljub temu pa je pomembno, da iz podatkov dobimo določeno znanje, do katerega lahko pridemo z uporabo različnih metod [12].

2.2.2.1 Klasifikacija

Klasifikacija (angl. classification) je postopek učenja, pri katerem so podatki razvrščeni v enega ali več vnaprej definiranih razredov. Je ena izmed najpogostejših uporabljenih metod v podatkovnem rudarjenju, uporabljena pa je bila tudi na primeru te zaključne naloge. Poznamo več algoritmov, ki jih lahko pri tem uporabimo. Nekateri izmed njih so predstavljeni v nadaljevanju. Na sliki 1 lahko vidimo grafični prikaz klasifikacije, kjer so podatki klasificirani v dve skupini, glede na prihodek in dolg. Osenčeni so tisti primeri, ki ne dosegajo pogojev za pridobitev kredita, neosenčeni pa tisti, ki ga dosegajo. [12]

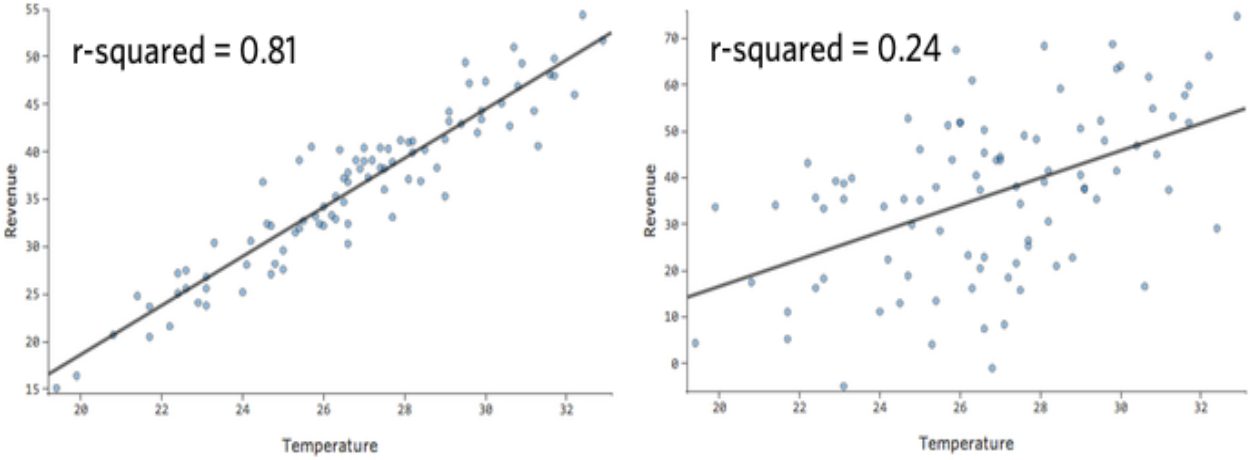


Slika 1: Primer klasifikacije. (vir: [12])

2.2.2.2 Regresija

Podobno kot pri klasifikaciji je **regresija** (angl. regression) postopek učenja, pri katerem podatke razvrstimo v razrede. Razlika pa je v tipu spremenljivke samega razreda. Pri klasifikaciji morajo biti vrednosti spremenljivke razreda že vnaprej definirane in so nominalne, pri regresiji pa je vrednost razreda številska vrednost. Z regresijo ugotovimo ali odvisnost med spremenljivkami sploh obstaja in kako močna je. To nam pove koeficient korelacije.

Z njim lahko npr. napovedujemo verjetnost, da bo pacient preživel, če zboli za določeno boleznijo, verjetnost, da se bo določen produkt dobro prodajal glede na kakovost in količino oglaševanja ipd. [12] Rezultat lahko predstavimo z grafom, ki je podan na sliki 2. Ta prikazuje odvisnost med dvema spremenljivkama. Bolj kot so podatki (predstavljeni s pikami) oddaljeni od črte, ki je bila narejena s pomočjo regresije, manjši je njihov korelacijski koeficient in s tem manjša medsebojna odvisnost. Tako vidimo, da sta spremenljivki na levem grafu bolj povezani, kot pa tisti na desnem grafu slike 2.

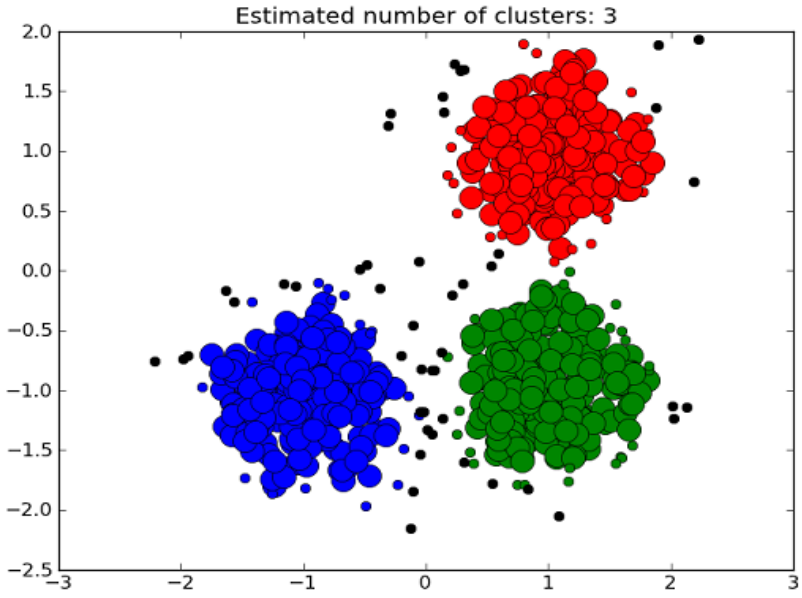


Slika 2: Primer regresije. (vir: [45])

2.2.2.3 Razvrščanje v skupine

Cilj **razvrščanja v skupine** (angl. clustering) je najti končne množice, v katere lahko razporedimo podatke, glede na njihove lastnosti. Množice so med seboj lahko disjunktne, kar pomeni, da ne vsebujejo skupnega elementa, lahko je ena množica podmnožica druge množice ali pa se množici sekata. Postopek se uporablja v primerih, kjer je bistveno podatke razdeliti na več podmnožic ali skupin.

Za razliko od klasifikacije, kjer primere razvrstimo v določen razred, se pri razvrščanju v skupine razredna spremenljivka ne uporabi, saj jo nadomestijo kar množice same [12]. V biologiji je znan primer uporabe v sistematiki, kjer lahko na podlagi lastnosti ugotovimo vrsto organizma, v medicini za razločevanje vrste tkiva na 3D sliki, na socialnih omrežjih pa za razvrščanje v skupine ljudi z enakimi interesi [62]. Na sliki 3 lahko vidimo primer razvrščanja v skupine, ki so ločene po barvi. V večini primerov se najdejo tudi primeri, ki ne spadajo v nobeno izmed skupin, zaradi različnih razlogov kot so napake, izjeme ali pri organizmih mutacije. Taki podatki so v našem primeru označeni s črno barvo in so predstavljeni v razdelku 2.2.2.6.



Slika 3: Primer razvrščanja v skupine. (vir: [8])

2.2.2.4 Povzemanje

Povzemanje (angl. summarization) vključuje metode, katerih namen je najti kratek in jedrnat opis množice podatkov. Enostaven primer takega postopka je izračun povprečja in standardne deviacije za vsako spremenljivko. V praksi pa se uporabljajo bolj kompleksne metode, ki vsebujejo odvajanje in druge zahtevnejše izračune. Cilj postopka je samodejno ustvarjeno poročilo, ki mora biti kratko in podprto z različnimi vrstami vizualizacije [12].

2.2.2.5 Modeliranje odvisnosti

Modeliranje odvisnosti (angl. dependency modelling) se ukvarja z iskanjem modela, ki opisuje ključne odvisnosti med spremenljivkami. Postopek se deli na dva nivoja:

- strukturni nivo (angl. structural level)
- kvantitativni nivo (angl. quantitative level)

Prvi, največkrat v grafični obliki, predstavlja spremenljivke, ki so med seboj odvisne. Pri drugem, kvantitativnem nivoju, pa je pomembna informacija o moči same odvisnosti [12]. Uporablja se lahko za ugotavljanje nakupovalnih navad strank v trgovinah z živili. Trgovina strankam ponudi kartico, ki prinaša popuste, v zameno pa v svojo bazo shranjuje podatke o kupljenih izdelkih. Tako trgovina pridobi vse podatke, s katerimi lahko ugotovijo verjetnost, da bo neka stranka kupila izdelek X ob predpostavki, da je kupila izdelek Y [47]. Na sliki 4 so predstavljeni primeri asociacijskih pravil, ki so del modeliranja odvisnosti. Opazimo lahko oba prej omenjena nivoja, saj so v stolpcu Transaction odvisnosti predstavljene na strukturnem, v ostalih treh pa na kvantitavnem nivoju.

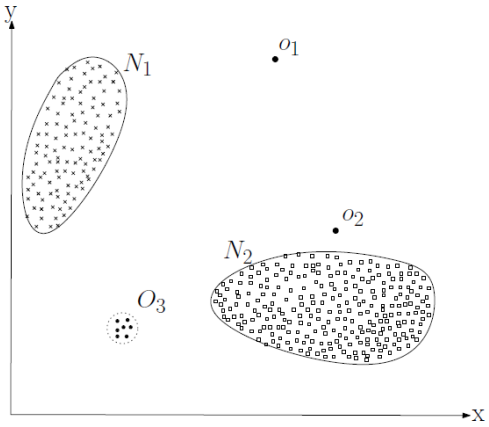
Transaction	Support	Confidence	Lift
Canned Beer → Soda	1%	20%	1.0
Canned Beer → Berries	0.1%	1%	0.3
Canned Beer → Male Cosmetics	0.1%	1%	2.6

Slika 4: Primer modeliranja odvisnosti. (vir: [21])

2.2.2.6 Zaznavanje spremembe

Zaznavanje sprememb (angl. change and deviation detection ali anomaly detection) daje glavni poudarek na iskanju podatkov z znatnim odstopanjem (anomalijo). Z drugimi besedami je to postopek iskanja primerov, ki ne ustrezajo pričakovanemu vzorcu v določeni množici podatkov. Ponavadi so to napake, ki pa lahko ključno vplivajo na ostale podatke, še posebej v medicini, kjer lahko taki podatki zaznajo tumor ali pa v bančništvu, kjer lahko taki podatki nakazujejo na poizkus goljufije [4]. Poznamo več vrst anomalije podatkov.

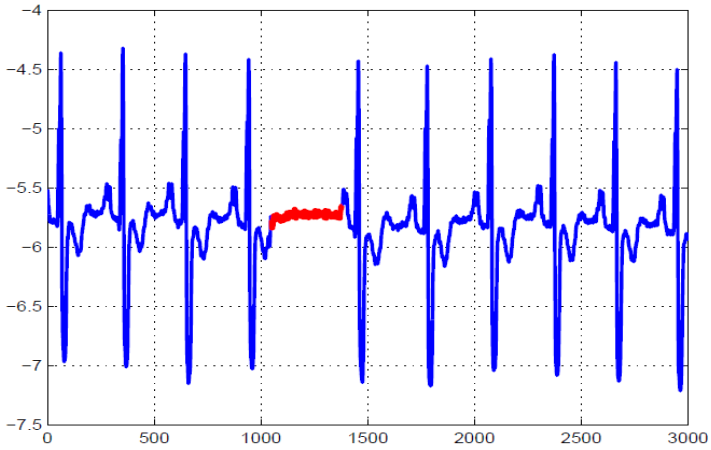
Točkovna anomalija (angl. point anomaly) predstavlja osamljen primer, ki odstopa v primerjavi z ostalimi podatki. Je najenostavnejša vrsta anomalije in zato se pri iskanju anomalij nanjo najbolj osredotočimo. Na sliki 5 je točkovna anomalija predstavljena s točkama o_1 in o_2.



Slika 5: Primer točkovne anomalije. (vir: [4])

Kontekstualna anomalija (angl. Contextual anomaly) predstavlja primer, ki odstopa v primerjavi z ostalimi podatki zgolj v določenih okoliščinah. Eden izmed takih primerov je temperatura ozračja, kjer je vrednost 30°C običajna vrednost v poletnem obdobju, v zimskem obdobju pa predstavlja anomalijo.

Skupinska anomalija (angl. Collective anomaly) je skupina povezanih primerov, ki odstopa v primerjavi z ostalimi podatki. To pomeni, da lahko posamezen primer ne predstavlja anomalije kot samostojen podatek, jo pa kot del skupine podatkov v neki množici. Primer predstavlja slika 6, kjer vrednost podatkov v skupini obarvanih z rdečo barvo odstopa [4].



Slika 6: Primer skupinske anomalije. (vir: [4])

Točkovna anomalija se lahko pojavi v vseh množicah podatkov, skupinska anomalija pa zgolj v med seboj povezanih podatkih. Vsak podatek lahko predstavlja več anomalij. Tako lahko del podatka v skupinski anomaliji predstavlja točkovno anomalijo, ki je ob dodanem kontekstu del kontekstualne anomalije. Tak primer je večkrat zapored izmerjena temperatura 60°C v zimskem letnem času. V primeru, da anomalijo v podatkih zaznamo, je dobro take podatke posebej označiti. Temu namenu služijo **podatkovne oznake** (angl. data labels). To so oznake, ki instancam določijo ali je vrednost podatka anomalija, ali ne. Tako označevanje je zahtevno in drago, saj je ponavadi delano ročno, s strani strokovnjaka na tem področju. Glede na obseg podatkov poznamo tri načine zaznavanja in določanja anomalije.

Nadzorovano zaznavanje anomalij poteka tako, da se podatke najprej razvrsti v dve skupini. To je skupina običajnih podatkov in tistih podatkov, ki odstopajo. Nato vsak nov podatek primerjamo s podatki iz obeh skupin. Nov podatek tako dobi oznako skupine, kateri je bolj podoben. Tak način zaznamujeta dve težavi. Prva je manjša količina podatkov, ki predstavljajo anomalijo v primerjavi s tistimi podatki, ki anomalije nimajo. Druga pa je iskanje značilnosti, ki bi povezovala vse anomalije. Teh je namreč zelo veliko in so si med seboj različne.

Polnadzorovano zaznavanje anomalij zahteva zgolj predhodno podatkovno zbirko običajnih podatkov. Iz njih naredi učni model, katerega primerja z novimi instancami. Model nato izračuna verjetnost podobnosti glede na podatkovno zbirko in določi ali je v podatku anomalija ali ne. Uporabi se lahko tudi zgolj podatkovna zbirka podatkov, ki predstavljajo anomalijo, vendar je v tem primeru učni model težje zgraditi, saj je vsaka vrsta anomalije različna od ostalih.

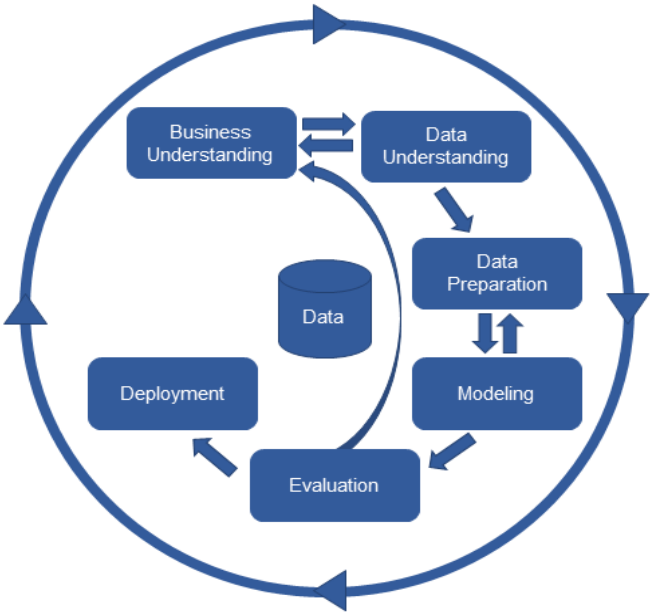
Nenadzorovano zaznavanje anomalij ne potrebuje predhodnih podatkov in je zato najbolj uporabljeno izmed vseh zaznavanj anomalij. Predvideva, da je običajnih podatkov dosti več kot tistih, ki vsebujejo anomalijo, zato instance, ki so najmanj podobne ostalim, določi za podatke z anomalijo. Tako zaznavanje ima veliko napako v primeru, da je podatkov z anomalijo zelo veliko [4] [59].

2.2.3 Procesni modeli

Podatkovno rudarjenje se je najprej uvrščalo pod **KDD** (Knowledge Discovery in Databases) proces. Ta je postopek ločil na pet faz, med katerimi je bilo tudi podatkovno rudarjenje. Z naraščanjem in širjenjem uporabe podatkovnega rudarjenja, se je na tem področju razvila potreba po novih standardih. Tako je najprej nastal model **SEMMA** (Sample, Explore, Modify, Model Assess), ki je bil razvit na SAS inštitutu. Postopek je podoben KDD procesu, saj je tudi ta sestavljen iz petih faz. Razlika pa je v njihovih imenih, zato imena podatkovno rudarjenje v procesu SEMMA ne najdemo. Danes je najbolj uporabljen postopek **CRISP-DM**, po katerem smo se pri izvajanju zaključne naloge tudi zgledovali, zato je bolj podrobno opisan v nadaljevanju [1].

2.2.3.1 CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) je procesni model, ki opisuje pristop za učinkovito podatkovno rudarjenje. Sestavljen je iz cikla šestih faz, med katerimi je možno vračanje na prejšnje faze, kar prikazuje slika 7.



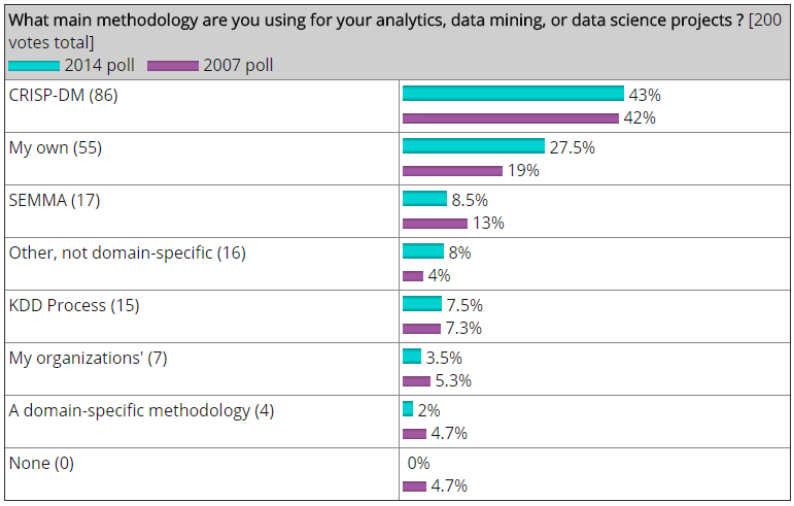
Slika 7: Shema CRISP-DM procesa. (vir: [24])

Omenjene faze so:

1. **Razumevanje poslovanja/problema** (angl. business/problem understanding) je začetna faza, ki se osredotoča na cilje in zahteve s stališča končnega uporabnika. Cilj faze je opredelitev problema samega projekta.
2. **Razumevanje podatkov** (angl. data understanding) se začne s pridobitvijo podatkov. Ponavadi so zelo neurejeni, zato jih je potrebno znati razumeti v širšem kontekstu in opisati. V tej fazi lahko podamo že določene hipoteze ali pa v njih najdemo primere, ki vsebujejo očitne anomalije ali napake.
3. **Priprava podatkov** (angl. data preparation) je namenjena urejanju in pripravljanju na obdelavo dobljenih neurejenih množic podatkov. Postopek se v večini primerov ponovi večkrat, kar vzame največ časa. V tej fazi dobimo končno množico podatkov, kar pomeni, da se podatkov več ne spreminja.
4. **Modeliranje** (angl. modelling) pokriva obdelavo podatkov. Najprej določimo tehnike in algoritme, ki jih bomo uporabili ter njihove parametre. Nato jih zaženemo nad podatki, ki smo jih pripravili v prejšnji fazi.
5. **Vrednotenje** (angl. evaluation) poteka tako, da modele iz prejšnje faze ovrednotimo. Pomembno je, da rezultate natančno analiziramo in jih znamo interpretirati. S tem dobimo tudi končni rezultat, ki ga predamo končnemu uporabniku.

6. **Uporaba** (angl. deployment) služi predstavitvi podatkov. Če bi rekli, da smo v prejšnjih fazah z delom že končali, bi se zmotili, saj ravno v tej fazi nastopi bistvo celotnega projekta. Podatke je treba znati razložiti in interpretirati tako, da jih bo končni uporabnik razumel in jih znal uporabiti v svojo korist [1].

Med leti 2002 in 2014 so bile opravljene štiri ankete [22] [23] [25] [26], ki so pokazale, da je CRISP-DM glavni procesni model uporabljen v večini podjetij. Rezultati ankete, ki jo je opravila skupnost KDNuggets [25], so predstavljeni na sliki 8.



Slika 8: Rezultati KDNuggets ankete. (vir: [25])

CRISP-DM pa ni zadnji večji procesni model, saj je IBM leta 2015 razvil **ASUM-DM** (Analytics Solutions Unified Method for Data Mining/Predictive Analytics), ki je razširjena različica modela CRISP-DM [18].

2.2.4 Podatkovno rudarjenje in etika

Eden izmed glavnih izzivov podatkovnega rudarjenja je tudi etika. Podatki pogosto vključujejo informacije, o katerih ljudje ne govorimo preveč radi, čeprav so v določenih primerih taki podatki bistveni. Na primeru določene bolezni so osebni podatki, kot so starost, spol, teža, pomembni, saj lahko ključno vplivajo na rezultate. Neetično pa rudarjenje postane takrat, ko podatke uporabljamo za stvari, ki nimajo neposredne povezave z njimi.

Višina in teža nimata neposrednega vpliva na znamko kupljenega telefona, oziroma tudi, če ga imata, je taka informacija nekoristna. Še bolj moramo biti na to pozorni, ko uporabljamo podatke o rasi, verskem prepričanju, finančnem stanju, itd. Na tem področju se poraja precej vprašanj o tem, katere podatke je sploh dovoljeno uporabiti in katere ne [68]. Pri podatkovnem rudarjenju moramo biti pozorni tudi na pravilno obveščenost udeležencev. Malo ljudi je namreč pripravljenih deliti svoje podatke z ostalimi ljudmi in organizacijami, če od tega nimajo nikakršne koristi. Priporočeno je, da pred dostopom do podatkov ljudi obvestimo o:

- razlogu za dostop do njihovih podatkov,
- kako bodo podatki uporabljeni,
- kdo bo imel dostop do podatkov,
- varnosti in zasebnosti podatkov,
- na kakšen način se lahko podatke posodobi [63].

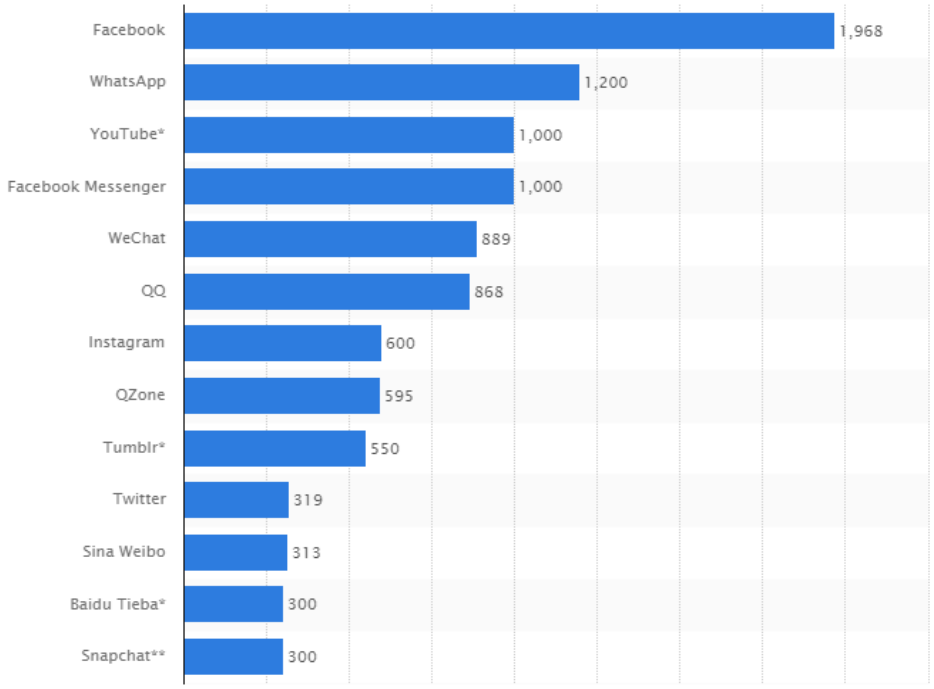
Pomembno je tudi, da varujemo zasebnost podatkov in da do njih dostopa omejeno število ljudi. Temu se lahko izognemo tako, da namesto imen in priimkov uporabimo šifre [63] [68].

3 UPORABLJENA PROGRAMSKA OPREMA

Pri izdelavi zaključne naloge so bile uporabljene objave uporabnikov socialnega omrežja Twitter in program Weka.

3.1 Twitter

Twitter je socialno omrežje, kjer uporabniki komunicirajo s pomočjo **tvitov** (angl. tweets). To so 140 znakov dolga sporočila, ki jih lahko objavijo zgolj registrirani uporabniki. Twitter so leta 2006 razvili Jack Dorsey, Noah Glass, Biz Stone in Evan Williams. Sedaj je v podjetju zaposlenih okoli 4000 ljudi, uporablja pa ga 328 milijonov aktivnih uporabnikov [67]. To ga uvršča na deseto mesto med socialnimi omrežji po številu aktivnih uporabnikov, kar prikazuje slika 9. Na Twitter je prijava možna preko spletnega brskalnika, aplikacije na pametnem telefonu, v določenih državah pa tudi preko SMS sporočila [55]. Preveden je v 34 jezikov, med katerimi trenutno ni slovenščine. [54]. Na dan ameriških predsedniških volitev leta 2016 je bilo objavljenih 40 milijonov tvitov povezanih z volitvami. S tem je Twitter premagal tudi največje socialno omrežje Facebook, kar nakazuje na to, da je Twitter najbolj uporabljeno socialno omrežje za novice [48].



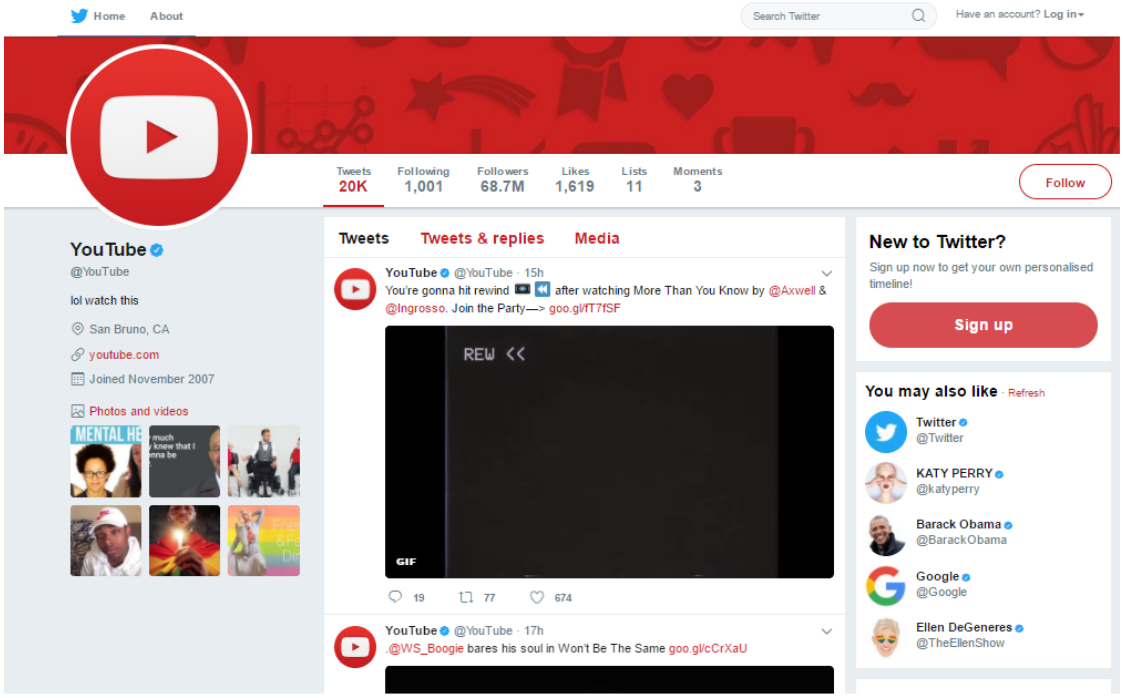
Slika 9: Število aktivnih uporabnikov na socialnih omrežjih (v milijonih). (vir: [44])

3.1.1 Osnovne značilnosti

Osnovni način sporazumevanja so **tviti**. Njihov namen lahko predstavlja navadno objavo ali odgovor na zapisano objavo. Ta ima posebno obliko, saj se začne z znakom @, za katerim sledi uporabniško ime osebe, ki ji želimo odgovoriti (primer @uporabniskoime). Objavo lahko tudi delimo, čemur pravimo »retvitanje«, ali »všečkamo« podobno kot na ostalih socialnih omrežjih. Na začetku je bila komunikacija omejena na 140 znakov ne glede na vsebino tvita. Leta 2016 so omejitev omilili in tako kot znake izključili URL povezave, slike in črke zapisane za @, ki označujejo določenega uporabnika [9]. Poleg @ ima tudi znak # posebno mesto, saj z njim označimo tvite s povezano vsebino. Tako lahko tvitu dodamo »#datamining«, kar pomeni, da se vsebina objave navezuje na podatkovno rudarjenje. Od leta 2015 lahko kot tvit objavimo tudi anonimno anketo z največ štirimi odgovori [10]. Komunikacija je mogoča tudi po privatnih sporočilih imenovanih DM (direct messages).

Slika 10: Primer tvita z odgovorom in tvita z anketo. (vir: [51] [50])

Ob registraciji si lahko vsak uporabnik oblikuje svoj **profil**. Določi lahko svoje ime, sliko, lokacijo, barvo teme, opis in rojstni dan. Na vsakem profilu se poleg teh podatkov prikaže število objavljenih tvitov in njihova vsebina, število oseb, ki jim oseba sledi in število sledilcev tega profila. To prikazuje slika 11. Vsak si lahko tudi svoje tvite zaklene, kar pomeni, da mu lahko sledijo in tvite berejo samo osebe, ki jih sam potrdi. V primeru, da oseba nima zaklenjenih tvitov, pa jih lahko beremo tudi, če v Twitter nismo prijavljeni. Tviti oseb, ki jim sledimo se nam prikazujejo na časovnici. V primerjavi s Facebookom in Instagramom, kjer se najprej izpišejo najbolj vroče objave, je na Twitterju to možnost mogoče izključiti. Tako so tviti prikazani po časovnem vrstnem redu, kar mnogim bolj ugaja [11].



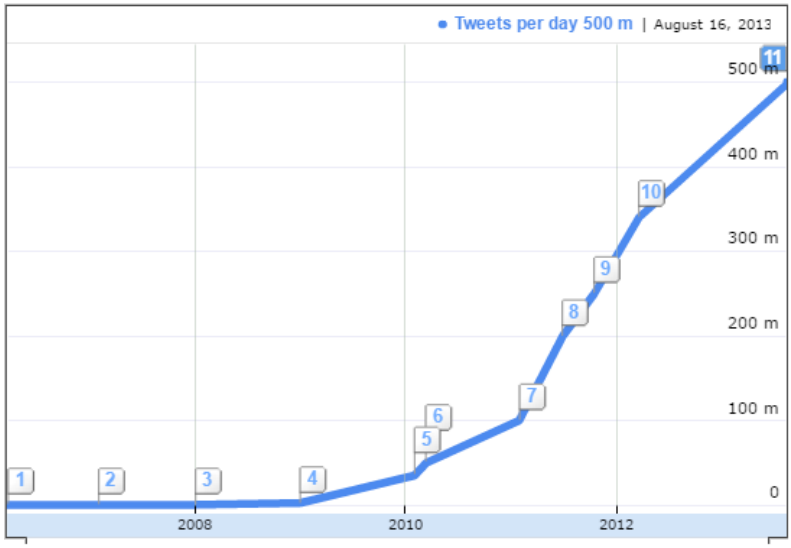
Slika 11: Primer Twitter profila. (vir: [52])

3.1.2 Dodatne funkcije

Uporabniki lahko ustvarijo tudi sezname, kamor dodajo profile z enakimi interesi. Tako imajo vse novice na enem mestu in šport na drugem mestu. Seznami so lahko javni (do njih lahko dostopajo vsi) ali pa zasebni (do njih dostopa zgolj oseba, ki seznam ustvari).
Profile lahko tudi utišamo, kar pomeni, da se kljub sledenju tviti na časovnici ne prikažejo ali pa blokiramo. Znane osebnosti lahko svoj račun tudi potrdijo, s čimer uporabnike prepričajo, da profil ni lažen. Leta 2016 je Twitter uvedel tudi predvajanje raznih prireditev v živo. Samo v prvi četrtini leta 2017 so predvajali za več kot 800 ur takih vsebin [56].

3.1.3 Rast popularnosti

Twitter je nastal leta 2006. Že leta 2007 je bilo poslanih več kot 5000 tvitov dnevno. Ta številka se je v letu 2008 povečala za 60-krat in narasla na 300000, v letu 2010 pa je bilo vsak dan poslanih že več kot 50 milijonov objav. Leta 2013 je število preseglo 500 milijonov, danes pa znaša okoli 650 milijonov [19] [49]. Število dnevno objavljenih tvitov prikazuje slika 12.



Slika 12: Prikaz števila dnevno objavljenih tvitov. (vir: [19])

3.1.4 API

Twitter je poznan tudi po zelo dobrem API-ju za razvijalce. Tako lahko preko HTTP protokola kdorkoli omogoči pretok tvitov na svoj računalnik v realnem času. To lahko stori zgolj pod določenimi omejitvami in pogoji uporabe. Na tak način so bili shranjeni tudi podatki, ki so bili uporabljeni v zaključni nalogi [53].

3.2 Weka

Weka (Waikato Environment for Knowledge Analysis) je programska oprema, ki vsebuje zbirko algoritmov za strojno učenje in podatkovno rudarjenje. Program je v celoti napisan v programskem jeziku Java in je odprtokoden, saj je na voljo pod GNU licenco. Ime je dobil po vrsti ptice, ki živi na Novi Zelandiji, saj je bil tam tudi program narejen [58]. Podpira splošne algoritme za klasifikacijo, regresijo, razvrščanje v skupine in ostale metode podatkovnega rudarjenja [13].

Weko lahko uporabljamo na več načinov, zato je razdeljena na 5 delov. To so raziskovalec (angl. explorer), preizkuševalec (angl. experimenter), tok znanja (angl. knowledgeFlow), delovno okolje (angl. workbench) in preprost vmesnik z ukazno vrstico (angl. simple CLI) [16]. Za potrebe zaključne naloge smo uporabili prva dva načina. Zadnja različica programa je 3.8.1 (za razvijalce 3.9.1), deluje pa na Windows, Linux in Mac OS X sistemih.

3.2.1 Zgodovina

Projekt Weka se je s finančno pomočjo novozelandske vlade začel razvijati leta 1993. Večji del programa je bil napisan v C-ju z deli kode v Prologu. Prva različica, ki je izšla leta 1994, ni bila namenjena javnosti, saj je šlo zgolj za beta različico programa. Za javno uporabo je program izšel leta 1996. Naslednja izdaja je izšla leta 1997 in je vsebovala 8 različnih algoritmov. Z večanjem kompleksnosti programa, ga je bilo v taki obliki čedalje težje vzdrževati, zato so se odločili program na novo implementirati. Tako je leta 1999 izšla različica 3.0, ki je bila v celoti napisana v Javi. Leta 2005 je razvojna ekipa prejela SIGKDD nagrado, kar je največja nagrada s področja podatkovnega rudarjenja [16]. 14. aprila 2016 je izšla zadnja stabilna različica 3.8.1.

3.2.2 ARFF format

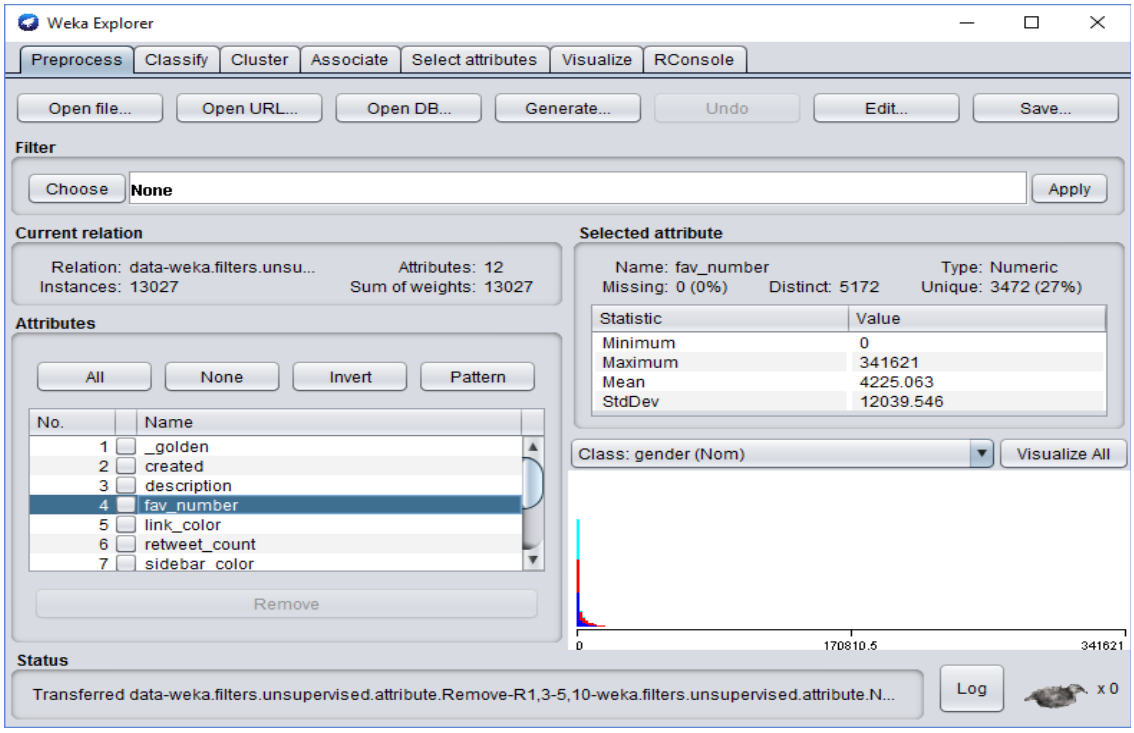
Algoritmi se izvajajo na podatkih, ki morajo biti v program naloženi. Weka podpira več formatov datotek, kot so CSV, LibSMV in C4.5, vendar vse pretvori v poseben format imenovan ARFF. Ta vsebuje glavo, v kateri podamo ime relacije in opišemo atribute, ki so lahko nominalni, numerični, datumi ali nizi znakov. Glavi sledijo z vejico ločeni podatki, ki jih navedemo v vrstnem redu kot smo jih razvrstili v glavi. V primeru, da so podatki nominalni, moramo v glavo navesti vse vrednosti, ki jih atribut lahko zavzame. Numerična vrednost je lahko vsako realno število, datum pa podamo v obliki kot ga navedemo v glavi. Ponavadi je to v obliki »LLLL-MM-DD-uu:mm:ss«. Niz lahko vsebuje vse možne znake ASCII kode [57] [68]. Na sliki 13 vidimo, da podatkovna zbirka z imenom iris vsebuje pet atributov. Štirje so numerični, zadnji pa nominalen in zavzema vrednosti »{Iris-setosa, Iris-versicolor, Iris-virginica}«. Zbirka vsebuje deset instanc.

```
@RELATION iris

  @ATTRIBUTE sepallength  NUMERIC
  @ATTRIBUTE sepalwidth   NUMERIC
  @ATTRIBUTE petallength  NUMERIC
  @ATTRIBUTE petalwidth   NUMERIC
  @ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA

  5.1,3.5,1.4,0.2,Iris-setosa
  4.9,3.0,1.4,0.2,Iris-setosa
  4.7,3.2,1.3,0.2,Iris-setosa
  4.6,3.1,1.5,0.2,Iris-setosa
  5.0,3.6,1.4,0.2,Iris-setosa
  5.4,3.9,1.7,0.4,Iris-setosa
  4.6,3.4,1.4,0.3,Iris-setosa
  5.0,3.4,1.5,0.2,Iris-setosa
  4.4,2.9,1.4,0.2,Iris-setosa
  4.9,3.1,1.5,0.1,Iris-setosa
```

Slika 13: Primer datoteke ARFF formata. (vir: [57])

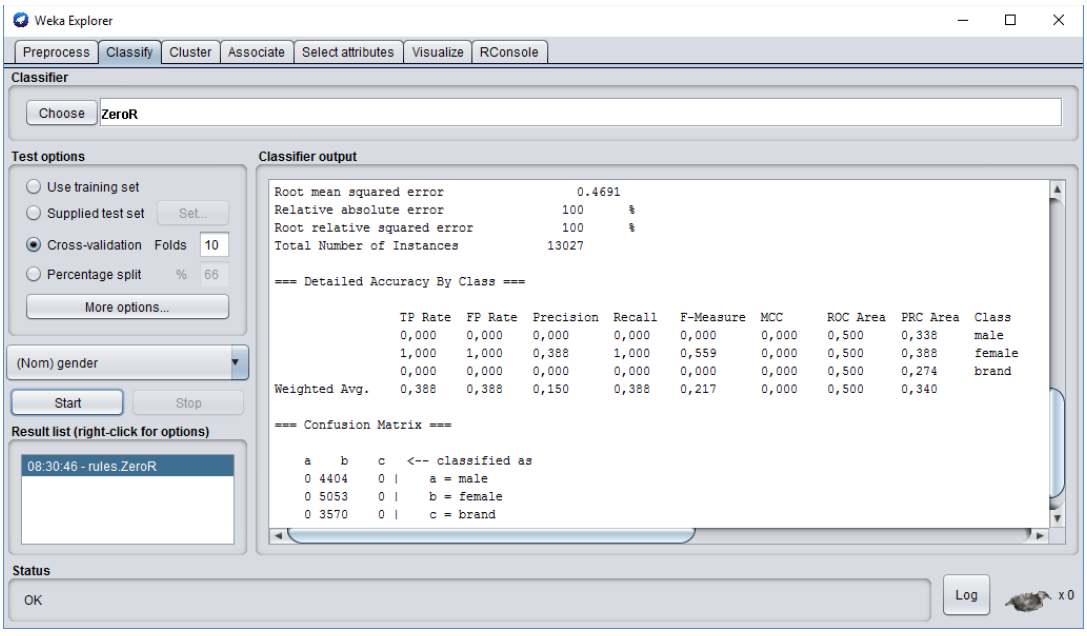
3.2.3 Raziskovalec

Glavni vmesnik v programu Weka je **raziskovalec**. Sestavljen je iz več zavihkov, kjer vsak ponazarja eno izmed metod podatkovnega rudarjenja. V program moramo najprej naložiti podatkovno zbirko. To lahko storimo preko URL povezave, SQL baze ali datoteke, ki je oblikovana po enem izmed veljavnih formatov [13]. V glavnem oknu se nato prikaže ime relacije, število instanc, število in vrednosti atributov za vsako instanco ter nekaj statističnih podatkov, kar vidimo na sliki 14. Podatke lahko nato prečistimo ročno ali pa z uporabo enega izmed algoritmov za prečiščevanje. Po vsakem narejenem koraku lahko podatkovno zbirko na novo shranimo ali pa korak razveljavimo.



Slika 14: Raziskovalec v Weki.

Zavihek Classify je namenjen klasifikaciji in regresiji. V njem izberemo algoritem, ki ga želimo uporabiti in način testiranja uspešnosti klasifikacije. Vsebuje tudi okno, kamor se izpiše rezultat, kar prikazuje slika 15. Ta rezultat lahko shranimo tudi v ločeno datoteko.

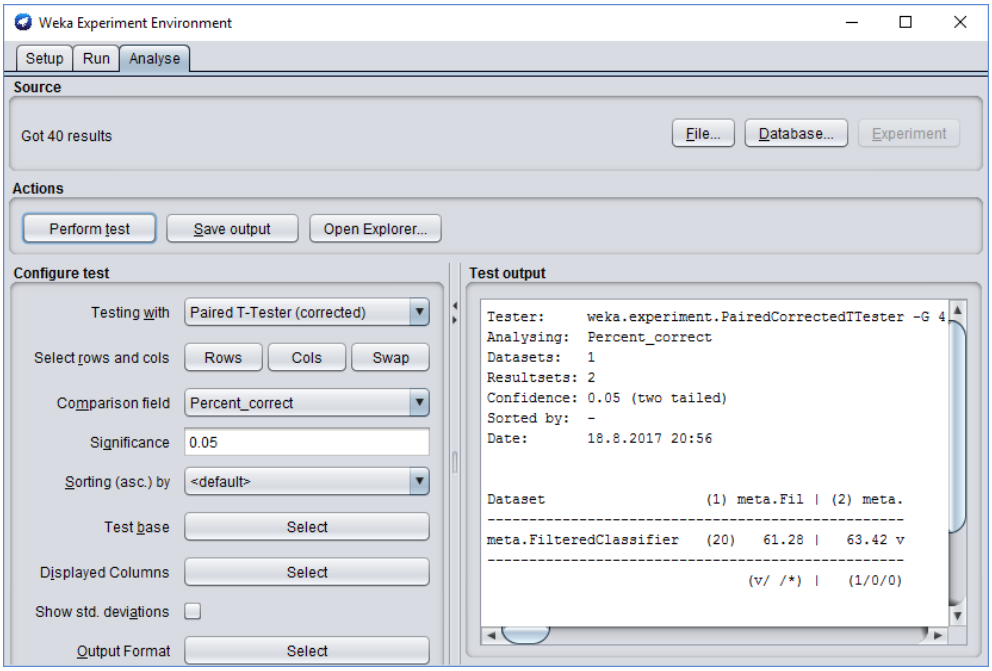


Slika 15: Prikaz rezultata algoritma

Poleg omenjenih zavihkov imamo še zavihek za razvrščanje v skupine, iskanje asociacijskih pravil, izbiro primernih atributov in zavihek, ki predstavlja vizualizacijo. Zaključna naloga dela s temi zavihki ne zajema.

3.2.4 Preizkuševalec

Preizkuševalec je namenjen primerjavi algoritmov na isti ali različnih podatkovnih zbirkah. Preizkusi se lahko več različnih algoritmov ali pa en algoritem z različnimi parametri. Vsebuje zavihek za pripravo izvajanja algoritmov, kjer se poleg algoritma določi način testiranja, število ponovitev in podatkovno zbirko nad katero se bo algoritem izvedel. Znak za začetek izvajanja se določi v drugem zavihku. V tretjem zavihku, ki je namenjen statistični analizi in je prikazan na sliki 16, pa se med seboj primerja rezultate in ugotovi ali se rezultati med seboj znatno razlikujejo. S tem je mogoče tudi statistično dokazati, da je določen algoritem boljši od ostalih algoritmov [42].



Slika 16: Preizkuševalec v Weki.

3.2.5 Ostali vmesniki

Weka vsebuje še druge vmesnike, ki v zaključni nalogi niso bili uporabljeni. To so tok znanja, s katerim lahko določimo tok izvajanja podatkov [17], delovno okolje, ki je zbirka vseh funkcij v Weki in preprost vmesnik z ukazno vrstico, kjer lahko poženemo algoritme neposredno z ukazom.

4 METODOLOGIJA DELA S PODATKI

Postopek operacij nad podatki je bil izveden po procesnem modelu CRISP-DM, ki je opisan v razdelku 2.2.3.1.

4.1 Razumevanje poslovanja/problema

Projekt je potekal individualno, brez sodelovanja z določenim podjetjem, zato ocena tveganja in poslovni cilji niso podani. Je pa v tem koraku podana kratka opredelitev problema.

Na določenih socialnih omrežjih je med osebne podatke možno določiti tudi spol. Kljub temu, nekateri tega podatka ne želijo deliti z drugimi ali pa spol pozabijo določiti. V primeru, da želimo izvedeti spol določene osebe, si moramo pomagati na drugačen način.
Cilj projekta je na podlagi tvitov določiti ali gre za osebo ženskega spola, osebo moškega spola ali za račun, iz katerega tvite objavlja več oseb. Tak primer so podjetja, novice in organizacije, ki so v zaključni nalogi imenovani blagovna znamka. Cilj zaključne naloge je predvsem pridobitev odgovorov na vprašanja kot so:

- Kako dobro besede v tvitih določijo spol uporabnika?
- Katere besede najbolj zaznamujejo določen spol?
- Ali obstaja povezava med barvo ozadja profila in spolom uporabnika?

Smiselno je poudariti tudi dejstvo, da so uporabljeni podatki v angleškem jeziku. Angleščina je za razliko od slovenščine jezik, v katerem oblika besed neposredno ne razkrije spola osebe. V slovenščini bi bilo napovedovanje lažje, saj bi ob uporabi besedne zveze »sem šla« z veliko verjetnostjo predpostavili, da gre za osebo ženskega spola in ob uporabi besedne zveze »sem šel« predpostavili, da gre za osebo moškega spola. Posledično diplomska naloga temelji predvsem na pomenskem delu besed.

4.2 Razumevanje podatkov

Podatki, ki so bili uporabljeni za potrebe diplomske naloge so javno dostopni na spletu [20]. Zbrani so bili 26. oktobra 2015 med 12:39 in 12:40 ter isti dan med 13:19 in 13:20 na način, ki je omenjen v razdelku 3.1.4.

Za vsako instanco, ki predstavlja en tvit je skupina ljudi pregledala profil in presodila spol osebe uporabnika profila. Na koncu so podali še vrednost, ki predstavlja prepričanost v pravilnost svoje odločitve.

4.2.1 Struktura podatkov

Podatkovna zbirka je shranjena v CSV formatu in vsebuje okoli 20000 instanc. Opisane so s 26 atributi, ki so:

- **unit_id:** identifikator vsake instance. Je celoštevilska vrednost, ki je unikatna za vsako instanco.
- **_golden:** vsebuje logično vrednost (TRUE, FALSE), ki določa ali instanca dosega zlati standard.
- **_unit_state:** vsebuje vrednost »finalized« za navadne instance in »golden« za tiste, ki dosegajo zlati standard.
- **_trusted_judgments:** predstavlja število oseb, ki so presodili spol osebe določenega tvita. Za zlati standard je vrednost večja, medtem ko je za ostale vrednost enaka 3.
- **_last_judgment_at:** določa čas in datum zadnje podane razsodbe o spolu instance.
- **gender:** predstavlja razredni atribut, ki je lahko vrednosti »male« za moškega, »female« za žensko, »brand« za blagovno znamko in »unknown« za instanco, kjer spol ni bil določen.
- **gender:confidence:** je realno število med 0 in 1, ki predstavlja verjetnost za podani spol.
- **profile_yn:** predstavlja nominalno vrednost, kjer »no« pomeni, da profil v času ocenjevanja spola ni bil dostopen (zaklenjen profil), »yes« pa, da je bil dostop do njega v tem času omogočen.
- **profile_yn:confidence:** je realno število med 0 in 1, ki predstavlja verjetnost v obstoj profila.
- **created:** vsebuje datum in čas, ko je bil profil ustvarjen.
- **description:** je opis profila.
- **fav_number:** predstavlja število tvitov, ki jih je uporabnik »všečkal«.
- **gender_gold:** v primeru, da instanca dosega zlati standard, predstavlja določen spol. V nasprotnem primeru vrednosti ni.

- **link_color:** predstavlja barvo povezave kot zapis v šestnajstiškem številskem sistemu.
- **name:** je uporabniško ime uporabnika.
- **profile_yn_gold:** predstavlja nominalno vrednost, kjer »no« pomeni, da instanca ne dosega zlatih standardov. »yes« pa pomeni, da dosega.
- **profileimage:** vsebuje povezavo do profilne slike uporabnika.
- **retweet_count:** predstavlja število retvitov tvita.
- **sidebar_color:** predstavlja barvo teme, kot zapis v šestnajstiškem številskem sistemu.
- **text:** predstavlja besedilo tvita.
- **tweet_coord:** predstavlja koordinate lokacije, od koder je bil tvit objavljen.
- **tweet_count:** predstavlja število tvitov, ki jih je uporabnik do tedaj objavil.
- **tweet_created:** predstavlja datum in čas, ko je bil tvit objavljen.
- **tweet_id:** je identifikator tvita.
- **tweet_location:** predstavlja lokacijo tvita.
- **user_timezone:** predstavlja časovni pas uporabnika, ki je tvit objavil.

4.3 Priprava podatkov

Faza priprave podatkov je bila razdeljena na dve podfazi. Za lažje in hitreje opravljeno delo sta bila uporabljena odprtokodna progama za urejanje besedil LibreOffice Calc [30] in Notepad++ [35].

4.3.1 Prva faza

Prvi del v sklopu priprave podatkov je bilo njihovo urejanje v tako obliko, da jih Weka pravilno prebere. Pri tem je nastalo več problemov. Največjo težavo so povzročale vejice, saj so bili podatki shranjeni v CSV formatu, ki vrednosti loči glede na položaj vejice. Problem nastane, če je znotraj ene vrednosti (npr. v atributu besedila tvita) več vejic, kar Weka razume kot več različnih vrednosti. Takih datotek program ne more prebrati, saj misli, da je v eni instanci več vrednosti kot atributov. Prav tako so narekovaji uporabljeni za označevanje atributov, ki vsebujejo niz, kar pomeni, da je bilo potrebno vse vejice in narekovaje, ki so bili uporabljeni kot vrednost atributa, nadomestiti z drugimi znaki.

Vejica je bila tako spremenjena v » @@@«, dvojni narekovaj v » *** «, enojni narekovaj pa v »€€«.

Problem so predstavljali tviti, ki so vsebovali neveljavne znake. To so znaki, ki jih Weka ne more prebrati. Takih primerov je bilo nekaj deset, zato so bili ročno odstranjeni.
Soočiti se je bilo potrebno tudi s tviti, ki so vsebovali veliko presledkov ali prehodov v novo vrstico. V ARFF formatu je vsaka instanca zapisana v novi vrstici, zato tvit, ki vsebuje prehod v novo vrstico Weki povzroča težave. To je bilo odpravljeno s pomočjo regularnih izrazov, ki so prehod v novo vrstico zamenjali s presledkom.

4.3.2 Druga faza

Druga faza je vsebovala dodatno čiščenje podatkov. Potrebno je bilo izločiti nekoristne in ponavljajoče se atribute ter pregledati ali so vse vrednosti na pravem mestu. Nekoristni so bili unikatni atributi (identifikator tvita) ali pa atributi, ki niso nudili za nas pomembnih informacij (atribut, ki določa čas in datum zadnje podane razsodbe o spolu instance). V uporabljeni podatkovni zbirki so bili tako odstranjeni atributi, ki so predstavljeni v tabeli 1.

Tabela 1: Odstranjeni atributi

Izbrisan atribut	Razlog izbrisa
_unit_id	Nekoristnost atributa
_unit_state	Vsebovan že v atributu _golden
_trusted_judgments	Nekoristnost atributa
_last_judgment_at	Nekoristnost atributa
profile_yn	Nekoristnost atributa
profile_yn:confidence	Nekoristnost atributa
gender_gold	Vsebovan že v atributu gender
name	Nekoristnost atributa
profile_yn_gold	Nekoristnost atributa
profileimage	Več kot 60% manjkajočih vrednosti*
tweet_coord	Več kot 80% manjkajočih vrednosti
tweet_id	Nekoristnost atributa
tweet_location	Nekoristnost atributa (podatek je možno ročno spremeniti, zato so bile v večini primerov lažne vrednosti)

*V atributu profileimage so kot vrednosti podane povezave do spletne strani, kjer se slika nahaja. V ta namen je bil napisan program, ki pregleda ali se na podani povezavi nahaja slika ali ne. Program je pokazal, da je več kot 60% povezav brez slik, zato je bil atribut odstranjen.

Odstranjene so bile tudi instance, ki so imele vrednost atributa »gender_confidence« enako 0 ali več kot 1, saj to predstavlja napako v vrednosti podatka. Prav tako so bile izločene instance, kjer je bila vrednost razrednega atributa »unknown«, saj take instance ne določajo spola in so nekoristne. Poleg tega so imele nekatere vrednosti atributa »user_timezone« napačne vrednosti (ime kraja namesto ime časovnega pasa), zato so bile ročno pretvorjene in spremenjene v časovne pasove od »UTC-12« do »UTC+13«, glede na njihov položaj.
Kot je opisano v razdelku 3.2.2, Weka sprejema zgolj atribute določenih tipov. Poleg tega, pa nekateri algoritmi delujejo zgolj nad podatki z numeričnimi in nominalnimi vrednostmi, zato je bilo potrebno podatke, ki so vsebovali niz znakov, pretvoriti. Pretvorjeni so bili tudi atributi, ki kot vrednost vsebujejo datumski zapis. Vrednost atributa v takih primerih ponazarja čas v milisekundah, ki je pretekel od 01.01.1970, 01:00:00 do vrednosti, ki jo datum predstavlja. Na koncu so bili odstranjeni še atributi, ki so vsebovali napačen vrstni red vrednosti. Mnogi pa so vsebovali prazne vrednosti, vendar taki podatki Weki ne povzročajo težav.

4.3.3 Podatkovne zbirke

Podatki, ki so bili v obeh fazah prečiščeni, so bili uporabljeni za histograme. Vsebujejo 13 atributov in 18420 instanc.

Za klasifikacijo pa so bili podatki še dodatno prečiščeni, saj so bile uporabljene zgolj instance, kjer je razredni atribut zagotovo pravilen. Tako so bili odstranjeni primeri, ki so vsebovali vrednost »gender_confidence« različno od 1. Po tem koraku je bila vrednost tega atributa povsod enolična, zato je postal tudi ta atribut nekoristen in je bil odstranjen. V podatkovni zbirki uporabljeni za algoritme, ki delujejo na podatkih z datumskimi vrednosti je tako ostalo 12 atributov in 13027 instanc.

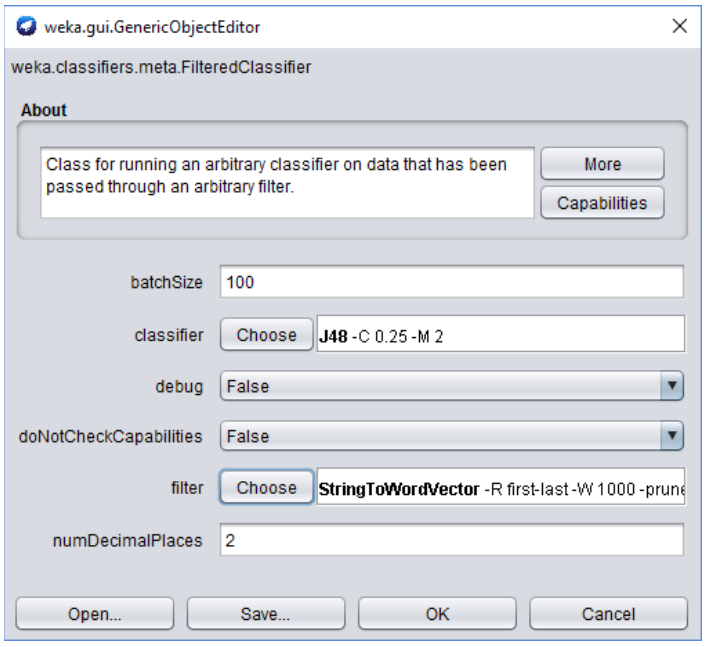
Za algoritme, ki pa ne podpirajo datumov, je bilo treba datume pretvoriti v nominalne vrednosti. Postopek je bil narejen z eno izmed funkcij za urejanje podatkov, ki samodejno pretvori datumske vrednosti v nominalne. Atribut »created«, ki predstavlja datum in čas, ko je bil račun ustvarjen, je tako izgubil svojo vrednost, zato je bil odstranjen. Ta podatkovna zbirka vsebuje 11 atributov in 13027 instanc.

5 MODELIRANJE

Za delo s podatki je bilo uporabljenih več algoritmov, ki jih Weka podpira. To so J48, PART, Naivni Bayes, SMO, RandomForest in IBk. Našteti algoritmi niso zmožni obdelave podatkov, ki kot vrednost vsebujejo niz znakov. Posledično je bilo potrebno uporabiti FilteredClassifier, ki omogoča, da so podatki pred klasifikacijo dodatno obdelani. Vlogo obdelovalnega algoritma je imel StringToWordVector, za vrednotenje algoritmov pa je bil uporabljen način k-kratnega prečnega preverjanja (angl. k-fold cross validation).

5.1 FilteredClassifier

FilteredClassifier omogoča delo algoritma na dodatno prečiščenih podatkih. Deluje tako, da najprej izvede algoritem, ki podatke obdela, nato pa nad obdelanimi podatki izvede algoritem za klasifikacijo. Kljub spremembi strukture podatkov med procesom, ostanejo podatki po koncu izvajanja nespremenjeni [36]. Slika 17 prikazuje grafični vmesnik za FilteredClassifier v programu Weka, kjer je kot filter izbran algoritem za obdelovanje, kot classifier pa algoritem za klasifikacijo podatkov.



Slika 17: Grafični vmesnik FilteredClassifierja

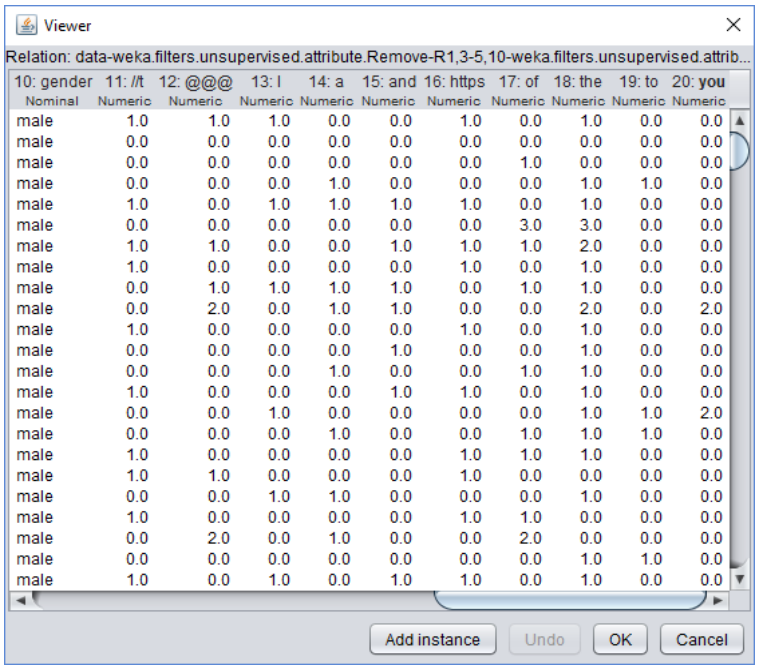
5.2 StringToWordVector

StringToWordVector je obdelovalni algoritem, ki obstoječim atributom v podatkovni zbirki doda nove atribute. Ti za vsako instanco predstavljajo frekvenco pojavitve besede, po kateri se atribut imenuje. Novi atribut nastane zgolj v primeru, da se beseda pojavi v vsaj eni instanci. V tem primeru je vrednost atributa enaka 1, v instancah, kjer se beseda ne pojavi, pa je vrednost enaka 0 [68]. Algoritem deluje s pomočjo »tokenizerja«, ki niz razdeli na več delov. V njem se določijo znaki, ki predstavljajo konec besede, ob vsakem zaznanem znaku pa se določi nov atribut. Na koncu se za vsako instanco preveri katere atribute (besede) vsebuje [65] [68].

Algoritem StringToWordVector ima še nekatere druge možnosti. Atribute lahko loči od ostalih atributov z določitvijo predpone, rezultat lahko shrani v novo datoteko, združi enake besede z različno začetnico, normalizira dolžino besedila za vsako instanco in omeji število novih atributov (besed). V tem primeru algoritem deluje tako, da kot rezultat prikaže najbolj pogoste besede. Lahko se tudi določi, da algoritem ne izpiše zgolj 0 ali 1, temveč število pojavitve besede v vsaki instanci. Ena izmed bolj pomembnih lastnosti algoritma je tudi možnost neupoštevanja besed, ki se pojavljajo velikokrat in nimajo bistvenega pomena. Take angleške besede so na primer »a«, »an«, »get« ipd.

Za bolj kompleksne primere se lahko uporabi tudi »TF-IDF« transformacija, kar vsakemu atributu določi vrednost o pogostosti pojavitve glede na ostale primere [6] [28].

Slika 18 prikazuje rezultat, ki nastane po uporabi StringToWordVector algoritma na določeni podatkovni zbirki. Za vsako instanco so dodani atributi besed in število njihovih pojavitev.



Slika 18: Rezultat algoritma StringToWordVector

S pomočjo tega algoritma so bili ustvarjeni tudi rezultati, predstavljeni v histogramih.

5.3 J48

J48 klasifikator spada med odločitvena drevesa. Je eden izmed najbolj uporabljenih pristopov v podatkovnemu rudarjenju, saj kot rezultat ustvari drevo, ki ga je enostavno brati in razumeti. Med gradnjo drevesa manjkajoče vrednosti ne upošteva. Algoritem je znan tudi po imenu C4.5 [40], J48 pa je zgolj njegova implementacija v Weki [37]. Deluje na podlagi količine informacije, ki jo določen atribut vsebuje, imenovane informacijski prispevek (angl. Information gain).

Postopek se začne z določitvijo atributa, ki najbolje razdeli podatkovno zbirko (ima največji informacijski prispevek). Tako dobimo več podmnožic, na katerih se postopek ponovi rekurzivno na najboljšem izmed preostalih atributov. Postopek se konča, ko vse instance pripadajo svojemu razredu ali po določenem številu opravljenih rekurzij [61]. V Weki ima algoritem tudi možnost rezanja drevesa, ki ga lahko določimo z minimalno količino primerov, ki jih mora za določen list vsebovati. To naredi drevo manjše in bolj pregledno, izgubi pa se natančnost algoritma.

5.4 PART

PART vsebuje kombinacijo algoritmov C4.5 (J48) [40] in RIPPER (Jrip) [7], kot rezultat pa ustvari odločitvena pravila. Od ostalih algoritmov se razlikuje po postopku nastanka pravil. Najprej zgradi odločitveno drevo, v katerem se list z največ instancami pretvori v pravilo. Deluje na način deli in vladaj, kar pomeni, da je postopek rekurziven. To poteka dokler vsaka izmed instanc ne pripada vsaj enemu izmed pravil. Glavna ideja je zgraditi delno odločitveno drevo, kar pomeni, da nekatere veje nimajo svojih poddreves [14].

5.5 Naivni Bayes

Naivni Bayes je implementiran s pomočjo verjetnostnega računa, ki izračuna verjetnost, da instanca pripada določenemu razredu na podlagi frekvenc in vrednosti v podatkovni zbirki. Algoritem predvideva, da so vsi atributi med seboj neodvisni, kar pa ni vedno res. Je zelo hiter algoritem [37]. Za točno določen problem je redko med najboljšimi, njegova dobra lastnost pa je konstantnost, saj se težko najde primer, kjer bi bil med slabšimi algoritmi. Za interpretacijo je enostaven tako, da ga lahko razumejo tudi ljudje, ki niso tako izkušeni na tem področju [69]. Formula po kateri se izračuna verjetnost je predstavljena na sliki 19.

$$P(A \mid B) = \frac{P(B \mid A)\,P(A)}{P(B)}$$

Slika 19: Bayesova formula. (vir: [60])

kjer je:

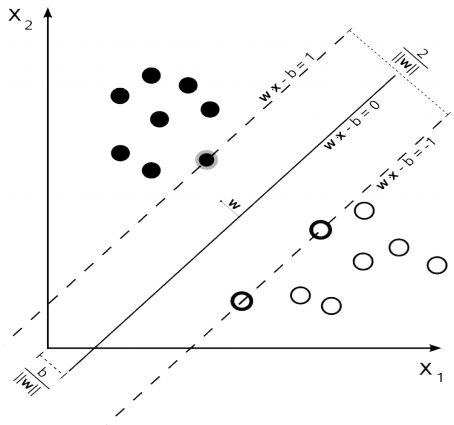
- A vrednost razreda,
- B vrednost atributov instance.

Klasifikator izračuna verjetnosti za vsak razred na podlagi vseh atributov za določeno instanco. Instanca je nato klasificirana v razred z najvišjo izračunano verjetnostjo [60].

5.6 SMO

SMO (angl. sequential minimal optimization) je optimizacijski algoritem, ki deluje po principu metode podpornih vektorjev (angl. SVM - support vector machines) [38].

Algoritem razdeli množico podatkov na dva dela, kjer vsak pripada svojemu razredu. To prikazuje slika 20. Vsako instanco predstavi z vektorjem, cilj postopka pa je najti hiperravnino, ki loči primere obeh razredov. Točke na robovih hiperravnine se imenujejo podporni vektorji. V primeru, da je razredov več, se postopek ponovi za vse kombinacije možnosti razredov. Algoritem je lahko uporabljen tako za klasifikacijo, kot za regresijo [66].



Slika 20: Prikaz delovanja SVM algoritma. (vir: [66])

5.7 RandomForest

RandomForest je algoritem, ki uporablja več tehnik strojnega učenja. Njegov cilj je izdelava množice naključnih odločitvenih dreves s postopkoma imenovanima »boosting« in »bagging«. To sta algoritma, ki na podlagi več klasifikatorjev izbereta najboljšega. Najprej se podatkovna zbirka razdeli na več podmnožic, na katerih se izvede več različic enakega algoritma. Algoritme se nato preveri na testni množici, kjer se izbere tistega, ki je dosegel najboljši rezultat. Pri »boostingu« uspešnost prejšnjih zgrajenih dreves (algoritmov) vpliva na izbiro novega algoritma, pri »baggingu« pa se drevo zgradi neodvisno od ostalih dreves. Za razliko od ostalih načinov gradnje dreves se pri RandomForest algoritmu postopek gradnje dreves nadaljuje na atributu, ki je najboljši izmed določene podmnožice naključnih atributov, ne pa na tistem, ki je najboljši izmed vseh atributov. Tako ni potrebno preverjati vseh atributov, kar pospeši postopek [29] [68].

5.8 IBk

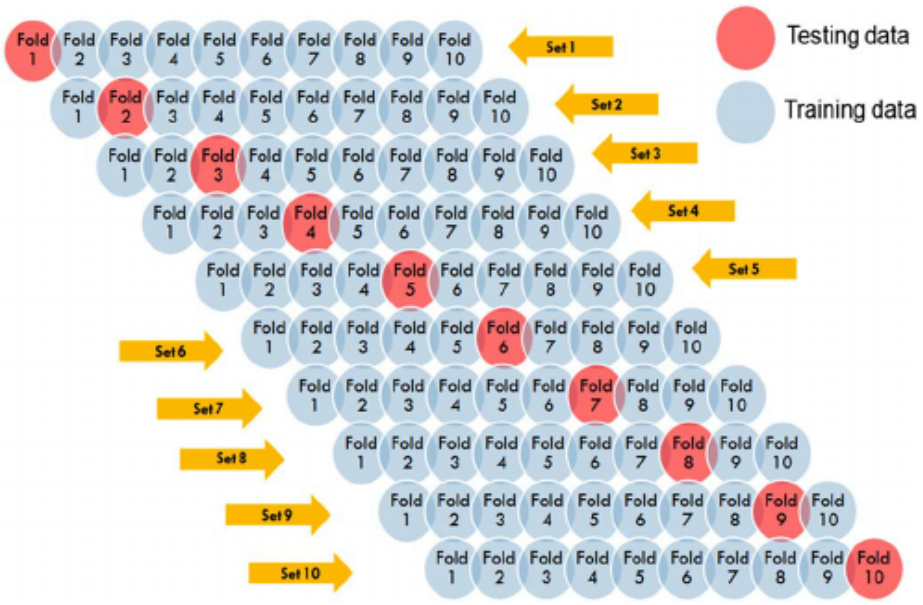
IBk deluje na principu k-NN algoritma. Ta deluje tako, da izračuna razdaljo do vseh instanc, ki imajo znano vrednost razreda. Klasifikacija poteka na podlagi k najbližjih instanc. Algoritem se ne uči, ampak za vsak nov primer, ki ga želi klasificirati na novo izračuna razdalje. Je časovno zahteven algoritem, ki zasede veliko prostora. Njegova prednost pa je njegova enostavnost [32].

Za izračun razdalje se lahko uporabi več različnih formul. V zaključni nalogi je bila uporabljena Evklidska razdalja, ki se jo izračuna po formuli:

$$d(a,b) = \sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\ldots+(a_n-b_n)^2} \quad [3]$$

5.9 K-kratno prečno preverjanje

K-kratno prečno preverjanje (angl. K-fold cross validation) je način vrednotenja algoritmov. Podatkovno zbirko razdeli na k enako velikih disjunktnih podmnožic. Nato uporabljen klasifikacijski algoritem zgradi učni model na k-1 podmnožicah, na zadnji pa ga testira. Postopek se ponovi k-krat, kjer se podmnožice vsakič zamenjajo, kar prikazuje slika 21. Natančnost algoritma izračuna tako, da število pravilno napovedanih instanc deli s številom vseh instanc v podmnožici [27]. V zaključni nalogi je bilo uporabljeno 10-kratno prečno preverjanje, ki se je izvedlo 10-krat. Kot rezultat je izpisano povprečje vseh desetih iteracij.



Slika 21: 10-kratno prečno preverjanje. (vir:[39])

6 REZULTATI IN VREDNOTENJE

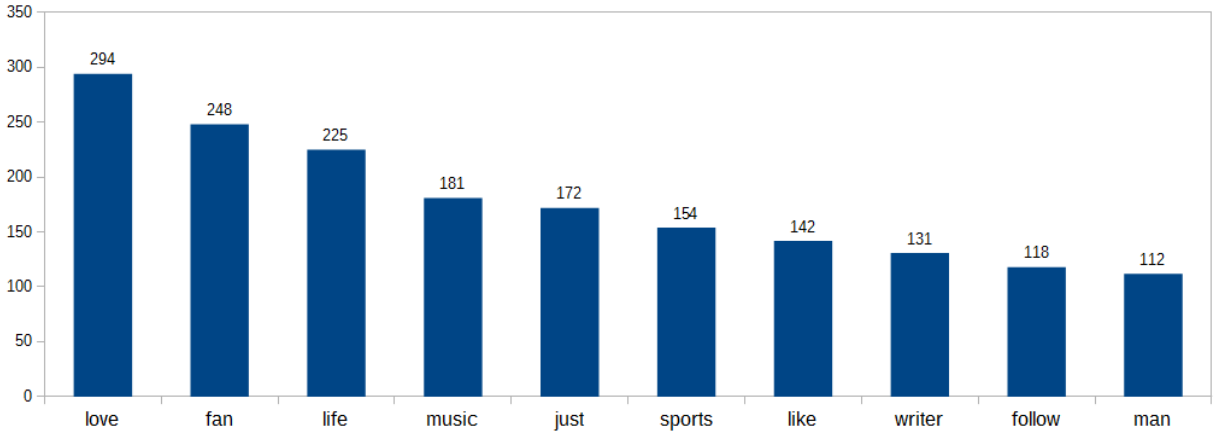
Zaradi lažjega razumevanja sta v zaključni nalogi zadnji fazi CRISP-DM procesa združeni.

6.1 Analiza podatkov

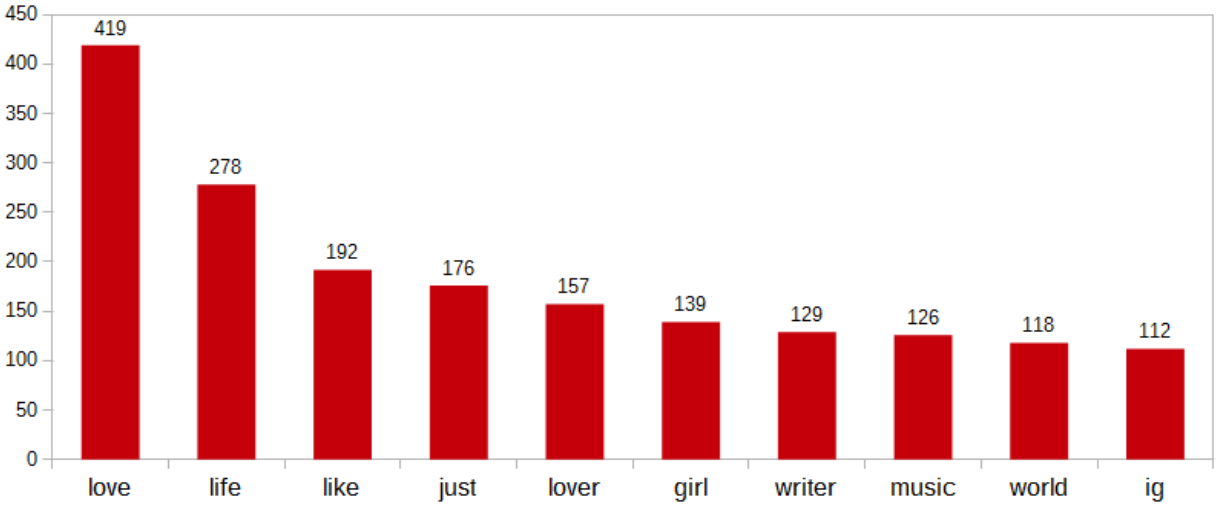
Kot je omenjeno v razdelku 4.3.3, je bila za analizo podatkov uporabljena različna podatkovna zbirka. Za atribute, katere vrednosti so bili nizi besed, je bil uporabljen algoritem StringToWordVector. Najpogosteje uporabljene besede so predstavljene v obliki histogramov na slikah 22, 23, 24, 25, 26 in 27. Algoritem je kot rezultat vrnil 100 besed, od katerih je zaradi pomanjkanja prostora predstavljenih zgolj najpogostejših 10. Prav tako je bilo algoritmu določeno, da ne razlikuje med malimi in velikimi začetnicami besed, besede, ki se pojavljajo velikokrat in nimajo pomena pa so bile odstranjene. Te besede so predstavljene v prilogi A.

Za ostale atribute je bil postopek analize lažji, saj je število posameznih vrednosti v nominalnih atributih v Weki že prešteto, pri numeričnih atributih pa je izračunana tudi povprečna vrednost.

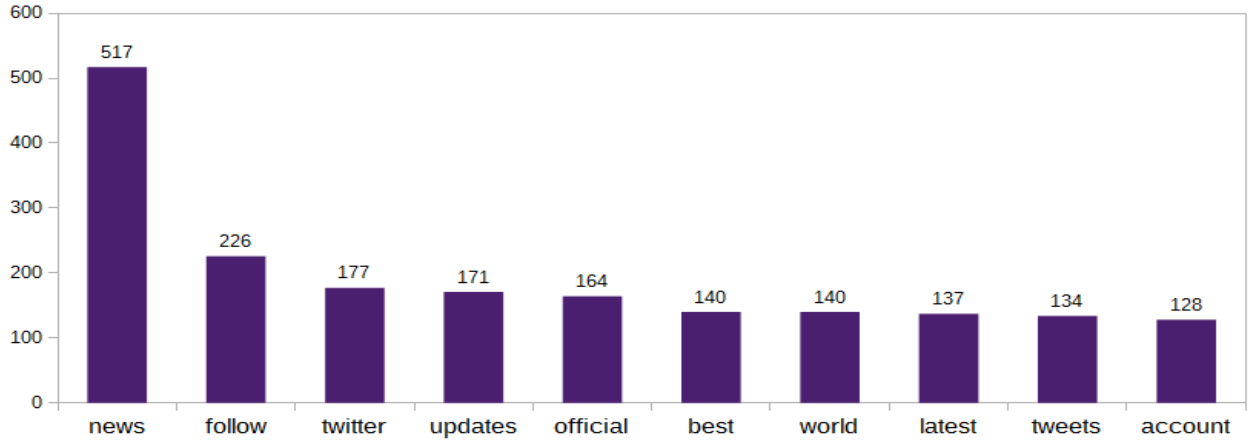
6.1.1 Opis profila



Slika 22: Histogram najpogostejših besed, ki jih uporabljajo moški v opisu profila.



Slika 23: Histogram najpogostejših besed, ki jih uporabljajo ženske v opisu profila.



Slika 24: Histogram najpogostejših besed, ki jih uporabljajo blagovne znamke v opisu profila.

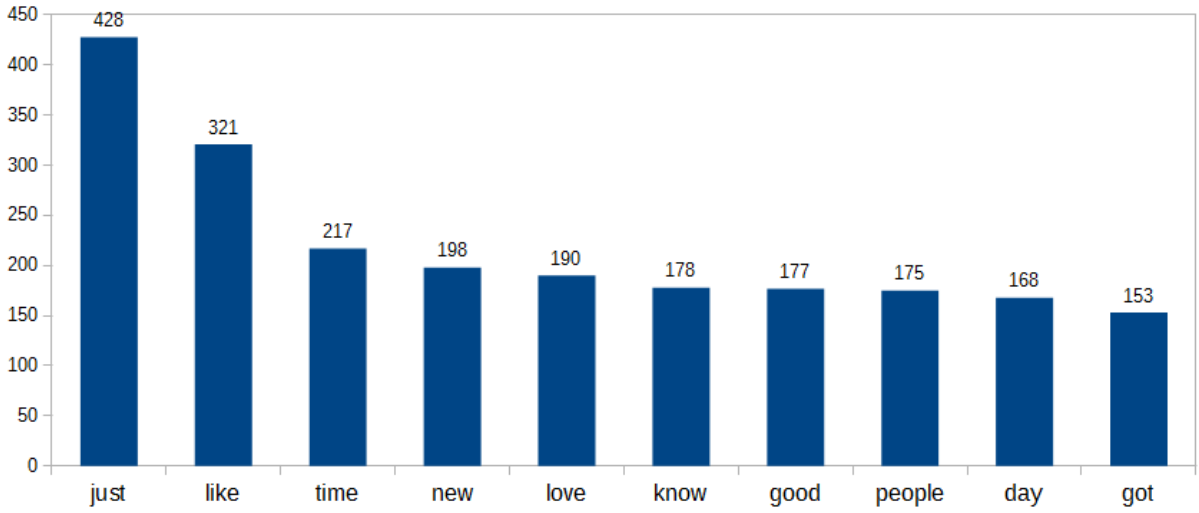
Histogrami nam že v atributu, ki predstavlja opis profila nakazujejo, da obstajajo določene besede, ki jasno določajo spol osebe. Tak primer je beseda »man«, ki jasno napove, da je oseba moškega spola in beseda »girl«, ki predstavlja osebe ženskega spola.

Zanimive pa so besede, ki so se na lestvico najpogostejših besed, pojavile le pri enem izmed obeh spolov. Za moški spol je to beseda »fan«, ki se pri ženskah pojavi šele na 15. mestu in beseda »sports«, ki je pri ženskah med 100 najpogostejšimi besedami sploh ne najdemo. Ob takih besedah bi se lahko reklo, da račun pripada osebi moškega spola.

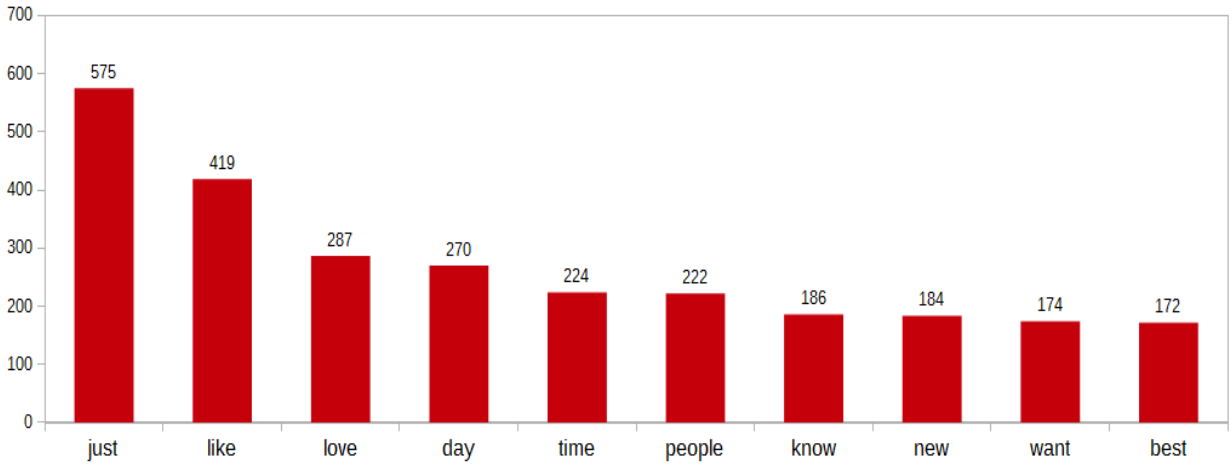
Pri ženskah so take besede »lover«, »world« in »ig« (okrajšava za Instagram), ki se pri moških pojavijo med 10. in 15. mestom. Za take besede lahko trdimo, da je račun osebe ženskega spola.

V opisu blagovnih znamk se pojavljajo popolnoma drugačne besede kot so »news«, »follow« »twitter« in »updates«, zato je take profile enostavno ločiti od osebnih profilov.

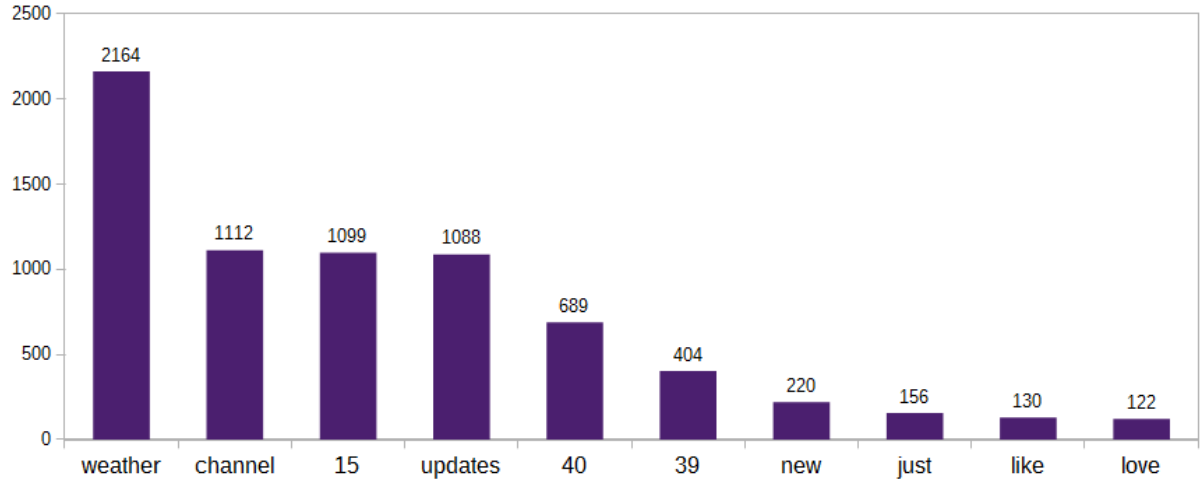
6.1.2 Besedilo tvita



Slika 25: Histogram najpogostejših besed, ki jih uporabljajo moški v besedilu tvita.



Slika 26: Histogram najpogostejših besed, ki jih uporabljajo ženske v besedilu tvita.



Slika 27: Histogram najpogostejših besed, ki jih uporabljajo blagovne znamke v besedilu tvita.

Iz same vsebine tvita pa se s stališča spola osebe težko karkoli napove. Ponovno pa se razlika opazi med osebnimi profili in profili blagovnih znamk. Zanimivo je dejstvo, da so med najpogostejšimi desetimi besedami kar tri števila. S skoraj polovično verjetnostjo pojavitve pa prevladuje beseda »weather«.

6.1.3 Ostali atributi

V tabeli 2 so za nominalne atribute predstavljene najpogostejše vrednosti in delež njihovih pojavitev, za numerične pa povprečna vrednost. Pri barvah je treba upoštevati, da sta 008484 in C0DEED vrednosti, ki jih Twitter samodejno uporabi pri registraciji novega profila, zato so uporabljene največkrat. Posledično so v tabeli 3 predstavljene druge najpogostejše vrednosti.

Tabela 2: Najpogostejše vrednosti po atributih.

Atribut	Moški	Ženske	Blagovna znamka
link_color	008484 (50,1%)	008484 (39,4%)	008484 (58,5%)
sidebar_color	C0DEED (44,9%)	C0DEED (37,1%)	C0DEED (53,6%)
user_timezone	UTC-5 (17,7%)	UTC-5 (15,3%)	UTC-5 (13,2%)
fav_number	4839	6068	2065
tweet_count	31644	26456	61342
created	4.3.12	7.8.12	18.3.13
retweet_count	0,093	0,049	0,121

Tabela 3: Druge najpogostejše vrednosti po atributih.

Atribut	Moški	Ženske	Blagovna znamka
link_color	9999 (4,7%)	F5ABB5 (4,3%)	3B94D9 (3,4%)
sidebar_color	0 (19,3%)	FFFFFF (22,5%)	0 (18,1%)
user_timezone	UTC-0 (10,7%)	UTC-8 (11,7%)	UTC-8 (11,2%)

Iz atributov predstavljenih v tabelah lahko ugotovimo, da dajo ženske večjo pozornost na videz svojega profila, saj kar 60% izmed njih zamenja obe barvi. Pri moških pa to stori zgolj vsaka druga oseba. Zanimiv je tudi podatek, da večina moških barvo zamenja v črno-sivo (0 in 9999), medtem ko večina žensk zamenja v belo-rdečo (FFFFFF in F5ABB5).

Atribut, ki predstavlja časovni pas je predstavljen zgolj kot zanimivost, saj je ta podatek odvisen predvsem od časa objave tvita, ne pa toliko od spola.

Iz atributov, ki predstavljajo čas nastanka profila, število všečkanih tvitov in število napisanih tvitov (tabela 2) lahko razberemo, da ženske veliko bolj všečkajo tvite kot moški, kljub temu da so v povprečju njihovi računi mlajši za približno pol leta. Upoštevajoč starost profilov, se število napisanih tvitov bistveno ne razlikuje. Pri številu retvitov je pomembno to, da so bili tviti v podatkovno zbirko preneseni le nekaj sekund po objavi, zato je številka zelo majhna. Lahko pa opazimo, da so moški tviti v povprečju dvakrat bolj retvitani, kot ženski.

Za profile blagovnih znamk se lahko izpostavi število povprečnih napisanih tvitov, ki za več kot dvakrat presega tvite osebnih profilov. Profili so v splošnem mlajši, tviti pa so bolj retvitani. Barva je zamenjana na manj kot polovici vseh profilov.

6.2 Algoritmi

Algoritmi so bili pognani z algoritmom FilteredClassifier, kjer je bil kot obdelovalni algoritem uporabljen StringToWordVector, kot algoritem za klasifikacijo pa različni algoritmi. Cilj je bilo najti najboljšo različico algoritma. To predstavlja razmerje med hitrostjo algoritma in število uspešno napovedanih primerov. V splošnem je pomemben tudi prostor, ki ga algoritem med izvajanjem zasede, vendar v tem primeru se s tem podatkom nismo ukvarjali.

Vsak algoritem je bil testiran z več različnimi parametri, izbran pa je bil najboljši. Pri večini algoritmov je sprememba parametrov na uspešnost vplivala največ do 5%, v povprečju pa zgolj 2-3%.

Tabela 4: Primerjava uporabljenih algoritmov.

Ime algoritma	Klasifikacijska točnost	Čas (s)	ROC površina
J48	59,32%	48,86	0,711
PART*	57,70%	62,23	0,743
Naivni Bayes	60,05%	5,7	0,761
SMO	**64,05%**	*414,45*	0,749
RandomForest*	63,42%	28,28	**0,808**
IBk*	*52,25%*	**0,78**	*0,683*

Tabela 4 prikazuje različico algoritma, ki je bila najuspešnejša glede na število pravilno klasificiranih primerov. Ob izboru algoritma v Weki so parametri že nastavljeni tako, da je algoritem pripravljen za uporabo. Ker je parametrov zelo veliko, so tukaj opisani zgolj tisti, ki so bili spremenjeni. Pri vseh razen pri Naivnemu Bayesu, je bil obdelovalni algoritem nastavljen tako, da atribut spremeni v 100 najpogostejših besed, ki jih ne razlikuje po veliki začetnici. Naivni Bayes pa je edini najboljše klasificiral z 200 najpogostejšimi besedami. Vrednost novih atributov (besed) je tako celoštevilska vrednost, ki predstavlja število pojavitve besed za vsak primer.

Algoritmi, ki v tabeli zvezdice ne vsebujejo, so bili zagnani z že predlaganimi parametri. Vrednosti napisane ležeče pomenijo najslabšo vrednost, krepko pa najboljšo vrednost.
Pri PART algoritmu so bili spremenjeni parametri, ki vplivajo na velikost drevesa. Tako je bilo minimalno število instanc, ki jih list mora vsebovati nastavljeno na 200, faktor rezanja drevesa pa na 0,15.

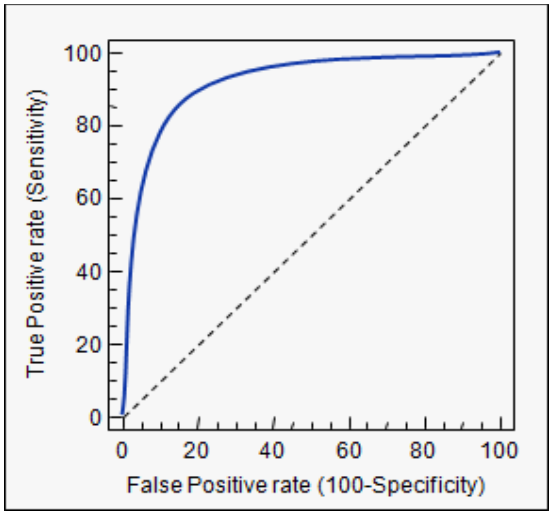
RandomForest je bil najboljši, ko je bila vrednost parametra minimalnega števila instanc nastavljena na 100.

Pri IBk pa je bila uporabljena Evklidska razdalja s tremi najbližjimi sosedi.

Rezultati kažejo, da je najučinkovitejši SMO in najhitrejši IBk algoritem. Kljub temu pa je SMO daleč najpočasnejši in IBk daleč najslabši algoritem po številu uspešno klasificiranih primerov. Zato bi lahko rekli, da je v splošnem najboljši RandomForest, ki ima zgolj za pol odstotka manjšo uspešnost kot SMO, je pa veliko hitrejši. To potrjuje tudi ROC površina, ki je opisana v razdelku 6.2.1.

6.2.1 ROC površina

ROC površina je dober indikator kakovosti določenega klasifikatorja. Pomaga predvsem pri primerjavi algoritmov, predstavlja pa področje pod ROC krivuljo. Ta je matematično gledano razmerje senzitivnosti in specifičnosti, kjer je senzitivnost delež pravilno razvrščenih pozitivnih instanc in specifičnost delež pravilno razvrščenih negativnih instanc. Z večanjem kakovosti algoritma, se tudi ROC površina veča [2]. Primer ROC krivulje predstavlja modra črta na sliki 28.



Slika 28: ROC krivulja. (vir: [31])

7 ZAKLJUČEK IN NADALJNJE DELO

Podatkovno rudarjenje je proces zbiranja in analize podatkov, iz katerih izluščimo koristne informacije. Pri tem procesu so pomembni predvsem algoritmi strojnega učenja, ki računalnik naučijo določene stvari na podlagi preteklih izkušenj. Poznamo več metod in procesnih modelov podatkovnega rudarjenja. V zaključni nalogi je bila uporabljena klasifikacija, ki je potekala po procesnem modelu CRISP-DM. Podatki so imeli obliko tvitov, cilj pa je bil napoved spola osebe. Za algoritme je bila uporabljena programska oprema Weka. To je program, ki vsebuje algoritme strojnega učenja. Zaradi neurejenosti podatkov, jih je bilo potrebno najprej predobdelati. Nato je sledil postopek njihove analize. Ta je pokazala, da v opisu Twitter profila obstajajo besede, ki so bolj pogoste za določen spol, v vsebini tvitov pa je ločevanje med spoloma bolj težavno. Pri moških so v atributu opisa profila izstopale besede »man«, »sports« in »fan«, pri ženskah pa besede »lover«, »world« in »ig«. Za profile, ki jih vodi več ljudi so bile take besede »news«, »follow«, »twitter« in »updates«. V atributu besedila tvita med moškimi in ženskimi profili ni bilo večjih razlik, opazna pa je bila razlika v besedah med osebnimi profili in profili blagovnih znamk. Pri slednjih je izstopala beseda »weather«, ki se pojavi v skoraj vsakem drugem tvitu in števila, ki jih med osebnimi profili ne najdemo. Ugotovljeno je bilo tudi, da osebe ženskega spola bolj skrbijo za izgled svojega profila kot ostali in da lahko razlike med profili najdemo tudi v številu všečkanih tvitov in drugih uporabljenih atributih. Sledila je uporaba različnih algoritmov z različnimi parametri. Uporabljeni so bili algoritmi:J48, PART, Naivni Bayes, SMO, RandomForest in IBk. Rezultati so pokazali, da so algoritmi zmožni določiti spol osebe v približno 60% primerov, kar je zadovoljivo. V splošnem se je najbolj izkazal RandomForest.

V diplomski nalogi je bila osredotočenost usmerjena predvsem na iskanje najboljšega algoritma, ki bi spol ločil na podlagi vsebine tvita. V bodoče bi se lahko naloga razširila in bi se za klasifikacijo uporabilo tudi profilne slike Twitter računov. Poleg tega se dandanes veliko uporablja emojije in morda bi se lahko na podoben način raziskalo, katere emojije najraje uporabljajo moški in katere emojije najraje uporabljajo ženske. Zanimivo bi bilo klasificirati osebe tudi na drugih socialnih omrežjih kot so Facebook, Instagram ipd.

8 LITERATURA IN VIRI

[1] A. Azevedo and M. F. Santos, "Kdd, Semma and Crisp-Dm: a Parallel Overview."

[2] A. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[3] M. Bramer, *Principles of Data Mining*. 2007.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. September, pp. 1–58, 2009.

[5] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mob. Networks Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[6] ClassStringToWordVector[Online]. Dosegljivo: http://weka.sourceforge.net/doc.stable/weka/filters/unsupervised/attribute/StringToWordVector.html. [Dostopano: 24. 6. 2017].

[7] W. Cohen, "Fast effective rule induction," *Twelfth Int. Conf. Mach. Learn.*, pp. 115–123, 1995.

[8] Fast Threshold Clustering Algorithm (FTCA), *CSSA* [Online]. Dosegljivo: https://cssanalytics.wordpress.com/2013/11/26/fast-threshold-clustering-algorithm-ftca. [Dostopano: 16. 6. 2017].

[9] Doing more with 140 characters, *Developer blog* [Online]. Dosegljivo: https://blog.twitter.com/developer/en_us/a/2016/doing-more-with-140-characters.html. [Dostopano: 18. 6. 2017].

[10] Introducing Twitter Polls, *Developer blog* [Online]. Dosegljivo: https://blog.twitter.com/official/en_us/a/2015/introducing-twitter-polls.html. [Dostopano: 18. 6. 2017].

[11] Never miss important Tweets from people you follow, *Developer blog* [Online]. Dosegljivo: https://blog.twitter.com/official/en_us/a/2016/never-miss-important-tweets-from-people-you-follow.html. [Dostopano: 18. 6. 2017].

[12] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, pp. 37–54, 1996.

[13] E. Frank, M. Hall, L. Trigg, G. Holmes, and I. H. Witten, "Data mining in bioinformatics using Weka," *Bioinformatics*, vol. 20, no. 15, pp. 2479–2481, 2004.

[14] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," *Proc. Fifteenth Int. Conf. Mach. Learn.*, pp. 144–151, 1998.

[15] Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage, *Gartner* [Online]. Dosegljivo: http://www.gartner.com/newsroom/id/3412017. [Dostopano: 14. 6. 2017].

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10, 2009.

[17] M. Hall and P. Reutemann, "Tutorial: WEKA KnowledgeFlow for Version 3-5-8," p. 15, 2008.

[18] Have you seen ASUM-DM?, *IBM* [Online]. Dosegljivo: https://developer.ibm.com/predictiveanalytics/2015/10/16/have-you-seen-asum-dm/. [Dostopano: 17. 6. 2017].

[19] Twitter Usage Statistics, *Internet live stats* [Online]. Dosegljivo: http://www.internetlivestats.com/twitter-statistics/. [Dostopano: 18. 6. 2017].

[20] Twitter User Gender Classification, *Kaggle* [Online]. Dosegljivo: https://www.kaggle.com/crowdflower/twitter-user-gender-classification. [Dostopano: 18. 6. 2017].

[21] Association Rules and the Apriori Algorithm: A Tutorial, *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html. [Dostopano: 16. 6. 2017].

[22] Data Mining Methodology, *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/polls/2004/data_mining_methodology.htm. [Dostopano: 17. 6. 2017].

[23] Data Mining Methodology, *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm. [Dostopano: 17. 6. 2017].

[24] Four Problems in Using CRISP-DM and How To Fix Them, *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html. [Dostopano: 17. 6. 2017].

[25] What main methodology are you using for your analytics, data mining, or data science projects? Poll (Oct 2014), *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/polls/2014/analyticsdata-mining-data-science-methodology.html. [Dostopano: 17. 6. 2017].

[26] What main methodology are you using for data mining?, *KDnuggets* [Online]. Dosegljivo: http://www.kdnuggets.com/polls/2002/methodology.htm. [Dostopano: 17. 6. 2017].

[27] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection" no. March 2001, 2016.

[28] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary, "Twitter trending topic classification," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 251–258, 2011.

[29] a Liaw and M. Wiener, "Classification and Regression by randomForest," *R news*, vol. 2, no. December, pp. 18–22, 2002.

[30] LibreOffice Calc [Online]. Dosegljivo: https://www.libreoffice.org/discover/calc/ [Dostopano: 10.8.2017]

[31] ROC curve analysis, *Medcalc* [Online]. Dosegljivo: https://www.medcalc.org/manual/roc-curves.php. [Dostopano: 3. 7. 2017].

[32] T. Mitchell, "Instance Based Learning," *Mach. Learn.*, pp. 199–214, 1997.

[33] T. Mitchell, *Machine Learning*. McGraw Hill. p. 2. (1997).

[34] A. Munoz, "Machine Learning and Optimization," *Courant Inst. Math. Sci.*, 2014.

[35] Notepad++ [Online]. Dosegljivo: https://notepad-plus-plus.org/download/v7.4.2.html [Dostopano: 10.8.2017]

[36] M. Panda, A. Abraham, and M. Patra, "Discriminative multinomial naive bayes for network intrusion detection," *2010 Sixth Int. Conf. Inf. Assur. Secur.*, pp. 5–10, 2010.

[37] T. R. Patil, "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," *Int. J. Comput. Sci. Appl. ISSN 0974-1011*, vol. 6, no. 2, pp. 256–261, 2013.

[38] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," *Adv. kernel methods*, p. 376, 1999.

[39] 10-fold cross-validation procedure, *ResearchGate* [Online]. Dosegljivo: https://www.researchgate.net/figure/239386696_fig3_Fig-4-10-fold-cross-validation-procedure. [Dostopano: 25. 6. 2017].

[40] S. Salzberg, "Book Review: C4.5: Programs for Machine Learning," *Mach. Learn.*, vol. 1, no. 16, pp. 235–240, 1994.

[41] R. Schapire, "Theoretical Machine Learning," pp. 1–7, 2013.

[42] D. Scuse, "WEKA Experimenter Tutorial for Version 3-4," *Test*, 2007.

[43] Uporaba interneta v gospodinjstvih in pri posameznikih, Slovenija, 2016, *STAT* [Online]. Dosegljivo: http://www.stat.si/StatWeb/News/Index/6263. [Dostopano: 4. 7. 2017].

[44] Most famous social network sites worldwide as of April 2017, ranked by number of active users (in millions), *Statista* [Online]. Dosegljivo: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/. [Dostopano: 17. 6. 2017].

[45] Interpreting residual plots to improve your regression, *Statwing* [Online]. Dosegljivo: http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression/. [Dostopano: 16. 6. 2017].

[46] K. Tan, Steinbach, "Data Mining: Introduction Lecture Notes," p. 31, 2004.

[47] P.-N. Tan, M. Steinbach, and V. Kumar, "Association Analysis: Basic Concepts and Algorithms," *Introd. to Data Min.*, pp. 327–414, 2005.

[48] For Election Day Influence, Twitter Ruled Social Media, *The New York Times* [Online]. Dosegljivo: https://www.nytimes.com/2016/11/09/technology/for-election-day-chatter-twitter-ruled-social-media.html?_r=0. [Dostopano: 17. 6. 2017].

[49] Twitter users send 50 million tweets per day, *The Telegraph* [Online]. Dosegljivo: http://www.telegraph.co.uk/technology/twitter/7297541/Twitter-users-send-50-million-tweets-per-day.html. [Dostopano: 18. 6. 2017].

[50] NZS | FA Slovenia tvit, *Twitter* [Online]. Dosegljivo: https://twitter.com/nzs_si/status/873557421763039232. [Dostopano: 18. 6. 2017].

[51] Pokemon GO tvit, *Twitter* [Online]. Dosegljivo: https://twitter.com/PokemonGoApp/status/876152476143169536. [Dostopano: 18. 6. 2017].

[52] YouTube profil, *Twitter* [Online]. Dosegljivo: https://twitter.com/YouTube. [Dostopano: 18. 6. 2017].

[53] Streaming APIs, *Twitter Developer Documentation* [Online]. Dosegljivo: https://dev.twitter.com/streaming/overview. [Dostopano: 18. 6. 2017].

[54] Twitter for Websites supported languages, *Twitter developer documentation* [Online]. Dosegljivo: https://dev.twitter.com/web/overview/languages. [Dostopano: 17. 6. 2017].

[55] Tweeting via text message, *Twitter support* [Online]. Dosegljivo: https://support.twitter.com/articles/14226. [Dostopano: 17. 6. 2017].

[56] Twitter Pushes Live-Video Deals With MLB, NFL, Viacom, BuzzFeed, Live Nation, WNBA and More, *Variety* [Online]. Dosegljivo: http://variety.com/2017/digital/news/twitter-pushes-live-video-deals-with-mlb-buzzfeed-live-nation-wnba-and-others-1202405236/. [Dostopano: 18. 6. 2017].

[57] ARFF (book version), *Weka* [Online]. Dosegljivo: https://weka.wikispaces.com/ARFF+ %28book+version%29. [Dostopano: 18. 6. 2017].

[58] Weka 3: Data Mining Software in Java, *Weka* [Online]. Dosegljivo: http://www.cs.waikato.ac.nz/ml/weka/. [Dostopano: 18. 6. 2017].

[59] Anomaly detection, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Anomaly_detection. [Dostopano:16. 6. 2017].

[60] Bayes' theorem, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Bayes%27_theorem. [Dostopano: 25. 6. 2017].

[61] C4.5 algorithm, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/C4.5_algorithm. [Dostopano: 24. 6. 2017].

[62] Cluster analysis, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Cluster_analysis. [Dostopano: 16. 6. 2017].

[63] Data Mining, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Data_mining. [Dostopano: 17. 6. 2017].

[64] Examples of data mining, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Examples_of_data_mining. [Dostopano: 15. 6. 2017].

[65] Lexical analysis, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Lexical_analysis. [Dostopano: 24. 6. 2017].

[66] Support vector machine, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Support_vector_machine. [Dostopano: 25. 6. 2017].

[67] Twitter, *Wikipedia* [Online]. Dosegljivo: https://en.wikipedia.org/wiki/Twitter. [Dostopano: 17. 6. 2017].

[68] I. H. Witten, E. Frank, and M. a Hall, *Data Mining: Practical Machine Learning Tools and Techniques (Google eBook)*. 2011.

[69] X. Wu *et al.*, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2008.

PRILOGE

PRILOGA A: Seznam besed, ki so najbolj uporabljene v angleškem jeziku

a about above across after afterwards again against all almost alone along already also although always am among amongst amoungst amount an and another any anyhow anyone anything anyway anywhere are around as at back be became because become becomes becoming been before beforehand behind being below beside besides between beyond bill both bottom but by call can cannot cant co computer con could couldnt cry de describe detail do done down due during each eg eight either eleven else elsewhere empty enough etc even ever every everyone everything everywhere except few fifteen fify fill find fire first five for former formerly forty found four from front full further get give go had has hasnt have he hence her here hereafter hereby herein hereupon hers herself him himself his how however hundred i ie if in inc indeed interest into is it its itself keep last latter latterly least less ltd made many may me meanwhile might mill mine more moreover most mostly move much must my myself name namely neither never nevertheless next nine no nobody none noone nor not nothing now nowhere of off often on once one only onto or other others otherwise our ours ourselves out over own part per perhaps please put rather re same see seem seemed seeming seems serious several she should show side since sincere six sixty so some somehow someone something sometime sometimes somewhere still such system take ten than that the their them themselves then thence there thereafter thereby therefore therein thereupon these they thick thin third this those though three through throughout thru thus to together too top toward towards twelve twenty two un under until up upon us very via was we well were what whatever when whence whenever where whereafter whereas whereby wherein whereupon wherever whether which while whither who whoever whole whom whose why will with within without would yet you your yours yourself yourselves