Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Ivan Banović

Prikupljanje i analiza podataka na Redditu

Završni rad

Mentor: prof. dr. sc. Sanda Martinčić – Ipšić dipl. Ing.
Mentor: dr. sc. Slobodan Beliga

Rijeka, 4.9.2023.

University of Rijeka
FIDIT Faculty of Informatics and Digital Technologies

Rijeka, 8.3.2023.

Zadatak za završni rad

Pristupnik: Ivan Banović

Naziv završnog rada: Prikupljanje i analiza poruka na Reditu

Naziv završnog rada na eng. jeziku: Analysis of Reddit posts

Sadržaj zadatka:

Cilj završnog rada je prikupljanje podataka o porukama na Reddit-u pomoću API-ja s hrvatskog subredita vezanog uz temu COVID-19 pandemije. U završnome radu će se opisati postupak prikupljanja podataka, izvest će se statistička analiza prikupljenih podataka te odgovarajuće vizualizacije. Na istom skupu podatka pokušat će se procijeniti udio poruka pozitivnog i negativnog polariteta s ciljem analize stavova i mišljenja na Redditu tijekom pandemije.

Mentor

Prof. dr. sc. Sanda Martinčić-Ipšić

dr. sc. Slobodan Beliga

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Zadatak preuzet: datum

Banović Ivan

(potpis pristupnika)

Sažetak

Završni rad se bavi analizom objava i komentara vezanih za COVID-19 pandemiju, prikupljenih sa hrvatskog subreddita u periodu od 1.1.2020. do 31.12.2020., te anotiranih sa pozitivnim, neutralnim ili negativnim sentimentom.

Opisat će se početni skup podataka, njegovo prikupljanje, te njegove transformacije, a nad podacima će se provesti statistička analiza i frekvencijska analiza, te će se prikazati vizualizacije u obliku histograma, korelacijskih grafova i matrica zabune. Koristit će se različiti algoritmi za klasifikaciju, kao što su naivni Bayesov algoritam, nasumične šume (eng. *random forest*), te stroja potpornih vektora (eng. *support vector machine*). Uz to, ukratko će se opisati korištene programske metode.

Ključne riječi: Reddit, analiza teksta, strojno učenje, naivni Bayes, nasumična šuma, stroj potpornih vektora, klasifikacija, sentiment, matrica zabune.

Abstract

Analysis of Reddit posts

The final thesis deals with the analysis of posts and comments related to the COVID-19 pandemic, collected from the Croatian subreddit during the period from January 1st, 2020, to December 31, 2020, and annotated with a positive, neutral, or negative sentiment.

The initial dataset, its collection, and its transformations will be described, and visualizations such as histograms, correlation graphs, and confusion matrices will be presented. Different classification algorithms will be used, such as the Naive Bayes algorithm, Random Forest, and Support Vector Machine. Furthermore, a brief description of the programming methods used will be provided.

Keywords: Reddit, text analysis, machine learning, Naive Bayes, Random Forest, Support Vector Classifier, classification, sentiment, confusion matrix.

Kazalo

1. Uvod

Analiza vršena u ovom radu rađena je na skupu podataka sakupljenih s društvene mreže Reddit. Skup podataka se sastoji od objedinjenih objava i komentara vezanih na neki način za koronavirus, te je svaka objava anotirana s oznakom za neutralan, negativan, ili pozitivan sentiment.

U idućim poglavljima proći ćemo kroz značajke potrebne za obradu skupa podataka prije statističke analize, kao i za tekstualnu klasifikaciju. Osvrnut ćemo se na razliku u rezultatima klasifikacije zbog korištenja različitih algoritama, i proći ćemo kroz vizualne prikaze različitih statistika u obliku histograma, oblaka riječi i grafova.

2. Skup podataka

Skup objava s hrvatskog subreddita[1], tzv. „hreddita” preuzet je pomoću Pythona te PRAW[2] i Pushshift[3] API-jeva. Kako bi se pokrila cijela 2020. godina, program sakuplja sve objave u rasponu od 1. siječnja do 31. prosinca, potom iz tog skupa filtrira objave vezane za COVID-19 pandemiju koristeći ključne riječi.

Program je strukturiran na način da pomoću Pushshift API-ja dobiva stalne poveznice (*eng. permalinks*) na sve objave u prijespomenutom vremenskom rasponu. Zatim te poveznice koristi kako bi iz svake objave preuzeo njen tekst (ukoliko je objava tekstualna), metapodatke kao što su broj bodova, vrijeme objave, ID korisnika i broj komentara, te potom i komentare i metapodatke vezane za njih. Sve to pohranjuje u .JSON formatu.

Nakon pohrane, program prolazi kroz podatke o objavama i traži spominjanja ključnih riječi kao što su „corona”, „virus”, „zaraza”, „pandemija” i varijacije na te riječi dobivene korištenjem regularnih izraza.

Skup podataka korišten u ovom radu sastoji se od sveukupno 5439 jedinstvenih objava.

2. Obrada skupa podataka

Za klasifikaciju i statističku analizu je potreban što „općenitiji” skup podataka. Inicijalni skup komentara je sintaktički diverzan, stoga je potrebno tekstove pripremiti za obradu prirodnog jezika i analizu teksta.

Tokenizacija (*eng. Tokenization*) je proces razdvajanja elemenata u tekstu na manje komponente. Ovdje tokeniziramo rečenice na njihove sastavne riječi koristeći razmak (*eng. whitespace*) kao mjesto podjele.

Tokenizacija je u našem slučaju prvi korak u obradi teksta iz skupa podataka, nakon čega iz istoga uklanjamo stop-riječi.

Stop-riječi (*eng. Stop words*) su često korištene riječi u jeziku koje same po sebi ne nose informaciju, primjerice „a”, „ako”, „joj”, ili „kroz”. Uklanjanjem takvih riječi ubrzavamo obradu, ali i izražavamo 'preostale' riječi koje ukazuju na suštinsko značenje teksta i pridonose generalnom shvaćanju teksta u ML modelima.

Kako bismo dodatno olakšali analizu teksta, možemo smanjiti varijaciju među riječima korjenovanjem.

Korjenovanje (*eng. Stemming*) je proces raščlane riječi na njen osnovni oblik, odnosno korijen, kako bi se grupirale riječi koje imaju isti korijen ili isto značenje. Primjerice, i „čokolada” i „čokolino” se svode na korijen „čoko”. Time se smanjuje varijacija riječi i olakšava analiza teksta.

Tablica 1 prikazuje primjer iz skupa podataka prije obrade, dok Tablica 2 pokazuje izlaz nakon navedenih postupaka i brisanja stupaca koji nisu relevantni.

Rbr.	Anotacija	Komentar	ID autora	ID objave	Alias autora	Bodovi
1.	0	„A gle praktički sve virusne infekcije su takve...”	g0bohiq	i3gfig	Flaminije	13
2.	1	„Zaraženih ima znatno više u odnosu na službene brojeve...”	g0kyyk3	i4ve2b	bearmother	10
3.	2	"Stožer se brine o nama...”	g0kyqsb	i4ve2b	MirkoZD	20

Tablica 1: Primjer iz skupa podataka prije obrade

Rbr.	Anotacija	Tekst	Bodovi
1.	0	Gle, praktič, virus, infek	13
2.	1	Zara, viš, služb, broj	10
3.	2	Stožer, brin	20

Tablica 2: Primjer iz skupa podataka nakon obrade

Za stemmanje je korišten *Stemmer for Croatian*[4], alat NLP grupe sa Filozofskog Fakulteta u Zagrebu, dostupan pod GNU Lesser General Public License.

3. Statistička analiza

U skupu podataka su korištene tri vrijednosti za anotacijske oznake: „0” za neutralan sentiment, „1” za negativan i „2” za pozitivan.

Promatranjem Tablice 3 uočavamo da je standardna devijacija od broja bodova objave u prosjeku visoka, uzevši u obzir i da 75% objava ima 14 ili manje bodova, dok prosjek diže malen broj objava sa izrazito visokim brojem bodova.

Gledajući Tablicu 4 vidimo da čak tri četvrtine sveukupnih objava imaju neutralan sentiment, dok su pozitivno i negativno nastrojene objave u manjini.

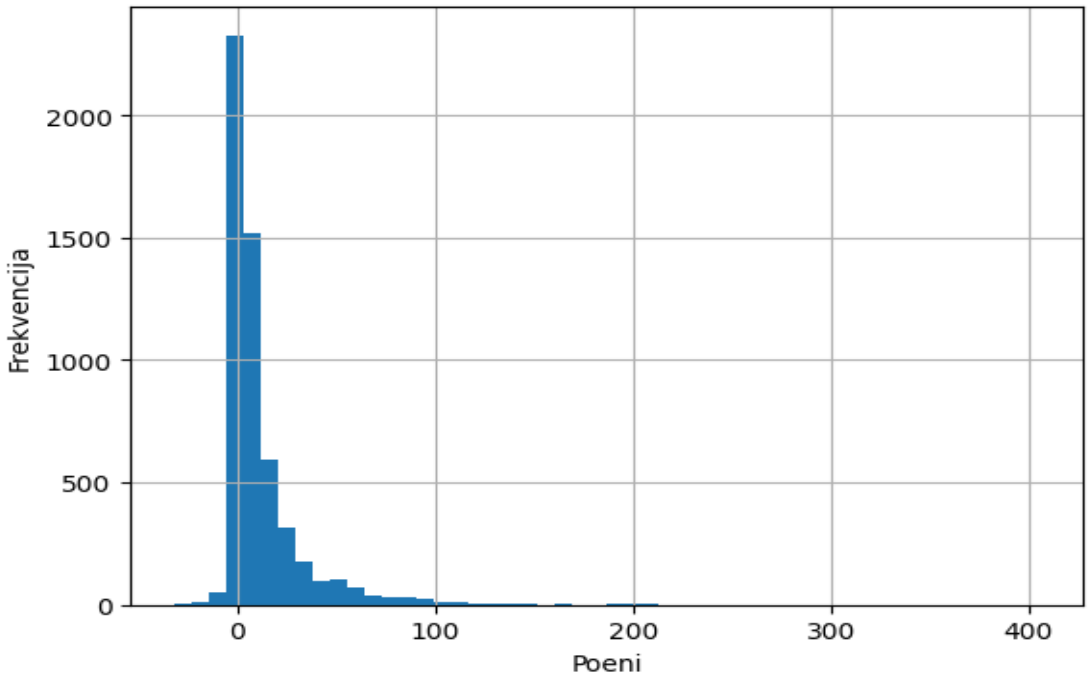
Sveukupno objava	5438
Prosjek bodova	12.43
Standardna devijacija	23.24
Minimum	-32
Maksimum	406

Tablica 3: Statistika broja bodova

Anotacija	Frekvencija	Postotak
Neutralan	4127	75.89%
Negativan	1077	19.81%
Pozitivan	234	4.30%

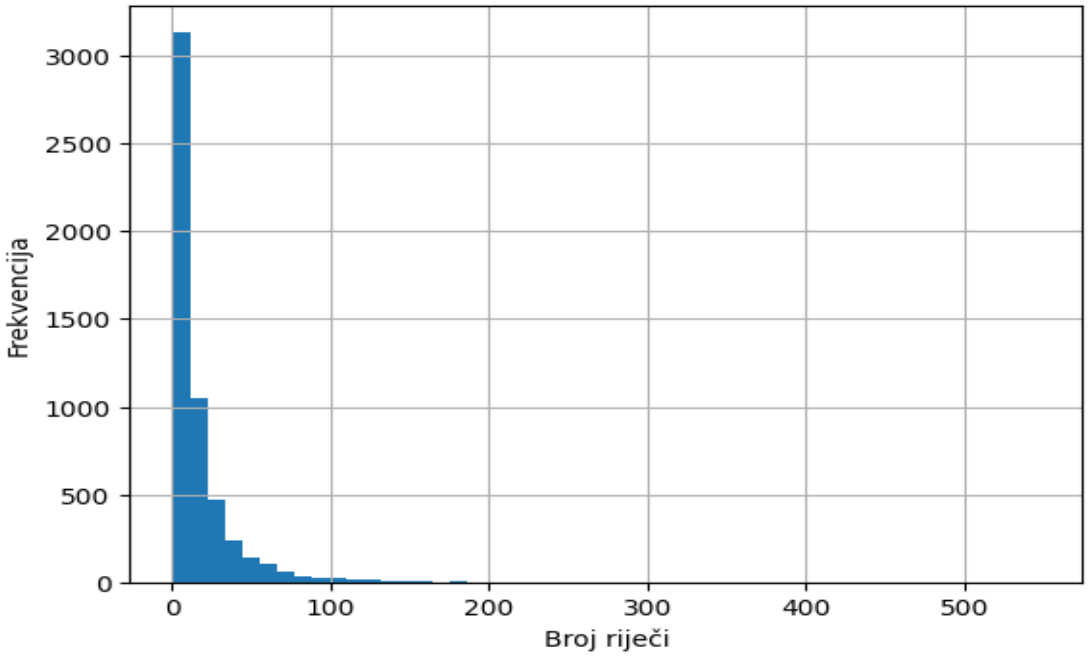
Tablica 4: Broj i postotak pojavljivanja svake anotacijske oznake

Pomoću Pythonovih modula Pandas[5] i matplotlib[6] crtamo histograme koji aproksimiraju distribuciju broja bodova po objavi (Slika 1), kao i broj riječi po objavi (Slika 2).



Slika 1: Histogram broja bodova po objavi

Na temelju skupa objava izrađujemo i **oblake riječi** (*eng. word cloud*), prikazane na slikama 3 do 9. Oblaci riječi prezentiraju učestalost riječi koje se pojavljuju u nekom od podskupova skupa objava (pozitivne, negativne, neutralne). „Riječi" prikazane u oblacima su ustvari tokeni dobiveni procesom korjenovanja riječi, kako bi frekvencija kontekstualno istih riječi bila prikazana preciznije.



Slika 2: Histogram broja riječi po objavi

Slika 3: Najčešće riječi iz cjelokupnog skupa objava

Slika 4: Najčešće riječi iz skupa negativno nastrojenih objava

Slika 5: Najčešće riječi koje se nalaze isključivo u skupu negativno nastrojenih objava

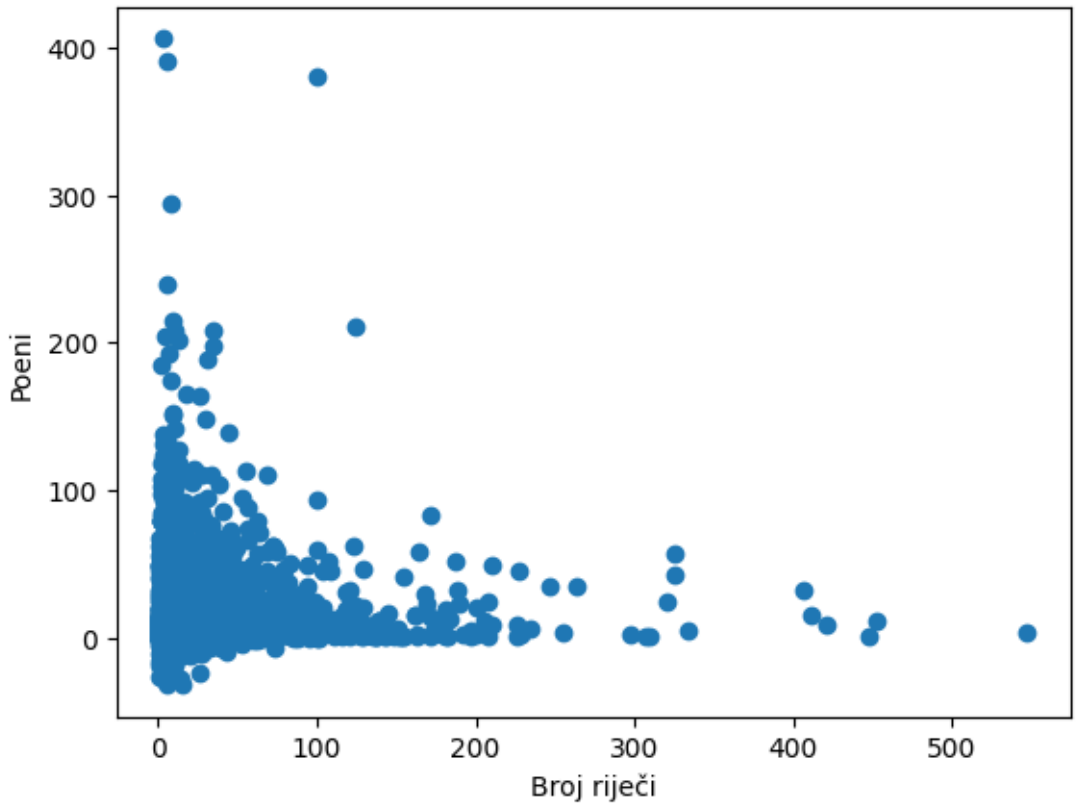
Slika 6: Najčešće riječi iz skupa neutralno nastrojenih objava

Slika 7: Najčešće riječi koje se nalaze isključivo u skupu neutralno nastrojenih objava

Slika 8: Najčešće riječi iz skupa pozitivno nastrojenih komentara

Slika 9: Najčešće riječi koje se nalaze isključivo u skupu pozitivno nastrojenih objava

Uz to, kreiramo i graf korelacije između broja tokena (riječi) i bodova. Na ovom je grafu svaka točka jedna objava, na x osi je broj riječi, dok je na y osi broj bodova. Vrijednost korelacije u našem slučaju iznosi 0.03028133, što nam ukazuje da najvjerojatnije ne postoji korelacija između broja riječi u objavi i broja bodova koje su drugi korisnici dodijelili objavi.



Slika 10: Graf korelacije

4. Klasifikacija

Za klasifikaciju je korišteno nekoliko različitih modela za strojno učenje. Prije nego modeli kreću učiti iz skupa podataka, podatke predstavljamo modelom „vreće riječi” (*eng. bag of words*), za što koristimo Pythonov sklearn[7] modul. Njime se riječi iz skupa pretvaraju u vektore brojeva kako bi ML algoritmi mogli raditi s takvim skupom.

Skup podataka se dijeli na skup za učenje i skup za testiranje. Skup za učenje je potreban kako bi se njime model trenirao, te da se na temelju skupa za testiranje aproksimira točnost modela. U našem slučaju je većina skupa (70%) dodijeljena na treniranje, jer skup nije balansiran – većinu objava čine neutralno nastrojeni tekstovi, dok su pozitivni i negativni u manjini.

Nebalansiran skup se može pomoći izbalansirati tzv. klasnim težinama (*eng. class weights*). Klasama koje imaju manji broj pojavljivanja u skupu se dodjeljuje određena „težina”, odnosno viši stupanj važnosti, kako ML model ne bi prioritizirao većinsku klasu zbog njene više učestalosti. Za evaluaciju svakime od modela strojnog učenja korištena je metoda balansiranja iz Python modula Imbalanced-Learn[], a za *random forest* i *support vector machine* modele korištene su i klasne težine. Usporedbe radi, evaluacija je provedena i nad skupom podataka bez prethodnog balansiranja.

Matrica zabune (*eng. confusion matrix*) je matrica čiji elementi označavaju broj puta koliko je predviđanje modela za svaku klasu bilo točno. Retci prestavljaju točnu, a stupci predviđenu anotaciju. Točno pogođene neutralne, negativne i pozitivne vrijednosti nalaze se na glavnoj dijagonali matrice. Na primjer, koristeći *Naive Bayes* algoritam, neutralnih je točno pogođeno 852, negativnih 116, a pozitivnih 29 (slika 11). Prikaz strukture matrice nalazi se u Tablici 5.

Točni neutralni	Lažni pozitivni	Lažni negativni
Lažni negativni	Točni negativni	Lažni pozitivni
Lažni pozitivni	Lažni negativni	Točni pozitivni

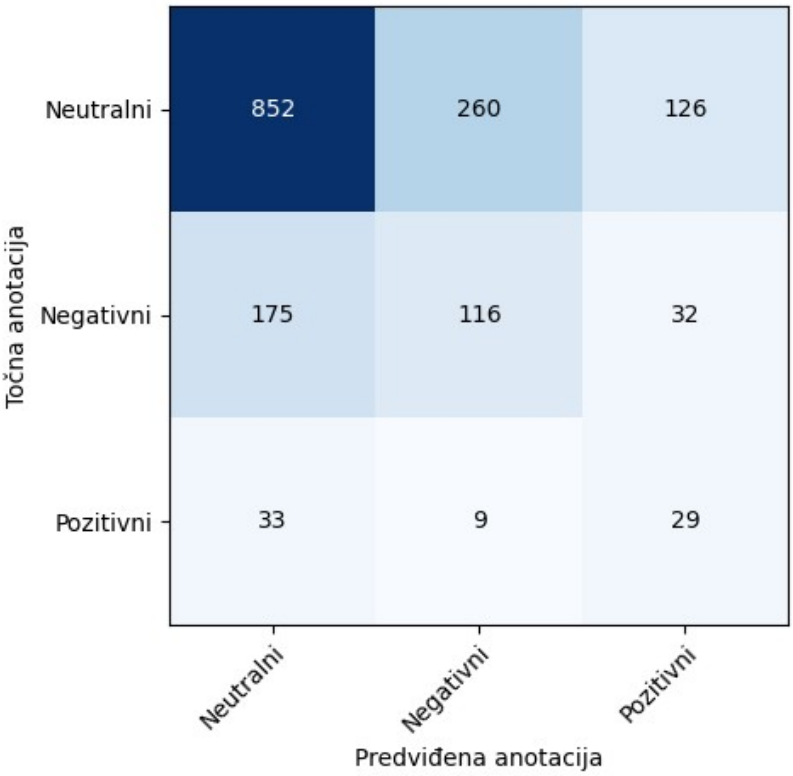
Tablica 5: Matrica zabune

Treniranje modela i izračun matice zabune rezultiraju idućim podacima:

Točnost (*eng. accuracy*)	Omjer točno klasificiranih instanci u odnosu na sveukupan broj predviđenih instanci
Preciznost (*eng. precision*)	Omjer točno pogođenih pozitivnih instanci u odnosu na sve instance predviđene kao pozitivne (i točne i lažne)
Odziv (*eng. recall*)	Omjer tolno pogođenih pozitivnih instanci u odnosu na sve točne pozitivne instance (uniju točno pozitivnih i lažno negativnih)

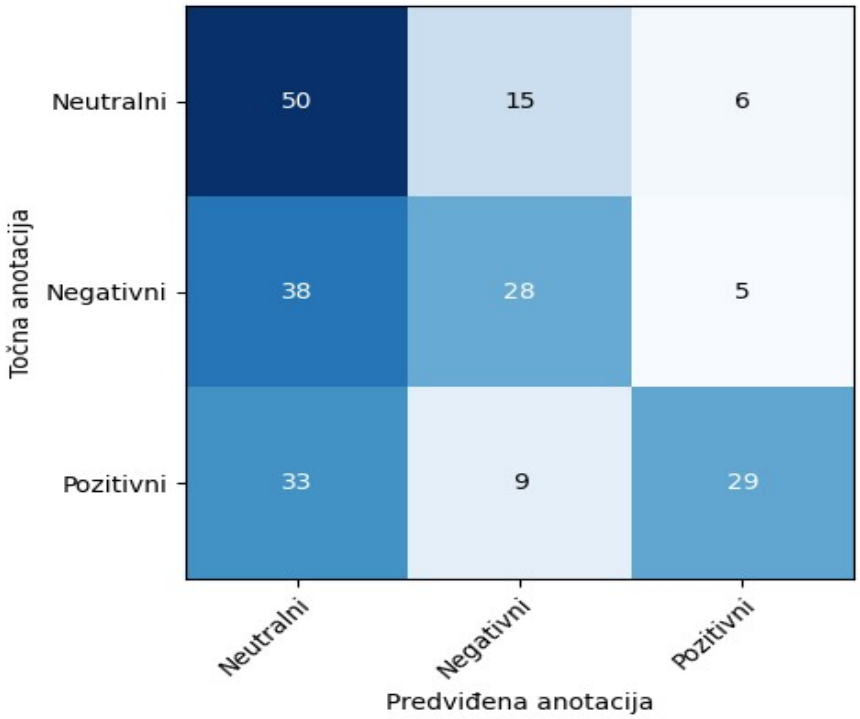
U nastavku su matrice zabune i rezultati evaluacije tri različita modela, sa i bez klasnih težina.

4.1. Naive Bayes



Slika 11: Matrica zabune - Naive Bayes

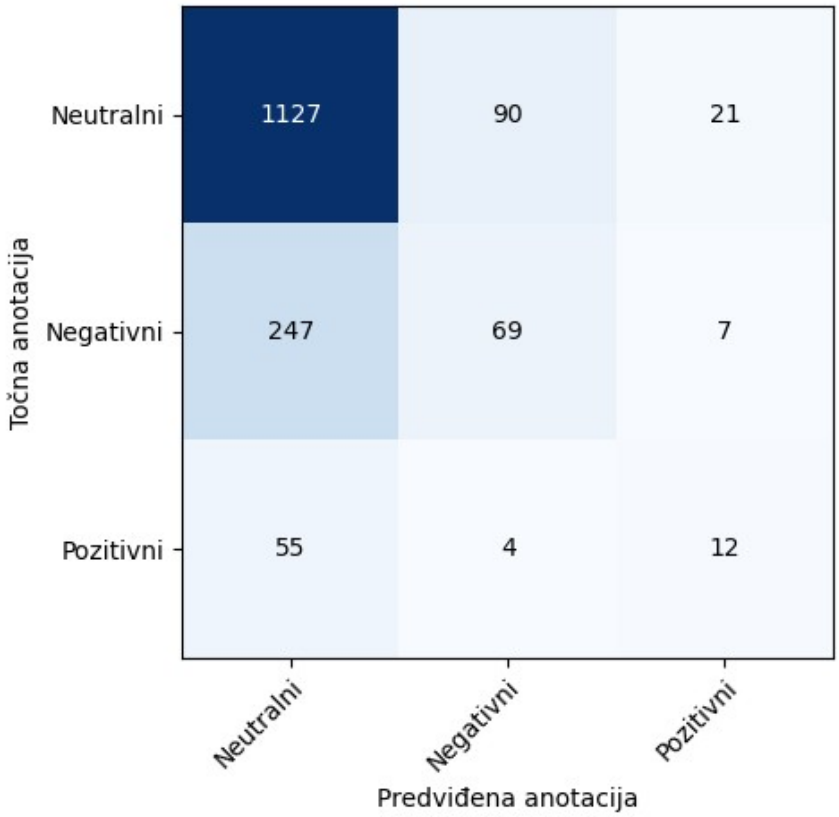
Točnost	0.6109 (61%)
Preciznost	0.6761 (67%)
Odziv	0.6109 (61%)



Slika 12: Matrica zabune - Naive Bayes (balansirane klase)

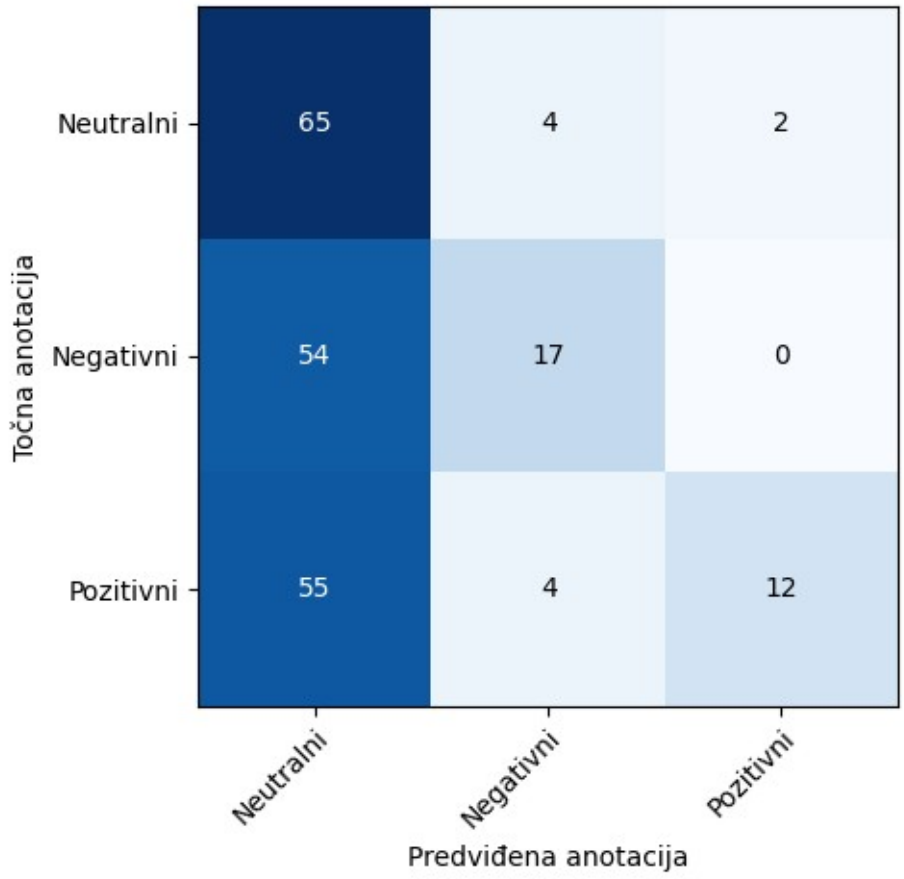
Točnost	0.5023 (50%)
Preciznost	0.5589 (56%)
Odziv	0.5023 (50%)

4.2. Random forest



Slika 13: Matrica zabune - Random forest

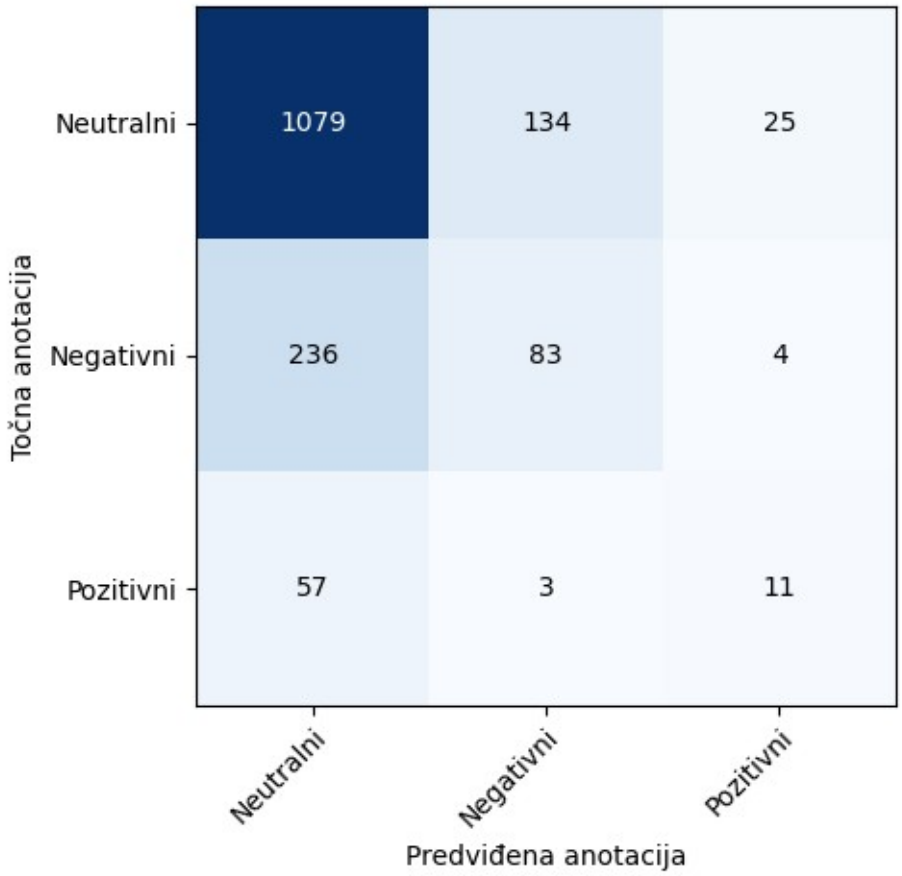
Točnost	0.7402 (74%)
Preciznost	0.6951 (70%)
Odziv	0.7402 (74%)



Slika 14: Matrica zabune - Random forest (balansirane klase)

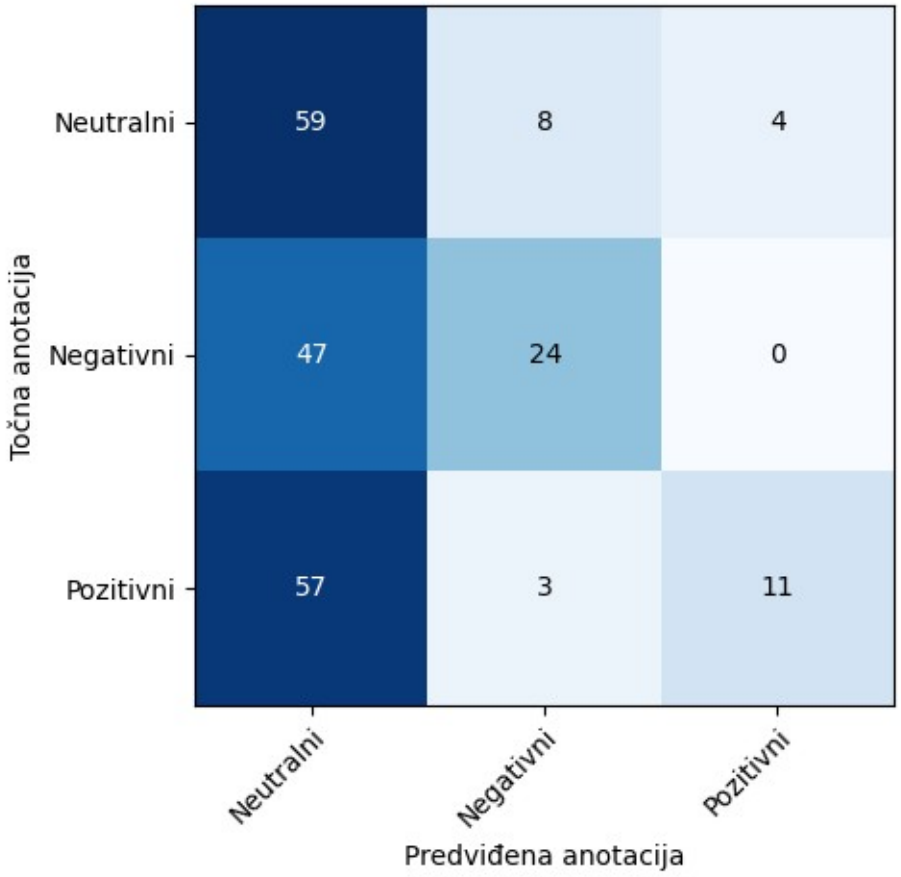
Točnost	0.4413 (44%)
Preciznost	0.6369 (64%)
Odziv	0.4413 (44%)

4.3. Support Vector Machine



Slika 15: Matrica zabune - Support Vector Machine

Točnost	0.7187 (72%)
Preciznost	0.6832 (68%)
Odziv	0.7187 (72%)



Slika 16: Matrica zabune - Support Vector Machine (balansirane klase)

Točnost	0.4413 (44%)
Preciznost	0.5937 (59%)
Odziv	0.4413 (44%)

5. Zaključak

U ovom radu proveli smo klasifikaciju Reddit objava pozitivnog, negativnog i neutralnog sentimenta korištenjem tri algoritma: *Naive Bayes, Random Forest i Support Vector Machine,* svakog od njih koristeći sa balansiranim i nebalansiranim skupovima podataka.

Rezultati klasifikacije pokazali su da je, od tri korištena algoritma, najveću efikasnost postigao *Random Forest.* Njegov rezultirajući model postiže točnost od 74%, preciznost od 70%, i odziv od 74%. S obzirom na nebalansiranost dataseta, u kojem su tri četvrtine objava označene neutralnima, taj postotak preciznosti je zadovoljavajuć.

Uz to je provedena i statistička analiza, kojom smo dobili uvid u nekoliko zanimljivosti. Motrenjem wordclouda zamjećujemo da su tokeni poput "zarazen", "virus", "koron", "mask" među najprevalentnijim tokenima, ako ignoriramo jezične uobičajenosti poput "ov", "im", ili "kad".
Također vidimo da u pravilu objave ne dobivaju mnogo bodova, jer je prosjek "upvoteova" svega 12.43. Na grafu korelacije broja riječi i bodova zamjećujemo da najveći broj bodova u pravilu stječu najkraće objave.

6. Popis literature

A. Rajaraman and J. D. Ullman, „Data Mining”, p. 1–17. Cambridge University Press, 2011.

Hastie, Trevor et al. „The Elements of Statistical Learning: Data Mining, Inference, and Prediction" Vol. 2. New York: Springer, 2009.

McKinney, Wes. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2017.

Krivičić, Armin; Sanda Martinčić-Ipšić. "Analyzing Sentiment of Reddit Posts for the Russia-Ukraine War." 2023 46th MIPRO ICT and Electronics Convention (MIPRO). IEEE, 2023.

[1] https://www.reddit.com/r/croatia/

[2] https://praw.readthedocs.io/en/stable/index.html

[3] https://github.com/pushshift/api

[4] http://nlp.ffzg.hr/resources/tools/stemmer-for-croatian/

[5] https://matplotlib.org/

[6] https://scikit-learn.org/stable/index.html

[7] https://imbalanced-learn.org/stable/