Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Sašo Rutar

# Empirična evalvacija procesa avtomatske klasifikacije sentimenta na finančni domeni

univerzitetno diplomsko delo

Mentor: izr. prof. dr. Marko Robnik-Šikonja
Somentor: doc. dr. Igor Mozetič

Ljubljana 2016

*Besedilo je oblikovano z urejevalnikom besedil LaTeX.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Klasifikacija sentimenta besedil se ukvarja z odkrivanjem mnenja piscev o predmetu pisanja. Pravilno določanje sentimenta je koristno za številne namene, npr. za napovedovanje uspešnosti izdelkov, borznih gibanj, izidov volitev ali v sociooloških raziskavah. Naloga za avtomatske sisteme ni enostavna, saj je potrebno iz besedila izluščiti bistveno semantično informacijo. Za označevanje sentimenta v velikem številu besedil uporabljamo avtomatske postopke, ki besedila najprej pripravijo s tehnikami obdelave naravnega jezika, nato pa klasificirajo s pomočjo algoritmov strojnega učenja. Celoten postopek sestavlja več faz, sestavljenih iz mnogih algoritmov, ki jih lahko prilagodimo na različne načine. Mnogokrat je postopek odvisen od vrste besedil in domene, ki jo analiziramo.

Za dane tvite s finančnega področja empirično analizirajte parametre, ki nastopajo v celotnem postopku določanja sentimenta. Postopek evalvirajte in predlagajte kar najboljši nabor parametrov za proces klasifikacije.

# Izjava o avtorstvu diplomskega dela

Spodaj podpisani Sašo Rutar, z vpisno številko **24035431**, sem avtor diplomskega dela z naslovom:

*Empirična evalvacija procesa avtomatske klasifikacije sentimenta na finančni domeni*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Robnika-Šikonje in somentorstvom doc. dr. Igorja Mozetiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, 6. junija 2016 Podpis avtorja:

*Na prvem mestu gre zahvala Mihi Grčarju, ki je s podporo na raziskovalnem področju in implementacijo obravnavanega sistema odprl pot tej nalogi. Poleg tega se zahvaljujem vsem v podjetju Sowa Labs, ki je prispevalo podatke za evalvacijo. Iskrena hvala mentorjema za izčrpno pomoč ob sproščenem vzdušju ter Radi za lekturo in razumevanje. Zahvaljujem se tudi družini in tistim prijateljem, ki so mi stali ob strani na tej maratonski poti do cilja.*

Svetu, da se ni prenehal vrteti, preden bi jaz diplomiral. S sentimentom.

# Kazalo

# Povzetek

V tem diplomskem delu obravnavamo specifične vidike sistema za avtomatsko analizo sentimenta v tvitih. Naš sistem za analizo sentimenta temelji na tehnikah strojnega učenja in tekstovnega rudarjenja, kot sta predstavitev besedil z vrečami besed in metoda podpornih vektorjev. S sistemom obdelamo podatkovni tok kratkih sporočil (tvitov) na temo finančnih trgov, specifično na temo trgovanja z delnicami, v razponu dveh let. Vsako sporočilo avtomatsko klasificiramo v pozitivni, negativni ali nevtralni razred, kar predstavlja sentiment oziroma stališče do delnice, ki je omenjena v sporočilu. Sentiment torej v našem primeru odraža stališče govorca in v primeru pozitivnega ali negativnega razreda predstavlja nagib k nakupu ali prodaji delnice.

Za izgradnjo klasifikacijskega modela uporabimo relativno velik nabor označenih podatkov, ki sestoji iz približno pol milijona tvitov, ki so jih ročno označili eksperti. Za potrebe analize smo razvili evalvacijsko platformo in pripadajočo metodologijo, ki nam omogoča, da z zaporedjem poskusov lahko odgovorimo na številna vprašanja, ki se pojavijo pri aplikacijah analize sentimenta v industrijskih okoljih. Pri analizah upoštevamo časovno sosledje sporočil v podatkovnih tokovih in tako omogočimo sprotno merjenje uspešnosti sistema tudi v produkcijskih okoljih.

Rezultati analize nam med drugim razkrijejo (i) najprimernejši algoritem za klasifikacijo, (ii) optimalno velikost in vzorčenje (redčenje) podatkov za ročno označevanje, (iii) odvisnost med uspešnostjo klasifikacije in časovno oddaljenostjo od označenih primerov, (iv) vpliv prisotnosti duplikatov v podatkih in (v) obnašanje izbrane klasifikacijske metode v območju negotovosti

ob hiper ravnini klasifikatorja z metodo podpornih vektorjev.

**Ključne besede**: analiza sentimenta, strojno učenje, rudarjenje mnenj, Twitter, obdelava naravnega jezika, klasifikacija, metoda podpornih vektorjev, empirična evalvacija, finančno trgovanje, delnice.

# Abstract

In this thesis, we explore several specific aspects of Twitter sentiment analysis. Our system for sentiment analysis is based on machine learning and text mining techniques, such as the bag-of-words representation of texts and support vector machine classifier. We employ our system to analyze a stream of short messages (tweets) about financial markets, specifically about stock trading, in the time span of two years. We classify each message into positive, negative, or neutral class, which represent the sentiment or stance towards the stock mentioned in the message. The term sentiment in our case thus denotes the stance of the author (speaker) and in the case of positive or negative class represents the author's leaning towards buying or selling the stock. To build the classification model, we employ a relatively large gold standard which consists of approximately a half million tweets hand-labeled by the domain experts.

For the purpose of this analysis, we developed an evaluation platform and a methodology that allow us, by conducting a series of experiments, to answer various questions which arise when applying sentiment analysis in industrial settings. In the evaluation processes, we take the temporal nature of the data into account and thus enable continuous monitoring of performance of live systems.

The results of the analysis reveal (i) the most appropriate classification algorithm, (ii) the optimal size of the labeled data and subsampling method, (iii) the relationship between the classifier performance and the time lag from the training data, and (iv) the effect of duplicated tweets (e.g., retweets), and

(v) the behavior of the employed classification method in the uncertainty area near the hyper-plane of support vector machine classifier.

**Keywords**: sentiment analysis, machine learning, opinion mining, Twitter, natural language processing, classification with support vector machine, empirical evaluation, financial trading, stocks.

# Poglavje 1

# Uvod

Mnenja si kot posamezniki izoblikujemo na podlagi lastnih vedenj in izkušenj ter izkušenj, o katerih so nam pripovedovali drugi. Gre za osebne zaključke, ki jih navadno spremlja določeno razpoloženje ali sentiment. Mnenje in sentiment, ki sta na nek način sopomenki, praviloma nosita pozitivno ali negativno konotacijo. O nekem izdelku imamo torej lahko slabo mnenje oziroma nas v odnosu do izdelka preveva negativen sentiment. Mnenja igrajo osrednjo vlogo v procesu odločanja, zato želijo biti podjetja in organizacije čim bolje seznanjena z mnenji, ki jih imajo potrošniki in javnost o njihovih produktih in storitvah. Po drugi strani želijo biti tudi posamezniki, bodisi kot potrošniki pred izbiro izdelka bodisi kot volivci v predvolilnem obdobju, seznanjeni z mnenjem drugi o izdelku ali političnih kandidatih.

Mnenja se širijo od ust do ust, kot pravimo *"dober glas seže v deveto vas"*, z uporabo klasičnih medijev, kot so časopis, radio in televizija. Z razvojem in vse širšo uporabo interneta, še posebno z vstopom v obdobje t. i. **spleta 2.0** (angl. *Web 2.0*) pa se je vzpostavila infrastruktura za nastajanje in vzdrževanje virtualnih omrežij prijateljev ali privržencev, s pomočjo katere si uporabniki lahko izmenjujejo mnenja in izkušnje. Primeri platform te infrastrukture so *Twitter*, *Facebook* ali *Pinterest*. Poleg njih pa obstaja vrsta forumov in spletnih mest za izmenjevanje mnenj o določeni tematiki. Znana sta recimo *RottenTomato* in *IMDB*, kjer uporabniki sporočajo svoje

komentarje in ocene filmov. Spletne trgovine, kot so *Amazon* ali *mimovrste*, omogočajo puščanje komentarjev o izdelkih, ki jih uporabniki lahko ocenijo tudi s številom zvezdic. Mnenja se torej širijo na podlagi obilice spletnih storitev, kar uporabnike postavlja pred nov izziv iskanja in izbiranja relevantnih informacij. Brez sodobnih sistemov, ki se razvijajo v okviru področij, kot so poizvedovanje po podatkih [14], tekstovno rudarjenje [2], spletno rudarjenje ali rudarjenje mnenj [13], bi bilo iskanje mnenja o neki stvari podobno *iskanju igle v kopici sena*.

Vsesplošna potreba po sistemih za rudarjenje mnenj in dostopnost velikih količin mnenjskih podatkov v digitalni obliki sta odprli novo poglavje v raziskovanju tehnik analize sentimenta. Pri reševanju problema analize sentimenta v tekstu uporabljamo znane metode s področja tekstovne klasifikacije. Ta se ukvarja s klasifikacijo tekstovnih dokumentov v različne tematske kategorije, kot so politične, znanstvene ali športne. Klasifikacija glede na tematiko se naslanja predvsem na ugotavljanje prisotnosti določenih tematskih besed v dokumentih, kar se izkaže kot zelo učinkovit pristop. Tudi pri analizi sentimenta obstajajo določene besede, ki so pomembne za izražanje pozitivnega ali negativnega sentimenta, recimo *super*, *odlično*, *neverjetno*, *grozno*, *slabo*, *najslabše* in podobno. Vendar se klasifikacija sentimenta izkaže za kompleksnejši problem od tematske klasifikacije, kar nakazuje tudi slabša uspešnost. Pri zanašanju na besede pri analizi sentimenta naletimo na probleme, kot so:

- Besede za izražanje pozitivnega ali negativnega sentimenta imajo na različnih področjih lahko različen pomen.

- Čeprav stavek nemara vsebuje besede za izražanje sentimenta, ni nujno, da slednjega tudi sporoča. Vzemimo stavek: *"Če bodo v trgovini imeli dobro kamero, jo bom kupil."*. Stavek vsebuje besedo za pozitivni sentiment, *dobro*, vendar očitno ne izraža sentimenta.

- Drži tudi nasprotno, saj lahko stavki izražajo sentiment kljub temu, da ne vsebujejo besed za izražanje sentimenta. V stavku *"Ta pomivalni*

*stroj porabi veliko vode"* lahko zaznamo negativno konotacijo o stroju, čeprav ne vsebuje besed s čustveno konotacijo.

- Sarkazem v besedilu predstavlja zahteven problem, saj je za njegovo prepoznavanje ponavadi potrebno poznavanje širšega konteksta. Recimo *"Res odličen telefon! Nehal je delovati po dveh dneh."* Prisotnost sarkazma sicer ni tako značilna za komentiranje izdelkov, ga je pa veliko v političnih debatah.

Opisane probleme lahko do neke mere uspešno premostimo s tehnikami, ki vključujejo uporabo slovarjev sentimenta ali algoritmov za strojno učenje. Pri strojnemu učenju se algoritem pravil za klasifikacijo teksta uči iz zbirke vnaprej označenih besedil ali primerov, kjer za vsak primer ročno določimo, kako naj ga algoritem klasificira. Za metode s takšnim načinom pogojevanja znanja uporabljamo izraz **nadzorovano učenje**. Največja omejitev pri aplikaciji takšnih metod je potreben vložek v pripravo zbirk označenih primerov, pri čemer ponavadi sodelujejo skupine ljudi, včasih tudi ekspertov na nekem področju.

Glavni cilj te naloge je evalvacija obstoječega sistema za analizo sentimenta. Gre za sistem za sprotno klasifikacijo sentimenta v toku sporočil platforme Twitter, ki se v okviru podjetja **Sowa Labs** izvaja in trži že nekaj let. Sistem uporablja nekatere izmed pristopov in tehnik, ki so opisani v nadaljevanju. Čeprav je izbira tehnik in njihove aplikacije v sistemu pravzaprav predvidljiva in se ne spreminja, se celoten sistem skupaj z modeli prilagaja nenehno spreminjajočim se podatkom v toku sporočil. Spremembam podatkov pri nadzorovanem učenju sledimo s pomočjo sprotnega posodabljanja zbirke označenih primerov, pri čemer se spreminjajo tudi modeli za klasificiranje. Zato je potreben nenehen nadzor nad kakovost klasifikacije in ustrezno prilagajanje parametrov sistema. Eden takšnih parametrov je intenzivnost označevanja primerov, ki, kot smo že omenili, predstavlja nezanemarljiv operativni strošek sistema. S tega stališča bi intenzivnost označevanja radi zmanjšali, vendar bi to lahko negativno vplivalo na kvaliteto modelov, torej moramo iskati ravnotežje med obema ciljema. Iskanje tega ravnotežja je

tudi predmet enega izmed poskusov, ki jih obravnavamo v tej nalogi. Poskusi bodo temeljili na podatkih, pridobljenih v obdobju približno dveh let delovanja sistema. Pred tem obdobjem takšna in tako obsežna empirična evalvacija sistema, kot jo predstavljamo v tej nalogi, ni bila mogoča.

Vir podatkov, s katerimi delamo, je splošno razširjena in zelo popularna platforma **Twitter**. Poglavitna lastnost te platforme so kratka sporočila, ki uporabnikom omogočajo hitro in sprotno beleženje dogajanja, kar je bogat vir podatkov o raznovrstnih dogodkih in razpoloženju udeleženih uporabnikov. Zaradi kratkosti besedil je za platformo Twitter značilno jedrnato podajanje vsebine in zatekanje k improvizacijam v izražanju. Zanj je značilno iskanje bližnjic pri črkovanju besed, neupoštevanje slovničnih pravil, pogosta uporaba kratic in posebnih zaporedij znakov, kot so emotikoni. Vse to otežuje klasifikacijo sentimenta, s čimer moramo računati pri implementaciji sistema.

Evalvacija je razdeljena v posamezne sklope ali vprašanja, ki obravnavajo različne vidike sistema za klasifikacijo sentimenta. Vsem sklopom je skupna domišljena metodologija izvajanja poskusov, ki zagotavlja pravilnost in konsistentnost dobljenih rezultatov. Poglavitni vidik, ki ga pri tem upoštevamo, je časovno sosledje sporočil v toku, ki ga opazujemo. To pomeni, da v poskusih modele vedno učimo le na preteklih in sedanjih sporočilih, da bi z njimi klasificirali sporočila v prihodnosti, kar ustreza pogojem v praksi. Zato kot del metodologije izberemo tudi mere za izražanje uspešnosti klasifikacije, ki so primerne za delo s časovno spremenljivimi podatki.

V 2. poglavju se bomo najprej seznanili s področjem analize sentimenta in podatkovno platformo Twitter ter njenih posebnostih pri analizi sentimenta. Predstavili bomo tudi vsebinski vidik podatkov, ki je finančno trgovanje z vrednostnimi papirji. V 3. poglavju bomo opisali metode, s katerimi klasificiramo sentiment v besedilih, in uporabljeno metodologijo poskusov. V 4. poglavju bomo opisali konkretni podatkovni nabor skupaj z načinom njegove pridobitve in priprave. V 5. poglavju bomo predstavili tehnično platformo za izvedbo eksperimentov ob pripadajočih vhodnih parametrih. V 6. poglavju bomo obdelali vsa postavljena vprašanja evalvacije z izvedbo ustreznih po-

skusov in interpretacijo dobljenih rezultatov. V sklepnem delu povzamemo bistvene ugotovitve, do katerih smo prišli v procesu evalvacije.

# Poglavje 2

# Analiza sentimenta

**Analiza sentimenta** ali **analiza razpoloženja** je novejše področje v domeni procesiranja naravnega jezika (angl. *natural language processing, NLP*), ki se ukvarja z analizo mnenj, razpoloženj, stališč in čustev, ki jih imajo ljudje do produkta, organizacij, posameznikov, dogodkov in tematik [13]. Gre za interdisciplinarno področje, ki se je v preteklosti razvijalo znotraj več različnih panog. Razvoj spleta in eksplozija uporabniških podatkov sta prinesla nove potrebe in izzive ter potrebo po združitvi različnih pristopov. Znotraj različnih panog so se uporabljala različna poimenovanja problemske domene. Izraz **rudarjenje mnenj** tako prihaja s področja informacijskega poizvedovanja (angl. *information retrieval, IR*), kjer je bil problem ekstrakcija in nadaljnje procesiranje uporabniških mnenj o produktih, filmih in drugih objektih. Po drugi strani je bil izraz analiza sentimenta najprej formuliran v okviru NLP za opis problema prepoznavanja sentimenta v prostem besedilu [28]. **Analiza subjektivnosti** se ukvarja z ločevanjem subjektivnega in objektivnega besedila. Poleg omenjenih obstajajo še izrazi, kot so ekstrakcija mnenj, analiza recenzij, rudarjenje čustev, analiza afekta, vendar danes vsi ti izrazi označujejo skupno področje analize sentimenta.

Področje analize sentimenta je po letu 2000 doživelo razmah z naraščajočo uporabo interneta in spletnih aktivnosti, kot so pogovori, rezervacija vstopnic, elektronsko trgovanje, udejstvovanje v družabnih omrežjih, forumih, pi-

sanje spletnih dnevnikov itd. Vse večje zahteve po obdelavi in analizi velikih količin strukturiranih in nestrukturiranih podatkov v kratkem času so pripeljale do začetka dobe t. i. velepodatkov (angl. *big data*). Številni forumi, blogi, socialna omrežja, portali za elektronsko poslovanje in novice s komentarji služijo kot platforma za izražanje mnenj, s pomočjo katerih lahko dobimo sprotni vpogled v dinamiko in značaj družbenih dogajanj. To odpira možnost avtomatizacije pri analizi javnega mnenja, potrošniškega mnenja o produktih, spremljanju ugleda podjetij ali osebnosti, izdelovanja tržnih strategij itd. Ogromna količina informacij povezanih s potrošniškimi mnenji in komentarji je zahteven analitični problem, zato moramo pri reševanju iskati enostavnejše pristope in poenostavitve problemov, tako pri analizi tekstov kot pri klasifikaciji sentimenta.

Dejavnike za razmah področja analize sentimenta lahko strnemo v tri točke: [20]

- vse večja veljava in uporaba pristopov strojnega učenja znotraj NLP in IR;
- razpoložljivost podatkov za uporabo algoritmov strojnega učenja z razcvetom interneta in znotraj tega spletnih portalov za širjenje uporabniških vsebin, kot so recenzije, komentarji in pogovori;
- možnost razvoja komercialnih aplikacij, ki jih odpira področje.

Osrednji problem analize sentimenta lahko po [13] formalno opredelimo kot iskanje peterk $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$, kjer je $e_i$ entiteta, na katero se sentiment nanaša, in $a_{ij}$ eden izmed njenih vidikov. Sentiment $s_{ijkl}$ se nanaša na vidik $a_{ij}$ entitete $e_i$, izražen pa je bil s strani osebe $h_k$ v času $t_l$. Sentiment $s_{ijkl}$ je lahko pozitiven, negativen ali nevtralen, lahko pa je tudi vrednost na lestvici intenzivnosti sentimenta. Entiteta $e_i$ in aspekt $a_{ij}$ skupaj predstavljata objekt sentimenta.

Najpogosteje se analiza sentimenta izvaja na **nivoju dokumenta**, kjer določamo prevladujoč ali odločilen sentiment znotraj celotnega besedila. Klasifikacija sentimenta na ravni dokumenta predpostavlja, da posamezen do-

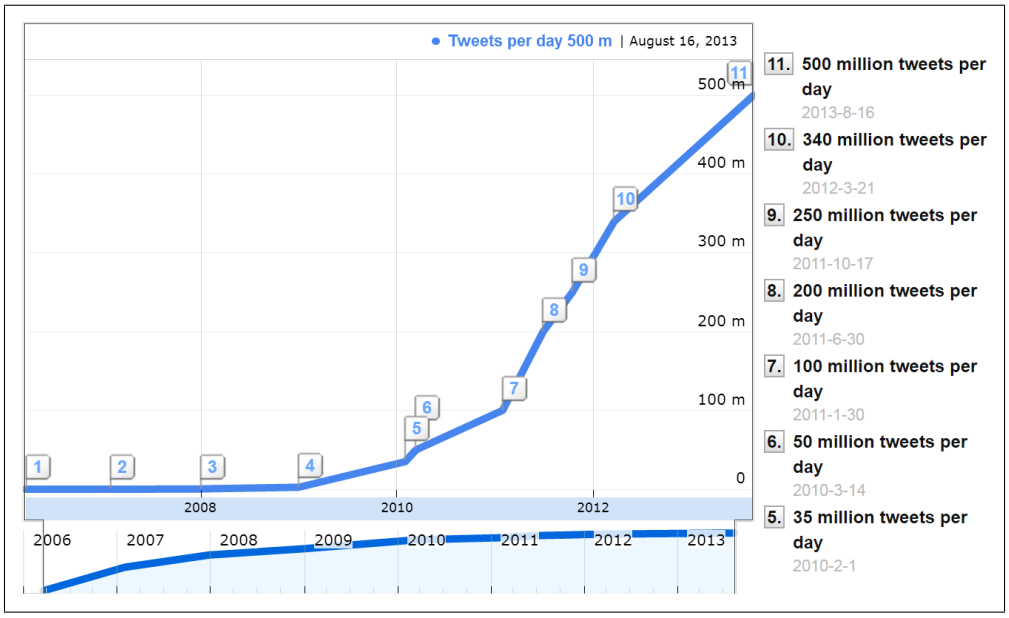
kument izraža mnenje ene osebe o eni entiteti, kar poenostavi problem. Formalno torej iščemo $(\_, \_, s, \_, \_)$ danega dokumenta, kjer prepoznavanje entitete, nosilca in časa sentimenta niso del problema. Analiza na **ravni stavka** je najbolj smiselna, kadar vemo, na katero entiteto se stavek nanaša, poznamo avtorja in nas zanima pozitivnost oziroma negativnost sentimenta v njegovem stavku. Najbolj podrobna in zahtevna je analiza sentimenta na **nivoju vidika**, kjer je poleg pripadajočega sentimenta v problem vključeno še odkrivanje entitet in njihovih vidikov znotraj besedila.

Kategorizacija sentimenta, ki se veliko uporablja znotraj domene NLP, temelji na sistemu FACS (*Facial Action Coding System*) [5], ki pozna šest osnovnih čustev: jeza, gnus, strah, veselje, žalost in presenečenje. Takšna kategorizacija zaradi kompleksnosti procesiranja in analize vhodnih podatkov ni vedno mogoča, še posebno pri velepodatkih. Pri analizah sentimenta zato v zadnjem času največkrat opazujemo **polarnost sentimenta** (*sentiment polarity*) [20], kjer sta razreda sentimenta samo dva: pozitivni in negativni. Da lahko pokrijemo še primere brez sentimenta ali brez prevladujočega sentimenta, razredoma sentimenta pridružimo še nevtralni razred. Takšno kategorizacijo uporabljamo tudi v naši nalogi.

V nadaljevanju opisujemo problem klasifikacije polarnosti sentimenta v tvitih. Objekt sentimenta predstavljajo posamezni finančni instrumenti, ki jih identificiramo ob zajemu podatkov, tako da je prisotnost objekta v vsakem tvitu zagotovljena. Besedila v tvitih so pretežno posamezne povedi, tako da naš problem najbolj ustreza problemu analize na ravni stavka.

## 2.1 Opis in značilnosti platforme Twitter

Twitter je ena najbolj popularnih platform za objavljanje kratkih sporočil (angl. *microblogging*). Njena uporaba zadnja leta strmo narašča in je kot takšna postala stičišče najrazličnejših skupin, od strokovnjakov, študentov do znanih osebnosti, politikov in podjetij. Takšna priljubljenost omogoča dostop do ogromne količine informacij, s širokim spektrom tematik, ki se-




Slika 2.1: Eksponentna rast dnevnega prometa na Twitterju skozi čas.

gajo od počutja ljudi in iskrivih zamisli do mnenj o blagovnih znamkah, političnih temah in družabnih dogodkih. V tem kontekstu se Twitter ponuja kot zelo močno orodje za napovedovanje in opazovanje družbenih pojavov. Prva objava na Twitterju se je zgodila 16. julija 2006, od takrat se je število dnevnih objav stopnjevalo in danes (leto 2016) se dnevno odvije 500 milijonov objav oziroma 200 milijard letno. Slika 2.1 prikazuje eksponentno rast števila objav na Twitterju skozi čas[1].

Uporabniki Twitterja pošiljajo sporočila, ki so omejena na dolžino 140 znakov. Sporočila se imenujejo **tviti** (angl. *tweet*), slovensko tudi čviki. S sporočili skušajo biti uporabniki čimbolj aktualni in zanimivi, da bi s tem pridobili čim večje število privržencev (angl. *follower*), ki njihovim vsebinam sledijo. Vsebina tvitov je zelo raznolika, lahko gre za osebne podatke, ki avtorju nekaj pomenijo, pa tudi splošnejše informacije s povezavami do fotografij, video posnetkov ali spletnih strani, ki jih avtor vidi kot zanimive za druge. Velik del vsebine na Twitterju tvorijo javni pogovori med uporabniki,

[1]vir: http://www.internetlivestats.com/twitter-statistics/

ki jih prepoznamo po tem, da so v tvitu vsebovane reference na uporabnike. Referenca se začne z znakom @ in nadaljuje z uporabniškim imenom, ki ne sme vsebovati presledkov. Pojavitev takšne reference se imenuje omemba (angl. *mention*) in povzroči, da tvit dobijo vsi omenjeni uporabniki.

Funkcija, ki v veliki meri pripomore k temu, da je Twitter postal orodje za razširjanje informacij, je možnost ponovitve tvita (angl. *retweet*). Uporabnik lahko tvit, ki se mu zdi zanimiv, ponovi oziroma pošlje svojim privržencem, bodisi v prvotni obliki ali z dodanim lastnim komentarjem. Takšna sporočila imajo predpisano obliko z znakoma RT na začetku in navedbo prvotnega avtorja sporočila. Zaradi ogromne količine informacij, ki se pretaka skozi Twitter, je na voljo mehanizem za označevanje informacij in povezovanje sporočil v vsebinske skupine. Oznaka tematike (angl. *hashtag*) se začne z znakom #, ki mu sledi ime tematike brez presledkov. Če uporabnik klikne na oznako, lahko dostopa do vseh sporočil z isto oznako tematike. Same oznake nastajajo spontano med pisanjem sporočil, ko se uporabnikom neka tematika zazdi splošno relevantna. Twitter vzdržuje katalog trendnih tematik, ki dosežejo dovolj veliko število pojavitev v določenem časovnem obdobju[2].

S stališča analize jezika je poglavitna značilnost tvitov njihova kratkost, ki avtorje sili k improviziranju in jedrnatosti v izražanju. Pri obdelavi besedil v tvitih moramo upoštevati naslednje lastnosti.

- **Dolžina besedil**
  Tviti so navadno zelo kratki. Po eni strani je to prednost, ker uporabniki težijo k jedrnatosti izražanja in zgoščevanju informacij. Po drugi strani pa je pomen celotnega besedila pogosto odvisen od ene same besede, kar poveča možnost napačne interpretacije [1].

- **Spremembe v črkovanju**
  Spontanost izražanja, neformalnost konteksta in omejenost dolžine tvitov uporabnike napeljujejo k uporabi nenavadnega črkovanja besed. To vključuje namerno napačno črkovanje, uporabo številk (npr. ju3

[2]https://twitter.com/trendingtopics

namesto jutri), čustveno uporabljanje velikih črk (SRAMOTA!!!), podaljševanje besed (kooooončno!) ter uporaba slenga, kratic in krajšav. Tako se povečuje razredčenost informacije v podatkih (angl. *sparsity*), kjer se za en pomen uporablja več različnih besednih oblik.

- **Posebni tekstovni elementi**
  V sporočilih so prisotne posebne sekvence znakov, kot so spletni naslovi, emotikoni itd., ki jih lahko izkoristimo v skladu z njihovo namembnostjo.

- **Količina podatkov**
  Dolžina posameznih besedil je sicer kratka, vendar je kumulativna količina vseh podatkov lahko ogromna. Ob večjih dogodkih, kot je npr. nogometno prvenstvo ali naravna nesreča, so za Twitter značilne eksplozije podatkov, kjer v kratkem času nastane veliko podatkov. To je poglaviten izziv za aplikacije, ki sprotno analizirajo tok sporočil.

- **Različni jezikovni slogi**
  Glede na obsežnost uporabniške baze in raznolikost kontekstov uporabe se v besedilih na Twitterju uporablja veliko različnih jezikovnih slogov. Spekter jezikovnega sloga sega od formalnega jezika novičarski portalov do neformalnega slenga s preklinjanjem. Jezikovni slogi se s časom tudi spreminjajo.

- **Mešanje jezika**
  Določene populacije uporabnikov za sporazumevanje uporabljajo več jezikov, preklapljanje med jeziki pa se dogaja tudi znotraj enega tvita. To skupaj s kratkostjo sporočil otežuje problem avtomatske detekcije jezika.

## 2.2 Analiza finančnih podatkov

Ogromna količina mnenjskih podatkov je za uporabnike spleta neprecenljiv vir informacij, potrebnih pri vsakodnevnem odločanju, po drugi strani

pa je postal izziv iskanja in zbiranja relevantnih informacij praktično neobvladljiv. Zato so razvili številne sisteme za avtomatsko analizo in zbiranje mnenjskih podatkov, ki vnašajo spremembe v način odločanja na številnih področjih. Na finančnem področju takšni sistemi služijo podpori pri investicijskem odločanju, boljšemu vpogledu v finančne trende in podpori napovedovanja teh trendov. Pomemben del finančne dejavnosti je trgovanje z vrednostnimi papirji. **Delnice** (*stock*) so vrednostni papirji, ki prinašajo pravico do deleža pri glavnici in dobičku delniških družb. Trgovanje z njimi se odvija na delniških borzah (angl. *stock exchange*), njihova vrednost pa je odvisna od ponudbe in povpraševanja. Na vrednost delnic vplivajo dogodki, kot so objave rednih finančnih poročil delniških družb, ali dogodki, ki vplivajo na poslovanje družb, kot so naravne nesreče, tehnološki razvoj in podobno. Dogajanje na trgu nenehno spremlja določeno razpoloženje v zvezi s posameznimi delnicami na trgu. Zaradi svojega formata, ki omogoča beleženje odzivov v zelo kratkem času, se platforma Twitter v finančnih krogih pogosto uporablja.

V [3] je opisana ena prvih raziskav, ki se je ukvarjala z vplivom sentimenta na vrednosti delnic. V raziskavi so na podlagi velike zbirke zbranih tvitov uspešno modelirali sentiment in ugotovili njegovo korelacijo z vrednostjo indeksa DOW za nekaj dni vnaprej. Delniški indeks DOW opazujemo tudi mi in ga podrobneje predstavimo v 4. poglavju. Na vzročno povezanost med sentimentom in pričakovano povrnjeno investicijo so pokazali tudi v [21]. Splošen pregled ekonomskega učinka analize sentimenta je na voljo v [20]. V tej nalogi se s spremljanjem gibanja delniških indeksov v povezavi s sentimentom ne ukvarjamo in smo omejeni na opazovanje sentimenta v besedilih sporočil.

Eden izmed ponudnikov sistemov za podatkovno rudarjenje, ki se ukvarja tudi z analizo sentimenta je podjetje **Sowa Labs**[3]. Podjetje komercialno trži sistem za avtomatsko klasifikacijo polarnosti sentimenta v kontekstu izbranih finančnih inštrumentov. V procesu klasifikacije implementirajo tehnike, ki so

[3]http://www.sowalabs.com/

delno pojasnjene v nadaljevanju. Del procesa je tudi zbiranje podatkov in ročno označevanje sentimenta v podatkih. Določen izbor tako pripravljenih podatkov, ki so last podjetja, uporabljamo tudi v tej nalogi.

Empirično evalvacijo torej izvajamo na podlagi omenjenih finančnih podatkov. Čeprav so rezultati in izsledki evalvacije splošne narave, ob tem ne smemo zanemariti izvora in vsebine konkretnih podatkov ter njihovega vpliva na izsledke raziskave. Previdni moramo biti recimo pri predpostavljanju stopnje šuma v podatkih zaradi avtomatsko generiranih sporočil ali napak pri prepoznavanju entitet v sporočilih. Tudi nepravilno ocenjena časovna dinamika spremenljivosti podatkov lahko ob prekratkem časovnem intervalu opazovanih podatkov privede do nezanemarljivih razlik med našo predstavo o podatkih in dejanskim stanjem, kar nas lahko zavede pri interpretaciji rezultatov evalvacije.

# Poglavje 3

# Modeliranje sentimenta besedil v naravnem jeziku

Pri reševanju problema klasifikacije sentimenta v besedilu sta najbolj uveljavljena **leksikalni pristop** in **pristop z uporabo strojnega učenja**. Leksikalni pristop se naslanja na uporabo slovarja sentimenta, ki je vnaprej pripravljena in urejena zbirka besed in besednih zvez z ustrezno oceno sentimenta. Tekom analize besedila posameznim besedam s pomočjo slovarja določimo oceno sentimenta in nato izračunamo sentiment za celotno besedilo. Pristop z uporabo metod strojnega učenja predvideva uporabo sofisticiranih algoritmov v dveh korakih: (i) učenje modela iz korpusa učnih podatkov (ii) klasifikacija novih podatkov na podlagi naučenega modela.

Korpus učnih podatkov je množica primerov, ki jim v procesu označevanja dodelimo razred. Učni primeri so lahko bodisi strukturirani bodisi nestrukturirani podatki. Strukturirani podatki so navadno v tabelarični obliki, kjer so posamezne vrstice primeri, polja vrstic pa atributi primerov. Nestrukturirane podatke, med katere spadajo tudi besedila, moramo pred uporabo pretvoriti v strukturirano obliko, s čimer se ukvarjamo v nadaljevanju. Učno množico z $n$ elementi lahko zapišemo kot $T_{učna} = \{(t_1, r_1), ..., (t_n, r_n)\}$, kjer je $t_i$ učni primer in $r_i$ prirejeni razred klasifikacije. Ob dani učni množici potrebujemo algoritem, ki bo iz učne množice $T_{učna}$ sposoben zgraditi model

$M$ za napovedovanje razreda oziroma klasifikacijo še ne videnih besedil. Primeri znanih učnih algoritmov so naivni Bayes, metoda podpornih vektorjev in odločitvena drevesa.

Slovarji sentimenta, ki se uporabljajo pri leksikalnem pristopu, večinoma vsebujejo besede z izraženim sentimentom v splošnem jeziku in kontekstu uporabe. V tej nalogi se ukvarjamo z besedili iz finančne domene in raziskave so pokazale, da je klasifikacija sentimenta močno odvisna od domene, iz katere izhajajo podatki [13]. Razlog za to leži v različnosti besednih ali celo jezikovnih konstruktov, ki se v različnih domenah uporabljajo za izražanje sentimenta. Zgodi se lahko, da ista beseda v eni domeni vsebuje pozitiven, v drugi pa negativen sentiment. Beseda *glasen* ima v stavku *"ventilator je zelo glasen"* negativno konotacijo za razliko od stavka *"za zvočnik je res glasen"*, kjer izraža pozitivno lastnost izdelka. Pristop s strojnim učenjem predvideva uporabo učne množice, ki jo lahko z relativno majhnim vložkom razvijemo za dano domeno ali primer uporabe. Pri tem ne potrebujemo posebnih lingvističnih znanj, le poznavanje obravnavane domene oziroma področja. To je eden glavnih razlogov, da v tej nalogi uporabljamo pristop z uporabo strojnega učenja.

## 3.1 Priprava učne množice

Učna množica določa temeljno znanje, iz katerega se algoritmi učijo in na podlagi katerega se testira njihova uspešnost. Zato učni množici rečemo **zlati standard**. Primere za učno množico dobimo z vzorčenjem celotne populacije primerov, ki jih želimo klasificirati. Pomembno je, da vzorec vsebinsko dovolj dobro odraža celotno populacijo, saj je od tega odvisno, kako ocene evalvacije odražajo dejansko uspešnost klasifikacije na celotni populaciji. Poseben primer je sprotno (angl. *real-time*) vzorčenje in klasifikacija sentimenta v toku podatkov, kjer se vsebina, porazdelitev sentimenta in gostota podatkov dinamično spreminja s časom. Tej dinamiki moramo slediti med drugim s

čim hitrejšim prilagajanjem vzorca učne množice in s tem klasifikacijskega modela.

Po pridobitvi vzorca je na vrsti korak označevanja primerov z želenimi razredi klasifikacije. To je pomemben korak, od katerega je odvisna uporabnost naučenega modela. V raziskavi [16] so opazovali vpliv kakovost učne množice na uspešnost klasifikacijskih modelov. Na voljo so imeli nabor relativno velikih učnih množic (od 50.000 do 200.000 primerov), ki jih je več skupin uporabnikov označevalo v različnih jezikih. S pomočjo merjenja stopnje medsebojnega strinjanja med označevalci so opazovali kakovost učnih množic in kako ta vpliva na uspešnost klasifikacijskih modelov. Potrdili so, da je kakovost učne množice pomembnejša od izbire algoritma za strojno učenje.

V procesu ročnega označevanja primerov ponavadi sodeluje skupina ljudi. Odločilnega pomena je, da vsi v skupini poznajo pravila za klasifikacijo ter jih dosledno upoštevajo. Pod izrazom **pravila za klasifikacijo** imamo v mislih interpretacijo sistema kategorizacije, ki označevalcu omogoča nedvoumno razvrstitev vsakega besedila v posamezno kategorijo. Za potrebe opazovanja medsebojnega ujemanja oznak vsaj del primerov v označevanje pošljemo več kot enemu označevalcu. To omogoča sprotno merjenje stopnje ujemanja in kakovosti označenih podatkov.

Pravila za klasifikacijo sentimenta v splošnem kontekstu so samoumevna vsakomur, v tej nalogi pa obravnavamo besedila iz domene finančnega trgovanja. Za označevanje potrebujemo finančno pismene ljudi, ki so seznanjeni z izrazoslovjem in načini izražanja sentimenta v tej domeni.

Javno dostopnih je več učnih korpusov z označenim sentimentom v splošnem kontekstu, nekaj primerov je naštetih v [28]. Korpusi v različnih jezikih za klasifikacijo tvitov so obravnavani v [16]. V [24] izvajajo evalvacijo osmih javno dostopnih korpusov. Poleg ročno označenih obstajajo tudi primeri avtomatsko označenih korpusov, kjer se za označevanje uporabijo določeni atributi sporočil, tak primer je opisan v [6], kjer so tviti avtomatsko označeni glede na to, ali vsebujejo pozitivne ali negativne emotikone (angl. *emoticon*).

Razreda sentimenta sta v tem primeru pozitivni in negativni, sami emotikoni pa so bili naknadno odstranjeni iz besedil, da ne bi vplivali na klasifikacijski model.

## 3.2 Algoritmi za klasifikacijo

Pri klasifikaciji tekstovnih dokumentov je uveljavljenih več različnih algoritmov. Najbolj popularna sta metoda podpornih vektorjev (angl. *support vector machine, SVM*) in naivni Bayes (angl. *Naive Bayes, NB*) [11]. Poleg teh se pogosto uporabljajo še metoda največje entropije (angl. *maximum entropy*), pogojno naključna polja (angl. *conditional random fields*), linearna regresija, algoritem najbližjih sosedov (angl. *nearest neighbor algorithm*) in drugi. V tej nalogi kot osrednji algoritem pri kategorizaciji tekstovnih dokumentov uporabljamo metodo SVM. SVM se namreč dobro obnese pri problemu klasifikacije besedil, kjer imamo opravka z vektorji v hiper-prostoru z velikim številom dimenzij (navadno več kot 10.000), in kjer je razredčenost informacije velika (angl. *sparseness*). Ob tem ne smemo zanemariti vpliva medsebojne soodvisnosti dimenzij, ki so na splošno razvrščene v linearno ločljive kategorije [8].

### 3.2.1 Metoda podpornih vektorjev

Metoda SVM [4] je binarni klasifikator, kar pomeni, da primere uvršča v enega izmed dveh razredov, navadno označena kot pozitivni in negativni razred. V učni fazi SVM poišče položaj hiper ravnine (t. j. ravnine v večdimenzionalnem prostoru), tako da ta ločuje pozitivne primere od negativnih. Hiper ravnino v evklidskem prostoru zapišemo kot $w \cdot \mathbf{x} - \mathbf{b} = 0$, pri čemer je $\mathbf{w}$ normalni vektor hiper ravnine in $b$ njen odmik od izhodišča. V osnovni formulaciji SVM ločitev primerov s hiper ravnino iščemo z dodatnim pogojem, ki določuje, da naj bo rob okoli nje, v katerem se ne nahaja noben primer, čim večji. Izrazimo jo kot iskanje vrednosti $\mathbf{w}$ in $b$ za najmanjši $|w|$

pri pogoju $y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1$, kjer je $y_i$ razred primera $\mathbf{x_i}$ in ima vrednost 1 za pozitivne in -1 za negativne primere.

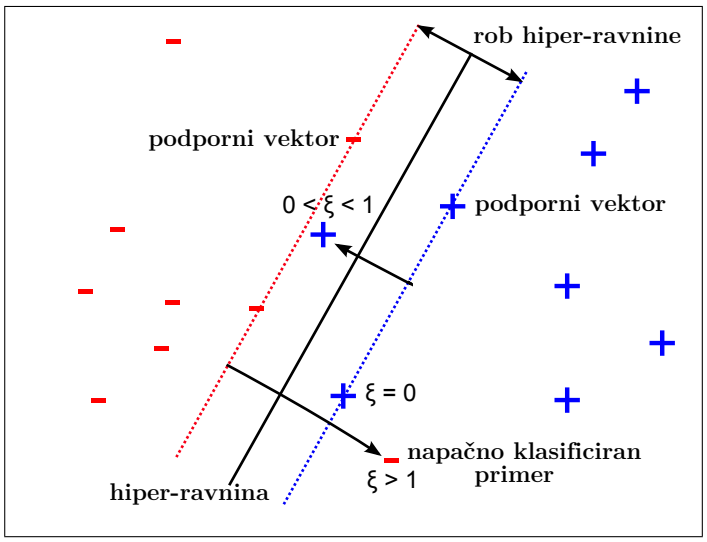
Ker ni vsaka množica primerov ločljiva z linearno ravnino, se zgodi, da v takih primerih opisani problem nima rešitve. Zato problem dopolnimo z vpeljavo t. i. mehkega roba, ki omogoča kaznovanje primerov, ki niso na pravilni strani hiper ravnine ali ležijo v območju njenega roba. Kazenska spremenljivka $\xi_i$ ima za vsak primer $\mathbf{x_i}$ določeno vrednost, ki je pri nepravilno umeščenih primerih večja od nič. Pri iskanju rešitve SVM želimo doseči čim manjšo vrednost vsote spremenljivk $\xi_i$ ob čim večjem robu hiper ravnine, kar uravnavamo s parametrom $C$. Formalno to zapišemo kot naslednji problem:

$$\textit{Iščemo } \mathbf{w} \textit{ in } b \textit{ in } \xi_i \textit{ ob najmanjši vrednosti } \frac{1}{2}|\mathbf{w}|^2 + C\sum_i \xi_i,$$
$$\textit{kjer za vsak } x_i \textit{ velja: } \quad y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Optimizacijski problem SVM je mogoče rešiti s tehnikami kvadratnega programiranja, kjer se izkaže, da lahko hiper ravnino izrazimo kot linearno kombinacijo relativno majhnega števila neničelnih primerov. Tem primerom pravimo podporni vektorji.

V fazi klasifikacije se neoznačen primer $\mathbf{x}$ klasificira glede na to, na kateri strani hiper ravnine se nahaja. Če leži na pozitivni strani ravnine, se označi s pozitivnim razredom, če leži na negativni strani ravnine, pa z negativnim. Če leži na sami hiper ravnini, njegovega razreda ne moremo določiti. V procesu klasifikacije primera izračunamo njegovo oddaljenost od hiper ravnine. To lastnost vključujemo v eksperimentu v poglavju 6.7, kjer opazujemo odvisnost oddaljenosti primera od ravnine in verjetnosti za pravilnost njegove klasifikacije.

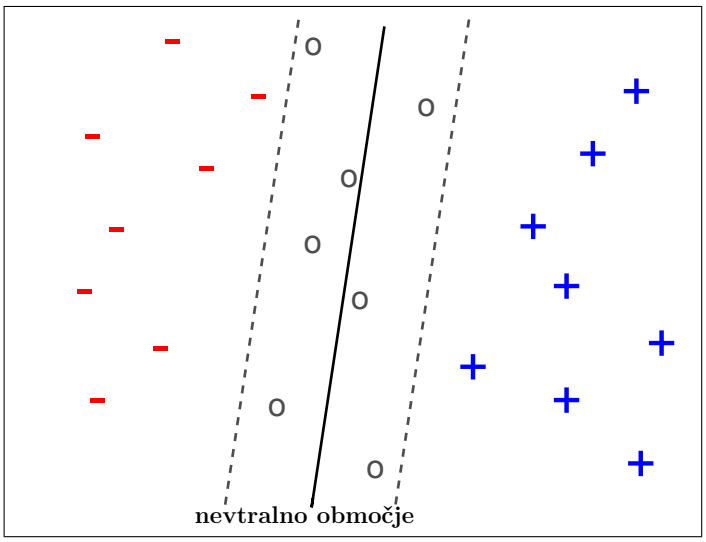
Osnovni SVM je namenjen linearni klasifikaciji, kjer se umestitev primera v določen razred določa na podlagi linearne kombinacije atributov primera. S pomočjo t.i. jedrnih matrik (angl. *kernel matrix*) lahko prvotni vektorski prostor preslikamo v prostor višjih dimenzij in s tem izvajamo nelinearno klasifikacijo. V praksi je bila pri klasifikaciji besedila najbolj učinkovita uporaba linearnega jedra SVM.




Slika 3.1: Hiper ravnina SVM, rob in kazenska spremenljivka $\xi$ za napačno klasificirane primere.

Kot rečeno je SVM formuliran kot binarni klasifikator, ki razločuje med pozitivnimi in negativnimi primeri, medtem ko imamo v tej nalogi opravka s tri-razrednim klasifikacijskim problemom. Pristopov za reševanje je več, izbira pa je med drugim odvisna od tega, ali so klasifikacijski razredi medsebojno urejeni. Pri polarnosti sentimenta je urejenost kategorij očitna: v skrajnostih sta pozitivni in negativni sentiment, medtem ko je nevtralni nekje vmes. Urejenost take kategorizacije so na podlagi analize označenih primerov pokazali tudi v [16]. Smailović je v [26] izhajala iz predpostavke, da so nevtralni primeri, preslikani v geometričen vektorski prostor, locirani med pozitivnimi in negativnimi primeri. Uporabila klasifikator SVM in ga učila samo iz pozitivnih in negativnih primerov, tako da se je ravnina postavila v simetrično sredino med oba razreda. Dejstvo, da se nevtralni primeri nahajajo prav v bližini sredinske ravnine, je z vpeljavo ločenega območja okoli ravnine uporabila za klasifikacijo vseh treh razredov. Prikaz tega je na sliki 3.2.

Drug pristop je kombiniranje več modelov SVM. Pang [19] je uporabil pristop, kjer za vsak razred vpelje svoj model SVM. SVM uči s primeri enega




Slika 3.2: Uporaba modela SVM in *območje nevtralnih primerov* za tri-razredno klasifikacijo sentimenta.

razreda na pozitivni in preostalimi primeri na negativni strani. Temu načinu rečemo učenje z razdelitvijo primerov eden-proti-vsem (angl. *one-versus-all*). Druga varianta je razdelitev eden-proti-enemu (angl. *one-versus-one*), kjer za vsak par razredov zgradimo svoj klasifikator. V nadaljevanju opazujemo tri različne načine sestavljanja modelov SVM za tri-razredno klasifikacijo sentimenta.

### Dvo-ravninski model

Kot je razvidno iz imena, sta v ta način vključena dva modela SVM, enemu lahko rečemo negativni, drugemu pa pozitivni model. Učni množici za oba modela pridobimo z razdelitvijo eden-proti-vsem. Pri pozitivnem modelu vzamemo razdelitev pozitivni primeri proti preostalim, pri negativnem pa negativni proti ostalim. Oba modela SVM lahko zapišemo kot $M_{poz}$ in $M_{neg}$ in ustrezni učni množici kot $T_{poz}$ in $T_{neg}$, za kateri velja

$$
\begin{aligned}
T_{poz} &= \{(t_i, poz) : (t_i, s_i) \in T_{učna} \wedge s_i = poz\} \\
&\quad + \{(t_i, neg) : (t_i, s_i) \in T_{učna} \wedge s_i \neq poz\} \\
T_{neg} &= \{(t_i, neg) : (t_i, s_i) \in T_{učna} \wedge s_i = neg\} \\
&\quad + \{(t_i, poz) : (t_i, s_i) \in T_{učna} \wedge s_i \neq neg\}
\end{aligned}
$$

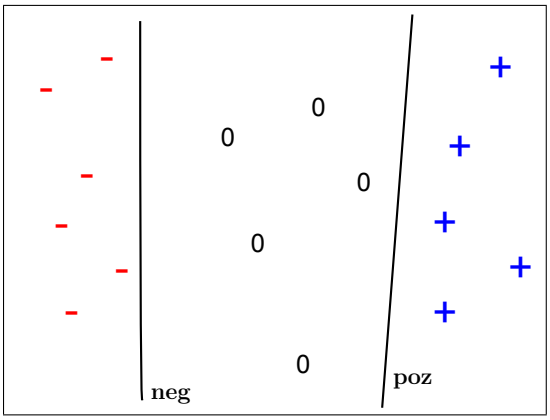
Takšni učni množici postavita ravnini SVM v vektorski prostor, kot ga prikazuje slika 3.3. Končna napoved za nek neoznačen primer $t_i$ se izračuna na podlagi napovedi obeh modelov $M_{poz}(t_i)$ in $M_{neg}(t_i)$ po sledečem postopku:

$$
M(t_i) = \begin{cases} poz; & \text{če } M_{poz}(t_i) = poz \wedge M_{neg}(t_i) = poz \\ nev; & \text{če } M_{poz}(t_i) \neq M_{neg}(t_i) \\ neg; & \text{če } M_{poz}(t_i) = neg \wedge M_{neg}(t_i) = neg \end{cases}
$$

Dvo-ravninski model s postavitvijo ravnin SVM na nevtralno-negativni in nevtralno-pozitivni meji opiše "nevtralni prostor" med prostoroma obeh sentimentov, kar ustreza urejenosti razredov sentimenta. Takšna postavitev uvrsti primere, kjer sta modela SVM v napovedovanju negotova, v nevtralni razred. Klasifikator tako (pre)več primerov uvrsti v nevtralni razred, vendar so zato napake pozitivnega in negativnega razredom manj pogoste. To potrjujejo tudi rezultati poskusa na sliki 6.2. V določenih kontekstih uporabe je takšna konzervativnost in tem manjše število napak pri uvrščanju v negativni ali pozitivni razred zaželena lastnost.

### Tri-ravninski model

Kot izhaja iz imena, so pri tem algoritmu v igri trije modeli SVM. Učimo se na treh podmnožicah učne množice, dobljenih po strategiji *eden-proti-enemu*. To pomeni, da vsaka izmed treh podmnožic vključuje primere dveh razredov iz učne množice, pri čemer izključuje primere tretjega razreda. Glede na




Slika 3.3: Urejenost primerov v vektorskem prostoru glede na njihov razred in položaj ravnin SVM pri dvo-ravninskem modelu.

vsebovane razrede jih poimenujemo: $T_{poz|neg}$, $T_{neg|nev}$ in $T_{poz|nev}$.

$$T_{poz|neg} = \{(t_i, s_i) \in T_{učna} : s_i \neq nev\}$$
$$T_{neg|nev} = \{(t_i, s_i) \in T_{učna} : s_i \neq poz\}$$
$$T_{poz|nev} = \{(t_i, s_i) \in T_{učna} : s_i \neq neg\}$$

Iz teh učnih množic najprej zgradimo tri modele SVM: $M_{poz|neg}$, $M_{neg|nev}$ in $M_{poz|nev}$. Nato po naslednjem postopku določimo funkcijo, ki na podlagi napovedi teh modelov izračuna končno napoved klasifikatorja. Najprej vse učne primere $t_i \in T$ klasificiramo s tremi modeli, tako da dobimo napovedi $M_{i,poz|neg} \in \{poz, neg\}$, $M_{i,neg|nev} \in \{neg, nev\}$ in $M_{i,poz|nev} \in \{poz, nev\}$. Dobimo množico trojic napovedi in dejanskih razredov primera:

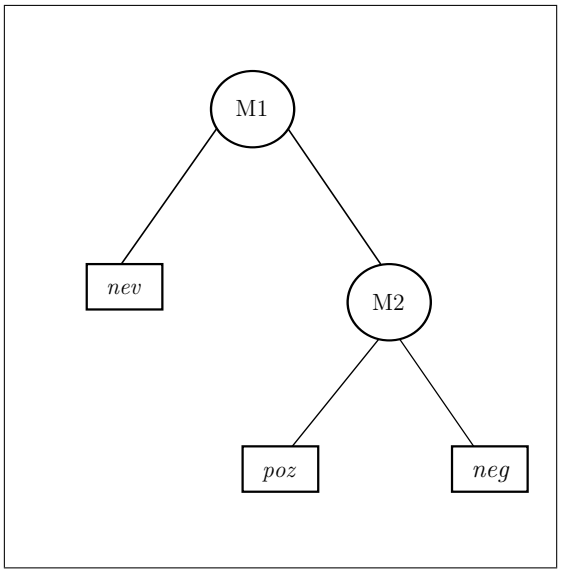
$$N = \{((M_{i,poz|neg}, M_{i,neg|nev}, M_{i,poz|nev}), s_i) : (t_i, s_i) \in T_{učna}\}$$

Množico N razbijemo na particije elementov z enakimi vrednostmi:

$$N[(s_{poz|neg}, s_{neg|nev}, s_{poz|nev}), s] = \{((M_{i,poz|neg}, M_{i,neg|nev}, M_{i,poz|nev}), s_i) \in N :$$
$$M_{i,poz|neg} = s_{poz|neg}, M_{i,neg|nev} = s_{neg|nev}, M_{i,poz|nev} = s_{poz|nev}, s_i = s\}$$

za vse možne vrednosti $((s_{poz|neg}, s_{neg|nev}, s_{poz|nev}), s)$. Funkcijo $f$, ki preslika trojico napovedi modelov SVM v končno napoved, zapišemo kot:

$$f(s_{poz|neg}, s_{neg|nev}, s_{poz|nev}) = argmax_s\{|N[(s_{poz|neg}, s_{neg|nev}, s_{poz|nev}), s]|\}$$




Slika 3.4: Hierarhija dveh modelov SVM kaskadnega modela.

Pri tem je $s \in \{neg, nev, poz\}$. Gre torej za izbiranje razreda končne napovedi po sistemu večinskega glasovanja.

### Kaskadni SVM

Sestavljen je iz dveh modelov SVM, ki sta postavljena v hierarhijo, kot prikazuje slika 3.4. Postopek klasifikacije izvajamo v dveh korakih. Začnemo pri modelu $M_1$, ki nevtralne primere ločuje od ostalih. V primeru rezultata napovedi $M_1(t_i) = nev$ je to že končna napoved in postopek končamo, sicer pa nadaljujemo z izračunom napovedi $M_2(t_i) \rightarrow \{poz, neg\}$, kar je končna napoved primera $t_i$. Učni množici za oba modela zapišemo kot:

$$\begin{aligned} T_1 &= \{(t_i, nev) : (t_i, s_i) \in T_{učna} \wedge s_i = nev\} \\ &\quad + \{(t_i, neg) : (t_i, s_i) \in T_{učna} \wedge s_i \neq nev\} \\ T_2 &= \{(t_i, s_i) \in T_{učna} : s_i \neq nev\} \end{aligned}$$

### 3.2.2 Naivni Bayes

Klasifikator NB je verjetnosti klasifikator, ki temelji na Bayesovem teoremu in predpostavki o medsebojni neodvisnosti atributov posameznih primerov. Ker ta neodvisnost v praksi ne drži povsem, se klasifikatorju reče naivni. Predpostavka o neodvisnosti omogoča pomembno poenostavitev algoritma, kar občutno poveča hitrost izvajanja algoritma.

Verjetnost, da primer $\mathbf{x}$ pripada razredu $y$, $P(y|\mathbf{x})$, lahko ob predpostavki o neodvisnosti izrazimo kot

$$P(y|\mathbf{x}) = 1/ZP(y)\prod_k P(k|y)$$

Pri tem je $1/Z$ faktor normalizacije, $P(y)$ verjetnost, da primer pripada razredu $y$, in $P(k|y)$ verjetnost, da atribut $k$ pripada razredu $y$. Klasifikacijo neoznačenega primera $\mathbf{x}$ dobimo s pomočjo funkcije

$$\hat{f}(\mathbf{x}) = argmax_{y\in Y}\{(1/Z)P(y)\prod_k P(k|y)\}$$

Ob tem velja, da je $Z = \sum_{y\in Y} P(y)P(\mathbf{x}|y) = \sum_{y\in Y} P(y)\prod_k P(k|y)$, kar pomeni, da je faktor $1/Z$ v kontekstu primera $\mathbf{x}$ konstanten in ga lahko odstranimo.

V fazi učenja algoritem NB oceni verjetnosti $P(y)$ in $P(k|y)$. $P(y)$ izračuna kot relativno frekvenco primerov, ki pripadajo razredu $y$: $P(y) = N_y/N$, kjer je $N_y$ število primerov razreda $y$ in $N$ število vseh primerov v učni množici. Po drugi strani obstaja več načinov za izraz verjetnosti $P(k|y)$. Pri tekstovni analizi pogosto uporabljamo t.i. multinominalni model NB, kjer lahko izrazimo frekvenco besede v dokumentu, tako da $P(k|y)$ izrazimo kot $P(k|y) = T_{t_{k,y}}/T_y$, pri čemer je $T_{t_{k,y}}$ število pojavitev besede $t_k$ v združenem besedilu vseh primerov razreda $y$ in $T_y$ število vseh besed v istem besedilu. NB se je v praksi izkazal kot uspešen in se pogosto uporablja za klasifikacijo besedila [17], zato je v to nalogo vključen kot referenčni klasifikator.

### 3.2.3 Večinski klasifikator

Gre za trivialen model, ki kot rezultat napovedi vedno vrne enak, in sicer večinski razred iz učne množice. Uporablja se kot referenčni model za določitev spodnje meje uporabnosti ostalih modelov.

## 3.3 Predobdelava in modeliranje besedil

### 3.3.1 Predobdelava

Prosto besedilo je pred postopkom klasifikacije potrebno očistiti nepotrebnih podatkov in šuma. Še posebej velja to za neformalna besedila z osebnimi vsebinami, kjer je razpon uporabljenih jezikovnih slogov zelo širok, od uporabe žargona in improviziranih besednih oblik do slenga, kletvic in podobno. Tviti s svojo kratkostjo še dodatno spodbujajo k iznajdljivosti v obliki krajšav, kratic in posebnih znakov. Uporaba posebnih znakov za izražanje emocij in drugih stvari je v zadnjem času postala del vsakdanjega pogovornega jezika.

Ena skupina takšnih znakov so emotikoni (angl. *emoticons*), ki so iz zaporedij nečrkovnih znakov sestavljene grafične upodobitve obraznih izrazov, npr. :-) kot izraz veselja ali :-( kot izraz žalosti. Beseda emotikon je sestavljenka iz besed emocija in ikona, iz česar lahko vemo, da se uporablja za označevanje čustev. Emotikoni so v neformalnem jeziku prisotni že dlje časa in so jih v raziskavah uporabili tudi za avtomatsko označevanje sentimenta besedil [26]. Druga skupina posebnih znakov so emodžiji (angl. *emoji*), ki so se pojavili s prihodom mobilnih platform za sporočanje. Emodži je grafični znak, ki se znotraj besedila uporablja za označevanje poljubnih pomenov, ki so opredeljeni tudi v Unicode standardu. Emodžiji imajo širši izrazni razpon kot emotikoni in lahko poleg čustev označujejo tudi druge stvari, kot npr. psa ali mačko. V raziskavi [18] so izmerili sentiment posameznih emodžijev na podlagi ročno označenega sentimenta v tvitih. V tej nalogi analiziramo besedila iz finančne domene, kjer uporaba emotikonov in emodžijev ni v navadi, zato jih nismo upoštevali.

Predobdelava besedila poteka po korakih v določenem vrstnem redu, ki je opisan v nadaljevanju. V tej nalogi so opazovana besedila tviti in temu je prilagojena tudi predobdelava, kar pomeni, da lahko nekatere običajne korake izpustimo. Tako smo npr. izpustili korak stavčne segmentacije (angl. *sentence segmentation*), s katerim besedila razbijamo na posamezne stavke, kar pri tvitih zaradi njihove kratkosti ni potrebno.

### Tokenizacija

Tokenizacija (angl. *tokenization*) je postopek, s katerim tekst razgradimo na simbole, ki navadno ustrezajo posameznim besedam, ločilom, številkam ali drugim elementom. V abecednih jezikih so besede navadno ločene s presledki, kar razčlenjevanje poenostavi. Pri tem je ključno, da pravilno upoštevamo dvoumna ločila (npr. pika na koncu stavka je samostojen člen, medtem ko pika za kratico pripada kratici) in večbesedne izraze (npr. datumi, naslov e-pošte). V našem primeru tekom tokenizacije ločil ne upoštevamo, pri ločevanju besed pa uporabimo vse znake, ki niso črke ali številke.

### Odstranjevanje neinformativnih besed

Neinformativne besede (angl. *stop words*) so splošne besede, ki se v besedilih pojavljajo pogosto in ne pripomorejo k razločevanju med tekstovnimi dokumenti. V angleškem jeziku so to določni in nedoločni členi (*a, an, the*), zaimki (*I, we, our*), oblike pogostih glagolov (*am, are, is, have, has, had, having, am, are, is ...* ), pomožni glagoli(*would, should, could, must ...* ), skrajšane glagolske oblike (*m, re, s, ve*) ter druge besedne oblike, kot so predlogi vezniki in prislovi (*and, but, or, if*).

### Krnjenje in lematizacija

Metodi krnjenja (angl. *stemming*) ali lematizacije (angl. *lemmatization*) se uporabljata pri krajšanju besednih oblik, kot so spregatve, sklanjatve in druge izpeljanke, na njihov koren ali osnovno obliko. Na ta način zmanjšamo

število različnih besed, ki jih moramo upoštevati, ter hkrati ohranimo vso informacijo in pomene, ki so vsebovani že v osnovnih besednih oblikah. Metodi se razlikujeta v načinu iskanja osnovne besedne oblike. Lematizacija se ukvarja z iskanjem leme (angl. *lemma*) oziroma oblike besede, kakršna se pojavlja v slovarju. To je lahko pri nekaterih jezikih, ki imajo veliko pregibnih oblik, zahteven problem. S tega vidika je krnjenje, ki iz besede izlušči njen koren, enostavnejši pristop. Krnjenje se v glavnem naslanja na niz preprostih pravil za odstranjevanje določenih pripon (v angleščini bi bile takšne pripone *-ed, -ing, -ly*). V tej nalogi obdelujemo besedila v angleškem in nemškem jeziku, modela za oba jezika pa sta na voljo v okviru lematizatorja *LemmaGen* [10], ki smo ga uporabili.

#### Obvladovanje besedil v tvitih

Tviti vsebujejo določene standardne tekstovne elemente, ki bi lahko vplivali na klasifikacijo, in jih želimo odstraniti. Za potrebe te naloge odstranimo vse vsebovane povezave na spletne strani (angl. *uniform resource locator, URL*). Poleg tega odstranimo uporabniška imena platforme Twitter, ker ne želimo, da si klasifikator pomaga z njimi. Glede na to, da obdelujemo finančna besedila, odstranimo še reference na delnice.

### 3.3.2 Modeliranje besedila

Po fazi predobdelave je besedilo očiščeno. Naslednji korak je pretvorba tekstovnih dokumentov v strukturirano obliko, kakršno na vhodu pričakujejo algoritmi za klasifikacijo. Za ta namen v tej nalogi uporabimo metodo za preslikavo besedil v vektorski prostor, ki jo imenujemo **vreča besed** (angl. *bag of words, BOW*). Na začetku postopka na podlagi zbirke tekstovnih dokumentov, ki jo zapišemo kot $T = \{t_1, t_2, ..., t_m\}$, določimo dimenzije vektorskega prostora. Dimenzije prostora ustrezajo tekstovnim atributom ali značilkam (angl. *feature*), prisotnih v $T$. Način njihove pridobitve je opisan v nadaljevanju, poenostavljeno pa lahko rečemo, da vsaka dimenzija oziroma

značilka ustreza eni besedi, ki je v zbirki dokumentov. Po preslikavi tekstovni dokument zapišemo kot vektor realnih števil $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, kjer $x_i$ predstavlja utež značilke *i-te* dimenzije vektorskega prostora, $n$ pa število vseh značilk v $T$. Načinov za računanje uteži je več, na splošno pa poskušamo upoštevati dva vidika:

- Pomembnost besede za neko tematiko je sorazmerna s številom pojavitev besede v dokumentu.
- Teža besede pri razlikovanju med dokumenti je manjša, če se pojavlja v večini dokumentov korpusa.

Primerjalna študija več načinov računanja uteži vektorskih komponent je predstavljena v [25]. Trije najbolj razširjeni načini so opisani v nadaljevanju.

- **Frekvenca pojavitve** (angl. *term frequency, TF*) je število pojavitev značilke v dokumentu. To zapišemo kot:

  $$x_i = TF_i,$$

  pri čemer je $TF_i$ število pojavitev *i-te* značilke v nekem dokumentu.

- Pri **binarnem označevanju** se označuje samo prisotnost značilke v dokumentu.

  $$x_i = \begin{cases} 1 & \text{če je } TF_i > 0 \\ 0 & \text{sicer} \end{cases}$$

- **TF-IDF** uteževanje je kombinacija frekvence pojavitve značilke in inverzne frekvence dokumentov (angl. *inverse document frequency, IDF*). IDF se izračuna z izrazom:

  $$IDF_i = log\frac{m}{m_i},$$

  pri čemer je $m_i$ število dokumentov, v katerih se pojavi *i-ta* značilka, in $m$ število dokumentov v korpusu. $IDF_i$ gre proti nič, ko se $m_i$ približuje $m$.

  $$x_i = TF_i \cdot IDF_i$$

Pristop TF-IDF upošteva pogostost pojavitve značilke v posameznem dokumentu skupaj z pogostostjo pojavitve značilke v zbirki dokumentov. Tako upoštevamo, da ima značilka, ki se pojavi v večini dokumentov, majhen pomen pri njihovem razločevanju. Pravzaprav gre za dinamičen način blokiranja neinformativnih besed, kjer neinformativne besede določamo glede na dani korpus besedil.

#### Pridobivanje značilk

Pri BOW prostoru govorimo o značilkah, ki jih je potrebno izluščiti iz besedil oziroma zaporedij besednih lem ali korenov, ki so rezultat predobdelave. Uporabimo preprost postopek za iskanje pogostih kratkih besednih zaporedij dolžine $n$. Takšnim zaporedjem rečemo ***n*-grami** in jih poleg posameznih besed vključimo med značilke pri oblikovanju BOW vektorskega prostora. Pri pridobivanju $n$-gramov upoštevamo dva parametra: (i) največja dolžina $n$-grama in (ii) najmanjša frekvenca $n$-grama. S prvim parametrom določimo največjo dolžino zaporedja besed v $n$-gramu. Z drugim pa določimo najmanjšo zahtevano število pojavitev $n$-grama v korpusu dokumentov.

Tako dobljenih značilk je lahko zelo veliko in nekatere izmed njih ne vsebujejo dovolj informacije za razločevanje dokumentov. Zato po koraku pridobivanja navadno sledi korak izbiranja značilk. V tej nalogi, kjer z uporabo modela SVM klasificiramo kratka besedila, izbiranje značilk ne prinese posebnega izboljšanja, zato ga izpuščamo. Dve najpogostejši metodi za izbiranje sta računanje medsebojne informacije in test $\widetilde{\chi}^2$, kar je opisano v [30].

## 3.4 Vrednotenje klasifikacijske uspešnosti

### 3.4.1 Metodologija evalvacije

Uspešnost klasifikacijskih algoritmov opazujemo znotraj zlatega standarda oziroma množice označenih primerov, ki nam je na voljo. V postopku množico primerov najprej razbijemo na učno in testno množico. Na prvi najprej

naučimo model, nato pa klasificiramo primere druge množice in rezultat primerjamo s pravilnimi oznakami. S štetjem pravilnih in nepravilnih klasifikacij primerov izračunamo mere, ki so opisane v nadaljevanju. Tipično razmerje med velikostjo učne in testne množice je 7 : 3. Število ponovitev postopka je odvisno od stopnje statistične gotovosti meritvenih rezultatov, ki jo pričakujemo pri eksperimentu. Dodatni omejitvi pri številu ponovitev sta lahko računska zahtevnost algoritma in količina podatkov glede na razpoložljive računalniške zmogljivosti. V tej nalogi razpolagamo z relativno velikim naborom podatkov, kar ob zadostnih računalniških kapacitetah omogoča veliko število ponovitev. Nabor podatkov v našem primeru je časovni tok tvitov, zato želimo v evalvaciji upoštevati časovno urejenost tvitov, kjer modele učimo na preteklih primerih in klasificiramo primere v sedanjosti ali prihodnosti.

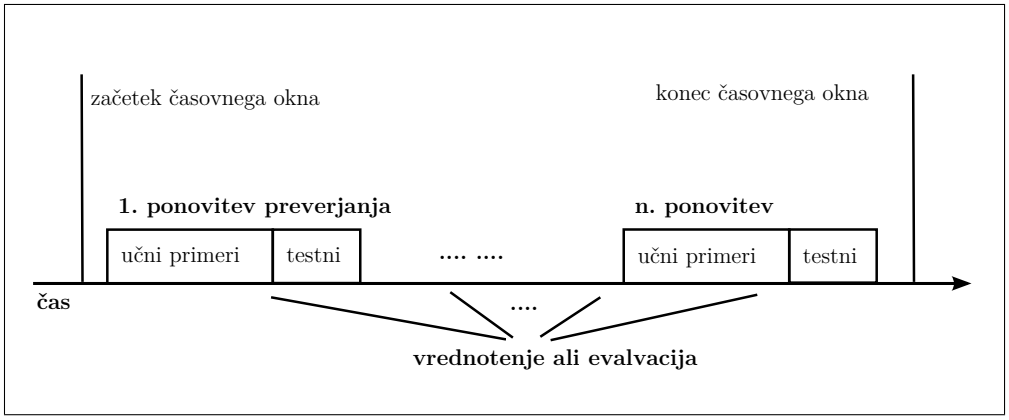
Na sliki 3.5 je shematski prikaz osnovne metodologije testiranja, ki jo z različnimi parametri uporabimo v vseh naslednjih poskusih. Opravila posameznega poskusa opredelimo na treh ravneh, hierarhično urejenih od najnižje ravni proti višjim:

1. **Preverjanje** je najosnovnejša procedura učenja in testiranja modela nad posamezno množico primerov. Vključuje razdelitev primerov na učne in testne, čemur sledi postopek učenja in testiranja modela. Na koncu so znani rezultati v obliki izbranih mer za uspešnost modela.

2. **Vrednotenje ali evalvacija** proceduro preverjanja večkrat ponovi s poljubnim številom neodvisnih vzorcev podatkov. Iz vseh dobljenih rezultatov na koncu izračunamo povprečja in standardne napake opazovanih mer evalvacije.

3. **Poskus ali eksperiment** lahko vsebuje eno ali več variant evalvacij. V okviru enega eksperimenta lahko recimo opazujemo skupino modelov v primerljivem kontekstu in na koncu primerjamo njihove rezultate.

Izraze **preverjanje**, **evalvacija** in **poskus** z opisanimi pomeni uporabljamo pri opisu eksperimentov v nadaljevanju.




Slika 3.5: Shematski prikaz časovnega prečesavanja podatkov znotraj posamezne evalvacije.

Posamezno množico zajetih primerov določimo s časovnim oknom, v katerem so primeri. Ob tem se moramo zavedati, da množice z enako dolžino časovnega okna nimajo nujno enakega števila primerov, vendar jih, ob zagotovitvi dovolj ponovitev preverjanja znotraj evalvacije, lahko obravnavamo kot ekvivalentne. Časovno okno množice za poljubno število ponovitev preverjanja naključno premikamo vzdolž celotne množice primerov, pri čemer ohranjamo konstantno velikost okna znotraj posamezne evalvacije. Vsako množico v proceduri preverjanja razdelimo na dva dela z izbiro časovne točke znotraj časovnega okna. Primere pred časovno točko uporabimo za učenje modela, preostale pa za testiranje. Razmerje velikosti učne in testne podmnožice je znotraj ene evalvacije konstantno. Z opisanim načinom izbiranja množic podatkov dosežemo medsebojno primerljivost in neodvisnost vzorcev evalvacije, kar nam po centralnem limitnem teoremu omogoča izračun standardne napake povprečja poljubnih mer. V vseh rezultatih prikazujemo standardno napako z intervalom gotovosti 95 %.

| dejanski razred | napovedani razred *pozitivni* | *nevtralni* | *negativni* | vsota |
|---|---|---|---|---|
| *pozitivni* | $PP$ | $PO$ | $PN$ | $\sum_i PS_i$ |
| *nevtralni* | $OP$ | $OO$ | $ON$ | $\sum_i OS_i$ |
| *negativni* | $NP$ | $NO$ | $NN$ | $\sum_i NS_i$ |
| vsota | $\sum_i S_iP$ | $\sum_i S_iO$ | $\sum_i S_iN$ | $\widetilde{N} = \sum_{i,j} S_iS_j$ |

Tabela 3.1: Matrika zmot za problem tri-razredne klasifikacije sentimenta. Indeksi vsot $i$ v matriki potekajo skozi tri razrede sentimenta.

### 3.4.2 Evalvacijske mere

Mere uspešnosti klasifikacije se izračunajo s štetjem pravilno klasificiranih testnih primerov. Za preglednejši prikaz preštetih primerov si ponavadi pomagamo z **matriko zmot** (angl. *confusion matrix*) [12]. Dimenzije te matrike tvorijo klasifikacijski razredi, tako da so dejanski razredi v stolpcu, klasificirani pa v vrstici. Vrednosti matrike so števci primerov, glede na ustrezne koordinate matrike. Primeri, kjer je klasificiran razred enak pravemu so zabeleženi na diagonali matrike. Matrika 3.1 je prikaz matrike zmot za problem tri-razredne klasifikacije. Uporabljeni so simboli $P$ za pozitivni, $N$ za negativni in $O$ za nevtralni. Simbol $S_i$ predstavlja enega izmed treh razredov sentimenta. Simbol $PP$ torej predstavlja število vseh v pozitivni razred pravilno klasificiranih (angl. *true-positive*) primerov pozitivnega razreda, medtem ko $ON$ predstavlja nepravilno klasificirane nevtralne primere v negativni razred. $\widetilde{N}$ je število vseh primerov v množici. S pomočjo elementov matrike zmot v nadaljevanju opisujemo evalvacijske mere uspešnosti klasifikacije.

Najpreprostejša mera je **klasifikacijska točnost** (angl. *classification accuracy*), ki kaže delež pravilno klasificiranih primerov izmed vseh primerov.

$$\text{Klasifikacijska točnost} = \frac{PP + OO + NN}{\widetilde{N}}$$

Ta mera ponavadi ne zadostuje za opredelitev tega, ali je klasikator dober ali ne. Njena glavna pomanjkljivost je, da ima ob različnih porazdelitvah razredov v testni množici različne vrednosti, čeprav bi denimo klasificirali z enakim modelom. Testna množica se v našem primeru pomika po toku sporočil, kjer se porazdelitev razredov dinamično spreminja, tako da bi se točnost spreminjala že samo zaradi tega. Pomanjkljivost točnosti lahko prikažemo še na primeru večinskega klasifikatorja, ki vedno vrne večinski razred učne množice. V primeru porazdelitve, kjer en razred zavzema 90 % populacije, bi bila točnost večinskega klasifikatorja 90 %, čeprav preostali razredi nikoli niso napovedani. V [23] je opisan primer, ki je podoben našemu in uporablja temu prilagojene mere. To sta meri **natančnosti** ali **preciznosti** (angl. *precision*) in **priklica** (angl. *recall, sensitivity*), ki se računata glede na posamezen razred klasifikacije. Natančnost je delež pravilno napovedanih primerov v nek razred. Priklic je delež pravilno napovedanih primerov nekega razreda. Kot mero, ki kombinira omenjeni meri poznamo **oceno F1**, ki je definirana kot njuno harmonično povprečje [29].

$$\text{Natančnost}(P) = \frac{PP}{PP + OP + NP}$$
$$\text{Priklic}(P) = \frac{PP}{PP + PO + PN}$$

$$F1(P) = 2 \cdot \frac{\text{Natančnost}(P) \cdot \text{Priklic}(P)}{\text{Natančnost}(P) + \text{Priklic}(P)}$$

Formule za izračun natančnosti, priklica in ocene $F1$ so zapisane za pozitivni razred, analogno velja še za preostala dva razreda. V praksi pogosto opazujemo povprečje mere $F1$ čez vse razrede. V tej nalogi je glavna mera za opazovanje klasifikacijske uspešnost povprečje $F1$ za pozitivni in negativni razred. Nevtralni razred izpustimo, ker je odvisen od ostalih dveh in ne daje

dodatne informacije. Povprečje $F1$ dveh razredov sentimenta je torej:

$$F1_{sent} = \frac{F1(P) + F1(N)}{2}$$

Zaradi urejenosti razredov sentimenta lahko tudi klasifikacijske napake razvrstimo glede na razdaljo med pravilnim in napovedanim razredom. Lahko rečemo, da ima razdalja med pozitivnim in negativnim razredom, z vmesnim nevtralnim, dolžino dve, medtem ko ima razdalja robnih razredov do nevtralnega dolžino ena. Mero, ki kaznuje samo napake dolžine dve, imenujemo $Napaka_1$.

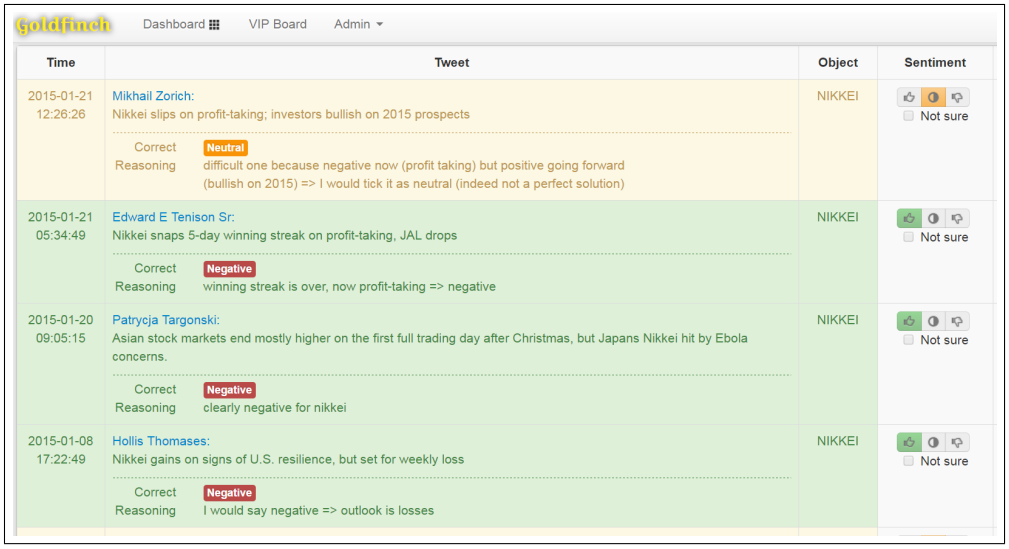
$$\text{Napaka}_1 = \frac{NP + PN}{\widetilde{N}}$$

# Poglavje 4

# Podatkovni nabor

Podatkovni nabor je zlati standard oziroma množica ročno označenih primerov, iz katerih se učijo algoritmi za strojno učenje. V našem kontekstu je vsak primer označen z enim izmed treh razredov sentimenta: *pozitivnim*, *negativnim* ali *nevtralnim*. Za označevanje smo uporabili aplikacijo z imenom *Goldfinch*, ki je namenjena označevanju tvitov ali krajših besedil s kategorijami sentimenta. *Goldfinch* omogoča usklajevanje in pregled nad potekom in kakovostjo dela skupine uporabnikov, ki tvite označujejo s sentimentom. Obenem je lahko aktivnih več projektov z različnimi skupinami označevalcev. Kot vir podatkov lahko v projektih določimo statično zbirko tvitov ali poljuben tok tvitov, ki prihaja neposredno s platforme Twitter. V primeru toka tvitov *Goldfinch* izvaja sprotno vzorčenje tvitov za označevanje, pri čemer lahko določimo parametre za stopnjo prekrivanja ali podvajanja tvitov, želeno časovno in tematsko porazdelitev tvitov in podobno. Podvajanje tvitov uporabljamo za nadzor označevanja s pomočjo opazovanja medsebojnega ujemanja označitev. Pridobivanje in priprava podatkov sicer nista del te naloge. Podatke je kot del komercialne dejavnosti pridobilo in pripravilo podjetje *SowaLabs*, ki se ukvarja z rudarjenjem podatkov s poudarkom na financah.

Sentiment v našem podatkovnem naboru izraža **signal za nakup ali prodajo** finančnih inštrumentov na borzah, zato so pri označevanju sodelo-




Slika 4.1: Posnetek zaslona aplikacije *Goldfinch*, kjer je viden modul za učenje z napotki za pravilno označitev primera.

vali strokovnjaki iz finančne domene. Pozitivni sentiment nosi pomen signala za nakup delnice, negativni pa za prodajo delnice. Nevtralni sentiment pomeni, da ne prevladuje ne signal za nakup ne za prodajo, kar pomeni, da lahko ni prisoten nobeden ali pa sta enakovredno prisotna oba. Za primer tvita s finančno tematiko vzemimo sporočilo, ki se nanaša na indeks borze v Tokiu *NIKKEI*:

> *"Azija z mešanimi občutki ob obratu dobička"*

Za razumevanje sporočila poleg splošnega jezika potrebujemo finančno predznanje. Strokovnjak je to sporočilo klasificiral kot negativno z interpretacijo: *"obrnjen dobiček pomeni, da gre ideks navzdol"*. Na sliki 4.1 je posnetek zaslona *Goldfinch* aplikacije, iz katerega lahko razberemo podatkovno strukturo posameznega primera. Sestoji iz polj za čas nastanka tvita, besedila tvita ter objekta in oznake sentimenta. Oznako sentimenta določamo glede na besedilo tvita in objekt sentimenta.

Objekt sentimenta je po definiciji iz poglavja 2 v besedilu prisotna entiteta, na katero se nanaša sentiment. V našem primeru so entitete izbrani

finančni inštrumenti, pri katerih spremljamo signale trgovanja. V tvitu je lahko prisotnih tudi več objektov, tako da moramo sentiment določiti za vsakega posebej. Različni objekti v istem tvitu imajo namreč med seboj lahko različen sentiment. Vzemimo hipotetičen primer:

*"Delnica A gre navzgor, medtem ko gre delnica B navzdol."*.

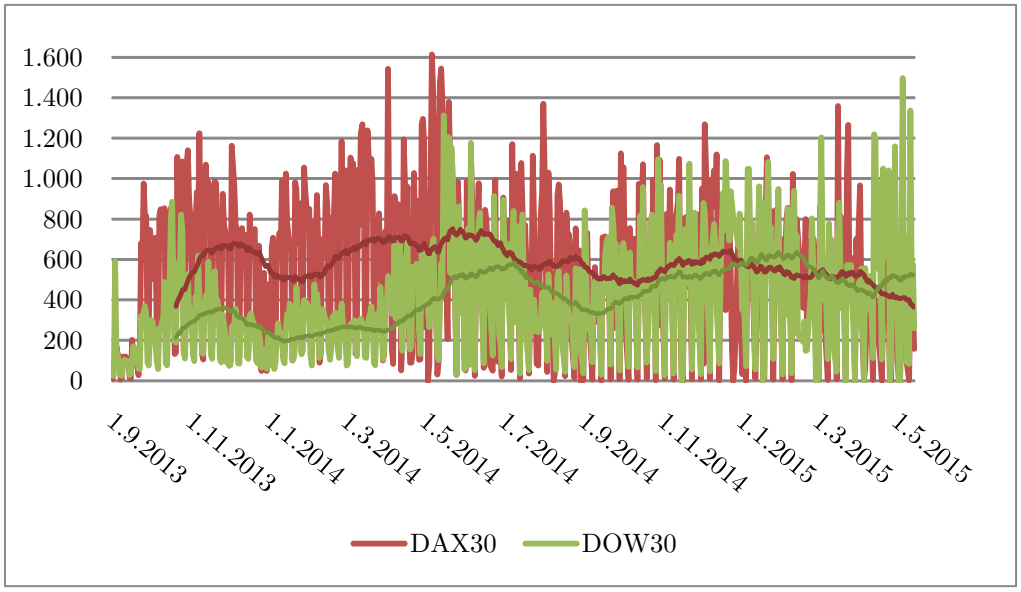
Sporočilo se nanaša na dva objekta, $A$ in $B$, razreda za sentiment pa sta očitno različna. Takšni primeri so sicer sila redki. V večini primerov se pojavljajo skupine objektov, ki si sentiment delijo. Vzemimo tvit:

*"7 Stocks To Be Careful With Going Into Earnings http://t.co/pgiZkLpH $AAPL $NFLX $GOOG $MA $GMCR $DECK $MCP"*

Tvit s finančno tematiko v tem primeru govori o sedmih delnicah, ki so naštete v istem delu stavka, kar večinoma pomeni, da se na vse nanaša enak sentiment. Eden izmed parametrov zlatega standarda je seznam obravnavanih objektov. Vzemimo, da jih izmed sedmih omenjenih objektov zlati standard obravnava pet. Na podlagi tega tvita torej dobimo pet primerov, vsakega s svojim objektom in označenim sentimentom.

V našem podatkovnem naboru so kot objekti vključene delnice dveh delniških indeksov: **DAX30** ali **DOW30**. DAX30 označuje nemški delniški indeks (nem. *Deutscher Aktienindex*, angl. *German stock index*), ki se izračunava na podlagi vrednosti delnic 30 izbranih nemških gospodarskih družb, s katerimi se trguje na Frankfurtski borzi (angl. *Frankfurt Stock Exchange*). V naših podatkih spremljamo vseh 30 delnic. Format za oznako je DAX30.simbol delnice, npr. DAX30.BMW. Poleg delnic spremljamo še sam indeks DAX30.INDEX. Podobno velja za *Dow Jones* indeks, ki se računa iz izbora 30 ameriških gospodarski družb, kjer prav tako spremljamo 30 delnic, npr. DOW30.CSCO, in indeks DOW30.INDEX.

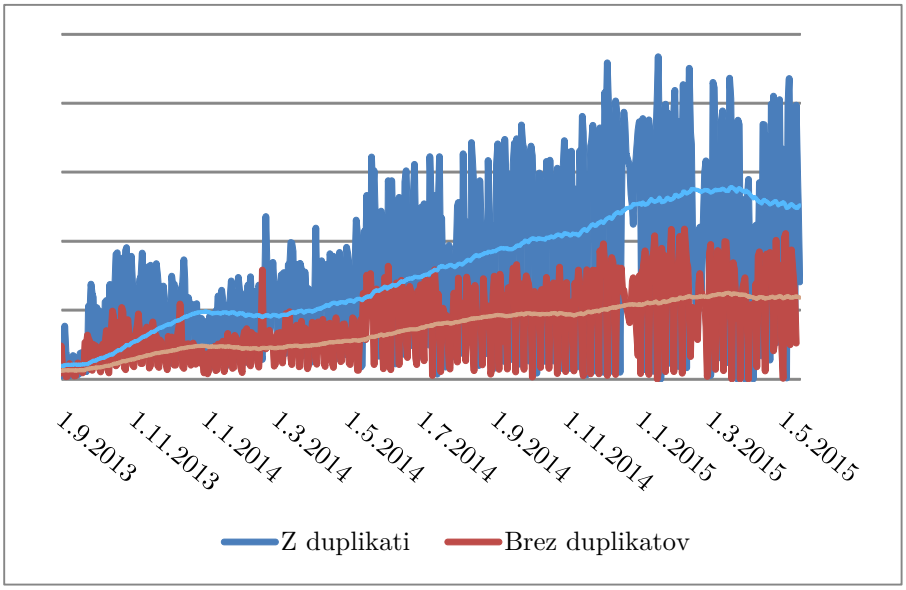
V vseh evalvacijah množici podatkov z objekti DOW30 in DAX30 modeliramo in opazujemo ločeno. Glavni razlog za to je različen jezik, ki se uporablja v obeh. Tviti DOW30 so pisani v angleškem jeziku in v glavnem




Slika 4.2: Velikost množice primerov oziroma števila tvitov skozi čas. Vidno nihanje je posledica vikendov, zato so zaradi preglednosti prikazane še vrednosti drsečega povprečja.

izvirajo iz ZDA ali Evrope, medtem ko so tviti DAX30 pisani v nemškem jeziku in pretežno prihajajo iz Nemčije. V tej nalogi smo se namreč odločili za ločeno modeliranje sentimenta za vsak jezik posebej. Z mešanjem jezikov v modelih sicer morda ne bi izgubili pri uspešnosti klasifikacije, predvidoma pa ne bi nič pridobili. Poleg tega se množici razlikujeta po količini in časovni dinamiki prihajanja tvitov. Kot je vidno na sliki 4.2, je množica DOW30 v prvi polovici opazovanega časovnega okvira izrazito manjša od množice DAX30, kar lahko vpliva na rezultate poskusov in na kar moramo biti posebej pozorni v eksperimentih.

Pri zlatem standardu je pomemben podatek o morebitnem deležu podvojenih primerov. Glede na to, da je ena glavnih operacij platforme Twitter ponovno pošiljanje tvitov, je podvojenost primerov pričakovana. S slike 4.3 je razvidno, da je v naših podatkih ponovljenih približno polovico tvitov, kar je precej. Kako to vpliva na kakovost klasifikacije, obravnavamo v poglavju 6.6.

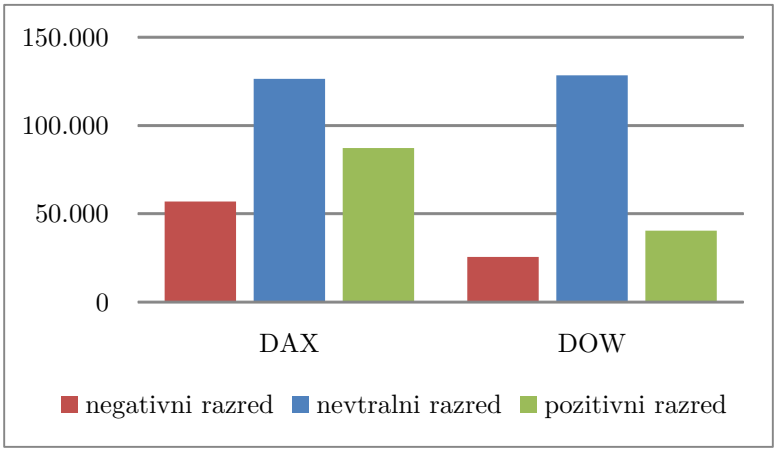



Slika 4.3: Velikost množice primerov skozi čas, z ali brez podvojenih primerov.

Razlike ležijo tudi v načinu nastanka tvitov, ki vpliva na vsebino sporočil. Tako je v množici *DAX30* občutno več računalniško generiranih tvitov, za katere je značilno ponavljanje kosov besedila s povezavo na spletno stran. Glavni motiv teh sporočil je ponavadi zvabiti uporabnika na določeno spletno stran, kar prinaša malo dodane vrednosti v vsebinskem smislu. Računalniško generirana sporočila vsebujejo bolj enolično besedišče, kar naj bi predvidoma olajšalo njihovo klasifikacijo. Na to kažejo tudi rezultati na sliki 6.1, kjer vidimo, da je klasifikacija tvitov pri podatkih *DAX30* uspešnejša kot pri podatkih *DOW30*, ki nimajo toliko avtomatsko poslanih tvitov. Razlika med množicama je tudi v porazdelitvi sentimenta, kot je razvidno s slike 4.4.

## 4.1 Zajem podatkov

Podatke so v okviru podjetja *Sowa Labs* pridobili z uporabo spletnega programskega vmesnika platforme Twitter za iskanje (angl. *Twitter Search*




Slika 4.4: Porazdelitev tvitov po razredih sentimenta za objekte DAX in DOW.

*API*)[1], ki omogoča poizvedovanje po tvitih za obdobje zadnjih nekaj dni. Statične poizvedbe izvajamo periodično, tako da sproti zajamemo vse objavljene tvite. Vsebina poizvedb so izrazi za izbrane finančne inštrumente, ki imajo predpisano obliko. Začnejo se z znakom za dolar ($), sledi pa jim borzni simbol za podjetje (npr. $GOOG, $APPL). Del zajetih tvitov aplikacija *Goldfinch* izbere za označevanje, s čimer zagotovimo učno množico za modele, preostalim tvitom pa sentiment določimo strojno s pomočjo naučenih modelov.

V tej nalogi opazujemo tvite iz časovnega obdobja med 1. 9. 2013 in 14. 5. 2015 (620 dni). Tviti so vsebinsko omejeni na delnice indeksov *DAX* in *DOW*. Skupna velikost izbrane množice je 561.176 tvitov. V tabeli 4.1 so za obe množici naštete vse delnice s pripadajočimi podatki. Naj omenimo, da se zbirka delnic, vsebovanih v obeh indeksih, zaradi ustrezanja različnim kriterijem s časom spreminja. Zbirka naštetih delnic v tabeli odraža stanje obeh indeksov v obdobju zajema tvtitov.

[1]https://dev.twitter.com/rest/public/search

| DOW | | | DAX | | |
|---|---|---|---|---|---|
| Delnica | Ime podjetja | Št. tvitov | Delnica | Ime podjetja | Št. tvitov |
| INDEX | Dow Jones Ind. Avg. | 62.377 | INDEX | Deutscher Aktienindex | 121.203 |
| AXP | American Express | 4.414 | ADS | Adidas | 6.195 |
| BA | Boeing | 5.369 | ALV | Allianz | 7.611 |
| CAT | Caterpillar | 5.054 | BAS | BASF | 4.940 |
| CSCO | Cisco Systems | 5.928 | BAYN | Bayer | 5.200 |
| CVX | Chevron | 4.981 | BEI | Beiersdorf | 4.504 |
| DD | Du Pont | 4.148 | BMW | BMW | 6.919 |
| DIS | Walt Disney | 5.694 | CBK | Commerzbank | 19.032 |
| GE | General Electric | 6.417 | CON | Continental | 3.375 |
| GS | Goldman Sachs | 7.179 | DAI | Daimler | 7.885 |
| HD | The Home Depot | 4.801 | DB1 | Deutsche Börse | 12.636 |
| IBM | IBM | 9.521 | DBK | Deutsche Bank | 19.123 |
| INTC | Intel | 8.762 | DPW | Deutsche Post | 3.938 |
| JNJ | Johnson & Johnson | 5.401 | DTE | Deutsche Telekom | 6.555 |
| JPM | JPMorgan Chase | 11.110 | EOAN | E.ON | 7.683 |
| KO | Coca-Cola | 5.786 | FME | Fresenius Medical Care | 1.254 |
| MCD | McDonald's | 5.685 | FRE | Fresenius | 3.787 |
| MMM | 3M | 3.749 | HEI | HeidelbergCement | 2.646 |
| MRK | Merck | 5.435 | HEN3 | Henkel | 3.420 |
| MSFT | Microsoft | 15.702 | IFX | Infineon Technologies | 4.965 |
| NKE | Nike | 4.779 | LHA | Deutsche Lufthansa | 10.379 |
| PFE | Pfizer | 6.184 | LIN | Linde | 2.832 |
| PG | Procter & Gamble | 4.460 | LXS | Lanxess | 3.866 |
| T | AT&T | 6.772 | MRK | Merck | 3.999 |
| TRV | Travelers | 3.105 | MUV2 | Munich Re | 2.648 |
| UNH | UnitedHealth Group | 3.885 | RWE | RWE | 7.910 |
| UTX | United Technologies | 3.494 | SAP | SAP | 5.815 |
| V | Visa | 5.589 | SDF | K+S | 7.177 |
| VZ | Verizon | 5.346 | SIE | Siemens | 9.890 |
| WMT | Wal-Mart | 5.990 | TKA | ThyssenKrupp | 5.581 |
| XOM | ExxonMobil | 5.930 | VOW3 | Volkswagen Group | 5.161 |
| | Vsota | 243.047 | | Vsota | 318.129 |

Tabela 4.1: Število tvitov po delnicah za množici *DOW* in *DAX*.

# Poglavje 5

# Platforma za izvajanje eksperimentov

Zasnovo platforme opredeljuje nabor nalog in algoritmov, ki smo jih morali implementirati za potrebe eksperimenta ter strojna in programska oprema, ki smo jo imeli na voljo za njihovo izvajanje. Za implementacijo smo uporabili **ogrodje .NET** (angl. *.NET Framework*), ki ga je razvilo podjetje Microsoft in se primarno izvaja v Microsoft Windows okolju [15]. *Ogrodje .NET* vsebuje nabor osnovnih programskih knjižnic za razvoj aplikacij in navidezni stroj (angl. *virtual machine*), v katerem se aplikacije izvajajo. Za razvoj algoritmov smo uporabili programski jezik C#, objektno orientiran jezik, ki je primarno namenjen za delo v okolju *.NET*.

Pri implementaciji algoritmov smo se naslonili na knjižnico **LATINO** za procesiranje in modeliranje besedil [7]. Uporabili smo elemente za vse faze procesiranja teksta, predstavitev teksta v vektorskem prostoru BOW in algoritme za klasifikacijo, med drugim algoritma za NB in SVM. Knjižnica LATINO se naslanja na knjižnico za lematizacijo *LemmaGen*[1] in knjižnico za *SVM*$^{light}$[2] [9]. Nad osnovnimi gradniki za tekstovno modeliranje so implementirane funkcije, potrebne za izvajanje eksperimentov, kot so zajem podatkov,

[1] `http://lemmatise.ijs.si/`

[2] `http://svmlight.joachims.org/`

faza učenja in testiranja modela, metoda drsečih oken, branje vhodnih parametrov, zapis rezultatov in podpora za vzporedno izvajanje nalog.

V okviru naloge izvajamo relativno veliko število eksperimentov. Vsi poskusi so bili izvedeni z enako ali nekoliko spremenjeno aplikacijo. Tekom procesa smo aplikacijo večkrat popravljali ali dopolnjevali, vendar smo poskrbeli, da spremembe niso vplivale na konsistentnost rezultatov. Vsak poskus v celoti opredelimo z naborom vhodnih parametrov. Teh naborov je toliko, kot je eksperimentov, ob vsaki ponovitvi poskusa pa se parametri spreminjajo. V tem je določen izziv pri vzdrževanju urejenosti vhodnih parametrov in pripadajočih rezultatov. Zato vhodne parametre zapišemo v eno samo datoteko (v formatu *JSON*), ki se ob vsaki izvedbi poskusa skupaj z izračunanimi rezultati prepiše v ciljno mapo. Tako imamo ob rezultatih vedno na voljo parametre, s katerimi smo rezultate pridobili.

## 5.1 Vhodni parametri

Za izvajanje poskusov uporabljamo enotno aplikacijo, tako da vhodni parametri v celoti opisujejo podatkovni in algoritmični vidik poskusov. Obenem velja, da lahko iz danih vrednosti parametrov nedvoumno razberemo vsebino izvedenih poskusov. Pri določanju vrednosti parametrov za posamezne poskuse si prizadevamo, da bi si bile vrednosti med seboj čim bolj podobne. Razlikoval naj bi samo tisti del parametrov, ki je potreben za opis specifičnega vidika posameznega poskusa. Tako se pri poskusu ukvarjamo samo z manjšim delom vseh parametrov. Po drugi strani so tudi izidi poskusov do neke mere medsebojno primerljivi. Tako npr. vemo, da v vseh poskusih, razen kjer to ni drugače določeno, nastopa dvo-ravninski klasifikator SVM. Standardne vrednosti parametrov smo določili postopoma, z izbiranjem tistih vrednosti, pri katerih so bili izidi najbolj zanimivi oziroma smiselni. Opisi parametrov so v nadaljevanju podani skupaj s standardnimi vrednostmi.

1. **Ime poskusa** – V enem teku aplikacije lahko opravimo več poskusov, ki jih med seboj ločimo po imenu.

2. **Časovni okvir (privzeto – od 01. 09. 2013. do 14. 05. 2015 - dolžina 620 dni)** – Časovni interval poskusa. Vzorčenje učnih in testnih množic izvajamo znotraj tega intervala.

3. **Število ponovitev (privzeto – 200)** – Število ponovitev evalvacije znotraj poskusa.

4. **Klasifikacijski algoritem (privzeto – dvo-ravninski SVM)** – Poleg dvo-ravninskega SVM so podprti še tri-ravninski SVM, kaskadni SVM, NB in večinski klasifikacijski algoritem. Znotraj poskusa lahko navedemo več algoritmov, kar pomeni izvajanje evalvacije za vsak algoritem posebej. Vsi modeli SVM imajo identične parametre, in sicer uporabljamo linearno jedro in parameter $C = 1$.

5. **Način izračuna uteži vektorjev BOW (privzeto – TF-IDF)** – Poleg TF-IDF je podprt še TF. Ostali parametri BOW prostora so za vse poskuse enaki, in sicer: *odstranjevanje neinformativnih besed* je izklopljeno; za značilke uporabljamo bigrame (*unigrami* + *bigrami*), pri čemer je *najmanjša frekvenca pojavitve* 5.

6. **Velikost množice za učenje (privzeto – 150 dni)** – Dolžina časovnega intervala za učno množico, kakor je opisano v poglavju 3.4.1. Znotraj poskusa lahko navedemo več dolžin, kar pomeni izvajanje evalvacije za vsako dolžino posebej.

7. **Velikost množice za testiranje (privzeto – 15 dni)** – Podobno kot pri učni množici.

8. **Podatkovni filtri (privzeto – *DAX30.** in *DOW30.**)** – Poskus je mogoče ponoviti za poljuben nabor podatkovnih filtrov. V tej nalogi se ukvarjamo z delnicami, ki pripadajo indeksoma *DAX30*, ki vsebuje sporočila v nemškem jeziku in *DOW30* s sporočili v angleškem jeziku. Razdelitev podatkov na ti dve podmnožici je prisotna v vseh poskusih. Znak * v vrednosti parametra je maskirni znak (angl. *wildcard*), ki pomeni zajem vseh znakov na njegovem mestu.

9. **Podvojeni primeri (privzeto – vključeni)** – Tvite, ki imajo enako ali glede na določeno normalizacijo podobno besedilo, razumemo kot podvojene. Poskuse lahko izvajamo na množicah podatkov, ki vsebujejo ali ne vsebujejo podvojenih tvitov.

10. **Učna množica z enotnim objektom (privzeto – izključeno)** – Primere učne množice lahko glede na njihove objekte uvrstimo v ločene podmnožice, iz katerih zgradimo posamezne modele. To nam omogoča opazovanje modelov za vsak objekt posebej. Način testiranja teh modelov je opredeljen s parametrom *testna množica z enotnim objektom*.

11. **Testna množica z enotnim objektom (privzeto – izključeno)** – Primere testne množice lahko glede na njihove objekte uvrstimo v ločene podmnožice, na katerih neodvisno izvajamo proceduro testiranja. Rezultate testiranja dobimo za vsak objekt posebej. Če je ob tem parametru vključen tudi parameter *učna množica z enotnim objektom*, se modeli, naučeni na primerih z enotnim objektom, testirajo na testnih podmnožicah z enakim objektom. Podrobnejša razlaga je na voljo v poglavju 6.5.

12. **Stopnja redčenja učne množice (privzeto – izključeno)** – Ob zajemu primerov v določen časovni interval lahko določimo, kakšen delež teh primerov naj se uporabi v učni množici. S tem simuliramo redčenje oziroma zmanjševanje gostote podatkov v toku (zmanjševanje števila tvitov v enem dnevu). Znotraj enega poskusa lahko določimo več stopenj redčenja, kar pomeni izvedbo več evalvacij, za vsako stopnjo posebej.

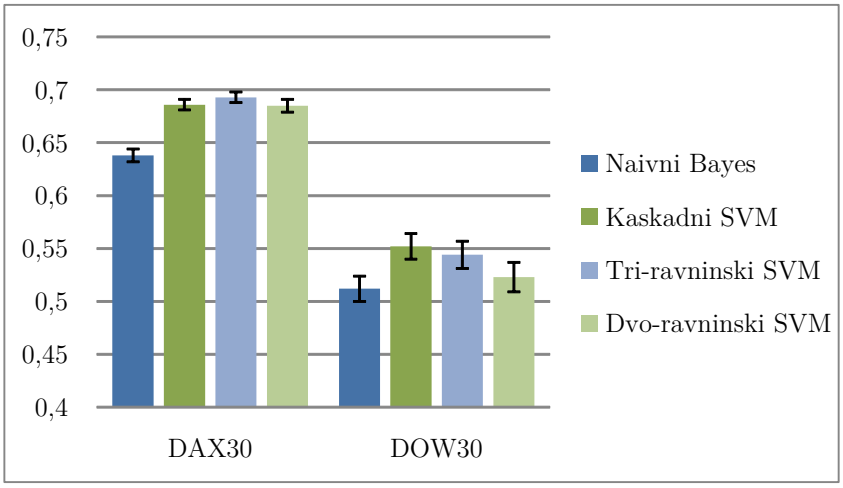
# Poglavje 6

# Opis in rezultati eksperimentov

V tem poglavju so po sklopih obravnavana različna vprašanja in predstavljeni rezultati o procesu klasifikacije sentimenta. Zlati standard in metodologija analize sentimenta izhajata iz dejanskega sistem za sprotno analizo sentimenta v sporočilih Twitter, ki se uporablja kot del poslovne aplikacije, kar pomeni, da imajo vprašanja tudi uporabno vrednost. Pristop k iskanju odgovorov na vprašanja je empiričen. To pomeni, da za vsako vprašanje poiščemo primerno simulacijo določenih vidikov sistema v nadzorovanih pogojih, odgovor pa poskušamo najti v interpretaciji izmerjenih rezultatov. Kot pristop je empirična evalvacija relativno enostaven vendar zelo uporaben način odkrivanja znanja. Glavni izziv je poiskati ustrezno simulacijo in pokazati, da njeni rezultati relevantno odgovarjajo na postavljeno vprašanje.

## 6.1 Različni klasifikacijski algoritmi

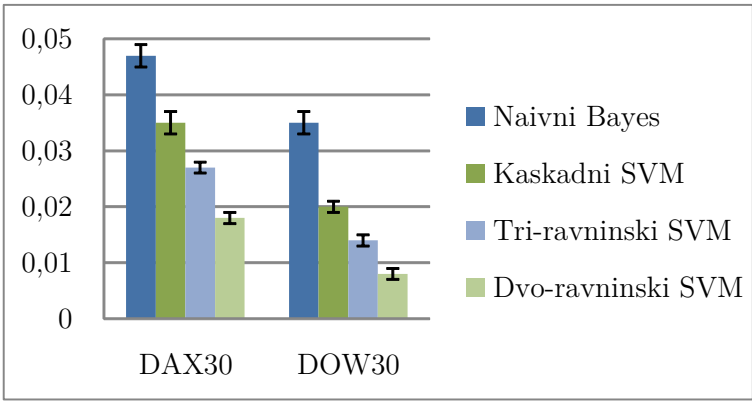
Pri klasifikaciji besedil uveljavljenih več različnih algoritmov. V tej nalogi smo osredotočeni predvsem na metodo SVM oziroma na njene izpeljanke, ki premoščajo omejitve SVM glede klasifikacije sentimenta. Medtem ko je SVM binarni klasifikator, je naš klasifikacijski problem tri-razredni. V poglavju 3.2.1 smo opisali tri načine za klasifikacijo sentimenta z uporabo SVM.




Slika 6.1: Prikaz ocen $F1_{sent}$ za vse klasifikatorje, ločene po primerih *DOW30* in *DAX30* z intervalom zaupanja 95 %.

Nekateri upoštevajo naravno urejenost razredov sentimenta, nekateri ne. Zanima nas primerjava klasifikacijske uspešnosti vseh načinov.

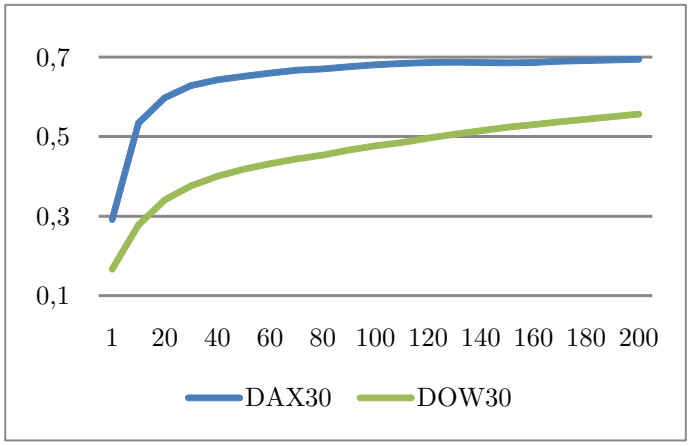
V sliki 6.1 primerjamo uspešnost vseh treh inačic modelov SVM in NB. Takoj je razvidno, da so vse izpeljanke SVM statistično signifikantno boljše od NB, medtem ko med njimi samimi ni velikih razlik. Dvo-ravninski model ima glede na ostala dva sicer relativno nizko oceno $F1_{sent}$, vendar se ob opazovanju mere $napaka_1$ na sliki 6.2 izkaže, da je delež napak med negativnim in pozitivnim razredom pri njem izrazito manjši kot pri ostalih. V praktični aplikaciji, kjer uporabniku ob tvitu prikazujemo tudi napovedan razred sentimenta, je to pomembna prednost. Velikost in percepcija napake med pozitivnim in negativnim razredom pri uporabniku namreč vzbudi precej večje nezaupanje v aplikacijo kot napaka, ki vsebuje nevtralni razred. Zaradi te lastnosti smo dvo-ravninski model uporabljali tudi pri vseh ostalih poskusih.




Slika 6.2: Prikaz ocen $Napaka_1$ za vse klasifikatorje, ločene po primerih *DOW30* in *DAX30* z intervalom zaupanja 95 %.

## 6.2 Velikost učne množice

Ocenjujemo gibanje mere uspešnosti klasifikacijskega modela glede na velikost učne množice. Empirična evalvacija je pravzaprav edini način za iskanje optimalne velikosti učne množice tako, da s postopnim povečevanjem množice opazujemo uspešnost klasifikacije. Iz rezultatov na sliki 6.3 je razvidno, da se naglo naraščanje ocene uspešnosti klasifikacije konča nekje pri 20 dneh časovnega razpona učne množice. To bi lahko vzeli za spodnjo mejo, kjer še dobimo uporaben klasifikacijski model. Od spodnje meje naprej ocena narašča počasneje in, vsaj v primeru DAX30, se pri 200 dneh že skoraj ustavi. To bi lahko vzeli za zgornjo mejo, od koder z nadaljnjim povečevanjem ne pridobimo več. V primeru delnic DOW30 se naraščanje ne ustavi in zgornje smiselne velikost učne množice ne moremo določiti. Razlaga za to bi lahko bila tudi večje število avtomatsko generiranih tvitov v primeru DAX30, ki zmanjšujejo količino uporabljene informacije. Nasploh bi lahko obstoj zgornje meje pri določeni velikosti učne množice razložili z omejenostjo besednjaka, ki se znotraj neke domene uporablja za izražanje sentimenta. Z drugimi besedami, ko učna množica enkrat zajame dovolj besedil, pride do nasičenja informacij in dodatno modeliranje sentimenta v besedilu ni več smiselno.

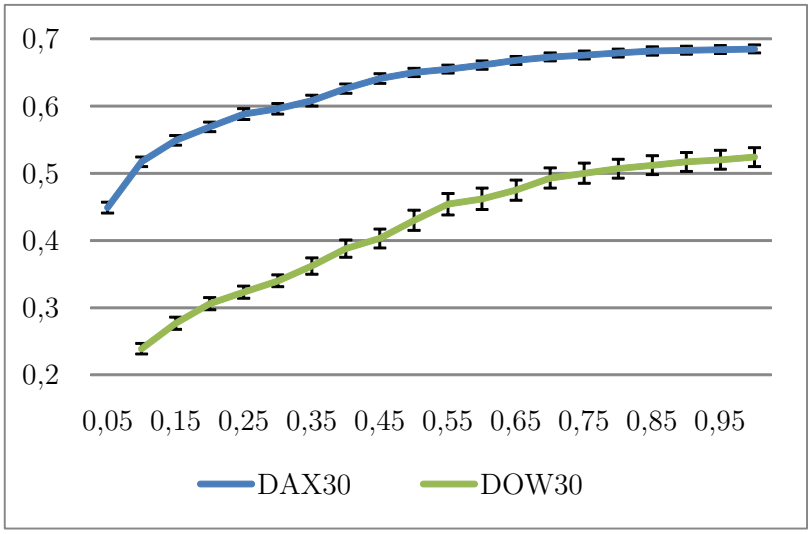



Slika 6.3: Ocena $F1_{sent}$ glede na velikost učne množice izražene v dnevih dolžine vzorca, ločeno za primere *DOW30* in *DAX30*.

## 6.3 Razredčena učna množica

Pridobivanje označenih primerov za učenje modelov je povezano s stroški, zato ni nepomembno vprašanje, koliko učnih primerov je potrebnih za dovolj dober klasifikacijski model. Na vprašanje, kolikšna naj bo absolutna velikost učne množice, smo odgovorili v prejšnjem poglavju. V tem poglavju se sprašujemo, kako spreminjanje velikosti toka označenih primerov vpliva na kakovost modela. To smo izvedli na način, da obstoječo množico primerov po času enakomerno redčimo in opazujemo spreminjanje modela.

V poskusu začnemo z najmanjšo, oziroma najbolj razredčeno, množico in potem v enakomernih korakih povečujemo delež primerov. Na sliki 6.4 je vidno naraščanje ocene $F1_{sent}$, ki je najprej hitrejše, potem se umiri, vendar se do konca (kjer redčenja več ni) ne ustavi povsem. Mogoče bi v primeru množice *DAX30* konec krivulje lahko interpretirali kot maksimum oziroma nasičenje. Hitrejše učenje modela bi lahko pojasnili z vsebino oziroma načinom nastanka tvitov. V poglavju 4 smo namreč omenili, da je v množici DAX30 več avtomatsko generiranih tvitov, kar bi lahko olajšalo modeliranje sentimenta. Po drugi strani je tudi testna množica bolj podobna učni, kar olajša klasifikacijo. Iz naraščajoče krivulje torej lahko sklepamo,




Slika 6.4: Ocena $F1_{sent}$ v odvisnosti od stopnje razredčenosti učne množice za dve množici primerov *DOW30* in *DAX30*, z intervalom zaupanja 95 %.

da si redčenja ne moremo privoščiti, vzorčenje toka tvitov lahko kvečjemu povečamo.

## 6.4 Časovna oddaljenost učne in testne množice

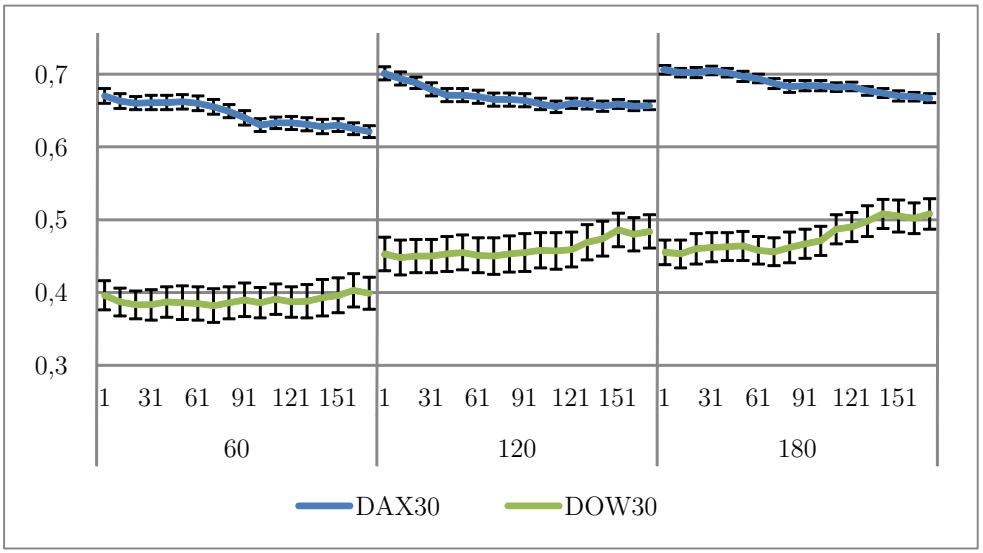
Teza o spremenljivosti uspešnosti klasifikacije glede na časovno oddaljenost testne množice od učne izhaja iz dejstva, da se besednjak, teme in način izražanja sentimenta znotraj vsebine sporočil s časom spreminja. Model naučen na starejših podatkih zato novejših ne klasificira pravilno. Ob večjem časovnem razmiku učne in testne množice, je večja tudi možnost, da se bo model zmotil. V poskusu, kjer je osnovna časovna enota en dan, smo razmik med učno in testno množico postopoma povečevali od nič do 180 dni in opazovali spremembe v uspešnosti modela. Razmik pomeni število dni od konca časovnega intervala učne množice do začetka časovnega intervala testne množice. To smo ponovili za tri različne velikosti učne množice (60, 120 in 180 dni), da bi opazovali, kako na pojav vpliva tudi absolutna velikost učne množice. Število vzorcev oziroma ponovitev poskusa smo iz 200,

ki je privzet pri preostalih poskusih, zmanjšali na 100. Ta poskus namreč zahteva izredno velik časovni razpon množice, iz katere se izdvojita učna in testna množica. Če gledamo največje razpone dobimo $maks_{razpon}(učna + razmik + testna\ množica) = 180 + 180 + 15 = 375$ dni, kar je približno polovica celotnega razpona, torej 620 dni.

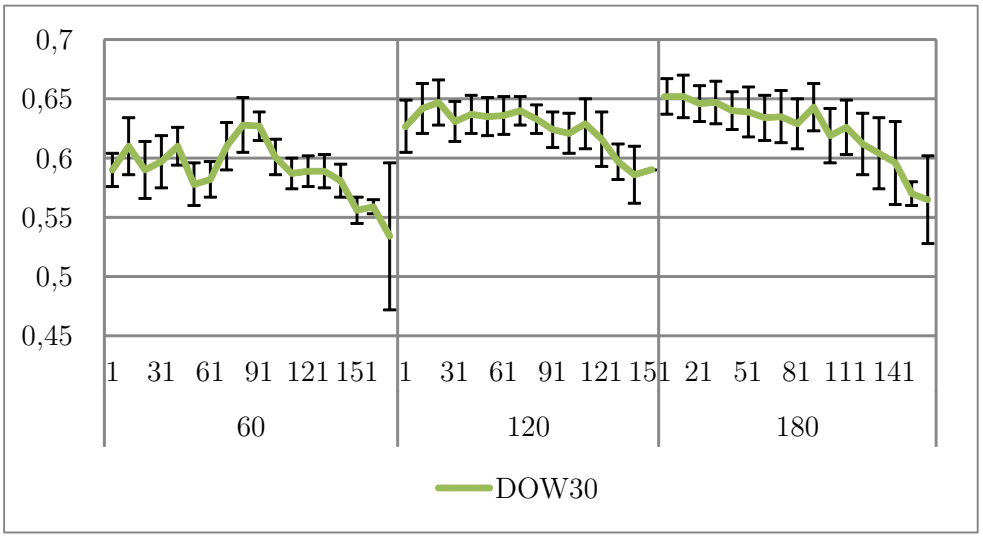
Slika 6.5 potrjuje tezo o padcu uspešnosti klasifikacije z razmikom, vendar samo za množico DAX30. Pri tej množici se lepo vidi postopno padanje ocene $F1_{sent}$ skupaj s časovnim razmikom, kar velja pri vseh treh velikostih učne množice. Vidimo tudi, kako velikost učne množice pripomore k dvigu ocen pri vseh razmikih. Pri množici DOW30 ne opazimo nič od tega, zato poskus za to množico v nekoliko spremenjeni obliki ponovimo. S slike 4.2 je za množico DOW30 razvidna neenakomernost porazdelitve tvitov skozi čas, na katero smo že opozorili. In sicer je v času pred 1. 11. 2014 količina tvitov občutno manjša, tako od množice DOW30 v času po 1. 11. 2014, kot od množice DOW30, ki ima enakomerno porazdelitev ves čas. Pri ponovitvi poskusa zato podatke opazujemo v časovnem okviru od 1. 11. 2014 do konca, število ponovitev pa zaradi manjšega razpona zmanjšamo na 40. Slika 6.6 kaže rezultate ponovljenega poskusa. Rezultati imajo zaradi manjšega števila ponovitev sicer večjo standardno napako, vendar ponavljajoči se vzorci dovoljujejo interpretacijo o padanju klasifikacijske uspešnosti s časovnim razmikom. Poleg tega smo pokazali tudi na nezanemarljivost dejavnika neenakomerne časovne porazdelitve tvitov pri izvajanju poskusa.

## 6.5 Sestava učne množice glede na objekt primera

Vsakemu primeru zlatega standarda je prirejen objekt, v našem primeru so to delnice borznih indeksov DOW30 in DAX30. Glede na objekt zlati standard sestavlja več podmnožic in sentiment lahko modeliramo za vsako podmnožico posebej. Namen tega eksperimenta je ugotoviti, ali se vsebina




Slika 6.5: Ocena $F1_{sent}$ glede na razmik med učno in testno množico. Na vodoravni osi imamo skalo na dveh ravneh: časovni razmik in tri velikosti učne množice. Prikaz je ločen za dve množici *DOW30* in *DAX30*. Interval zaupanja je 95 %.




Slika 6.6: Ocena $F1_{sent}$ glede na razmik med učno in testno množico ob zmanjšanem časovnem okvirju od 1. 11. 2014 do 14. 5. 2015 za množico *DOW30*. Na vodoravni osi imamo skalo na dveh ravneh: časovni razmik in tri velikosti učne množice. Označen interval zaupanja je 95 %.

tvitov v podmnožicah po posameznih delnicah dovolj razlikuje od vsebine splošnih tvitov, da to lahko vpliva na uspešnost modelov sentimenta.

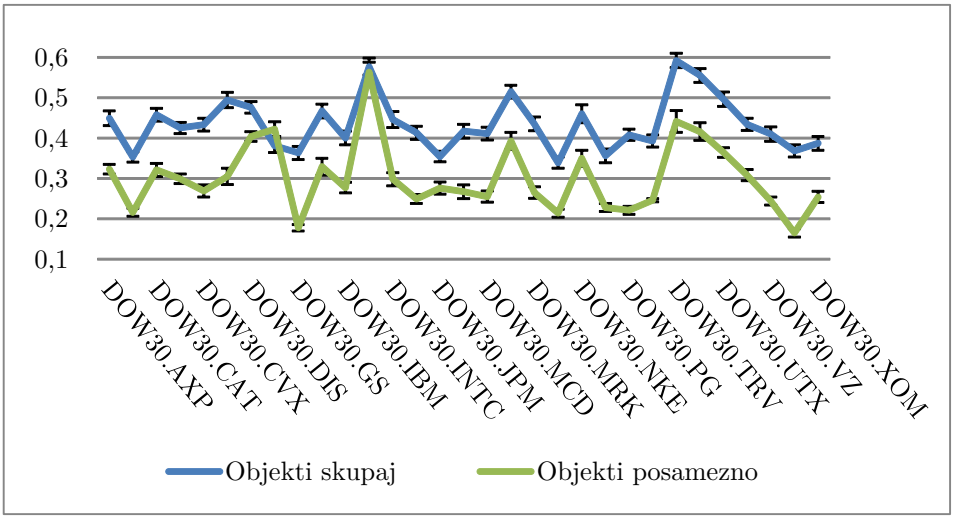
Množico vseh tvitov $T$, objektov $O$ in primerov $P$ zlatega standarda zapišemo kot:

$$T = \{t_1, t_2, ..., t_l\}, \quad O = \{o_1, o_2, ..., o_m\}$$
$$P = \{(t_i, o_j) : t_i \in T, o_j \in O\},$$

pri čemer je $l$ število tvitov, $m$ število različnih objektov (delnic). Množico, ki vsebuje samo primere z objektom $o$, zapišemo kot $P(o) = \{(t_i, o_j) \in P : o_i = o\}$. Učno in testno množico, ki jih pridobimo iz $P$, označimo kot $P_{učna}$ in $P_{testna}$.

V poskusu imamo na eni strani klasifikacijski model $M$, ki ga učimo na množici $P_{učna}$, in na drugi strani množico modelov $M(o_i)$, ki jih učimo na podmnožicah $P_{učna}(o_i)$ za vsak $o_i \in O$. Uspešnost modelov na obeh straneh medsebojno primerjamo $M \leftrightarrow M(o_i)$ za vsak $o_i \in O$ posebej. Rezultati poskusa so prikazani na slikah 6.7 in 6.8. Iz njih je razvidno, da je model $M$ konsistentno boljši od modelov $M(o_i)$. To se ne ujema z domnevo, da določene posebnosti, ki veljajo samo za primere vezane na nek objekt, modelu pri klasifikaciji primerov istega objekta omogočajo določeno prednost. Eden izmed razlogov je zagotovo različna velikost množic $P_{učna}$ in $P_{učna}(o)$. Povprečna velikost $|P_{učna}|$ se namreč giblje okoli 60.000 primerov, $|P_{učna}(o)|$ pa le okoli 2000 primerov. Glede na rezultate iz poglavja 6.2 takšna razlika že bistveno vpliva na kakovost modela.

Ostaja odprto vprašanje, kaj se zgodi, če velikost učne množice $|P_{učna}(o)|$ povečujemo do točke, ko uspešnost modela $M(o)$ doseže svoj maksimum. Na sliki 6.9 so rezultati poskusa, kjer velikost množice postopno povečujemo. Uspešnost $M(o)$ ob tem raste in se sicer približuje uspešnosti $M$, vendar nam zaradi premajhnega števila primerov ne uspe pokazati, ali bi model $M(o)$ v klasifikacijski uspešnosti dejansko dosegel ali celo prehitel model $M$. Lahko bi ubrali tudi obratno pot in zmanjševali množico $P_{učna}$ ob ohranjanju podmnožic $P_{učna}(o)$, vendar bi ob zmanjševanju odstranili tudi del primerov iz

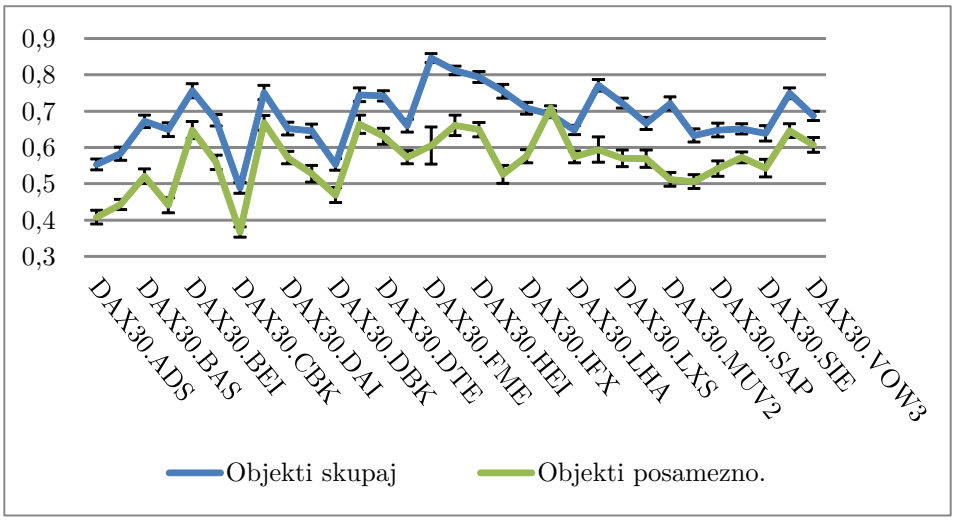



Slika 6.7: Ocena $F1_{sent}$ glede na način sestave učne množice za izbrano skupino delnic *DOW30*. Označen interval zaupanja je 95 %.

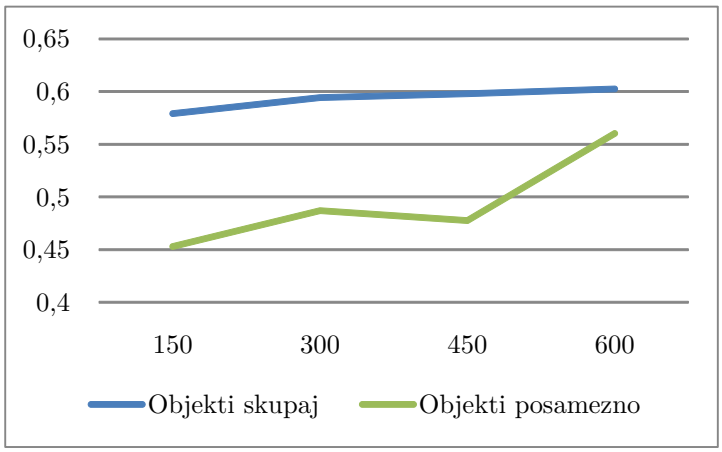
podmnožic $P_{učna}(o)$, ki naj bi, glede na našo hipotezo, igrali pomembno vlogo pri klasifikaciji testnih primerov $P_{testna}(o)$. Ob upoštevanju tega postane primerljivost uspešnosti obeh pristopov vprašljiva.

## 6.6 Izločitev podvojenih primerov

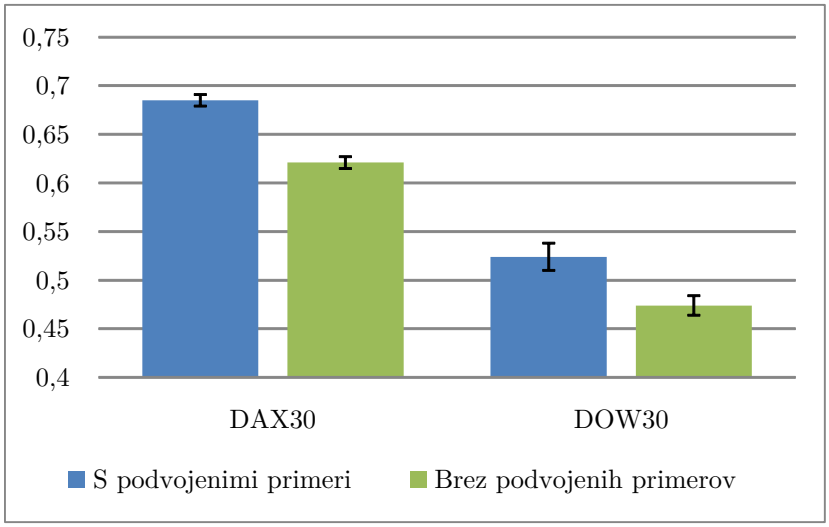
Podatkovni množica je pridobljena z vzorčenjem tematskega toka tvitov v določenem časovnem obdobju. Glede na to, da Twitter omogoča ponovno objavljanje tvitov (angl. *retweet*), lahko v vzorcu pričakujemo podvojena sporočila. Poleg tega se v fazi preslikave besedila v vektorski prostor izgubijo nekatere razlike med sporočili, kar povzroči nove podvojitve. Podvojeni primeri so vsi primeri, katerih tviti se v vektorskemu prostoru vreče besed preslikajo v enak vektor in imajo obenem enak objekt. Kot vidimo na sliki 4.3, je v našem podatkovnem naboru približno polovico tvitov podvojenih in zanima nas, kako to vpliva na uspešnost klasifikacije. Pri takšnem deležu podvojenih primerov se namreč pri razdelitvi na učno in testno množico zelo veliko enakih primerov pojavi v obeh. Manjša raznolikost besedil na splošno




Slika 6.8: Ocena $F1_{sent}$ glede na način sestave učne množice za izbrano skupino delnic *DAX30*. Označen interval zaupanja je 95 %.




Slika 6.9: Ocena $F1_{sent}$ v odvisnosti od velikosti učne množice, izražene z dolžino časovnega intervala (150, 300, 450, 600 dni). Vrednosti serije *objekti posamezno* predstavljajo povprečje ocen modelov čez vse objekte.



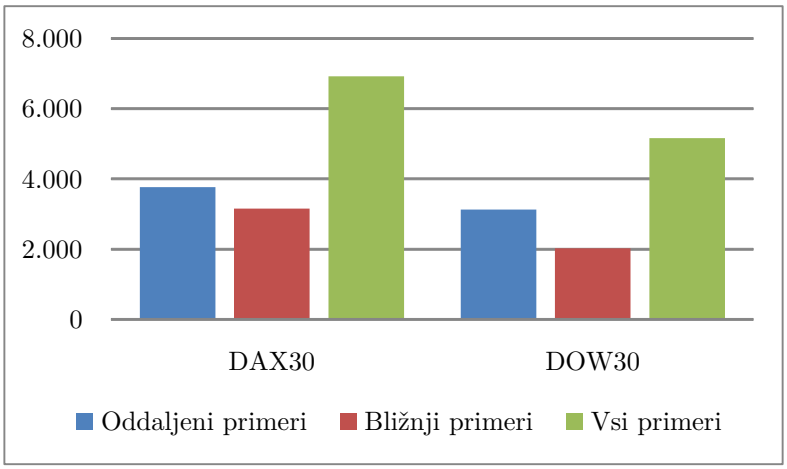

Slika 6.10: Ocena $F1_{sent}$ za množici z in brez podvojenih primerov, ločeno za *DOW30* in *DAX30*. Interval zaupanja je 95 %.

olajša klasifikacijo primerov. Poleg tega na podlagi izkušenj vemo, da je SVM uspešnejši pri klasifikaciji testnih primerov, ki jih je že videl v učni množici.

Pri poskusu primerjamo uspešnost klasifikacije dveh podatkovnih množic: (i) vključuje podvojene primere (ii) podvojeni primeri so odstranjeni. Rezultati so prikazani na sliki 6.10. Vidimo pričakovan padec ocene $F1_{sent}$ za množico brez podvojenih primerov, in sicer za približno 15 %.

## 6.7 Ločevanje primerov glede na oddaljenost od ravnine SVM

Delovanje modela SVM temelji na umestitvi ravnine v večdimenzionalni vektorski prostor tako, da med seboj loči primere dveh različnih razredov. Vsak primer, preslikan v vektorski prostor, leži na določeni oddaljenosti od ravnine. Primeri, ki so bolj oddaljeni od ravnine so posledično bolj oddaljeni tudi od primerov nasprotnega razreda ter se od njih jasneje ločijo. Sklepamo lahko, da je klasifikacija od ravnine bolj oddaljenih primerov zanesljivejša in s tem tudi uspešnejša. Namen tega poskusa je ugotoviti, kako oddaljenost




Slika 6.11: Povprečna velikost podmnožic glede na oddaljenost od ravnine SVM, ločeno za *DOW30* in *DAX30*.

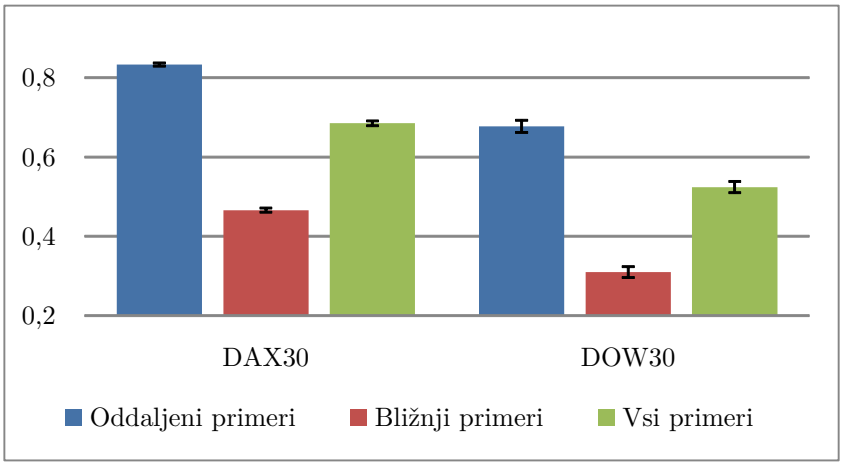
primera od ravnine vpliva na točnost njegove klasifikacije. Pri izvedbi poskusa, se naslanjamo na lastnost modela SVM, da ob klasifikaciji primera vrne tudi njegovo oddaljenost od ravnine. Glede na njihovo oddaljenost, primere v testni fazi razdelimo na dve disjunktni podmnožici bližjih in bolj oddaljenih primerov. Mere uspešnosti klasifikacije nato opazujemo za vsako podmnožico posebej.

Pri poskusu uporabimo dvo-ravninski model SVM, v katerem za vsak primer dobimo dve razdalji od ravnin. Ker za posamezen primer potrebujemo eno razdaljo, izmed obeh razdalj vzamemo tisto, ki je šibkejši člen klasifikacije. Ob predpostavki, da z oddaljenostjo od ravnine gotovost klasifikacije narašča, je to manjša razdalja. Testne primere uvrstimo v eno izmed podmnožic glede na primerjavo njihove oddaljenosti od ravnine z referenčno razdaljo, ki jo izračunamo v učni fazi glede na porazdelitev oddaljenosti vseh učnih primerov. Na sliki 6.11 je prikazana povprečna velikost celotne testne množice in podmnožic bližnjih in oddaljenih primerov. Vidimo, da je razmerje med velikostjo podmnožic približno $4:6$ v korist podmnožice oddaljenih primerov.

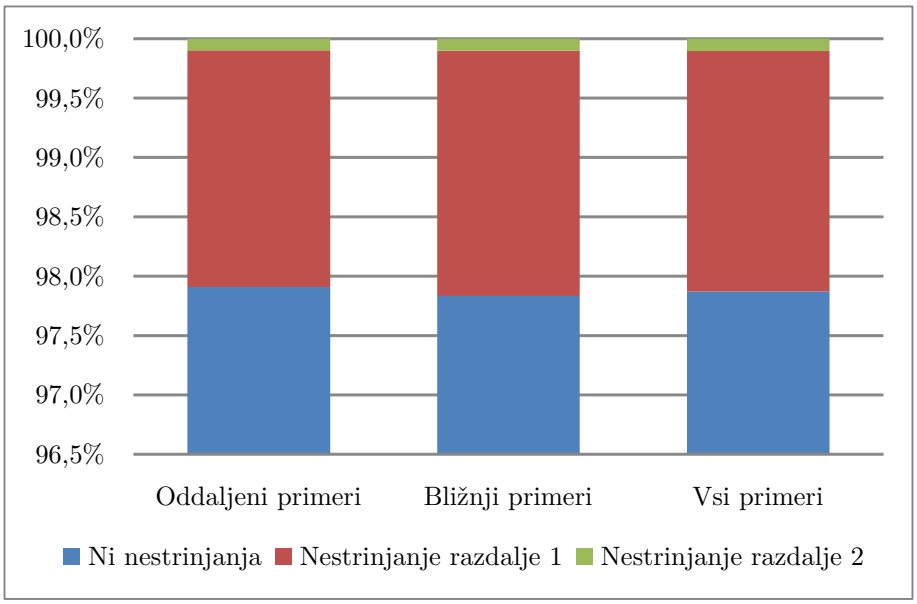
Rezultati primerjave klasifikacijske uspešnosti na obeh podmnožicah so prikazani na sliki 6.12. Ocena $F1_{sent}$ je podana za podmnožici bližnjih in oddaljenih primerov ter za celotno množico. S slike je razvidno, da ima najvišjo oceno $F1_{sent}$ particija oddaljenih primerov, in sicer mnogo višjo od ocene za celotno množico. Po drugi strani je ocena $F1_{sent}$ za množico bližnjih primerov prav tako mnogo nižja. Rezultati torej pritrjujejo tezi, da je z večjo oddaljenostjo primerov od ravnine klasifikacija uspešnejša.

Postavi se logično vprašanje, kakšna je torej razlika med primeri obeh podmnožic. Pričakovali bi, da vsebujejo besedila ravnini bližjih primerov več šuma in protislovij. Ali pa da so primeri bliže ravnini vsebinsko težji za klasifikacijo in zato pri ročnem označevanju pride do več napak in protislovij. Obstajajo tudi primeri, ko ljudje, ki označujejo, sploh ne morejo najti konsenza glede pravilne klasifikacije. Za potrebe opazovanja medsebojnega strinjanja med označevalci, naključno izbrane primere večkrat pošljemo v označevanje. Glede na to, da so večkrat označeni primeri naključno izbrani, bi morala biti njihova porazdelitev med obema podmnožicama enakomerna – toda le v primeru, da oddaljenost od ravnine ne vpliva na težavnost klasifikacije in s tem na stopnjo nestrinjanja med različnimi oznakami za isti primer. S slike 6.13 je razvidno, da je porazdeljenost v resnici skoraj povsem enakomerna in da težavnost ročne klasifikacije torej ni povezana z oddaljenostjo od ravnine.

Na zadnje vprašanje torej poskus ne da končnega odgovora. Odgovor ostaja skrit v notranjosti modela SVM, ki ob rezultatu klasifikacije ne ponuja človeku prijazne razlage razlogov za klasifikacijo, kar je ena izmed njegovih slabih lastnosti. Pri iskanju odgovora bi si lahko pomagali s splošno metodo za razlago klasifikatorjev, ki je opisana v [22] in [27].




Slika 6.12: Ocena $F1_{sent}$ za podmnožici primerov glede na njihovo oddaljenost od ravnine SVM, ločeno za *DOW30* in *DAX30*. Interval zaupanja je 95%.




Slika 6.13: Porazdelitev večkrat označenih primerov glede na stopnjo nestrinjanja. Nestrinjanje razdalje 1 pomeni nestrinjanje med nevtralnim in pozitivnim ali negativnim razredom. Nestrinjanje razdalje 2 pomeni nestrinjanje med negativnim in pozitivnim razredom. Porazdelitve so prikazane za podmnožici bližnjih in oddaljenih primerov ter celotno množico.

# Poglavje 7

# Sklepne ugotovitve

V tej nalogi smo se seznanili s področjem analize sentimenta ter z razlogi za popularnost in razmah tega področja v zadnjem času. Podrobneje smo predstavili tehnike in podrobnosti metod strojnega učenja pri klasifikacije sentimenta v tekstovnih dokumentih. Opisali smo osrednji algoritem pri evalvaciji, metodo SVM, in razloge za njeno razširjenost. Pri izbiri najbolj primernega nabora orodij in načinov obdelave besedil igrajo pomembno vlogo posebnosti in značilnosti obravnavanih podatkov. V našem primeru je vir podatkov platforma Twitter za izmenjavo kratkih sporočil, za katere je značilna jedrnatost izražanja in improvizirane besedne forme, ki dodatno zmanjšujejo podatkovno zgoščenost v besedilih. Analizo sentimenta v besedilih smo definirali kot tri-razredni problem klasifikacije dokumentov, ki vsebujejo znan objekt sentimenta.

Postavili smo metodološke temelje empirične evalvacije, pri kateri smo morali upoštevati časovno urejenost v toku sporočil. To pomeni, da smo modele vedno učili na primerih, ki so bili v času pred testnimi primeri. Poskrbeli smo za pravilno vzorčenje učnih in testnih množic primerov, kar je omogočalo izračun standardne napake povprečja mere uspešnosti klasifikacije v večini poskusov. Za opazovane mere smo upoštevali dinamiko spreminjanja porazdelitve klasifikacijskih razredov v časovnem toku sporočil.

Seznanili smo se z obstoječim sistemom za sprotno analizo sentimenta v toku sporočil. Podatki, ki so se nabrali v letih delovanja sistema, so služili kot podlaga pričujoči evalvaciji. Sistem za evalvacijo smo implementirali v programskem jeziku C# ob uporabi programskih knjižnice za tekstovno rudarjenje *LATINO*, ki vsebuje specializirane zunanje knjižnice za posamezna področja. Sistem je zasnovan kot enotna aplikacija, ki potrebne parametre za opis vseh eksperimentov dobi na vhodu in na izhodu vrne izračunane rezultate.

Nalogo smo sklenili z opisi sedmih eksperimentov, ki naj bi odgovorili na postavljena vprašanja:

1. Primerjali smo uspešnost različnih algoritmov za reševanje problema tri-razredne klasifikacije sentimenta, ki bodisi upoštevajo bodisi ne upoštevajo urejenost razredov sentimenta. Ugotovili smo, da se pri klasifikaciji tvitov izpeljanke metode SVM mnogo bolje obnesejo od metode NB. Ocena $napake_1$ je najboljša pri dvo-ravninski modelu SVM, ki edini upošteva urejenost razredov sentimenta.

2. Izmerili smo spreminjanje kakovosti klasifikacijskega modela ob različnih velikostih učne množice podatkov in ugotovili asimptotično približevanje mere uspešnosti klasifikacije določeni zgornji meji ob povečevanju množice učnih primerov. Po dosegu zgornje meje nadaljnje izboljšanje samo s povečevanjem učne množice ni več mogoče.

3. Poskušali smo ugotovi, ali lahko ob zmanjševanju gostote označenih primerov v določenem časovnem razdobju prihranimo pri stroških označevanja in obenem obdržimo zadovoljivo raven kakovosti modelov. Prav tako kot pri prejšnjem poskusu smo ugotovili asimptotično približevanje uspešnosti klasifikacije ob povečevanju gostote primerov, vendar pri opazovanih učnih množicah zgornja meja še ni bila dosežena. To pomeni, da si redčenja učne množice ne moremo privoščiti.

4. Opazovali smo, kako povečevanje časovnega razkoraka med učnimi in testnimi primeri vpliva na kakovost modela. Dobljeni rezultati kažejo, da se s povečevanjem časovnega razkoraka uspešnost klasifikacije manjša.

5. Opazovali smo vpliv grupiranja učnih podatkov glede na objekt primerov na uspešnost klasificiranja primerov z enakimi objekti. Ugotovili smo, da se ob grupiranju primerov po objektih zmanjšuje absolutna velikost učne množice, kar zmanjša tudi uspešnost klasifikacije.

6. Ker so v realnem scenariju v podatkovnem naboru tudi podvojeni primeri, nas je zanimalo, kako bi odstranitev podvojenih primerov vplivala na uspešnost modelov. Po pričakovanju se je uspešnost klasifikacije ob odstranjenih primerih vidno zmanjšala.

7. Nazadnje smo raziskovali uspešnost klasifikacije v odvisnosti od oddaljenosti od ravnine, kakršno ob klasifikaciji vrne metoda SVM. Testne primere smo glede na oddaljenost od ravnine SVM razdelili v dve podmnožici. Kot smo lahko videli, je uspešnost klasifikacije pri podmnožici oddaljenih primerov mnogo višja glede na celotno testno množico in obratno pri podmnožici bližjih primerov. To je v kontekstu vektorske predstavitve primerov in algoritma SVM pričakovan rezultat. To spoznanje bi lahko imelo uporabno vrednost v aplikacijah, pri katerih klasifikacija vseh sporočil ni potrebna in zadostuje (kvalitetnejši) prikaz porazdelitve sentimenta nad reprezentativnim vzorcem sporočil.

Uporabljen pristop in izvedba empirične evalvacije nam ob opazovanju danih podatkov ponudita določen vpogled v posamezne vidike sistema za avtomatsko klasifikacijo sentimenta v tvitih. Kot kaže, bi ob večjem naboru podatkov o posameznih vidikih klasifikacije lahko izvedeli še več. Zdaj bi to pravzaprav bilo celo že izvedljivo, saj je od odvzema podatkov za potrebe evalvacije preteklo že skoraj leto dni, pri čemer se tempo označevanja novih podatkov v podjetju *Sowa Labs* medtem ni zmanjšal. Čeprav implementiran sistem za evalvacijo omogoča relativno nezahtevno ponovitev evalvacije za poljuben nabor podatkov, pa bi ga bilo smiselno umestiti v platformo za

neprestano opazovanje lastnosti sistema, kjer bi se avtomatsko upoštevali najnovejši podatki.

# Literatura


[1] A. Bermingham and A. F. Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1833–1836. ACM, 2010.

[2] M. W. Berry and M. Castellanos. Survey of text mining. *Computing Reviews*, 45(9):548, 2004.

[3] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] P. Ekman and W. V. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Ishk, 2003.

[6] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12, 2009.

[7] M. Grčar. Latino - a light-weight library for building text mining applications in c#. [`https://github.com/LatinoLib/`; dostopano 31-03-2016].

[8] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Technical report, DTIC Document, 1996.

[9] T. Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999.

[10] M. Juršic, I. Mozetic, T. Erjavec, and N. Lavrac. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214, 2010.

[11] O. Kolchyna, T. T. Souza, P. Treleaven, and T. Aste. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv preprint arXiv:1507.00955*, 2015.

[12] I. Kononenko and M. R. Šikonja. *Inteligentni sistemi.* Založba FE in FRI, 2010.

[13] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[14] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[15] "Microsoft". ".net - powerful open source cross platform development". "[Online; dostopano 31-03-2016]".

[16] I. Mozetič, M. Grčar, and J. Smailović. Multilingual Twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036, 2016.

[17] V. Narayanan, I. Arora, and A. Bhatia. Fast and accurate sentiment classification using an enhanced naive Bayes model. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013*, pages 194–201. Springer, 2013.

[18] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič. Sentiment of emojis. *PloS one*, 10(12):e0144296, 2015.

[19] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics, 2005.

[20] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[21] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, and I. Mozetič. The effects of twitter sentiment on stock price returns. *PloS one*, 10(9):e0138441, 2015.

[22] M. Robnik-Sikonja and I. Kononenko. Explaining classifications for individual instances. *Knowledge and Data Engineering, IEEE Transactions on*, 20(5):589–600, 2008.

[23] S. Rosenthal, A. Ritter, P. Nakov, and V. Stoyanov. Semeval-2014 task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, 2014.

[24] H. Saif, M. Fernandez, Y. He, and H. Alani. Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold. In *Proceedings of the First International Workshop on Emotion and Sentiment in Social and Expressive Media: approaches and perspectives from AI (ESSEM 2013)*, 2013.

[25] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[26] J. Smailović. *Sentiment analysis in streams of microblogging posts*. PhD thesis, PhD Thesis, Jozef Stefan International Postgraduate School. Ljubljana, Slovenia, 2015.

[27] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[28] M. Tsytsarau and T. Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514, 2012.

[29] C. Van Rijsbergen. *Information Retrieval*. Butterworth, London, UK, 1979.

[30] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.