SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. xxxx

SUSTAV ZA ANALIZU I ANALIZA KOMENTARA NA NOVINSKE ČLANKE

Silvana Bakula

Zagreb, rujan 2022.

Student: Bakula, Silvana – 0036511066

Studij: Računarstvo

Profil: Računarska znanost

Zadatak: Sustav za analizu i analiza komentara na novinske članke

Zadatak (EN): System for analysis and analysis of comments on news articles

Opis zadatka:

Internet je omogućio nastanak mnogih portala koji prenose ili generiraju vijesti iz lokalnog, regionalnog, nacionalnog ili svjetskog života. Internet je također omogućilo rast društvenih mreža koje nude uslugu komentiranja vijesti koje se objave. Te podatke zanimljivo je analizirati iz raznih razloga. Analize mogu biti, primjerice, koje vijesti su najkomentiranije, tko komentira i kada, na koliko portala se istovremeno komentira. To su primjeri samo jednostavnijih analiza, a moguće su i složenije analize koje bi dale uvid u stanje tog dijela društvenog života.

U sklopu ovog diplomskog rada potrebno je složiti sustav za analizu komentara koji se ostavljaju na novinske članke korištenjem ELK stoga. Korištenjem tog sustava potom treba napraviti analizu virtualnih osoba koje komentiraju – cilj je utvrditi mogu li se korelirati lažni profili koje otvara jedna osoba te doći do podatka koliko je to prisutno. Omogućiti vizualizaciju dobivenih podataka. Sustav mora biti napravljen na takav način da omogući jednostavno dodavanje novih analiza. Citirati korištenu literaturu i navesti dobivenu pomoć.

Sadržaj

1. Uvod 1
2. Stilometrija i pripisivanje autorstva 2
2.1. Leksički koncepti i mjere 3
2.1.1. Zipfov zakon 3
2.1.2. Mjera raznovrsnosti vokabulara 4
2.1.3. Stilističke mjere 6
2.2. Ekstrakcija značajki 7
2.2.1. Analiza riječi 7
2.2.2. Ostale strategije ekstrakcije značajki 8
2.2.3. Značajke bazirane na frekvenciji 9
2.3. Modeli strojnog učenja 9
2.3.1. Naivan Bayesov klasifikator 10
2.3.2. Stroj potpornih vektora 11
2.3.3. Logistička regresija 12
3. Metodologija 14
3.1. Kreiranje seta podataka 14
3.2. Predprocesiranje podataka 15
3.2.1. Selekcija komentara i izbacivanje stršećih vrijednosti 16
3.2.2. Analiza ulaznog seta podataka 17
3.3. Ekstrakcija značajki 23
3.3.1. Lematizacija 24
3.3.2. BOW model 24
3.3.3. Zaustavne riječi 25
3.3.4. POS tagovi 25
3.3.5. Brojanje tipfelera 27

3.3.6. CountVectorizer 27
3.3.7. TF-IDF vektor 28
3.3.8. Redukcija dimenzionalnosti 29
3.3.9. Konačni izgled matrice 30
3.4. Klasifikacija 32
3.4.1. Naivni Bayesov klasifikator 32
3.4.2. Stroj potpornih vektora 32
3.4.3. Pasivno agresivni Bayesov klasifikator 33
3.4.4. Logistička regresija 33
4. Rezultati i evaluacija 34
5. Zaključak 41
6. Literatura 43
Sažetak 45
Summary 46

1. Uvod

Internet je omogućio brzo informiranje javnosti o aktualnim događanjima u svijetu putem društvenih mreža, raznih portala i foruma. Određene teme privlače visok interes javnosti i pokreću lavinu rasprava i komentara. Poboljšanje računalne snage i povećanje kapaciteta memorije za pohranu podataka omogućili su prikupljanje i spremanje velike količine komentara koje je moguće analizirati na razne načine, primjerice, koje teme izazivaju najveći interes u javnosti, tko komentira kakve članke, u koje vrijeme u danu se napiše najviše komentara.

Jedna osoba može koristiti više profila za ostavljanje komentara. Glavni cilj ovog rada je pokušati odgovoriti na pitanje je li moguće prepoznati i kvantificirati stil pisanja koji jedinstveno pripada svakom čovjeku i na taj način povezati napisani komentar s njegovim autorom. Stil pisanja je razlikuje od osobe do osobe pa je potrebno pronaći obilježja po kojima je moguće diferencirati različite autore. Analiziranjem i proučavanjem stila pisanja bavi se jezična disciplina koja se zove stilometrija. Stilometrija se prvotno razvijala na knjigama, odnosno na velikim količinama teksta. Budući da su komentari na društvenim mrežama obično jako kratki, sama stilometrija nije dovoljna pa se kombinira s različitim metodama iz domene strojnog učenja

Rad je strukturiran na sljedeći način. U drugom poglavlju je dan pregled leksičkih mjera te poznatih stilometrijskih metoda. Osim toga, opisani su algoritmi nadziranog strojnog učenja koji se često koriste kod pripisivanja autorstva. U trećem poglavlju je opisan postupak pripreme ulaznog seta podataka te implementacija pripisivanja autorstva. Implementacija se sastoji od definiranja i računanja značajki te od implementacije algoritama nadziranog strojnog učenja. U četvrtom poglavlju su dani rezultati klasifikacije. Rad završava zaključkom u petom poglavlju te popisom literature.

2. Stilometrija i pripisivanje autorstva

Stilometrija je definirana kao primjena statističke analize na nekom tekstu u svrhu otkrivanja i kvantificiranja jezičnog stila [1]. Jezični stil je način izražavanja, počevši od izbora riječi, učestalosti pojedinih riječi, punktacije, strukture rečenice, gramatičkih obrazaca i ostalih elemenata koje pojedini autor preferira kod pisanja [2].

Grana stilometrije koja se bavi identifikacijom autora dokumenta se zove pripisivanje autorstva. Pitanje autorstva dokumenta je staro koliko i sami dokumenti. Glavna ideja pripisivanja autorstva je prepoznati karakteristike pisanog teksta preko kojih ga je moguće povezati s određenim autorom. Neki od najpoznatijih primjera otkrivanja autorstva pomoću stilometrije su identifikacija James Madisona i Alexandra Hamiltona kao autora *Federalističkih spisa* koji su objavljeni anonimno i otkrivanje J. K. Rowling kao autorice knjige *Zov kukavice* [1].

Da bi se riješio bilo koji problem iz domene pripisivanja autorstva, pristupi analizi moraju biti utemeljeni na kvantitativnim lingvističkim i statističkim modelima [4]. Nad danim tekstom se provodi ekstrakcija značajki koje je potrebno kvantificirati kako bi se nad njima mogli obaviti potrebni izračuni. Drugim riječima, stil pisanja se reprezentira kao niz brojeva, najčešće u vektorskom prikazu, od kojih svaki predstavlja numeričku vrijednost neke značajke. Leon Batista Alberti (1404.-1472.) je prva osoba koja je predložila korištenje kvantitativnih mjera u stilometriji. Problem kojim se bavio je kako razlikovati prozu od poezije u latinskom jeziku. Brojanjem suglasnika je zaključio da u poeziji suglasnici čine oko 44% znakova. Razlikovanje poezije od proze po broju suglasnika je kasnije potvrdio Ycart pa se Alberti smatra pionirom kvantifikacije u stilometriji [4].

Značajke se mogu pronaći na leksičkoj, gramatičkoj te semantičkoj razini. Ovisno o ulaznom setu i klasifikatoru koji se koristi, neke značajke će biti diskriminativnije u odnosu na druge pa je za svaki model potrebno napraviti selekciju značajki.

Pripisivanje autorstva se obično temelji na uspoređivanju vektora. Ulazni tekst se nakon postupka vektorizacije, odnosno kvantifikacije jezičnog obrasca, uspoređuje sa setom poznatih vektora koji predstavljaju pojedinačne jezične stilove. Proces usporedbe i kategorizacije nekog seta podataka se naziva klasifikacija [4]. U strojnom učenju, klasifikacija je vrsta nadziranog učenja u kojem trenirani model, tj. klasifikator, ulaznim primjerima pridjeljuje oznaku koja predstavlja klasu ili grupu. Kod pripisivanja autorstva,

ulazni primjeri su komentari, a oznake, tj. izlazi klasifikatora, predstavljaju autore komentara.

Cilj ovog poglavlja je predstaviti različite osnovne koncepte pripisivanja autorstva. U prvom potpoglavlju su opisane leksičke mjere, tj. mjere temeljene na analizi riječi koje imaju neovisno značenje. Opisan je Zipfov zakon i njegova implikacija na analizu najčešćih riječi te mjere temeljene na raznolikosti vokabulara i stilističkim obilježjima. U drugom potpoglavlju su opisane stilometrijske metode koje se često koriste kao značajke za algoritme strojnog učenja. U trećem potpoglavlju je dan pregled algoritama nadziranog strojnog učenja koji se koriste kod pripisivanja autorstva.

2.1. Leksički koncepti i mjere

Jezični stil nekog teksta je potrebno prikazati na jasan i učinkovit način. Ova paradigma zahtjeva reprezentaciju jezičnog stila kao broj ili niz brojeva koji predstavljaju kvantificiranu vrijednost neke značajke [4]. Potrebno je pronaći značajke koje nedvosmisleno opisuju neki jezični stil. Ako je set komentara podijeljen na grupe od kojih svaka predstavlja nekog autora, takve grupe je potrebno moći lako regrupirati na šire grupe, primjerice, dobne, spolne.

Reprezentacija jezičnog stila mora biti takva da joj se lako može izračunati udaljenost, tj. razlika u odnosu na druge jezične stilove. Iz tog razloga se najčešće koristi vektorski prikaz. Tekst se dijeli na jedinice nad kojima se izvlače i računaju značajke koje mogu biti na leksičkoj, sintaktičkoj, semantičkoj i gramatičkoj razini. U nastavku su opisane leksičke mjere. To su mjere koje se temelje na analizi leksema, odnosno riječi od kojih je ulazni tekst sačinjen.

2.1.1. Zipfov zakon

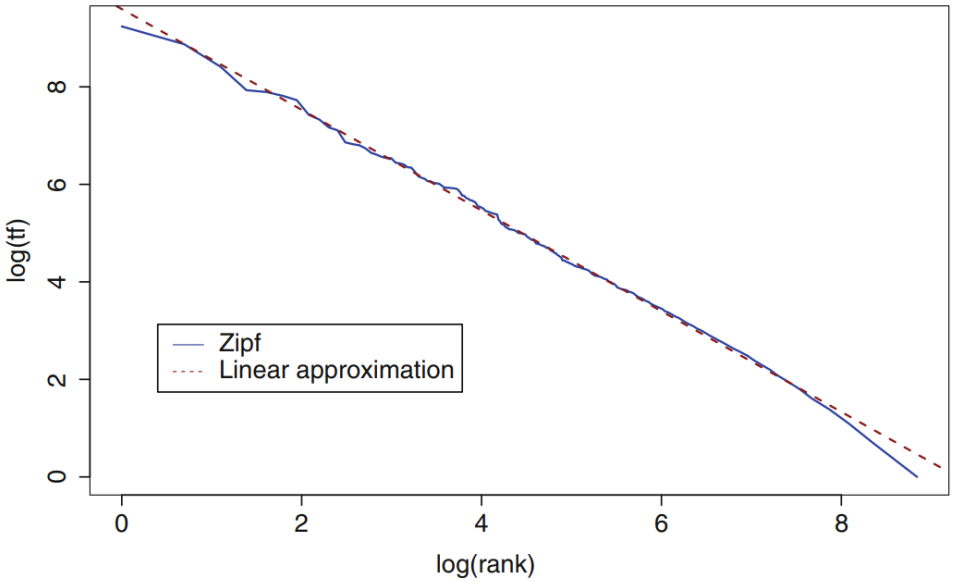
Zipfov zakon opisuje linearnu zavisnost logaritma ranga riječi i logaritma frekvencije pojedine riječi. Drugim riječima, ako se nad nekim tekstom pronađu najučestalije riječi i rangiraju od najčešćih do najrjeđih, tada su logaritam ranga i logaritam frekvencije tih riječi linearno zavisni (2.1).

$$\log(t f_{r,i}) = c - a \times \log(r_i) \tag{2.1}$$

r_i je riječ na i-tom rangu, a $f_{r,i}$ je frekvencija riječi r na i-tom rangu. c je konstanta koja ovisi o setu podataka, a prema Zipfu uvijek ima vrijednost približnu 1. a je parametar koji određuje nagib relacije (2.2). Zipfova relacija je dobivena nakon logaritmiranja i sređivanja izraza (2.2) [15].

$$tf_{r,i} = \frac{c}{r_i^a} \tag{2.2}$$

Izraz 2.2 kaže da je frekvencija riječi na i-tom rangu inverzno proporcionalna rangu potenciranom brojem a [15]. U nastavku je dan primjer Zipfovog zakona na djelu *Federalistički spisi*. Izdvojeno je 10 najučestalijih riječi, za konstantu c je uzeta vrijednost 9.6, a za konstantu a -1.03. Na slici 2.1 crvena isprekidana linija predstavlja linearnu aproksimaciju logaritma ranga riječi i logaritma frekvencije riječi, a plava linija Zipfovu relaciju na setu riječi iz *Federalističkih spisa.* Na slici 2.1 je vidljivo da se linearna aproksimacija i Zipfova relacija gotovo preklapaju [4].



Sl. 2.1 Apsolutna frekvencija kao funkcija ranga riječi

2.1.2. Mjera raznovrsnosti vokabulara

Različite mjere raznovrsnosti vokabulara se primjenjuju kako bi se izračunala amplituda vokabulara pojedinog autora. Mjere raznovrsnosti vokabulara služe za razlikovanje autora po elokvenciji, tj. bogatstvu vokabulara. Jedna od najpoznatijih mjera iz navedene skupine

je TTR (engl. *Type-Token Ratio*) mjera koja predstavlja omjer broja različitih riječi i ukupnog broja riječi (2.3).

$$TTR(T) = \frac{|Voc(T)|}{n} \tag{2.3}$$

Visoka vrijednost TTR mjere ukazuje na veliku raznovrsnost riječi, odnosno bogat vokabular. Suprotno, mala vrijednost TTR mjere ukazuje na šturi vokabular, odnosno sklonost autora ponavljanju sličnih izraza. Pripisivanje autorstva pomoću TTR mjere se zasniva na računanju TTR mjere za svakog autora te TTR mjere neviđenih tekstova koji se onda pridjeljuju autorima s kojima se mjera poklapa [4].

TTR mjera se nije pokazala kao pretjerano koristan alat kod pripisivanja autorstva u praksi. Najveći problem je osjetljivost mjere jer ovisi o duljini teksta pa su vrijednosti često jako male. Kako raste količina teksta, tako mora rasti i potreba za ponavljanjem određenih riječi, što dovodi do smanjivanja TTR mjere. Zbog toga se TTR najčešće računa tako da se dogovori fiksna duljina odlomka i onda se unutar odlomaka iste duljine pobroje različite riječi. Drugi pristup poboljšanja TTR mjere je razlamanje teksta na manje cjeline fiksne duljine te računanje prosječne TTR mjere za sve cjeline [5].

Osim TTR mjere, za mjerenje raznovrsnosti vokabulara se koriste i *Guiraudov R* i *Herdanov C*. Guiraudova mjera je omjer između kardinaliteta vokabulara datog teksta i korijena ukupnog broja riječi (2.4). Herdanova mjera je omjer prirodnog logaritma kardinaliteta vokabulara te prirodnog algoritma ukupnog broja riječi (2.5) [16].

$$Guiraudov\ R(T) = \frac{|Voc(T)|}{\sqrt{n}} \tag{2.4}$$

$$Herdanov\ C(T) = \frac{\ln|Voc(T)|}{\ln n} \tag{2.5}$$

Nijedna od navedenih mjera ne daje zadovoljavajuće rezultate u praksi, a problem leži u distribuciji riječi. Distribucija riječi se ne ponaša u skladu s poznatim distribucijama, kao što su, primjerice, Gaussova ili Poissonova. Što veći komad teksta razmatramo, to imamo više rijetkih riječi, odnosno riječi koje se pojavljuju jednom ili dva puta. Rijetke riječi najčešće

predstavljaju 50% vokabulara razmatranog teksta pa je gotovo nemoguće odrediti kompletan skup riječi koji predstavlja vokabular jednog autora.

2.1.3. Stilističke mjere

Jedna od najpoznatijih stilističkih mjera je leksička gustoća (engl. *lexical density,* LD). Leksička gustoća se iskazuje u postotcima, a računa se kao omjer količine leksički značajnih riječi i ukupnog broja riječi.

$$LD(T) = \frac{leksičke\ riječi\ (T)}{n} \tag{2.6}$$

Leksički značajne riječi su riječi koje nose neko značenje. Nazivaju se još i *funkcijskim riječima*. Suprotne leksičkim riječima su *zaustavne riječi* (engl. *stopwords*). *Zaustavne riječi* same po sebi nose malo ili gotovo nikakvo leksičko značenje. *Funkcijske riječi* su „ljepilo" koje veže leksički značajne riječi, odnosno imenice, glagole i pridjeve. Ekstrakcija leksički značajnih riječi se najčešće provodi tako da se tokenizira tekst nakon čega se izbace sve zaustavne riječi. Problem kod takvog pristupa je taj što se često razdvoje izrazi koji bi se trebali promatrati kao cjelina. Primjerice, u engleskom jeziku izraz *give up* bi se rastavio na dva tokena, *give* i *up*, nakon čega bi *up* bio izbačen zbog toga što se bez konteksta promatra kao zaustavna riječ.

Leksička gustoća mjera pisanih tekstova najčešće iznosi između 0.4 i 0.5, a u govornom jeziku oko 0.3. Često je proporcionalna s godinama autora, odnosno vrijednost je veća što je autor stariji [8].

Druga često korištena stilistička mjera je zastupljenost dugih riječi (engl. *big words,* BW). Zastupljenost dugih riječi se računa kao omjer broja dugih riječi i ukupnog broja riječi. Dugom riječju se smatra riječ koja ima šest ili više slova. Tekstovi koji imaju visoku zastupljenost dugih riječi su obično teži za razumjeti, jezični stil je sofisticiran i obično su vezani uz znanost [4].

Slična BW mjeri je mjera prosječne duljine riječi u tekstu. Ovu mjeru je prvi predložio Augustus de Morgan (1806.-1871.), ali u to vrijeme primjena nije bila moguća zbog nedostatka računalnih resursa. Mjeru prosječne duljine riječi je koristio Mendenhall kod pripisivanja autorstva Shakespeareovih dijela. Ustvrdio je da bi Christopher Marlowe mogao stajati iza Shakespeareovih dijela jer je prosječna duljina riječi kod oba autora četiri [9].

Srednja duljina rečenice (engl. *mean sentence length,* MSL) predstavlja prosječnu duljinu rečenice u tekstu. Ulazni tekst se razdvaja na mjestima s interpunkcijskim znakovima koji označavaju kraj rečenice i na kraju se računa prosjek duljine svih rečenica. Što je vrijednost ove mjere veća, to je tekst kompliciraniji i teži za razumjeti [4].

2.2. Ekstrakcija značajki

Ekstrakcija značajki je jedan od najvažnijih koraka u procesu pripisivanja autorstva. Moguće je izvući bezbroj značajki, ali to nije garancija za kvalitetan klasifikator. Kombinacija više različitih značajki poboljšava efikasnost modela, ali prevelik broj značajki često bespotrebno povećava složenost, a ne utječe bitno na kvalitetu modela. Značajke mogu biti redundantne ili netočne ako je u podatcima prisutan šum. Smanjenjem broja značajki dobivamo jednostavniji klasifikator koji obično bolje generalizira te brži izračun. Ideja je zadržati samo najdiskriminativnije značajke. U nastavku je dan pregled najčešće korištenih leksičkih i sintaktičkih te znakovnih značajki.

2.2.1. Analiza riječi

Ponekad se riječi izvornog teksta analiziraju u izvornom obliku, ali češća i bolja praksa je riječi prvo lematizirati ili korjenovati.

Lematizacija je proces svođenja promjenjivih vrsta riječi na lemu. Lema je dakle leksikografska natuknica, odnosno riječ svedena na osnovni oblik. Na primjer, riječi: *primljeno*, *primam, primaju* imaju zajedničku lemu, a to je glagol *primati*. Leme omogućuju takvu strukturu i organizaciju teksta koja omogućuje sustavno koncentriranje informacija [6].

Korjenovanje je slično postupku lematizacije, ali je zapravo širi postupak jer se veći broj riječi svede na isti oblik nego postupkom lematizacije. Korjenovanje je proces svođenja riječi na korijen riječi, odnosno proces uklanjanja afiksa. Afiksi su jezične jedinice koje se dodaju na osnovni oblik riječi kako bi se promijenilo njezino značenje. Afiksi sami po sebi nisu samostalne riječi, a mogu biti prefiksi ili sufiksi, ovisno o tome dodaju li se na početku ili na kraju riječi [7]. Na primjer, u riječi *djedovi* (pridjev), korijen riječi bi bio *djed*, sufiks je *ov*, a *i* je nastavak zbog deklinacije. Lema riječi *djedovi* bi bila *djedov*.

Analiza izoliranih riječi, bilo da su u izvornom obliku, lematizirane ili korjenovane, najčešći je pristup stilometrijske analize. Osim izbora riječi, često se promatra i kombinacija

korištenih riječi. Metoda koja kombinira susjedne riječi i grupira ih u tokene se zove n-gram metoda. Za *n* se najčešće uzima vrijednost 2 ili 3 [4]. Primjerice, 2-gram tokeni rečenice *Danas je lijep dan,* bi bili *danas je, je lijep* i *lijep dan.* Prije generiranja n-gram tokena se obično prvo izbaci punktacija. Kod neformalnog pisanja je česta praksa izostavljanje dijakritičkih znakova pa nije uvijek jasno je li potrebno raditi konverziju nad svim dijakritičkim znakovima (*č* i *ć* u *c* i sl.) Kad se makne kontekst, ponekad je teško odrediti značenje riječi, npr. *suma* može biti sinonim za zbroj, ali može nastati i od riječi *šuma* nakon konverzije dijakritika.

Analizom n-gram tokena je moguće otkriti izraze koje pojedinci često ponavljaju, npr. *pametnom dosta, na aparatima, spavaš li mirno.* n-gram tokeni čuvaju kontekst riječi.

n-gram tokeni se mogu generirati i po slovima. Primjerice, 2-gram tokeni rečenice *Danas je lijep dan* bi bili *da,an,na,as,sj,je,el,li,ij,je,ep,pd,da,dn.* Upitna je korisnost takvih tokena jer ih se generira jako puno, a nose malo informacija. Bolja varijanta je grupiranje slogova, npr. *da nas, nas je, je li, li jep, jep dan.* Grupiranje po slogovima u nekim jezicima može biti teško za implementirati.

Česta metoda kod analize riječi je POS (engl. *Parts of Speech*) tagiranje. POS tagiranje je postupak mapiranja riječi s njezinom vrstom, primjerice, imenica, glagol, pridjev, znamenka. POS oznake se najčešće prebrojavaju, odnosno računa se frekvencija svake oznake u tekstu [3].

Još jedna poznata reprezentacija ulaznog teksta je BOW (engl. *bag of words*) reprezentacija. BOW reprezentacija je mapa kojoj su ključevi riječi, odnosno najčešće leme ili korijeni riječi, a vrijednosti broj pojavljivanja u tekstu [10]. Primjerice, BOW reprezentacija teksta: „*John likes to watch movies. Mary likes movies too.*“ bi bila {"John":1," likes":2, "to":1, "watch":1, "movies":2, "Mary":1, "too":1}.

2.2.2. Ostale strategije ekstrakcije značajki

Analiza riječi teksta omogućuje ekstrakciju jako velikog broja značajki, no to nije jedini pristup. Mnoge studije zagovaraju da je bolji granularniji pristup, odnosno analiza znakova. Kod analize znakova je uobičajena praksa sve znakove svesti na mala slova te ukloniti interpunkcijske znakove. Jako česta metoda koja se temelji na analizi znakova je računanje frekvencije znakova: samoglasnika, suglasnika, određenih slova i sl. Često se koristi i brojanje interpunkcijskih znakova u izvornom tekstu [10].

Najčešće korištena metoda kod analize znakova su već spomenuti n-grami, gdje se kao i kod analize riječi za *n* uzima broj 2 ili 3. Često se koristi i analiza slogova, a ta metoda je posebno popularna kod klasične stilometrije na književnim djelima. Primjerice, Shakespeare je u svojim djelima često koristio jampski pentametar, a to je vrsta metričke linije koja se sastoji od 5 jambova. Jamb označava nenaglašeni slog iza kojeg slijedi naglašeni slog [4].

2.2.3. Značajke bazirane na frekvenciji

Značajke bazirane na frekvenciji govore koliko se često pojavljuje neka jedinica teksta. Za jedinicu teksta se obično uzimaju leme, korijeni riječi, n-gram tokeni ili POS oznake. Izbor jedinice koja se analizira ovisi o definiciji problema, npr. ako analiziramo pozitivne emocije, brojat ćemo POS oznake ADJ, tj. pridjeve dobar, lijep i slično.

Za broj jedinica koje se prebrojavaju se najčešće uzima broj iz raspona 50-300. Rezultat računanja frekvencije pojedine jedinice u tekstu je matrica kojoj su redci ulazni tekstovi, a stupci promatrane jedinice (n-gram tokeni, leme i sl.) Takva matrica se često prikazuje u TF-IDF (engl. *term-frequency-inverse-document-frequency*) reprezentaciji. Ideja kod TF-IDF reprezentacije je smanjiti utjecaj tokena koji se pojavljuju jako često, zbog čega se smatra da su empirijski manje informativni u odnosu na one koji se pojavljuju rijetko. TF-IDF mjera je umnožak TF i IDF mjere. TF mjera je frekvencija pojavljivanja jedinice u dokumentu, a IDF mjera je definirana kao logaritam omjera ukupnog broja dokumenata *n* i broja pojavljivanja termina *t* u svim dokumentima.

$$IDF(t,d) = \log\left(\frac{n}{df(t)}\right) \tag{2.7}$$

2.3. Modeli strojnog učenja

Svaka kvantitativna i stilometrijska studija se sastoji od šest glavnih koraka. Prvi korak je definicija problema ili definiranje hipoteze koja se pokušava dokazati ili opovrgnuti. Drugi korak je prikupljanje podataka. Treće, potrebno je obaviti predprocesiranje podataka kako bi se homogenizirali i pročistili od šumova i stršećih vrijednosti. Četvrti korak je definiranje i računanje značajki. Peto, potrebno je obaviti određene radnje nad značajkama, kao što su selekcija, smanjenje dimenzionalnosti, standardizacija vrijednosti i sl. Konačno, potrebno je izabrati model strojnog učenja koji će na osnovu značajki znati klasificirati ulazne podatke. Nakon računanja reprezentacije svake kategorije, model mora znati izračunati udaljenosti između neviđenog podatka i svake od mogućih kategorija i na taj način ga ispravno

klasificirati, ako odgovarajuća klasa postoji. Izlaz klasifikatora ne mora biti uniforman, odnosno poželjno je vratiti listu kandidata s izračunatim vjerojatnostima pripadanja ulaznog primjera, ako takvi kandidati postoje.

2.3.1. Naivan Bayesov klasifikator

Naivan Bayesov klasifikator je klasifikacijski algoritam strojnog učenja koji se temelji na Bayesovom teoremu uz pretpostavku o uvjetnoj nezavisnosti varijabli. Kategorije u koje se ulazni podatak može klasificirati se nazivaju hipotezama i označavaju se s $y_j, j = 1,2,3$. Bayesov klasifikator računa vjerojatnost pripadanja klasi *y* za zadani ulazni primjer, odnosno vektor značajki x_1 do x_n. (2.8). Vjerojatnost se označava velikim slovom *P*.

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1,\dots,x_n|y)}{P(x_1,\dots,x_n)} \tag{2.8}$$

Naivan Bayesov klasifikator se naziva naivnim zbog pretpostavke o uvjetnoj nezavisnosti varijabli. Varijable X i Y smatramo uvjetno nezavisnim ako poznavanje ishoda varijable Y ne utječe na vjerojatnost ishoda varijable X i obrnuto. Zbog uvjetne nezavisnosti vjerojatnosti se množe pa izraz (2.8) prelazi u izraz (2.9).

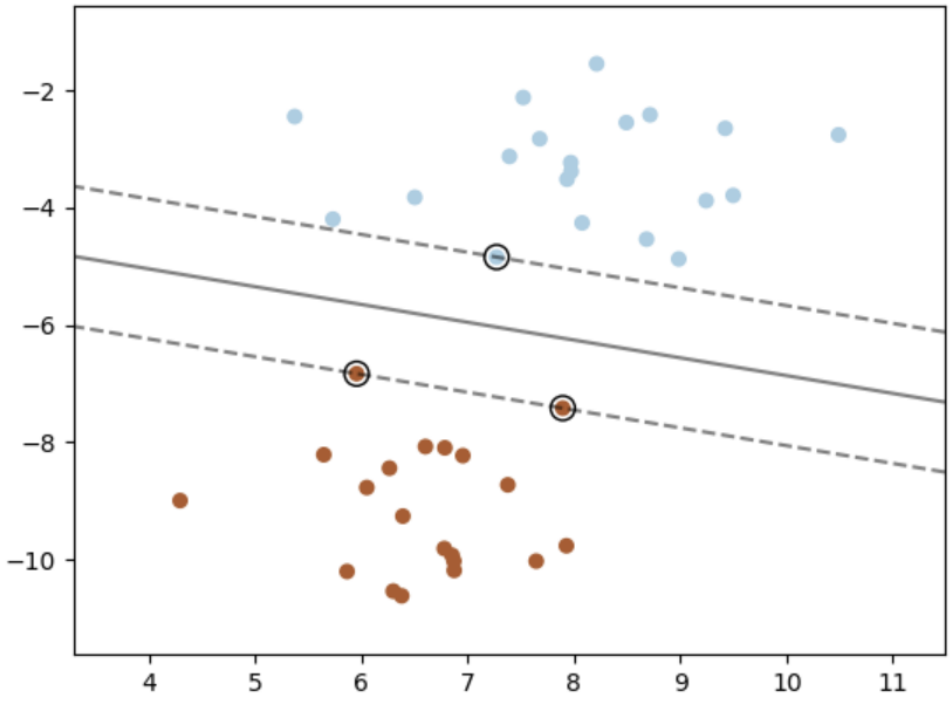
$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\dots,x_n)} \tag{2.9}$$

Bayesov klasifikator je probabilistički model, a učenje probabilističkog modela je ekvivalentno procjenjivanju parametara. Konkretno, potrebno je procijeniti parametre svih distribucija koje se koriste u modelu, odnosno parametre apriorne distribucije i parametre za vjerojatnost klasa. Kod klasifikacije nas ne zanima egzaktna vjerojatnost pripadanja nekoj klasi, nego za koju klasu je najveća vjerojatnost pripadanja. To je maksimum aposteriori hipoteza, odnosno uzima se hipoteza koja maksimizira izraz (2.10) [12].

$$\hat{y} = \arg max_y P(y)\Pi_{i=1}^{n} P(x_i|y) \tag{2.10}$$

2.3.2. Stroj potpornih vektora

Stroj potpornih vektora (SVM, engl. *support vector machine*) je metoda nadziranog strojnog učenja koja se koristi za regresiju, klasifikaciju ili identifikaciju stršećih vrijednosti. SVM se temelji na ideji maksimalne margine. Maksimalna margina je ona margina koja ima najveću udaljenost od primjera iz dviju klasa. Ideja je pronaći onu hiperravninu koja maksimizira marginu, pri čemu je margina udaljenost između te hiperravnine i najbližeg primjera sa svake strane.



Sl. 2.2 Stroj potpornih vektora

Na slici 2.2 se nalazi primjer klasifikacije korištenjem stroja potpornih vektora. Puna linija predstavlja granicu između dviju klasa. Ta granica je takva da maksimizira svoju udaljenost od najbližih primjera, odnosno zapisa s obje strane. Maksimizacija margine je optimizacijski problem. Izraz (2.11) je formulacija optimizacijskog problema maksimalne margine, uz određena ograničenja (2.12). Maksimizacija udaljenosti najbližih primjera od normale w, odnosno hiperravnine je jednaka minimizaciji L2 norme, odnosno udaljenosti vektora w od ishodišta. w_0 je početna težina.

$$argmin_{\boldsymbol{w},w_0} \frac{1}{2} \|\boldsymbol{w}\|^2 \tag{2.11}$$

$$y^{(i)}\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + w_0\right) \geq 1, \quad i = 1, \ldots, N \tag{2.12}$$

Izraz (2.12) znači da je za primjere najbliže ravnini izraz s lijeve strane jednak 1, a za sve udaljenije primjere veći od 1.

Problem kod ovako definiranog klasifikatora je taj što ulazni primjeri nisu uvijek linearno odvojivi, odnosno nije moguće pronaći hiperravninu koja dijeli ulazne primjere. Formulacija koja se uvodi je meka margina. Meka margina znači da se primjeri mogu naći unutar granica margine. U ograničenje se uvodi rezervna varijabla, odnosno varijabla koja govori koliko je neki primjer ušao u marginu (2.13) [13].

$$y^{(i)}\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + w_0\right) \geq 1 - \mathcal{E}_i, \quad i = 1, \ldots, N \tag{2.13}$$

Formulacija modela s mekom marginom je dana u izrazu (2.14), pri čemu je *C* hiperparametar koji određuje tvrdoću margine. Veći C znači tvrđa margina, i obrnuto [13].

$$argmin_{w,w_0,\varepsilon} \left\{\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \mathcal{E}_i\right\} \tag{2.14}$$

2.3.3. Logistička regresija

Logistička regresija je diskriminativan probabilistički model strojnog učenja. Unatoč nazivu, logistička regresija je zapravo klasifikacijski algoritam. Diskriminativan je jer definira samo granicu između klasa, a probabilistički jer je izlaz modela vjerojatnost da je primjer označen pozitivno. Za aktivacijsku funkciju se koristi sigmoidna funkcija (2.15).

$$\sigma(\alpha) = \frac{1}{1+\exp(-\alpha)} \tag{2.15}$$

$$h(x; w) = \sigma\left(w^T \phi(x)\right) = \frac{1}{1+\exp(-w^T \phi(x))} = P(y = 1|x) \tag{2.16}$$

Izraz (2.16) je definicija logističke regresije za slučaj kada imamo dvije klase. *h(x;w)* je funkcija preslikavanja ulaznog primjera *x*, odnosno vektora značajki *w* u odgovarajuću klasu. Tako definirana funkcija se naziva hipoteza. w^T je transponirana matrica značajki, a $\phi(x)$ je funkcija preslikavanja ulaznog primjera *x* u prostor značajki.

Poopćenje logističke regresije na slučaj s više klasa se naziva multionomijalna logistička regresija ili klasifikator maksimalne entropije. Kod multinomijalne logističke regresije, za svaku od K klasa se koristi zaseban vektor težina w_k, ali se onda skalarni umnožak transponirane matrice vektora težina w_k^T i ulaznog primjera $\phi(x)$ propušta kroz aktivacijsku funkciju koja omogućava da suma svih vjerojatnosti daje jedan. Aktivacijska funkcija koja se koristi se naziva softmax (2.17). Model multinomijalne logističke regresije za klasu *k* je vjerojatnost da ulazni podatak pripada klasi *k.* Definicija modela je dana u izrazu (2.18), gdje je **W** matrica od K vektora težina [14].

$$softmax_k(x_1, \dots, x_n) = \frac{\exp(x_k)}{\sum_j exp(x_j)} \tag{2.17}$$

$$h_k(x; \boldsymbol{W}) = \frac{\exp(w_k^T \phi(x))}{\sum_j \exp(w_j^T \phi(x))} = P(y = k|x, \boldsymbol{W}) \tag{2.18}$$

3. Metodologija

Primjena stilometrije na kratkim komentarima se bitno razlikuje od klasične stilometrije. Komentari su, za razliku od književnih djela ili dokumenata, najčešće nestrukturirani, neformalni i često gramatički neispravni. Osim toga, broj potencijalnih autora je puno veći pa je samim time vjerojatnost ispravnog klasificiranja komentara puno manja. U ovom poglavlju je dan pregled svih implementiranih koraka u procesu pripisivanja autora. To su sakupljanje podataka, obrada podataka, ekstrakcija i računanje značajki te implementacija algoritama strojnog učenja.

3.1. Kreiranje seta podataka

Set podataka se sastoji od sakupljenih HTML kodova stranica web portala *24sata*, *Večernjeg lista* te društvene mreže *Facebook*. Podatci su pohranjeni u MongoDB [17] bazu na udaljenom serveru. Na MongoDB bazu i udaljeni server se spaja uspostavom SSH konekcije pomoću MongoDB Compass [17] alata.

Iz HTML koda je potrebno isparsirati komentare koji čine ulazni set podataka. Nad dohvaćenim podatcima iz baze se poziva metoda *parse_comments* koja ovisno o tipu komentara poziva odgovarajuću metodu za parsiranje, a parsirane komentare vraća u obliku liste.

```
def parse_comments(json):
     if json['comment_type'] == 'facebook':
          all_comments.extend(parse_facebook(json))

     elif json['comment_type'] == 'vecernji':
          all_comments.extend(parse_vecernji(json))

      elif json['comment_type'] == '24sata':
           all_comments.extend(parse_24sata(json))

    return all_comments
```

Kod 3.1 Metoda za parsiranje komentara

Komentar se sastoji od jedinstvenog identifikatora, oznake autora, teksta komentara, vremena objavljivanja, izvora (npr. *24sata*) te naslova članka. Metode za parsiranje su napravljene pomoću *BeautifulSoup* [18] biblioteke. U nastavku je dana metoda za parsiranje komentara s portala *Večernji list*.

```
def parse_vecernji(json):
    result = []
    comment_type = 'vecernji'
    comments = json['comments']
    for com in comments:
        page = com['page']
        parsed_html = BeautifulSoup(page)
        article = process_string(parsed_html.find('title').text)
        commentList = parsed_html.body.findAll('div',
attrs={'class':'js_oneComment'})


        for comment in commentList:
            username = process_string(comment.find('a',
attrs={'class':'commbox__user'}).text.strip()[:-1])
            text = process_string(comment.find('span',
attrs={'class':'js_onecommentVisible'}).text)
            authorId = comment.find('a',
attrs={'class':'commbox__user'}).get('href').split("/")[3]
            date = comment.find('span', attrs={'class':'commbox__time--
date'}).text
            time = comment.find('span', attrs={'class':'commbox__time--
time'}).text
            timestamp = date + time
            result.append(Comment(authorId, username, text, timestamp,
comment_type, article))


    return result
```

Kod 3.2 Metoda za parsiranje komentara s portala *Večernji list*

3.2. Predprocesiranje podataka

Parsirani komentari portala *24sata* i *Večernjeg lista* te komentari s *Facebooka* čine ulazni set podataka. Ulazne podatke je prvo potrebno procesirati i homogenizirati. Procesiranje podataka se sastoji od uklanjanja znakova za prelazak u novi red, zamjene uzastopnih

razmaka jednim razmakom, uklanjanja razmaka prije interpunkcijskih znakova, izbacivanja znakova koji nisu alfanumerički, osim interpunkcijskih znakova točke, zareza, upitnika, uskličnika i crtica, konverzije u mala slova te konverzije dijakritičkih znakova (č i ć u c, š u s, ž u z).

```
def process_string(text):
      text = re.sub("\s\s+", " ", text)
      text = re.sub(r'[^A-Za-z0-9\s\.\-\!\?\,]', '', text)
      text = text.replace('\n', ' ')
      text = text.replace('\r', '')
      text = text.replace('\t', '')
      text = text.replace(' .', '.')
      text = text.replace(' !', '!')
      text = text.replace(' ?', '?')
      text = text.replace(' ,', ',')
      text = unicodedata.normalize('NFD', text).encode('ascii',
      'ignore').decode("utf-8")
      text = text.lower()
      return str(text)
```

Kod 3.3 Metoda za predprocesiranje komentara

3.2.1. Selekcija komentara i izbacivanje stršećih vrijednosti

Iz normaliziranog seta podataka se dalje uklanjaju podatci sa stršećim vrijednostima, kao i podatci koji ne ulaze u set za treniranje. Stršećim podatcima se smatraju oni komentari s premalim ili prevelikim brojem znakova i riječi. Komentari koji ulaze u set za treniranje su komentari onih autora koji su ostavili dovoljno velik broj komentara. Što je donja granica veća, to imamo kvalitetniji ulazni set komentara. U ovom radu je uzeta vrijednost 500, dakle komentari onih autora koji su ostavili manje od 500 komentara su odbačeni jer se ne mogu iskoristiti za učenje klasifikatora.

```
author_list =[data.authorId for data in dataset]
A = Counter(author_list)
train_set_author = {x: count for x, count in A.items() if count >= min_threshold}
train_set = [comment for comment in dataset if comment.authorId in train_set_author.keys()]
```

Kod 3.4 Filtriranje ulaznog seta podataka

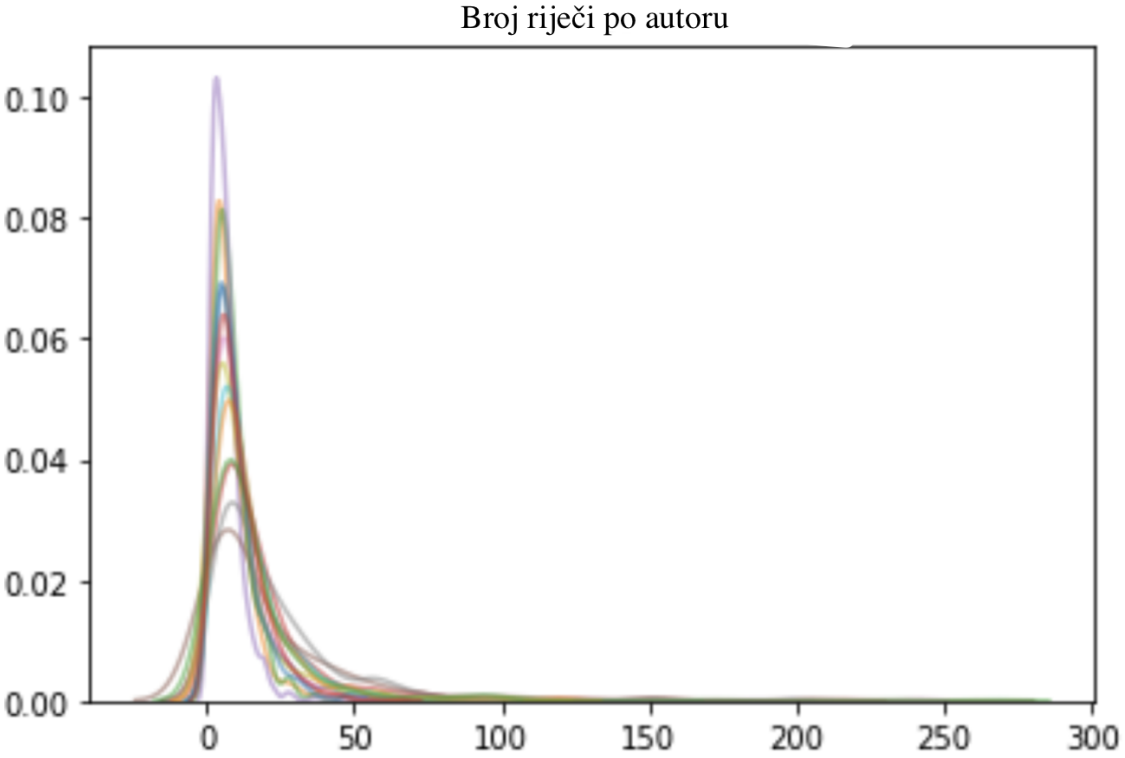
Kako bi prepoznali stršeće vrijednosti, potrebno je prvo primijeniti kalkulacije temeljene na duljini. Dobivene vrijednosti će se također iskoristiti kao značajke za klasifikator za one komentare koji ulaze u ulazni set podataka. Iskorištene su jedne od najpoznatijih stilometrijskih metoda temeljenih na duljini, a to su duljina rečenice (ukupan broj znakova), broj riječi u rečenici te uprosječena duljina riječi.

3.2.2. Analiza ulaznog seta podataka

Analiza ulaznog seta podataka je napravljena uz pomoć metoda temeljenih na duljini. Značajke temeljene na duljini su standardne i najpoznatije metode koje se primjenjuju u stilometriji. Korištene mjere su broj riječi u komentaru, ukupan broj znakova te prosječna duljina riječi. Set podataka je to kvalitetniji što su komentari duži i sadržajniji, ali ako je samo nekolicina takvih, što je realan slučaj, promatramo ih kao vrijednosti koje odskaču. Dakle, prekratke i preduge komentare je potrebno odbaciti. Za grafičku analizu komentara su iskorišteni linijski grafikoni za prikaz funkcije gustoće te box-plot grafikon za prikaz karakteristične petorke podataka. Karakteristična petorka je skupni naziv za medijan, minimalnu i maksimalnu vrijednost te gornji i donji kvartil. Donji kvartil je broj od kojeg 25% podataka ima manju ili jednaku vrijednost, a gornji kvartil broj od kojeg 75% podataka ima manju ili jednaku vrijednost. Kružići na grafikonu označuju stršeće vrijednosti, tj. podatke koje je potrebno odbaciti. Funkcija gustoće opisuje relativnu vjerojatnosti da varijabla poprimi određenu vrijednost.

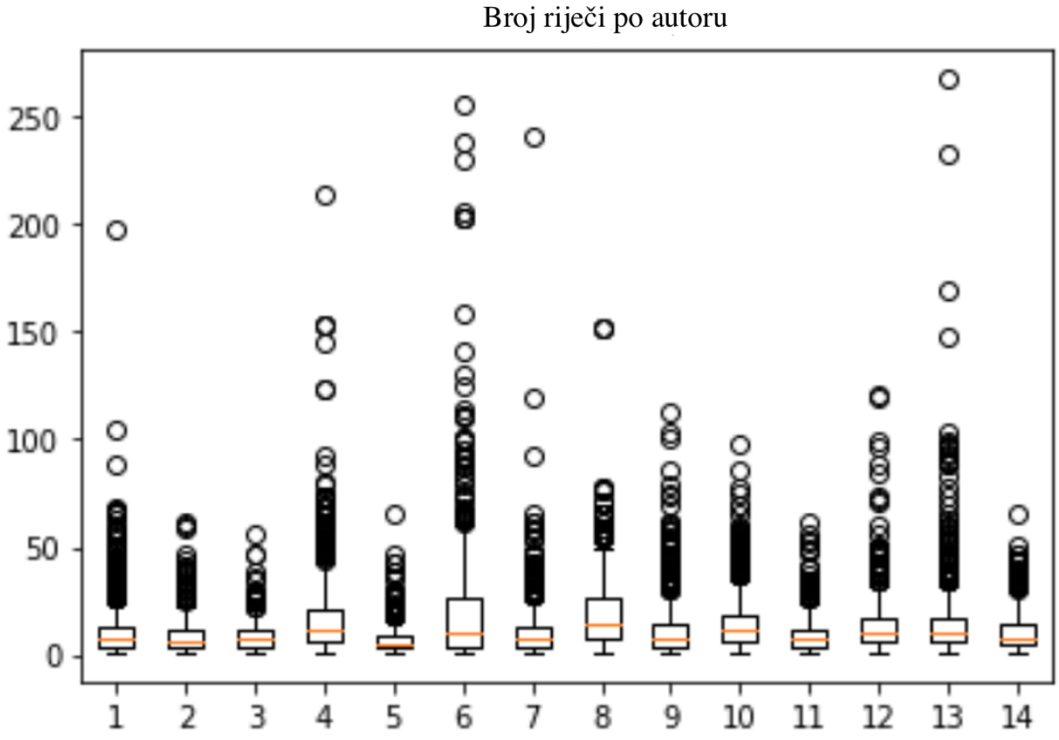
Metoda *count_words* vraća ukupan broj riječi u komentaru. Na grafikonu 3.1 je prikazana funkcija gustoće za broj riječi po autoru. Na grafikonu je vidljivo da ukupan broj riječi po autoru uglavnom poprima vrijednosti iz intervala <0,25>. Vrh krivlje se nalazi iznad vrijednosti na osi x koju varijabla najčešće poprima. Vidljivo je da svi autori imaju na sličnoj

poziciji vrh krivulje, ali ono po čemu se krivulje razlikuju je širina. Viša i uža krivulja znači da komentari nekog autora uvijek imaju isti ili sličan broj riječi. Šira i niža krivulja označava da broj riječi u komentaru varira. Primjerice, funkcija gustoće s najvišim vrhom, označena ružičastom bojom na slici, predstavlja autora čiji se komentari uglavnom sastoje od 6 riječi pa je vjerojatnost da komentar tog autora ima 6 riječi jako velika. Funkcija označena ljubičastom bojom i čiji vrh ima najnižu vrijednost na slici predstavlja autora kojem se također komentar najčešće sastoji od 6 riječi, ali često ostavlja i kraće i duže komentare pa je za njega vjerojatnost da mu se komentar sastoji od 6 riječi puno manja.



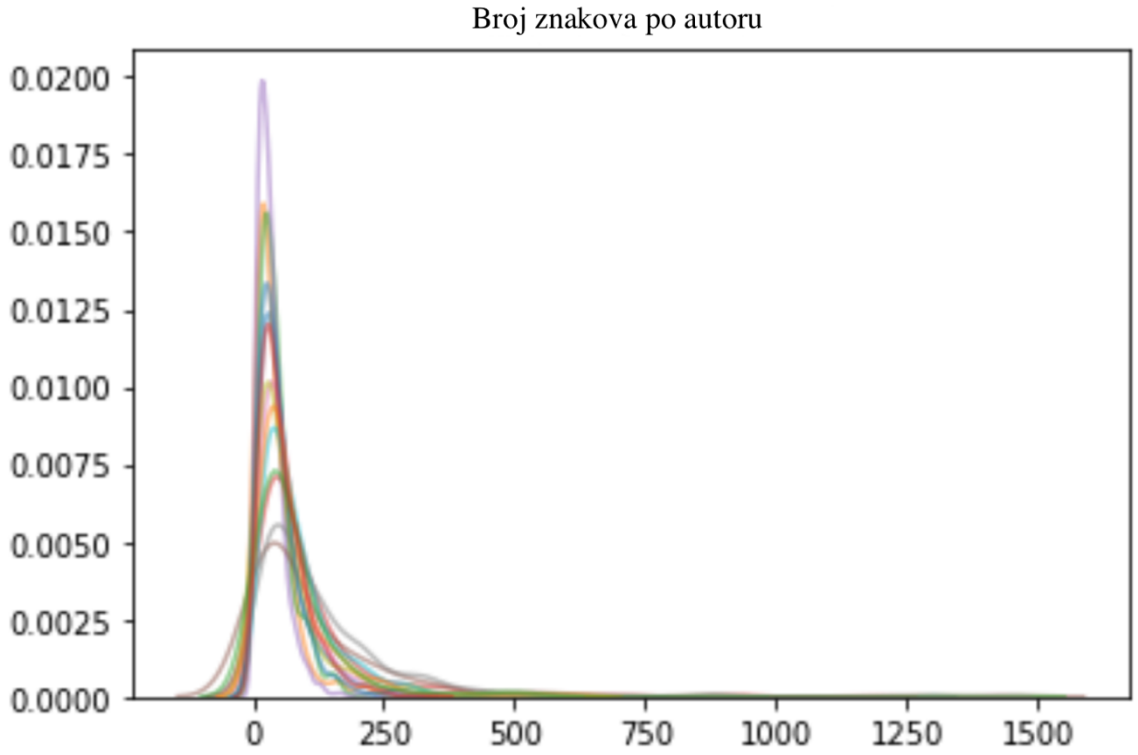
Sl. 3.1 Graf gustoće za funkciju broja riječi u komentaru

Na slici 3.2 je prikazan box plot grafikon za broj riječi u komentaru po autoru. Na grafikonu je vidljivo da je vrijednost medijana slična za sve autore i da većina autora uglavnom ostavlja kratke komentare zbog čega su komentari s većim brojem riječi prikazani kao stršeće vrijednosti. Najveću vrijednost gornjeg kvartila imaju autori 6 i 8, a iznosi 19. Gornji kvartil je medijan gornje polovice podataka, što znači da autori 6 i 8 imaju 25% komentara s više od 19 riječi. Gornji kvartil za ostale autore ima uglavnom vrijednost oko 10.



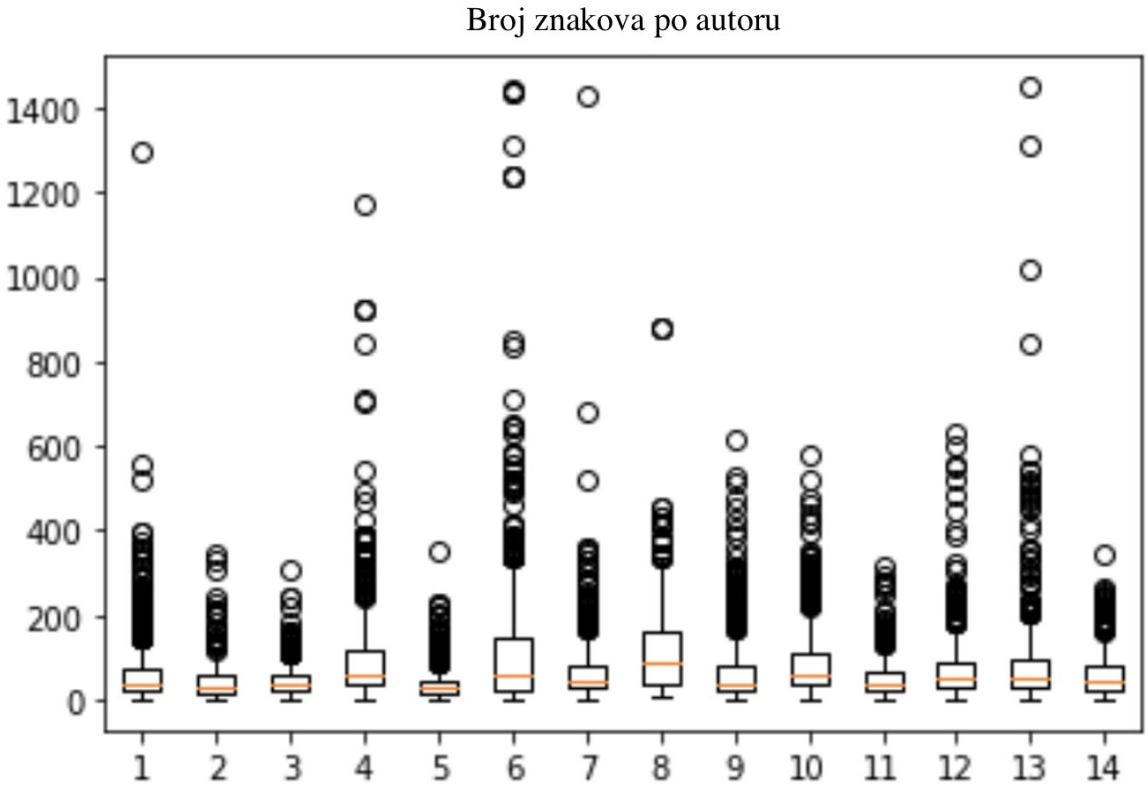
Sl. 3.2 Box plot graf za funkciju broja riječi u komentaru

Metoda *count_chars* vraća ukupan broj znakova u komentaru. Grafikon 3.3 prikazuje funkcije gustoće za broj znakova u komentaru. Iz grafikona je vidljivo da se vrh krivulje za sve autore nalazi unutar intervala [20, 25], odnosno da je najveća vjerojatnost da broj znakova u komentaru iznosi nešto više od 20 za sve autore.



Sl. 3.3 Graf gustoće za funkciju broja znakova u komentaru

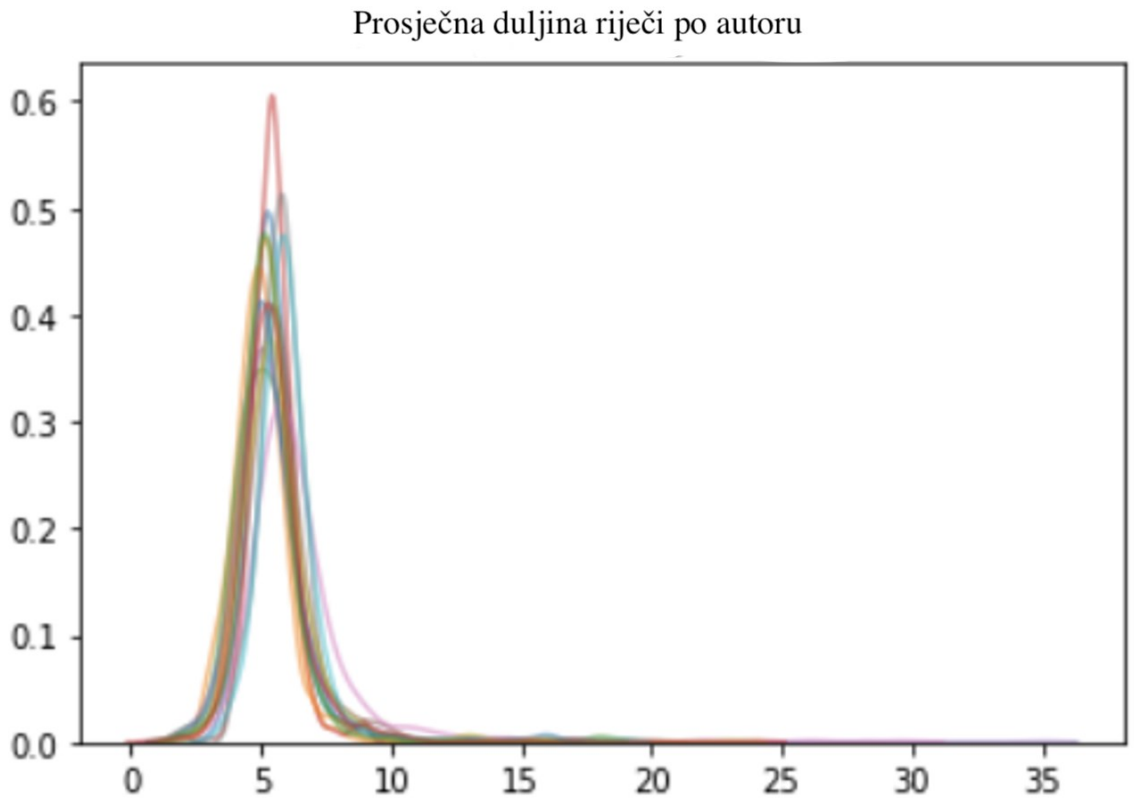
Na grafu 3.4 je prikazan box plot graf za broj znakova u komentaru. Vrijede slična svojstva kao i za mjeru broja riječi u komentaru. Najveću vrijednost gornjeg kvartila imaju autori 6 i 8, a iznosi nešto više od 160. To znači da imaju 25% komentara s više od 160 znakova. Kod ostalih autora gornji kvartil uglavnom iznosi nešto više od 20.



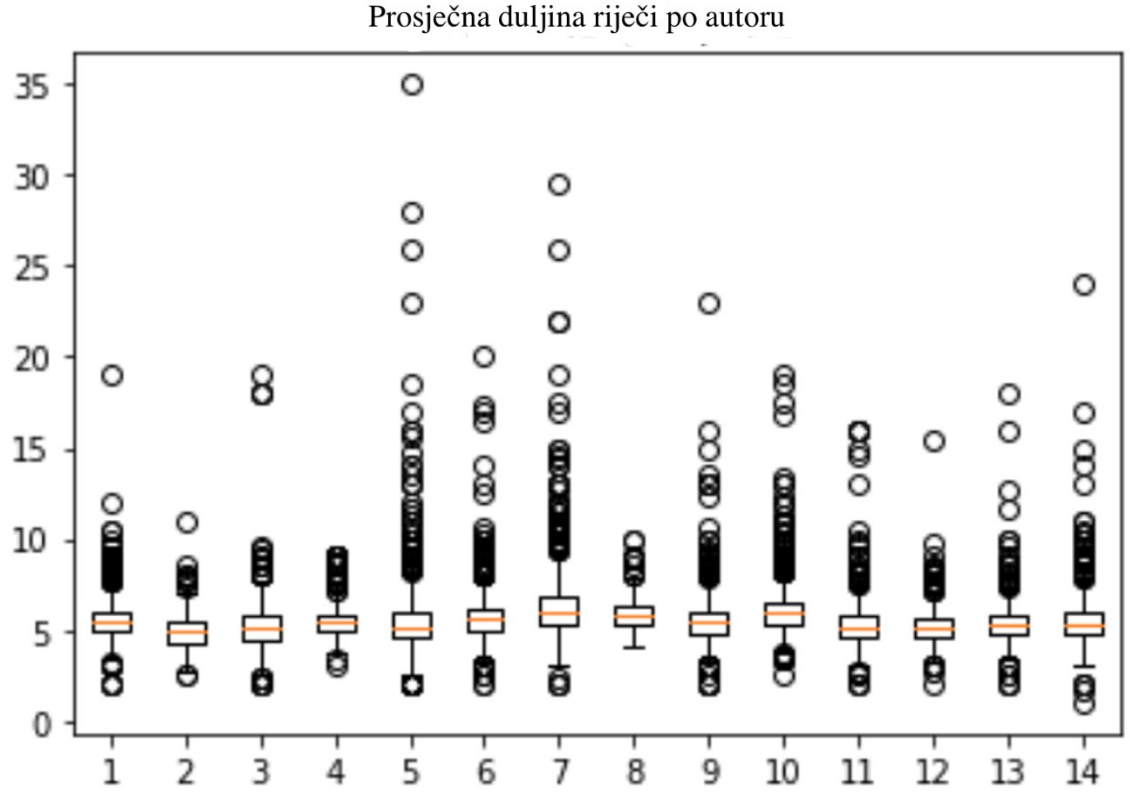
Sl. 3.4 Box plot graf za funkciju broja znakova u komentaru

Metoda *average_length* vraća prosječnu duljinu riječi u komentaru. Grafikon 3.5 prikazuje funkcije gustoće, a grafikon 3.6 box plot graf za prosječnu duljinu riječi u komentaru. Za većinu autora se vrh funkcije gustoće nalazi u intervalu [5,6]. Drugim riječima, za sve autore je najviše vjerojatno da im prosječna duljina riječi ima vrijednost između 5 i 6.

Medijani i kvartili su gotovo ujednačeni za sve autore na box plot grafikonu, što znači da svi autori najčešće koriste riječi čija je duljina oko 5 znakova.



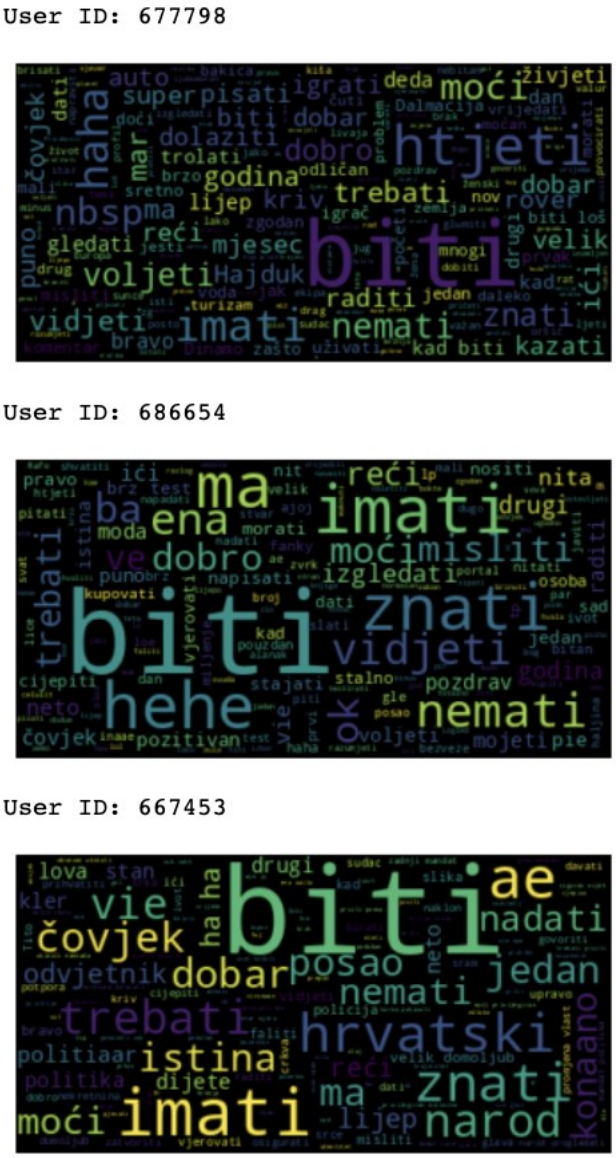
Sl. 3.5 Graf gustoće za prosječan broj riječi u rečenici po autoru



Sl. 3.6 Box plot graf za funkciju prosječne duljine riječi

Generator oblaka riječi generira graf s prikazom najčešće korištenih riječi. Riječ je veća na grafu što se češće pojavljuje unutar teksta. Nasumično su izabrana tri autora nad čijim komentarima su generirani grafovi. Komentari su prethodno lematizirani.

Najčešće korištena lema kod sva tri autora je pomoćni glagol biti. Kod sva tri autora se nalaze i glagoli imati, nemati, znati. Česti su izrazi iz neformalnog govora poput: *haha, he he, ae*.



Sl. 3.7 Graf oblaka riječi

3.3. Ekstrakcija značajki

U nastavku su opisani postupci izdvajanja i računanja značajki.

3.3.1. Lematizacija

Lematizacija je postupak svođenja riječi na kanonski oblik. Kanonski oblik imenice je nominativ jednine, glagola infinitiv, a pridjeva nominativ jednine muškog roda. Lematizacija se često miješa s korjenovanjem (engl. *stemming*). Korjenovanje je postupak izdvajanja korijena riječi. Algoritmi za korjenovanje obično rade samo tako da uklone prefiks ili sufiks riječi, dok lematizacija obuhvaća i morfološku analizu. Također, lematizacija radi ispravnije od korjenovanja kod biblioteka koje podržavaju rad s hrvatskim jezikom.

Lematizacija je napravljena pomoću biblioteke *classla* [19]. *Classla* je biblioteka koja služi za procesiranje hrvatskog, slovenskog, srpskog, makedonskog i bugarskog jezika. Osim lematizacije i tokenizacije ulaznog teksta, *classla* omogućuje i prepoznavanje tipa riječi, tzv. POS (engl. *part of speech*) tagiranje.

Metoda *extract_lemmas* vraća listu izvučenih tema te listu POS tagova.

```
def extract_lemmas(text):
    lemmas = []
    pos_tags = []
    doc = nlp(text)
    for sentence in doc.sentences:
        for word in sentence.words:
            lemmas.append(word.lemma)
            pos_tags.append(word.upos)

    return lemmas, pos_tags
```

Kod 3.5 Metoda za ekstrakciju lema i POS tagova

3.3.2. BOW model

BOW (engl. *bag of words*) je reprezentacija teksta kao parova leme i broja pojavljivanja pojedine leme. Često se koristi kod metoda iz područja obrade prirodnog jezika, pri čemu se brojevi pojavljivanja određenih riječi uzimaju kao značajke za model strojnog učenja.

```
def get_bag_of_words(lemmas):
    filtered_lemmas = list([x for x in lemmas if x not in stopwords])
    return Counter(filtered_lemmas)
```

Kod 3.6 Metoda za generiranje BOW modela

Na primjer, BOW reprezentacija teksta „Ivan voli gledati filmove. Marija također voli filmove.“ bi bila:

```
{'voljeti':2, 'film':2, '.':2, 'Ivan':1, 'gledati':1, 'Marija':1}
```

BOW reprezentaciju koristi metoda *count_unique_terms* koja vraća ukupan broj lema koje se pojavljuju samo jednom u komentaru. Metoda prima BOW reprezentaciju teksta, odnosno rječnik kojem su ključevi leme, a vrijednosti broj pojavljivanja i vraća ukupan broj stavki koje imaju vrijednost 1. Mjera jedinstvenih lema je u pozitivnoj korelaciji s bogatstvom vokabulara.

3.3.3. Zaustavne riječi

Zaustavne riječi su riječi koje same po sebi, bez konteksta, ne nose gotovo nikakvo značenje. U metodama iz domene obrade prirodnog jezika se obično uklanjaju prije treniranja modela, jer se na taj način smanjuje količina podataka bez velikog utjecaja na točnost modela. Suprotno od zaustavnih riječi, odnosno riječi koje nose značenje, nazivaju se funkcijske riječi.

Zaustavne riječi za hrvatski jezik su preuzete iz biblioteke *text_hr* [20]. Kod obrade komentara su iskorištene za filtriranje lema. Izvedena je i značajka *count_stopwords*, tj. broj zaustavnih riječi u izvornom komentaru.

3.3.4. POS tagovi

POS (engl. *part of speech*) tagiranje je postupak pridjeljivanja POS taga tokenima nekog teksta. POS tagiranje omogućuju razne biblioteke, ali tagovi su uglavnom standardni, a to su: ADJ (engl. *adjective*, pridjev), ADP (engl. *adposition*, prijedlog), ADV (engl. *adverb*, prilog), AUX (engl. *auxiliary verb*, pomoćni glagol), CCONJ (engl. *coordinating conjunction*, veznici nezavisno složenih rečenica), DET (engl. *determiner*, dodatci imenici, u engleskom jeziku *a, an* ili *the*), INTJ (engl. *interjection*, usklici), NOUN (engl. *noun,* imenica), NUM (engl. *numeral*, brojčana vrijednost), PART (engl. *particle*, čestica), PRON (engl. *proper noun*, vlastita imenica), PUNCT (engl. *punctation*, interpunkcija), SCNOJ (engl. *subordinating conjunction*, veznici zavisno složenih rečenica), SYM (engl. *symbol*,

simboli), VERB (engl. *verb*, glagol) te X (nepoznato, token ne pripada nijednoj od prethodno navedenih skupina).

Na slici 3.8 je dan ulazni tekst za metodu *extract_lemmas* te izlaz te metode, odnosno leme i POS tagovi.

```
Text:
Matematički je naprosto nemoguće da desetine kompanija u ionako grčevitoj konkurenciji kakva, primjerice, vlada u aut
omobilskoj industriji, ugrabi iole krupniji tržišni udio koji bi opravdao (pre)napregnute valuacije njihovih dionica.

lemma: matematički  POS tag: ADJ
lemma: biti  POS tag: AUX
lemma: naprosto  POS tag: ADV
lemma: nemoguć  POS tag: ADJ
lemma: da  POS tag: SCONJ
lemma: desetina  POS tag: NOUN
lemma: kompanija  POS tag: NOUN
lemma: u  POS tag: ADP
lemma: ionako  POS tag: ADV
lemma: grčevit  POS tag: ADJ
lemma: konkurencija  POS tag: NOUN
lemma: kakav  POS tag: DET
lemma: ,  POS tag: PUNCT
lemma: primjerice  POS tag: ADV
lemma: ,  POS tag: PUNCT
lemma: vlada  POS tag: NOUN
lemma: u  POS tag: ADP
lemma: automobilski  POS tag: ADJ
lemma: industrija  POS tag: NOUN
lemma: ,  POS tag: PUNCT
lemma: ugrabiti  POS tag: VERB
lemma: iole  POS tag: ADV
lemma: krupan  POS tag: ADJ
lemma: tržišni  POS tag: ADJ
lemma: udjel  POS tag: NOUN
lemma: koji  POS tag: DET
lemma: biti  POS tag: AUX
lemma: opravdati  POS tag: VERB
lemma: (  POS tag: PUNCT
lemma: pre  POS tag: ADP
lemma: )  POS tag: PUNCT
lemma: napregnuti  POS tag: ADJ
lemma: valuacija  POS tag: NOUN
lemma: njihov  POS tag: DET
lemma: dionica  POS tag: NOUN
lemma: .  POS tag: PUNCT
```

Sl. 3.8 Ulazni tekst i rezultati metode extract_lemmas

POS tagiranje je napravljeno pomoću biblioteke classla, a broj pojavljivanja svakog taga predstavlja značajku za model strojnog učenja. Nad pojedinim komentarom se prvo pozove metoda za generiranje POS tagova, nakon čega se lista tagova šalje u metode za brojanje pojavljivanja pojedinog taga.

```
def count_adj(pos_tags, tag):
    return pos_tags.count(tag)
```

Kod 3.7 Metode za računanje broja pojavljivanja pojedinog POS taga

3.3.5. Brojanje tipfelera

Tipfeler ili tiskarska pogreška je pogreška nastala tijekom tipkanja. Broj tipfelera je diskriminativna značajka budući da su pojedini ljudi skloniji pravljenju tipfelera od drugih. Za prepoznavanje tipfelera je korištena biblioteka *phunspell* [21]. *Phunspell* uključuje riječnike za sve jezike koje podržava LibreOffice i općenito se koristi za provjeru pravopisa. Metoda *lookup_list* vraća listu kandidata, tj. potencijalne tiskarske pogreške, a metoda *suggest* pomoću generatora kreira listu prijedloga, odnosno listu potencijalno ispravnih riječi. Na slici 3.9 je prikazan rad biblioteke *phunspell* za hrvatski jezik. Prije poziva metode za prepoznavanje tiskarskih pogrešaka je potrebno ukloniti interpunkcijske znakove jer ih biblioteka označuje kao tipfeler.

```
Example: Imali smo svakvih predcjednika, i ljevih i desnih .›
Found tipfellers:  ['svakvih', 'predcjednika,', 'ljevih', '.›']

Tipfeler: svakvih
Suggestions:
svakih
ovakvih
svakakvih
svakakvi
kakvih

Tipfeler: predcjednika,
Suggestions:
predsjednik

Tipfeler: ljevih
Suggestions:
lijevih

Tipfeler: .›
Suggestions:
```

Sl. 3.9 Rad biblioteke phunspell

3.3.6. CountVectorizer

CountVectorizer [22] je klasa iz paketa *sklearn* [23], a služi za konverziju dokumenta u matricu u koju su pohranjeni brojevi pojavljivanja pojedinog tokena (stupci matrice) u pojedinoj particiji teksta (redci matrice). Značajke su 1-gram, 2-gram i 3-gram tokeni, pri čemu je jedinica riječ, a ukupan broj značajki je 30. Postavljanjem zastavica *min_df* i *max_df*

je moguće izbaciti tokene koji se pojavljuju manje od *min_def*, odnosno više od *max_def* vrijednosti. Za učenje modela se koriste lematizirani i predprocesuirani komentari.

```
def get_count_vectorizer_features(data, sentences):
    tfidf_vec   =   CountVectorizer(ngram_range=(1,3),   max_features=30,
min_def = 10)
    tfidf_vec.fit(data)
    return pd.DataFrame(tfidf_vec.transform(sentences).toarray())
```

Kod 3.8 Metoda za generiranje CountVectorizer matrice

3.3.7. TF-IDF vektor

TF-IDF mjera raste proporcionalno s brojem pojavljivanja riječi u korpusu. Prema istraživanju iz 2015. čak 83% sustava za preporučivanje koji se temelje na analizi teksta koriste TF-IDF mjeru [4].

TF-IDF vektorizacija je napravljena pomoću klase *TfidfVectorizer* [24] iz *sklearn* biblioteke. *TfidfVectorizer* vraća matricu TF-IDF značajki, a ekvivalent je slijednom korištenju klasa *CountVectorizer* i *TfidfTransformer*. CountVectorizer stvara matricu kojoj su stupci riječi ili n-gram tokeni, ovisno o tome koja se mjera zada, a redci particije dokumenta. Vrijednosti u matrici su broj pojavljivanja segmenta teksta u datoj particiji. *TfidfVectorizer* transformira matricu u TF-IDF reprezentaciju. Glavni cilj korištenja TF-IDF mjere je smanjiti utjecaj onih tokena koji se jako često pojavljuju zbog čega su empirijski manje informativni od tokena koji se rijetko pojavljuju. Računa se kao umnožak mjere za učestalost tokena i mjere za inverznu učestalost.

Prije računanja TF-IDF vektora, potrebno je izbaciti interpunkcijske znakove i izvući leme iz svakog komentara. Te leme se spajaju u jedan zajednički dokument koji se predaje kao parametar metodi *get_tf_idf_features* tako da su stupci matrice najučestaliji tokeni svih komentara. Za stupce matrice su korišteni 1-gram, 2-gram i 3-gram tokeni, a ukupan broj značajki je 30.

```
def get_tf_idf_features(sentences):

      vectorizer = TfidfVectorizer(tokenizer=None, preprocessor=None,
      ngram_range=(1, 3), use_idf=True, smooth_idf=False, norm=None,
      decode_error='replace', max_features=30)

      tf_idf_features = vectorizer.fit_transform(sentences).toarray()
      tf_idf_features = pd.DataFrame(tf_idf_features)
      print("Tf idf features:")
      print(vectorizer.get_feature_names_out())
      return tf_idf_features
```

Kod 3.9 Metoda za TF-IDF vektorizaciju

3.3.8. Redukcija dimenzionalnosti

Za redukciju dimenzionalnosti je korištena klasa *TruncatedSVD* [25] iz paketa *sklearn.* Ovaj transformator izvodi linearnu redukciju dimenzionalnosti pomoću krnje singularne dekompozicije. Krnji SVD (engl. *singular value decomposition*) je aproksimacija dekompozicije matrice. Dekompozicija matrice je rastavljanje matrice na lijevu ortonormalnu i desnu ortogonalnu matricu. Stupci lijeve matrice su lijevi singularni vektori, a stupci desne matrice desni singularni vektori. Ideja redukcije dimenzionalnosti pomoću dekompozicije matrice je smanjiti broj dimenzija kombiniranjem postojećih vrijednosti.

Redukcija dimenzionalnosti je primijenjena na TF-IDF te *CountVectorizer* matrici, pri čemu je broj značajki smanjen na 15.

```
def SVD_Reduce(features, no_of_components):
    svd = TruncatedSVD(n_components=no_of_components, algorithm='arpack')
    svd.fit(features)
    return pd.DataFrame(svd.transform(features))
```

Kod 3.10 Redukcija dimenzionalnosti uz pomoć *TruncatedSVD* klase

Parametar *no_of_components* je broj značajki, odnosno stupaca, na koji smanjujemo ulaznu matricu.

3.3.9. Konačni izgled matrice

Konačno, sve izvučene značajke je potrebno spojiti s reduciranim TF-IDF i *CountVectorizer* matricama. Dobije se matrica čiji su stupci značajke i tokeni, a redci komentari iz seta za učenje i testiranje.

```
train_df = pd.concat([feature_train_df,tf_idf_svd_train,
count_vectorizer_svd_train], axis=1)

test_df = pd.concat([feature_test_df,tf_idf_svd_test,
count_vectorizer_svd_test], axis=1)
```

Kod 3.11 Kreiraranje finalnih matrica

Na slici 3.10 je prikazana transponirana finalna matrica za podskup od 10 komentara.

	0	1	2	3	4	5	6	7	8	9
words_count	20.000000	18.000000	20.000000	15.000000	22.000000	11.000000	6.000000	8.000000	7.000000	18.000000
chars_count	129.000000	102.000000	146.000000	86.000000	153.000000	67.000000	35.000000	39.000000	68.000000	94.000000
average_word_length	6.450000	5.666667	7.300000	5.733333	6.954545	6.090909	5.833333	4.875000	9.714286	5.222222
punctuation_count	3.000000	0.000000	6.000000	1.000000	4.000000	4.000000	1.000000	1.000000	2.000000	1.000000
stopword_count	5.000000	11.000000	6.000000	6.000000	8.000000	6.000000	2.000000	3.000000	4.000000	8.000000
mispelled_count	5.000000	2.000000	3.000000	4.000000	8.000000	1.000000	2.000000	1.000000	2.000000	3.000000
unique terms count	15.000000	7.000000	13.000000	11.000000	11.000000	7.000000	5.000000	6.000000	4.000000	12.000000
ADJ	3.000000	1.000000	0.000000	1.000000	2.000000	1.000000	0.000000	0.000000	0.000000	1.000000
ADV	0.000000	3.000000	2.000000	1.000000	3.000000	1.000000	1.000000	0.000000	0.000000	1.000000
INTJ	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000
NOUN	8.000000	3.000000	9.000000	3.000000	8.000000	2.000000	1.000000	1.000000	1.000000	5.000000
PROPN	3.000000	0.000000	0.000000	1.000000	0.000000	0.000000	2.000000	0.000000	1.000000	0.000000
VERB	1.000000	3.000000	4.000000	2.000000	3.000000	1.000000	0.000000	2.000000	0.000000	2.000000
ADP	1.000000	5.000000	2.000000	1.000000	2.000000	0.000000	1.000000	1.000000	1.000000	2.000000
AUX	1.000000	0.000000	2.000000	1.000000	1.000000	2.000000	0.000000	0.000000	0.000000	1.000000
CCONJ	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	2.000000
DET	2.000000	1.000000	1.000000	1.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000
NUM	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
PART	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000
PRON	0.000000	1.000000	0.000000	0.000000	0.000000	2.000000	0.000000	1.000000	1.000000	3.000000
SCONJ	0.000000	0.000000	0.000000	2.000000	1.000000	0.000000	0.000000	2.000000	1.000000	1.000000
UNKOWN	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
0	1.171103	2.850632	3.051249	2.853621	1.600208	3.280972	0.000000	1.416081	1.336214	2.826196
1	-0.272060	1.599164	0.549505	0.427399	0.436693	-1.122436	0.000000	1.878034	-2.127847	2.177511
2	0.152143	-1.840566	0.137005	0.415130	0.512445	1.077852	0.000000	0.230439	-1.284745	-0.135058
3	-0.104264	2.260470	-1.447059	0.311562	0.316452	-1.382772	0.000000	-1.538124	0.619737	-0.975128
4	0.226805	1.123346	0.543898	0.426129	-0.489712	-4.231131	0.000000	0.553627	-0.387790	1.433706
5	0.046351	0.295537	0.840032	-1.598056	-0.891982	-0.675332	0.000000	0.028029	-2.335267	0.635401
6	-0.005046	0.502977	0.561910	0.937769	1.046607	-1.560217	0.000000	0.657728	-1.415065	-0.311085
7	0.237743	0.825982	1.152646	-0.879211	-0.417891	0.140065	0.000000	-0.872180	-3.217372	0.940135
8	0.402278	1.694530	-2.181690	0.731123	0.339030	1.120262	0.000000	-1.469928	0.042818	-0.283170
9	-0.269190	0.388080	1.101594	-1.018239	-0.580684	0.348993	0.000000	0.571744	3.837918	1.128505
10	-0.826850	-0.365704	-1.266407	-0.192267	0.502585	-1.940026	0.000000	-0.689342	-0.241534	-0.194250
0	0.919365	0.748343	2.093190	1.439536	1.094209	2.143016	0.000000	0.399304	0.270842	1.294672
1	-0.259718	1.017762	-0.099122	0.415326	0.168881	-0.786655	0.000000	0.795022	-0.219561	0.610444
2	0.062411	1.094956	0.052789	-0.461009	-0.520403	-0.735918	0.000000	-0.493247	-0.036604	0.477970
3	-0.124191	-0.110856	0.082546	-0.500127	-0.369386	0.570331	0.000000	0.196075	0.176491	0.037392
4	0.123734	0.038437	0.751698	0.329787	0.086884	-1.280177	0.000000	0.593402	-0.339709	0.242347
5	-0.070867	0.176211	-0.955007	0.571456	0.211711	0.053311	0.000000	-0.144810	1.027743	-0.618512
6	-0.102277	-0.188157	0.995162	-0.739339	-0.139855	0.474847	0.000000	0.193422	0.107082	0.412662
7	0.217020	0.595880	0.066026	0.149231	-0.331561	0.545497	0.000000	0.035243	0.782200	0.270816
8	-0.082805	-0.543413	-0.151939	-0.345658	-0.149860	-0.331988	0.000000	0.010143	0.628470	0.093518
9	-0.126769	0.346234	0.783037	-0.100470	0.054111	-0.064261	0.000000	0.326439	0.883869	-0.155886
10	-0.293899	0.852835	0.134333	-0.340969	0.044142	-0.908845	0.000000	0.084322	-0.645320	0.030391

Sl. 3.10 Transponirana finalna matrica

3.4. Klasifikacija

Za klasifikaciju su korišteni algoritmi nadziranog strojnog učenja koji se standardno koriste kod pripisivanja autorstva [4]. Korišteni su naivni Bayesov klasifikator i pasivno agresivni Bayesov klasifikator, stroj potpornih vektora te logistička regresija.

3.4.1. Naivni Bayesov klasifikator

Naivni Bayesov klasifikator je probabilistički klasifikator koji pretpostavlja uvjetnu nezavisnost ulaznih značajki. Značajke su skalirane uz pomoć *MinMax* [26] normalizatora. Min-Max normalizacija skalira podatke u raspon od 0 do 1. Naivni Bayesov klasifiktor radi s vjerojatnostima pa je ova normalizacija nužna.

```
model =
Pipeline([('Normalizing',MinMaxScaler()),('MultinomialNB',MultinomialNB()
)])
model.fit(train_df, author_train)
y_pred = model.predict(test_df)
pred_test_y = model.predict_proba(test_df)
```

Kod 3.12 Naivni Bayesov klasifikator

3.4.2. Stroj potpornih vektora

Stroj potpornih vektora je linearni klasifikacijski algoritam. Treniranje modela znači pronalaženje hiperravnine između dvije klase na način na maksimizira udaljenost od najbližih primjera obje klase. Za implementaciju je korištena linearna jezga te algoritam pretraživanja rešetke za pronalazak optimalnog hiperparametra *C*. To je hiperparametar koji definira tvrdoću margine.

```
svm_model = SVC()
parameters = {'kernel': ['linear'], 'C':[1, 10, 100]}
scorer = make_scorer(accuracy_score)
grid_search = GridSearchCV(svm_model, parameters, scoring = scorer,
verbose = 50)
model = grid_search.fit(train_df, author_train)
y_preds_svm = model.predict(test_df)
y_preds_svm_proba = model.predict_proba(test_df)
```

Kod 3.13 Stroj potpornih vektora

3.4.3. Pasivno agresivni Bayesov klasifikator

Pasivno agresivni Bayesov klasifikator spada u algoritme s *online* učenjem. *Online* učenje znači da ulazni podatci dolaze sekvencijalno i model se ažurira korak po korak. Ako je ulazni podatak ispravno klasificiran, model ne radi ništa, tj. ponaša se pasivno. Ako je ulazni podatak neispravno klasificiran, onda se rade promjene na modelu [11].

```
pa_classifier=PassiveAggressiveClassifier(max_iter=50)
pa_classifier.fit(train_df, author_train)
y_pred=pa_classifier.predict(test_df)
```

Kod 3.14 Pasivno agresivni Bayesov klasifikator

3.4.4. Logistička regresija

Logistička regresija je klasifikacijski algoritam koji se koristi za određivanje vjerojatnosti pripadnosti kategoriji zavisne varijable na temelju više nezavisnih varijabli. Za odabir kategorije koristi procjenu maksimalne vjerojatnosti.

```
pipe = Pipeline(
        [('select',
SelectFromModel(LogisticRegression(class_weight='balanced', penalty="l1",
C=0.01, solver='liblinear'))),
        ('model',
LogisticRegression(class_weight='balanced',penalty='l1',
solver='liblinear'))])X_train = train_df.to_numpy()
X_test = test_df.to_numpy()
param_grid = [{}]
grid_search = GridSearchCV(pipe, param_grid,
cv=StratifiedKFold(n_splits=5).split(train_df, author_train), verbose=2)
model = grid_search.fit(train_df, author_train)
```

Kod 3.15 Logistička regresija

4. Rezultati i evaluacija

Ukupan broj ulaznog seta komentara iznosi 212 453. Iz ulaznog skupa podataka su izbačeni svi autori koji su ostavili manje od 500 komentara, pa se finalni set podataka sastoji od 14 autora s 500 i više komentara. Komentari su pretežno jako kratki pa je samim time zadatak poprilično izazovan.

U nastavku su dane konfuzijske matrice i tablice s rezultatima za pojedini algoritam. Konfuzijska matrica je matrica čiji su redci predviđene vrijednosti, a stupci stvarne vrijednosti. Prvi redak i prvi stupac predstavljaju prvu klasu i tako redom. Primjerice, ako je klasifikator pronašao 10 komentara koji uistinu pripadaju autoru koji se nalazi u prvom stupcu, tada će vrijednost elementa matrice koji se nalazi u prvom retku i prvom stupcu iznositi 10.

Redci tablica predstavljaju autore, a u stupci preciznost, odziv, f1-rezultat i ukupan broj komentara testnog skupa po autoru. Testni skup predstavlja 30% ukupnog ulaznog skupa podataka, što znači da je broj komentara za svakog autora na kojem se učio klasifikator 2.333 puta veći od skupa za testiranje.

Preciznost je definirana kao omjer točno označenih primjera (TP, engl. *true positive*) i ukupnog broja primjera koje je klasifikator pridijelio klasi. Primjerice, neki autor ima 30 komentara, a klasifikator označi 15 komentara oznakom tog autora, od čega 10 uistinu pripada autoru (TP), a 5 komentara pripada nekim drugim autorima i klasifikator ih je pogrešno označio (FP, engl. *false positive*). U tom slučaju preciznost iznosi 10/15.

$$Preciznost = \frac{TP}{TP+FP} \tag{4.1}$$

Odziv je omjer točno označenih primjera (TP) i ukupnog broja primjera iz klase. Ukupan broj primjera iz klase je zbroj točno označenih primjera i primjera koje je klasifikator propustio označiti (FN, engl. *false negative*).

$$Odziv = \frac{TP}{TP+FN} \tag{4.2}$$

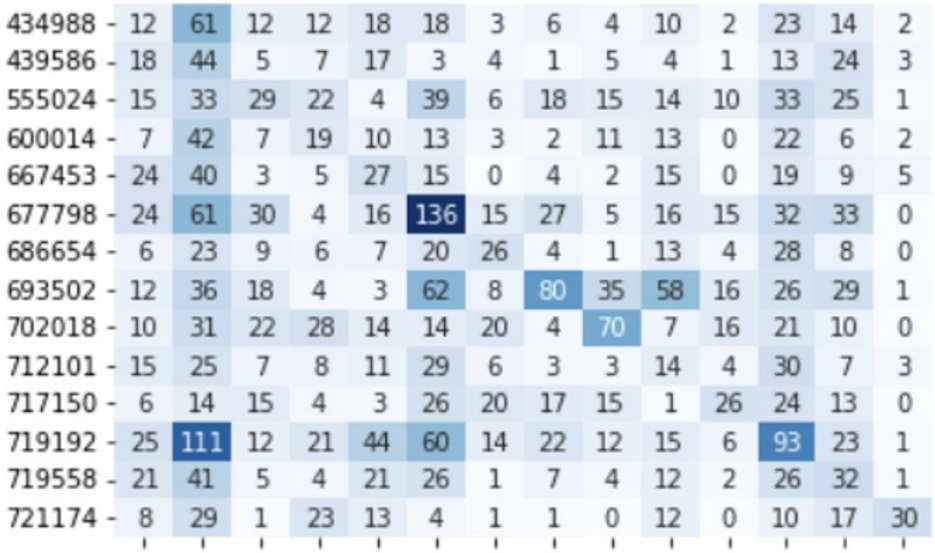
F1 mjera je definirana kao kombinacija preciznosti i odziva. Računa se prema izrazu (4.3).

$$F1 = 2 * \frac{Preciznost*Odziv}{Preciznost+Odziv} \tag{4.3}$$

U tablici 4.1 su dani rezultati za stroj potpornih vektora. Ukupna točnost modela iznosi 0.19. Najveća vrijednost F1 mjere je ostvarena kod dva autora, a iznosi 0.31. Preciznost kod prvog autora iznosi 0.29, što znači da 29% komentara koje je klasifikator označio da pripadaju tom autoru zaista pripada. Odziv iznosi 0.33, što znači da ti komentari čine 33% svih komentara tog autora iz testnog skupa. Kod drugog autora je preciznost nešto veća i iznosi 0.38, ali je odziv manji i iznosi 0.26. To znači da je klasifikator uspješno prepoznao 26% komentara tog autora. Na slici 4.1 je prikazana konfuzijska matrica za stroj potpornih vektora.

ID autora	**Preciznost**	**Odziv**	**F1 mjera**	**Veličina testnog skupa**
434988	0.06	0.06	0.06	197
439586	0.07	0.30	0.12	149
555024	0.17	0.11	0.13	264
600014	0.11	0.12	0.12	157
667453	0.13	0.16	0.14	168
677798	0.29	0.33	0.31	414
686654	0.20	0.17	0.18	155
693502	0.41	0.21	0.27	388
702018	0.38	0.26	0.31	267
712101	0.07	0.08	0.08	165
717150	0.25	0.14	0.18	184
719192	0.23	0.20	0.22	459
719558	0.13	0.16	0.14	203
721174	0.61	0.20	0.30	149

Tablica 4.1 Rezultati klasifikatora stroja potpornih vektora



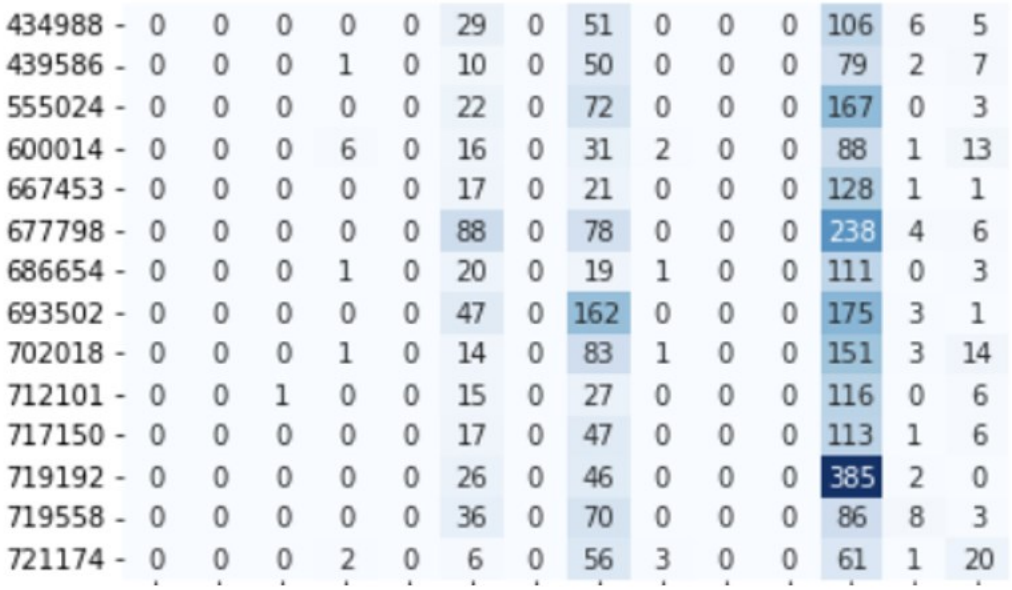
Sl. 4.1 Konfuzijska matrica za stroj potpornih vektora

U tablici 4.2 se nalaze rezultati naivnog Bayesovog klasifikatora. Ukupna točnost modela iznosi 0.2. Najveća točnost, odnosno vrijednost F1 mjere je ostvarena kod autora koji je ukupno ostavio 1530 komentara, od čega je 459 iskorišteno za testiranje. Ona iznosi 0.31. Preciznost iznosi 0.19, što znači da 19% pridruženih komentara uistinu pripada tom autoru. Odziv iznosi 0.84, što znači da je to 84% svih komentara tog autora iz testnog skupa. Razlog tako visoke vrijednosti odziva je taj što je klasifikator većinu komentara pridijelio tom autoru, što je vidljivo na slici 4.2. Stupac s najviše plavih kvadratića u konfuzijskoj matrici predstavlja tog autora.

ID autora	**Preciznost**	**Odziv**	**F1 mjera**	**Veličina testnog skupa**
434988	0.00	0.00	0.00	197
439586	0.00	0.00	0.00	149
555024	0.00	0.00	0.00	264
600014	0.55	0.04	0.07	157
667453	0.00	0.00	0.00	168

677798	0.24	0.21	0.23	414
686654	0.00	0.00	0.00	155
693502	0.20	0.42	0.27	388
702018	0.14	0.00	0.01	267
712101	0.00	0.00	0.00	165
717150	0.00	0.00	0.00	184
719192	0.19	0.84	0.31	459
719558	0.25	0.04	0.07	203
721174	0.23	0.13	0.17	149

Tablica 4.2 Rezultati naivnog Bayesovog klasifikatora



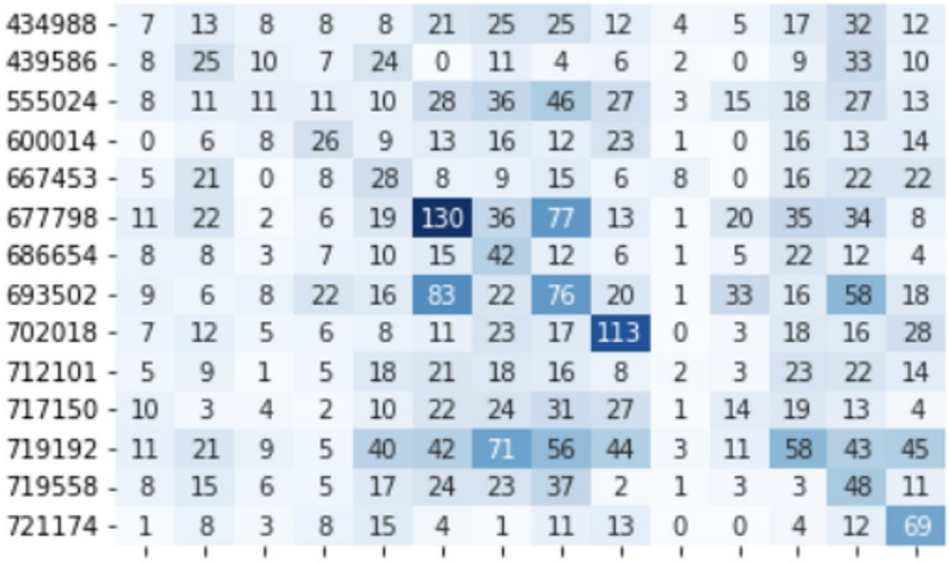
Sl. 4.2 Konfuzijska matrica za naivni Bayesov klasifikator

U tablici 4.3 su prikazani rezultati za logističku regresiju. Točnost klasifikatora iznosi 0.2. Najveća vrijednost F1 mjere je ostvarena kod autora koji je ukupno ostavio 890 komentara, a iznosi 0.39. Preciznost iznosi 0.35, što znači da je 35% komentara koje je klasifikator pridijelio tom autoru uistinu njegovo. Odziv iznosi 0.42, što znači da ti komentari čine 42%

svih komentara tog autora koji se nalaze u testnom skupu. Na slici 4.3 je prikazana konfuzijska matrica za logističku regresiju.

ID autora	Preciznost	Odziv	F1 mjera	Veličina testnog skupa
434988	0.07	0.04	0.05	197
439586	0.14	0.17	0.15	149
555024	0.14	0.04	0.06	264
600014	0.21	0.17	0.18	157
667453	0.12	0.17	0.14	168
677798	0.31	0.31	0.31	414
686654	0.12	0.27	0.16	155
693502	0.17	0.20	0.18	388
702018	0.35	0.42	0.39	267
712101	0.07	0.01	0.02	165
717150	0.12	0.08	0.09	184
719192	0.21	0.13	0.16	459
719558	0.12	0.24	0.16	203
721174	0.25	0.46	0.33	149

Tablica 4.3 Rezultati logističke regresije



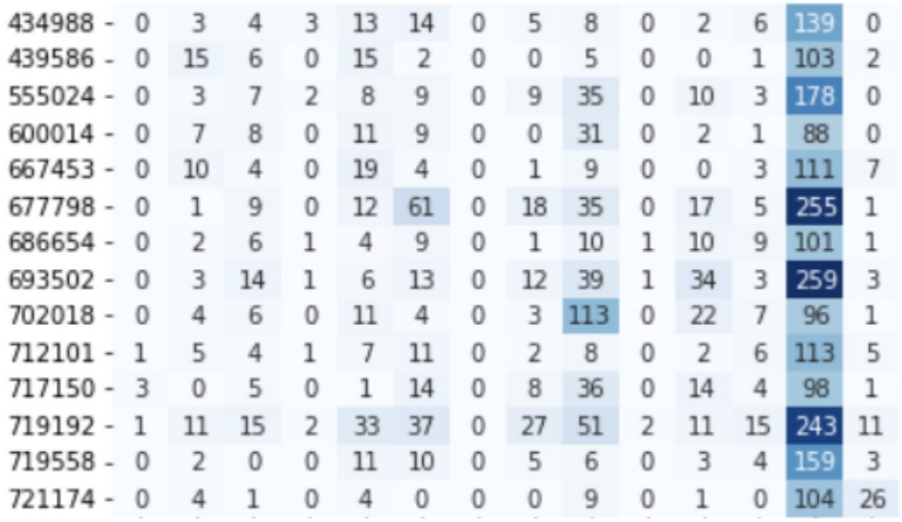
Sl. 3.1 Konfuzijska matrica za logističku regresiju

U tablici 4.4 su dani rezultati za pasivno agresivni Bayesov klasifikator. Iako je najmoderniji među implementiranim algoritmima, ostvario je najmanju točnost, a ona iznosi 0.13. Najveća vrijednost F1 mjere je ostvarena kod autora koji je ostavio ukupno 1317 komentara, od čega je 395 korišteno za testiranje, a iznosi 0.34. Preciznost iznosi 0.42, a odziv 0.29.

ID autora	Preciznost	Odziv	F1 mjera	Veličina testnog skupa
434988	0.00	0.00	0.00	197
439586	0.10	0.21	0.14	149
555024	0.03	0.08	0.04	264
600014	0.00	0.00	0.00	157
667453	0.11	0.12	0.12	168
677798	0.15	0.31	0.20	414
686654	0.00	0.00	0.00	155
693502	0.03	0.13	0.05	388

702018	0.42	0.29	0.34	267
712101	0.00	0.00	0.00	165
717150	0.08	0.11	0.09	184
719192	0.03	0.22	0.06	459
719558	0.78	0.08	0.14	203
721174	0.17	0.43	0.25	149

Tablica 4.4 Rezultati pasivno agresivnog Bayesovog klasifikatora



Sl. 4.4 Konfuzijska matrica za pasivno agresivni Bayesov klasifikator

5. Zaključak

Tema ovog diplomskog rada je bila pokušati odgovoriti na pitanje je li moguće jasno prepoznati karakteristike kratkog pisanog teksta po kojima je moguće diferencirati autore. Primijenjene su razne stilometrijske metode u svrhu ekstrakcije i kvantificiranja jezičnih obilježja, a koje su korištene kao značajke za modele strojnog učenja. Na malom skupu kratkih komentara je teško prepoznati obilježja koja karakteriziraju nekog autora, pa su iz ulaznog seta podataka izbačeni svi autori koju su ostavili manje od 500 komentara. Kombiniranje nadziranog strojnog učenja i metoda iz područja stilometrije na korištenom setu podataka nije dalo dobre rezultate kod pripisivanja autorstva, ali primarni problem je u malom setu podataka. Autori su imali između 500 i 1500 komentara, što prema predloženoj shemi nije dovoljno za prepoznavanje univerzalnog jezičnog obrasca.

Glavna pretpostavka kod pripisivanja autorstva je da sam autor mora ostaviti dobar trag, odnosno veliku količinu komentara. Također, da bi identificirali koja osoba stoji iza lažnog profila, mora biti ispunjena i pretpostavka da je osoba s drugog, odnosno pravog, profila ostavila dovoljno komentara. Većina objavljenih znanstvenih radova na temu pripisivanja autorstva ima strukturirane ulazne podatke, pa su i same točnosti modela iznimno visoke. Primjerice, u studiji Nirkhi i suradnici [2], ulazni set podataka je napravljen podjelom novela od 50 autora na odlomke od 500 riječi. Korištenjem algoritama strojnog učenja ostvarena je točnost od oko 90%. Komentari s društvenih mreža i portala su nestrukturirani i najčešće kratki pa je teže pronaći obrasce koji definiraju određenog korisnika. Komentari analizirani u radu su pisani na hrvatskom jeziku koji je kompleksan pa se kod obrade tekstova potrebno pozabaviti uklanjanjem raznih prefiksa i sufiksa nastalih deklinacijama ili konjugacijama kako bi se pronašao kanonski oblik riječi. Također, puno je manji spektar biblioteka nego za engleski jezik.

U svrhu poboljšanja rezultata potrebno je napraviti korekcije na modelu strojnog učenja u vidu pronalaska optimalnijih parametara ili uvođenja još diskriminativnijih značajki. Vjerujem da bi se predložena shema, uz određena poboljšanja, u skorijoj budućnosti mogla koristiti u stvarnim forenzičkim aktivnostima, poput pripisivanja autorstva nekog pisanog teksta. Važno je primijetiti da je za klasifikaciju iskorišteno samo 45 značajki od stotina značajki koje je moguće izvesti iz teksta. Naravno, nije ideja imati puno beskorisnih značajki, jer se nagomilavanjem značajki proširuje dimenzionalnost prostora te povećava

vrijeme potrebno za izračun. Zbog toga je od velike važnosti selekcija značajki. Također, potrebno je pronaći način kako poboljšati robusnost modela da može rukovati s kratkim komentarima.

6. Literatura

[1] Bhargava, M., Mehndiratta, P., Asawa, K. (2013). *Stylometric Analysis for Authorship Attribution on Twitter*, Big Data Analytics, 2013.

[2] Brady, P.T. *A Statistical Analysis of On-off Patterns in 16 Conversations*, Bell System Technical Journal, 47,1, str. 55-62, 1998.

[3] Kopel, M., Scheler, J. *Exploiting Stylistic Idiosyncrasies for Authorship Attribution*, Bar-Ilan University, Ramat-Gan, Izrael, 2003.

[4] Savoy, J., *Machine Learning Methods for Stylometry*, Springer, Department of Computer Science, University of Neeuchatel, Switzerland, 2020.

[5] Popescu, I.I., Altmann, G., Grzybek, P., Jayaram, B.D, Köhler, D., Krupa, V., Macutek, J., Pustet, R., Uhlírovà, L., Vidya, M.N., *Word Frequency Studies*, De Gruyter Mouton, Berlin, 2009.

[6] Hartman, R. R. K., Gregory, J., *Dictionary of Lexicography* (2nd ed.), Taylor & Francis e-Library, New York, 2002.

[7] Breitinger, C., Gipp, B., Langer, S., *Research paper recommender systems: a literature survey*, International Journal on Digital Libraries, 2016.

[8] Gregori-Signes, C., Clavel-Arroitia, B., *Analysing lexical density and lexical diversity in the university students' written discourse*, Proceedings International Conference on Corpus Linguistics, 2015.

[9] T. Mendenhall, *The characteristic curves of composition,* Science, 1887.

[10] Sari, Y., *Neural and non-neural Approaches to Authorship Attribution*, PhD thesis, Department of Computer Science, The University of Sheffield, 2018.

[11] Crammer, K., *Online Passive-Aggressive Alghoritms*, Journal of Machine Learning Research, The Hebrew University, Izrael, 2006.

[12] Naive Bayes - scikit-learn 1.1.2 dokumentacija, https://scikit-learn.org/stable/modules/naive_bayes.html (pristupljeno 15.9.2022).

[13] Smola, A. J., & Schölkopf, B., *A tutorial on support vector regression. Statistics and Computing*, 199–222, 2004.

[14] Bishop, C. M., *Pattern Recognition and Machine Learning*, New York, Springer, 2006.

[15] Powers, D.M.W, *Applications and Explanations of Zipf's Law*, Department of Computer Science, The Flinders University of South Australia, 1998.

[16] Siskova, Z., *Lexical Richness in EFL Students' Narratives*, University of Reading Language Studies Working Papers, 2012.

[17] MongoDB dokumentacija, https://www.mongodb.com/docs/ (pristupljeno 13.9.2022.)

[18] *BeautifulSoup* dokumentacija, https://beautiful-soup-4.readthedocs.io/en/latest/ (pristupljeno 11.9.2022.)

[19] *Classla* dokumentacija, https://pypi.org/project/classla, (pristupljeno 9.9.2022.)

[20] *Text_hr* dokumentacija, https://pypi.org/project/text-hr, (pristupljeno 31.8.2022.)

[21] *Phunspell* dokumentacija, https://pypi.org/project/phunspell, (pristupljeno 10.9.2022.)

[22] *CountVectorizer*, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html, (prisupljeno 20.9.2022.)

[23] *sklearn* dokumentacija, https://scikit-learn.org/stable, (pristupljeno 1.9.2022.)

[24] *TfIdfVectorizer* dokumentacija, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, (pristupljeno 3.9.2022.)

[25] TruncatedSVD dokumentacija, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html, (pristupljeno 5.9.2022.)

[26] *MinMaxScaler* dokumentacija, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html, (pristupljeno 7.9.2022.)

Sažetak

Internet je omogućio nastanak mnogih portala i društvenih mreža preko kojih se prenose vijesti iz svih domena. Određene teme pobuđuju visok interes javnosti i pokreću lavinu komentara. Porast računalne snage i memorijskih kapaciteta je omogućio pohranjivanje i analiziranje tih komentara. Cilj ovog rada je pokušati odgovoriti na pitanje je li moguće prepoznati i kvantificirati stil pisanja koji jedinstveno pripada svakom čovjeku i na taj način povezati komentar sa stvarnim autorom. Korištene su metode iz domene stilometrije u kombinaciji s algoritmima nadziranog strojnog učenja. Potrebno je pronaći način kako poboljšati robusnost modela da može rukovati s kratkim komentarima jer prema predloženoj shemi mali skup kratkih komentara nije dovoljan za prepoznavanje univerzalnog jezičnog obrasca.

Ključne riječi: pripisivanje autora, stilometrija, strojno učenje, obrada prirodnog jezika

Summary

The flow of news is growing every day on social media. Some topics arouse high interest in the public and trigger an avalanche of comments. Processors and memory capacity improvements have made it possible to save and process these comments. The main goal of this thesis is to answer if it is possible to extract and quantify linguistic features by which it is possible to distinguish different authors. Authorship attribution is implemented using stylometric techniques and supervised machine learning algorithms. It is necessary to improve the robustness of the model to be able to handle short comments because according to the proposed scheme a small set of short comments is not enough to recognize a universal writing style.

Key words: authorship attribution, stylometry, machine learning, natural language processing