

Detekcija širenja lažnih vijesti vezanih uz COVID-19

Jašarević, Arijan

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:893636>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-08-08**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Arijan Jašarević

Detekcija širenja lažnih vijesti vezanih uz COVID-19

Diplomski rad

Mentor: Izv. prof. dr. sc. Ana Meštrović

Rijeka, lipanj 2021.

Zadatak diplomskog rada



Rijeka, 15.6.2021.

Zadatak za diplomski rad

Pristupnik: Arijan Jašarević

Naziv diplomskog rada: Detekcija širenja lažnih vijesti vezanih uz COVID-19

Naziv diplomskog rada na eng. jeziku: Detection of COVID-19 fake news spreading

Sadržaj zadatka:

Pandemija uzrokovana koronavirusom donosi brojne izazove i opasnosti koji zadiru u različite aspekte ljudskog života. Jedna od opasnosti je širenje lažnih vijesti u medijima i infodemija koja je popratila pandemiju COVID-19. Danas postoje brojni algoritmi koji omogućavaju automatsko otkrivanje lažnih vijesti. Zadatak diplomskog rada je istražiti dosadašnje radove iz tog područja, analizirati postojeće algoritme za otkrivanje širenja lažnih vijesti, te primijeniti i usporediti algoritme na zadanom skupu tekstova vezanih uz domenu koronavirusa.

Mentor:

Izv. prof. dr. sc. Ana Meštrović

Voditelj za diplomske radove:

Izv. prof. dr. sc. Ana Meštrović

Zadatak preuzet: 25.6.2021.

(potpis pristupnika)

Sažetak

U diplomskom radu čiji je naziv „Detekcija širenja lažnih vijesti vezanih uz COVID-19“ govori se o problemu širenja lažnih vijesti vezanih uz COVID-19. Prvo je dat uvod u područje lažnih vijesti i područje vezano uz COVID-19. Zatim je pružen pregled dosadašnje literature koji se prvo fokusira općenito na temu širenja lažnih vijesti, a zatim i na širenje lažnih vijesti vezanih uz koronavirus. Nakon toga opisani su alati i algoritmi strojnog učenja koji se koriste u praktičnom dijelu izrade diplomskog rada te je naposljetku prikazan i tijek izrade praktičnog dijela rada te njegovi rezultati.

Ključne riječi : *COVID-19, koronavirus, lažne vijesti, dezinformacije, algoritmi za detekciju lažnih vijesti, matrica konfuzije, Logistička regresija, Stohaistički gradijenti silazak, Stablo odlučivanja, SVM, Naivni-Bayes, KNN, Slučajna šuma, AdaBoost, Podizanje gradijenata, Točnost, Preciznost, Odziv, F1*

Abstract

This graduate thesis which title is „Detection of COVID-19 fake news spreading“ deals with the problem of spreading fake news related to COVID-19. First, an introduction to the area of fake news and the area related to COVID-19 is provided. After that comes review of the literature related to the topic, focusing first on the area of spreading fake news and then on the area of spreading fake news related to coronavirus. After that, the tools and algorithms of machine learning used in the practical part of the thesis are described, and in the end the workflow of the practical part of the thesis and its results are presented.

Keywords: COVID-19, coronavirus, fake news, misinformation, fake news detection algorithms, confusion matrix, Logistic regression, Stochastic Gradient Descent, Decision Tree, SVM, Naive-Bayes, KNN, Random Forest, AdaBoost, Gradient Boosting, Accuracy, Precision, Recall, F1

SADRŽAJ

1.Uvod	7
2.Pregled dosadašnjih istraživanja	8
2.1.Analiza širenja lažnih vijesti.....	8
2.1.1.Lažne vijesti	8
2.1.2.Infodemija COVID-19.....	10
2.2.Sustavi za otkrivanje lažnih vijesti	11
2.2.1.Pregled sustava za otkrivanje lažnih vijesti u različitim domenama	11
2.2.2.Pregled sustava za otkrivanje lažnih vijesti vezanih uz pandemiju COVID-19	13
2.3.Opis metodologije	17
2.3.1.Algoritmi strojnog učenja	17
2.3.2.Mjere vrednovanja klasifikacije.....	29
2.3.3.Alati	33
2.3.4.Biblioteke koje se koriste	34
3.Praktični dio.....	36
3.1.Statistika skupa podataka.....	36
3.1.1.Instalacija i pokretanje jupyter notebook-a	36
3.1.2.Statistika	37
3.2.Algoritmi strojnog učenja	45
3.2.1.Funkcija za crtanje matrice konfuzije	45
3.2.2.Logistička regresija	48
3.2.3.Stohastički gradijentni silazak	49
3.2.4.Stablo odlučivanja	50
3.2.5.SVM	51
3.2.6.Naivni-Bayes	52
3.2.7.KNN.....	54
3.2.8.Slučajna šuma.....	55
3.2.9.Adaboost	56
3.2.10.Podizanje gradijenata.....	57
3.2.11.Podizanje gradijenata temeljeno na histogramu	58
4.Rezultati.....	59
5.Tablica rezultata	65
5.Zaključak.....	66
6.Literatura	68

7. Popis slika	72
8. Popis tablica	73

1.Uvod

Razne društvene mreže i novinski portali razvili su se zahvaljujući Internetu, te su pridonijeli uklanjanju vremenskih i geografskih prepreka, koje su kočile razmjenu informacija i povezivanje ljudi širom svijeta. Međutim, unatoč svim svojim prednostima, brža sredstva komunikacije i razmjene informacija također su rezultirala velikim širenjem lažnih vijesti.

Svijet trenutno prolazi kroz smrtonosnu pandemiju virusa COVID-19, a lažne vijesti o bolesti, njezinim lijekovima, prevenciji i uzrocima emitirane su širom svijeta milijunima ljudi. Prethodne 2020. godine vijesti vezane uz pandemiju COVID-19 dominirale su u medijima, kako u Hrvatskoj, tako i u inozemstvu. Uz povećanu pozornost koju je pandemija koronavirusa privukla na sebe, paralelno je došlo i do eskalacije širenja lažnih vijesti vezanih za COVID-19 putem Interneta. Širenje lažnih vijesti vezanih uz koronavirus tijekom takvog nesigurnog vremena može imati ozbiljne posljedice što dovodi do raširene panike i pojačavanja prijetnje pandemije, pa čak i do kobnih posljedica kao što je smrt pojedinaca. Baš zbog toga je od presudne važnosti ograničiti širenje lažnih vijesti i osigurati da se točne informacije prenose javnosti.

Upravo ovdje važnu ulogu imaju algoritmi za detekciju širenja lažnih vijesti koji pomažu u detekciji i sprečavanju širenja lažnih vijesti vezanih uz COVID-19. Korištenjem algoritama strojnog učenja lažne vijesti se mogu lako i automatski otkriti. Jednom kada netko objavi lažne vijesti, algoritmi strojnog učenja provjerit će sadržaj tog posta i detektirati će ga kao lažne vijesti [1]. Ovdje je veoma bitno uzeti u obzir točnost algoritma, jer ako algoritam nije uspio otkriti lažne vijesti, onda potencijalno može imati štetne posljedice na živote ljudi. Točnost algoritma najviše ovisi o procesu treniranja, stoga model koji je dobro treniran daje veću točnost. Dostupni su različiti algoritmi strojnog učenja koji se mogu koristiti za otkrivanje lažnih vijesti, a u ovome se diplomskom radu koriste sljedeći: „Logistička regresija“, „Stohastički gradijenti silazak“, „Stablo odlučivanja“, „SVM“, „Naivni Bayes“, „K-Najbližih-Susjeda“, „Slučajna šuma“, „AdaBoost“, „Podizanje gradijenata“, te „Podizanje gradijenata temeljeno na histogramu“.

Ovdje također treba spomenuti značaj koji umjetna inteligencija(AI) i računalna analiza prirodnog jezika(NLP) imaju u doba pandemije koronavirusa. Jay Carriere i suradnici su proveli istraživanje o upotrebi AI-a i NLP-a za pomoć u procjeni, dijagnozi i liječenju akutnih i kroničnih mišićno-koštanih, neuroloških i drugih stanja tijekom pandemije COVID-19. Naveli su i neke konkretne primjere uporabe tih tehnologija na područja kao što su analiza slike u medicinske svrhe, robotska rehabilitacija i procjena te NLP sustavi koji omogućuju daljinsko upravljanje, rukovanje, liječenje i procjenu osoba s akutnim i kroničnim stanjima [2].

Tema ovoga diplomskog rada vezana je upravo uz širenje lažnih vijesti vezanih uz COVID-19. Na početku će biti obrađeni ključni pojmovi kao što su lažne vijesti i COVID-19. Zatim će biti dat pregled dosadašnjih radova vezanih uz ovu temu, te će naposljetku biti data usporedba prethodno navedenih algoritama za detekciju lažnih vijesti na odabranom skupu podataka.

2.Pregled dosadašnjih istraživanja

Ovaj dio rada obuhvaća: pregled teorije potrebne za razumijevanje rada, pregled literature koja se bavi tim problemom, te opis metodologije koja će se koristiti.

2.1.Analiza širenja lažnih vijesti

Ovo poglavlje sastoji se od poglavlja: „Lažne vijesti“ i „COVID-19“, u kojima je dat teorijski uvod u ta područja.

2.1.1.Lažne vijesti

Lažne vijesti su bilo koji članci, objave ili videozapisi koji sadrže neistinite informacije, a predstavljaju se kao vjerodostojan izvor informacija [3]. Lažne vijesti mogu se podijeliti u dvije kategorije: "lažne vijesti" uz implicitno razumijevanje čitatelja da je sadržaj lažan i "lažne vijesti" kod kojih čitatelji nisu svjesni da su informacije lažne. Nedavna istraživanja pokazuju kako većina lažnih vijesti vezanih uz COVID-19 spadaju u kategoriju koja podrazumijeva da čitatelji nisu svjesni da su informacije lažne što je svakako razlog za uzbunu [4]. Ova se kategorija može dalje podijeliti u dvije skupine: dezinformacije koje se odnose na „nenamjerno širenje lažnih informacija“ i dezinformacije koje se odnose na „namjerno stvaranje i širenje informacija za koje se zna da su lažne“. Na slici ispod možemo vidjeti smjernice koje nam ukazuju na koje sve načine možemo prepoznati lažne vijesti: „razmisli o izvoru“, „pročitaj sve“, „provjeri autora“, „dodatni izvori“, „provjeri datum objave“, „je li to zafrkancija?“, „osvijesti svoje stavove“ te naposljetku „pitaj stručnjake“.

KAKO PREPOZNATI LAŽNE VIJESTI



RAZMISLI O IZVORU

Dobro prouči stranicu, istraži njezinu svrhu i podatke o kontaktu.



PROČITAJ SVE!

Senzacionalistički naslovi privlače klikove. O čemu je zapravo riječ?



PROVJERI AUTORA

Na brzinu provjeri autora! Može li mu se vjerovati i je li to stvarna osoba?



DODATNI IZVORI

Klikni na ponuđene linkove i dodatne izvore. Provjeri sadrže li informacije koje zaista podupiru vijest.



PROVJERI DATUM OBJAVE

Ako je nešto nedavno objavljeno, ne znači da je nova vijest.



JE LI TO ZAFRKANCIJA?

Ako je previše neobično i čudno, možda je šala. Istraži stranicu i autora da bi bio siguran.



OSVIJESTI SVOJE STAVOVE

Uzmi u obzir da bi tvoji stavovi mogli utjecati na prosudbu.



PITAJ STRUČNJAKE

Pitaj knjižničara ili provjeri informacije na *fact-checking* stranicama.

Prijevod: Hrvatsko knjižničarsko društvo

IFLA

International Federation of Library Associations and Institutions

With thanks to www.factcheck.org

Slika 1. Na koje sve načine možemo prepoznati lažne vijesti?

(Izvor:[61])

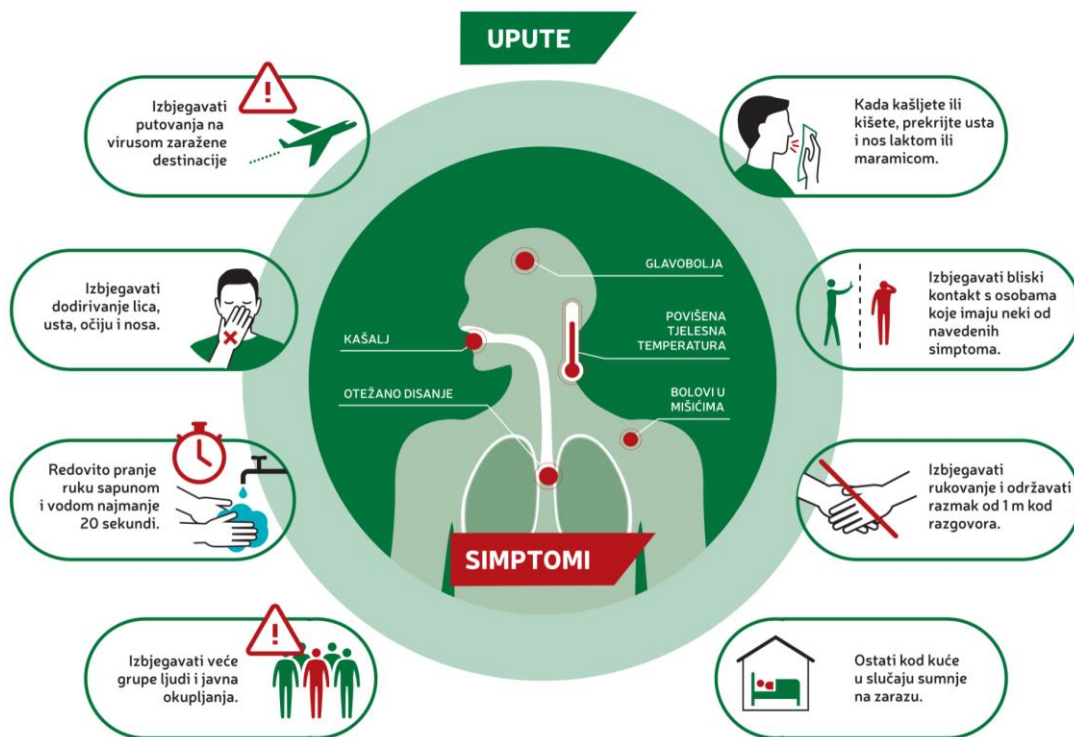
2.1.2. Infodemija COVID-19

Krajem prosinca 2019. Svjetska zdravstvena organizacija (eng. „World Health Organization“ ili „WHO“) je obaviještena o nizu slučajeva upale pluća nepoznatog uzroka koji su otkriveni u gradu Wuhan, koji se nalazi u provinciji Hubei, unutar države Kine. U početku su ti pacijenti dijagnosticirani simptomima akutne upale pluća. Većina njih radila je na tržnici u Wuhanu i pokazivala je uobičajene simptome kao što su suhi kašalj, umor i u težim slučajevima poteškoće s disanjem. Međutim, ti simptomi nisu bili posljedica akutne upale pluća kao što se prvo mislilo. S obzirom na sve veći broj slučajeva, Kina je početkom siječnja 2020. obavijestila WHO o situaciji i njenom nepoznatom uzroku [5].

WHO je virus nazvao „Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)“ i bolest kao „Coronavirus Disease (COVID-19)“. COVID-19 globalni je zdravstveni problem koji zahtijeva krajnji oprez, strogo održavanje osobne i opće higijene te čistoću svih mjesta. Ove prakse pomažu u izbjegavanju pojave mutacija kako bi se virus mogao kontrolirati i zadržati [6]. Do 2. svibnja 2021. COVID 19 proširio se u 222 države te pritom inficirao 153 milijuna od kojih je potvrđeno više od 3 milijuna smrti [7]. U veljači 2020. generalni direktor WHO-a Tedros Adhanom rekao je „Ne borimo se samo protiv epidemije; borimo se protiv infodemije. Lažne vijesti šire se brže i lakše od ovog virusa i jednako su opasne.“ [8]. Sylvie Briand, arhitektica strategije WHO-a za suzbijanje rizika širenja lažnih vijesti, tvrdi da se „s društvenim mrežama ovaj se fenomen pojačava, ide brže i dalje, poput virusa koji putuju s ljudima“ [9].

Infodemija se opisuje kao prekomjerno gomilanje informacija i dezinformacija koje ometaju pronalazak pouzdanih izvora i pouzdanih činjenica [10]. Infodemiologija (nastala prema riječima informacija i epidemiologija) se definira kao znanost o distribuciji informacija u elektroničkim medijima ili u populaciji s ciljem informiranja o javnom zdravstvu i politici [11]. Krajnji cilj istraživanja infodemiologije je razviti, prikupiti i procijeniti metriku, metode i okvire za praćenje i upravljanje infodemijskim pojavama [12].

Ova infodemija dovela je do ubrzanja širenja virusa, povećanja nasilja nad određenim skupinama pa čak i dovela do nanošenja tjelesnih ozljeda. Lažne vijesti vezane uz COVID-19 mogu izravno ugroziti živote. Tako neke lažne vijesti predlažu štetne „lijekove“, poput pijenja aditiva za riblje akvarije, izbjeljivača ili kravljeg urina [13]. Nadalje, postoji prijetnja da lažne vijesti povećaju postotak ljudi koji će odbiti cjepivo. Već postoji zajednica protiv cijepljenja vezana za COVID-19, kojoj se svaki dan pridružuje sve veći broj ljudi [14]. Šire se i lažne vijesti da su ljudi s tamnom kožom možda imuni na COVID-19, što je u SAD-u, početkom travnja 2020., dovelo do toga da se približno 70% smrtnih slučajeva u Chicagu i Louisiani odnosilo na Afroamerikance, koji čine samo otprilike 30% stanovništva [15]. Na slici ispod su prikazani simptomi koje osobe ima ukoliko je zaražena koronavirusom, te upute kojih bi se u tom slučaju striktno trebala držati.



Slika 2. Upute u slučaju zaraze koronavirusom

(Izvor: [62])

2.2.Sustavi za otkrivanje lažnih vijesti

2.2.1.Pregled sustava za otkrivanje lažnih vijesti u različitim domenama

Većina sustava za otkrivanje lažnih vijesti primjenjuje tehnike strojnog učenja kako bi se pomoglo korisnicima u klasificiranju jesu li podaci koje čitaju lažni ili ne. Ova se klasifikacija vrši usporedbom danih podataka s nekim poznatim korpusima koji sadrže obmanjujuće i neistinite podatke. Kod primjene tehnika strojnog učenja u izradi modela detekcije lažnih vijesti, svi podaci moraju proći kroz ove faze: priprema podataka i pred-procesiranje, ekstrakcija značajki te odabir i izrada modela [16]. Prethodno navedene faze olakšavaju rukovanje velikom količinom podataka koji su potrebni za izradu modela detekcije. Postoji veliki broj web stranica koje se mogu koristiti za traženje unaprijed provjerenih podataka (npr. Snopes.com, PolitiFact.com, Factcheck.org). Međutim, ove web stranice uglavnom se temelje na ljudskome faktoru, što znači da se analiza podataka vrši ručno. Tu analizu izvode stručni analitičari koji su blisko upoznati s predmetnim kontekstom. Ručni pristup je spor, skup, vrlo subjektivan i pristran, te je postao nepraktičan zbog ogromne količine dostupnih podataka na društvenim mrežama [17]. Upravo zbog svega toga se postupak automatizirane klasifikacije podataka formirao kao jedan od zanimljivijih i produktivnijih područja proučavanja.

Do danas su predloženi mnogi automatizirani sustavi za otkrivanje lažnih vijesti. Kaliyar i Singh napravili su sveobuhvatno istraživanje detekcije lažnih vijesti na različitim platformama društvenih mreža [18]. Zhangand Ghorbani predstavio je sveobuhvatan pregled nedavnih pronalaska vezanih uz lažne vijesti [19]. U tome radu dana je karakterizacija utjecaja lažnih vijesti na društvenim mrežama, te su predstavljene najmodernije metode otkrivanja i dan je pregled najčešće korištenih skupova podataka koji se koriste za izgradnju modela za klasifikaciju lažnih vijesti. Collins i Erascu također su dali izvanredan pregled raznih modela o otkrivanju lažnih vijesti te su otkrili da tehnike koje kombiniraju ljude i strojeve donose dosta bolje rezultate u usporedbi sa sustavima koji ovise samo o jednom od navedenih faktora [20].

Al Asaadet i suradnici su predložili model provjere vjerodostojnosti vijesti koji kombinira nekoliko tehnika strojnog učenja za klasifikaciju teksta. Učinkovitost modela je testirana na skupu podataka koji je sadržavao lažne i stvarne vijesti koristeći pritom algoritme: „Naive Bayes“ i „Lagrangeov Support Vector Machine“ [21]. Nakamura i kolege su predložili hibridni sustav za otkrivanje lažnih vijesti koji koristi algoritam multinomialnog glasanja. Potom su sustav testirali na višestrukim skupovima podataka, koristeći pet algoritama strojnog učenja: „Naivni-Bayes“, „Slučajna šuma“, „Stablo odluke“, „SVM“ i „K-Najbližih Susjeda“ [22]. Li i Ibrishimova su proučavale različite definicije lažnih vijesti te su predložile definiciju koja se temelji na apsolutnoj činjeničnoj točnosti i relativnoj pouzdanosti izvora. Osim toga, predložili su i „framework“ za otkrivanje lažnih vijesti, koji koristi ručnu i automatiziranu provjeru znanja i stilske značajke [23].

Posadas-Durán et al. i suradnici su predstavili novu metodu za otkrivanje lažnih vijesti na društvenim mrežama. Prikupili su skup podataka koji je sadržavao vijesti na španjolskom jeziku izvučenih s nekoliko web stranica. Podaci korpusa označeni su kao lažni ili istiniti u svrhu učenja nadziranih modela za otkrivanje lažnih vijesti. Također, svakom primjeru skupa podataka dodijeljena je kategorija vijesti: znanost, sport, obrazovanje, ekonomija, politika, zabava, zdravlje, sigurnost ili društvo. Svoju metodu testirali su koristeći četiri algoritma strojnog učenja: „Support Vector Machine“ s linearnim jezgrom, „Logistic Regression“, „Random Forest“ i „XG Boosting“ na predloženom skupu podataka. Rad je kao klasifikacijsku mjeru uzeo mjeru „Preciznosti“, te je od testiranih algoritama najbolji rezultat imao SVM s preciznošću od 73.22% [24].

Wang je predstavio novi skup podataka za lažne vijesti. Skup podataka se sastojao od 12,8 tisuća ručno označenih kratkih izjava u različitim kontekstima sa stranice PolitiFact.com. Značajno je da je ovaj skup podataka smatran prvim velikim skupom podataka koji se odnosi na otkrivanje lažnih vijesti, te je za red veličine veći od prethodnih javnih podataka o lažnim vijestima [25]. Thorneet i suradnici su predložili još jedan skup podataka koji je prikupljen sa Wikipedije. Ovaj skup podataka uglavnom služi aplikacijama koje su vezane uz detekciju nečijeg stava. Sadrži oko 185.000 dokumenata s istinitim tekstom, odnosno članaka sa Wikipedije, dok se lažni dio vijesti prikupljao putem publike [26]. Asr i Taboada pregledali su dostupne skupove podataka za otkrivanje lažnih vijesti i uveli repozitorij pod nazivom „MisInfoText“ kako bi se riješio problem skupova podataka koji nemaju pouzdane oznake. Spremište MisInfoText sadrži tri kategorije podataka: veze do svih javno dostupnih

tekstualnih skupova lažnih vijesti, značajke za izravno prikupljanje podataka s web lokacija za provjeru činjenica i skupove podataka [27].

Možemo zaključiti kako su se mnogi prethodni radovi bazirali na izgradnji sustava za otkrivanje lažnih vijesti. Većina ih se oslanja na ručno označene podatke za postupak otkrivanja, kao što je slučaj i u ovome diplomskom radu.

2.2.2. Pregled sustava za otkrivanje lažnih vijesti vezanih uz pandemiju COVID-19

Nakon izbijanja COVID-19 pojavila se ogromna količina literature koja se bavila količinom dezinformacija, vrstom dezinformacija i pristupima borbi protiv dezinformacija. Tako su se neki od tih radova bavili ispitivanjem globalnih trendova na Twitteru po zemljama, analizirajući pritom količinu tweetova prema određenim temama vezanim uz koronavirus, te objava koje su povezane s određenim mitovima koji okružuju virus, a krajnji rezultat bi bio broj tweetova koji sadrže informacije koje se smatraju lažnima. Dok su se drugi radovi fokusirali na pronalaženje znakova upozorenja koji bi mogli biti ključni indikatori da će određena zemlja doživjeti infodemiju. Ono što je svim tim studijama jednako, je to da su sve one pokazale da se lažne vijesti vezane uz COVID-19 razvijaju i šire nevjerojatnom brzinom.

Značajno otkriće koje potkrepljuje postojanje infodemije oko COVID-19 jest činjenica da su web-stranice koje se koriste za provjeru istinitosti činjenica preopterećene. Jedno istraživanje pokazalo je da se nakon izbijanja COVID-19 broj provjera istinitosti činjenica na engleskom jeziku povećao za 900% od siječnja do ožujka 2020. [28]. Unatoč povećanju broja provjera činjenica na takvim web-stranicama, one za to nisu u potpunosti adekvatne, te imaju ograničene resurse i stoga ne mogu provjeriti sav problematičan sadržaj. Pored istraživanja koja karakteriziraju vrste i količine informacija koje se dijele kao vijesti, bilo je i sličnih istraživanja koja karakteriziraju osobine COVID-19 informacija koje se dijele na platformama društvenih mreža, poput Twittera. Jedno je istraživanje otkrilo da su neke ključne riječi povezane s lažnim vijestima. Studija koja je obuhvaćala 673 tweeta, od koji je svaki sadržavao preko 14 hashtagova i ključnih riječi povezanih s epidemijom COVID-19 pokazala je da je 24,8% uzorka uključivalo dezinformacije, 17,4% nepovjerljive informacije, a tweetovi s neprovjerenih Twitter profila sadržavali su najviše lažnih vijesti [29]. Drugi radovi ukazuju na to da se strojno učenje može koristiti za dobivanje informacija o ključnim frazama odnosno značajkama koje koriste ljudi koji raspravljaju o pandemiji koronavirusa [30].

Upravo je u središtu eksplozije dezinformacija prisutnih u infodemiji COVID-19 pitanje o tome što ljude tjera da dijele te lažne vijesti. Tako je jedna studija pokazala da su ljudi dijelili lažne tvrdnje povezane s COVID-19 zato što nisu dovoljno promišljali o tome je li sadržaj točan prije nego što su ga odlučili objaviti. Kada su sudionici na početku studije bili potaknuti na razmišljanje o točnosti vijesti koje šire, broj onih koji su namjeravali podijeliti te članke se smanjio više od dvostruko. Zaključak te studije je bio da poticanje pojedinaca na razmišljanje o tome što dijele može značajno utjecati na njihovo daljnje dijeljenje lažnih vijesti [31]. Ovo istraživanje također ukazuje na to da bi poticanje pojedinaca na razmišljanje o točnosti

sadržaja vijesti moglo poslužiti kao vrlo učinkovit oblik borbe s dezinformacijama, istodobno ne zahtijevajući gotovo nikakvu pomoć računala. To također znači da računalne algoritme za detekciju lažnih vijesti treba pažljivo ispitati kako bi se osiguralo da oni ne daju lažnu sliku istinitosti. Ako algoritam tvrdi da je sadržaj "lažnih vijesti" stvaran, to može smanjiti vjerojatnost da će ga korisnik kritički ispitati, što bi zauzvrat moglo povećati šansu da korisnik dalje dijeli tu lažnu vijest.

Wash Karo je proveo istraživanje koje koristi pristupe koji se baziraju na strojnoj obradi prirodnog jezika i strojnom učenju, kako bi se pružili autentični izvori informacija s dnevnim vijestima vezanim uz COVID-19 na engleskom jeziku. Model pruža autentične vijesti putem provjere sličnosti između novih vijesti i postojećih članaka vijesti u njihovom skupu podataka koje su prethodno potvrđene kao istinite [32]. Jedno istraživanje se bavilo ispitivanjem zlonamjernog COVID-19 sadržaja i otkrilo da se „hateful“ COVID-19 vijesti akumuliraju na platformama koje omogućuju određene značajke zajednice poput Facebook stranica. Studija je također otkrila da se stvarno, rasno motivirano nasilje dogodilo nakon izbijanja COVID-19, sugerirajući potencijalne implikacije tih „hateful“ objava [33]. Potencijalno rješenje ovoga problema je implementacija algoritama otkrivanja osjećaja, odnosno određivanje prevladavajuće emocije u tekstu, korištenjem strojne obrade prirodnog jezika za identificiranje potencijalnih vijesti koje u sebi sadržavaju govor mržnje.

Istraživanja također pokazuju da masivan broj novih vijesti na društvenim mrežama koji se odnose na COVID-19 čini ručnu analizu istinitosti neisplativom, implicirajući pritom kako je ispravan pristup tome korištenje automatiziranih pristupa koji se baziraju na strojnom učenju [34]. Još nije utvrđeno da li bi ciljani oglasi koji daju protu informacije lažnim vijestima, odnosno daju istinite vijesti korisnicima bili učinkoviti u borbi protiv širenja dezinformacija vezanih uz COVID-19. Naravno ono što je jasno je da je gotovo nemoguće ciljati sve „lažne vijesti“ samo jednim od prethodno navedenih pristupa, već bi najbolje bilo kombiniranje istih.

Provedeno je i nekoliko studija koje su se fokusirale na ulogu koju Facebook ima u internetskim raspravama oko COVID-19. Tako je jedno od tih istraživanja proučavalo glasine koje tvrde da je uvođenje 5G tehnologije povezano s izbijanjem pandemije koronavirusa, te je pokazalo da teorije zavjere mogu započeti kao ideje koje se kreću od par pojedinaca, međutim naposljetku dolazi do prelaska na veliku populaciju ljudi, a sve se to odvija na Facebooku [35]. Drugo istraživanje koje je ispitivalo ulogu Facebooka u širenju vijesti vezanih uz koronavirus otkrilo je razne slučajeve dezinformacija koje se tamo šire, od teorija zavjere o napadu bio-oružjem do nepovjerljivih tvrdnji koje su došle od strane političara [36]. Na sljedećoj slici možemo vidjeti jedan primjer oglašavanja lažnih vijesti na Facebooku, koji tvrdi kako socijalno distanciranje uopće ne pomaže u sprečavanju širenja koronavirus. Ono što je ovdje veoma zabrinjavajuće je to što je Facebook prihvatio, odnosno dopustio objavljivanje i širenje tog oglasa.



Slika 3. Lažna vijest vezana uz COVID-19 koja se širila Facebookom

(Izvor: [63])

Također postoji veliki broj istraživanja koja su proučavala ulogu Twittera u razgovorima na društvenim mrežama vezanim uz COVID-19. Jedno istraživanje je proučavalo dvije skupine korisnika koje objavljuju takve vijesti te su ih podijelili na sljedeći način: jednu skupinu činili su korisnici koji aktivno objavljuju dezinformacije dok su drugu skupinu činili korisnici koji šire istinite informacije o koronavirusu. Rezultat toga istraživanja je pokazao kako prva skupina korisnika, odnosno ona koje namjerno i aktivno širi lažne vijesti vezane uz COVID-19, ima mnogo veći broj pripadnika i bolje je organizirana nego druga skupina koja je dijelila istinite vijesti [37]. Studija od 43,3 milijuna tweetova na engleskom jeziku povezanih s COVID-19 pronašla je dokaze o prisutnosti botova u raspravama o COVID-19 na Twitteru. Utvrđeno je da korisnički računi na Twitteru koji su ocijenjeni kao „high bot“ koriste sadržaj i hashtagove koji se odnose na COVID-19, a sve s namjerom širenja raznih vijesti što je zasigurno zabrinjavajuće [38]. Istraživanje koje se bavilo proučavanjem visoko kvalitetnih i nisko kvalitetnih URL-ova podijeljenih na Twitteru vezanih uz COVID-19 otkrilo je da više tweetova sadrži URL-ove s web stranica koje su niske kvalitete i sadrže lažne informacijama u usporedbi s visoko kvalitetnim web stranicama koje sadrže istinite informacije [39]. Uz Twitter bih naveo još jedno istraživanje koje je koristilo skup podataka od 67 milijuna

tweetova od 12 milijuna korisnika, koje je utvrdilo da su većinu utjecajnih tweetova objavili novinski portali te vladini službenici, ali najutjecajnije tweetove ipak su objavili obični ljudi, odnosno svakodnevni korisnici. Također ono što je primijećeno je da postoji veća vjerojatnost da ti obični ljudi šire lažne tweetove, ali također je utvrđeno i da su mnogi od tih korisnika botovi. Na slici ispod možemo vidjeti samo neke primjere lažnih vijesti koje su se širile društvenom mrežom Twitter: „Ljudi s bradom imaju veći rizik infekcije koronavirusom“, „Doći će do nestašice toaletnog papira“, „Vrući napitci i vitamin C sprječavaju zarazu“, „Sunčanje ubija koronavirus“, „Fenovi i češnjak ubijaju koronavirus“, „Trljanje alkohola po tijelu je najbolji način sprječavanja bolesti“, „Jedenje mesa će ojačati vaš imunitet“, „COVID-19 je rezultat svjetske zavjere, „Virus ne može preživjeti temperature veće od 27 celzijevih stupnjeva“.



Slika 4. Neke od lažnih vijesti koje su se širile putem Twittera

(Izvor: [64])

2.3. Opis metodologije

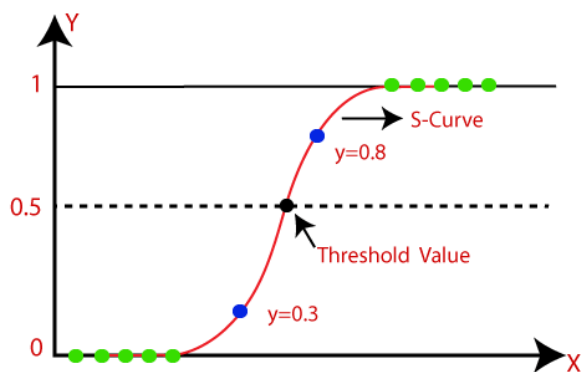
U ovome poglavlju bit će dat detaljan teorijski opis alata, metoda i algoritama koji će se koristiti u izradi praktičnog dijela ovoga diplomskog rada.

2.3.1. Algoritmi strojnog učenja

U ovome poglavlju bit će dat detaljan teorijski opis algoritama strojnog učenja koje će se testirati, odnosno koji će se koristiti pri izradi praktičnog dijela diplomskog rada.

2.3.1.1. Logistička regresija

Logistička regresija je klasifikacijski algoritam koji se koristi za procjenu diskretnih vrijednosti (binarne vrijednosti poput 0/1, da/ne, točno/netočno) na temelju zadanog skupa neovisnih varijabli [40]. Jednostavnim riječima, predviđa vjerojatnost pojave događaja prilagođavanjem podataka logaritamskoj funkciji. Stoga je poznata i kao logaritamska regresija. Budući da predviđa vjerojatnost, izlazne vrijednosti se nalaze u rasponu između 0 i 1. Logistička regresija veoma je slična Linearnoj regresiji, samo što se koriste za različite stvari. Linearna regresija koristi se za rješavanje problema regresije, dok se Logistička regresija koristi za rješavanje problema klasifikacije. U Logističkoj regresiji, umjesto da ugradimo regresijsku liniju, ugrađujemo logističku funkciju u obliku slova "S" koja predviđa dvije maksimalne vrijednosti (0 ili 1). Krivulja iz logaritamske funkcije ukazuje na vjerojatnost nečega, primjerice jesu li stanice kancerogene ili ne, je li miš pothranjen ili ne na temelju njegove težine. Logistička regresija važan je algoritam strojnog učenja jer ima mogućnost pružanja vjerojatnosti i klasifikacije novih podataka pomoću kontinuiranih i diskretnih skupova podataka. Logistička regresija može se koristiti za klasificiranje opažanja pomoću različitih vrsta podataka i lako može odrediti najučinkovitije varijable korištene za klasifikaciju. Sljedeća slika prikazuje logističku funkciju:

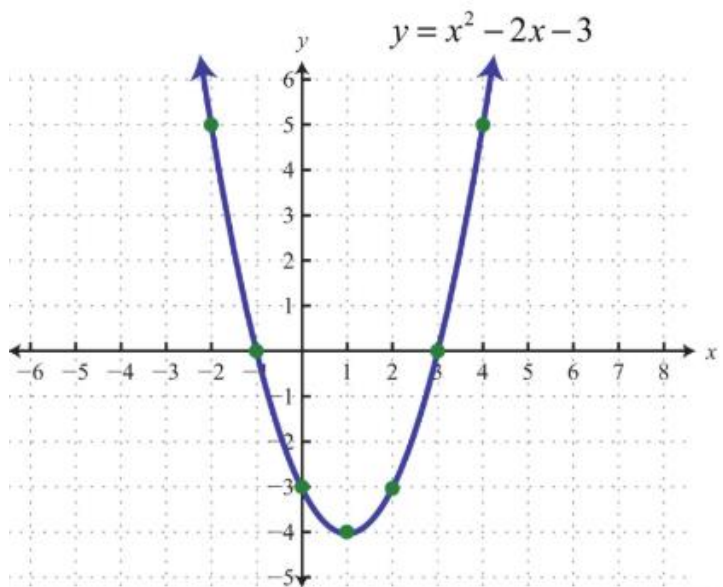


Slika 5. Model Logističke regresije

(Izvor: [40])

2.3.1.2. Stohastički gradijentni silazak

„Stohastički gradijentni silazak“ vrlo je popularan i čest algoritam koji se koristi u raznim algoritmima strojnog učenja, te čini osnovu neuronskih mreža. Gradijent ima značenje nagib ili nagib površine. Dakle, gradijentni silazak doslovno znači spuštanje niz padinu da bi se došlo do najniže točke na toj površini. Zamislimo dvodimenzionalni graf, poput parabole na koja je prikazana na donjoj slici:



Slika 6. Funkcije parabole s parametrima x i y

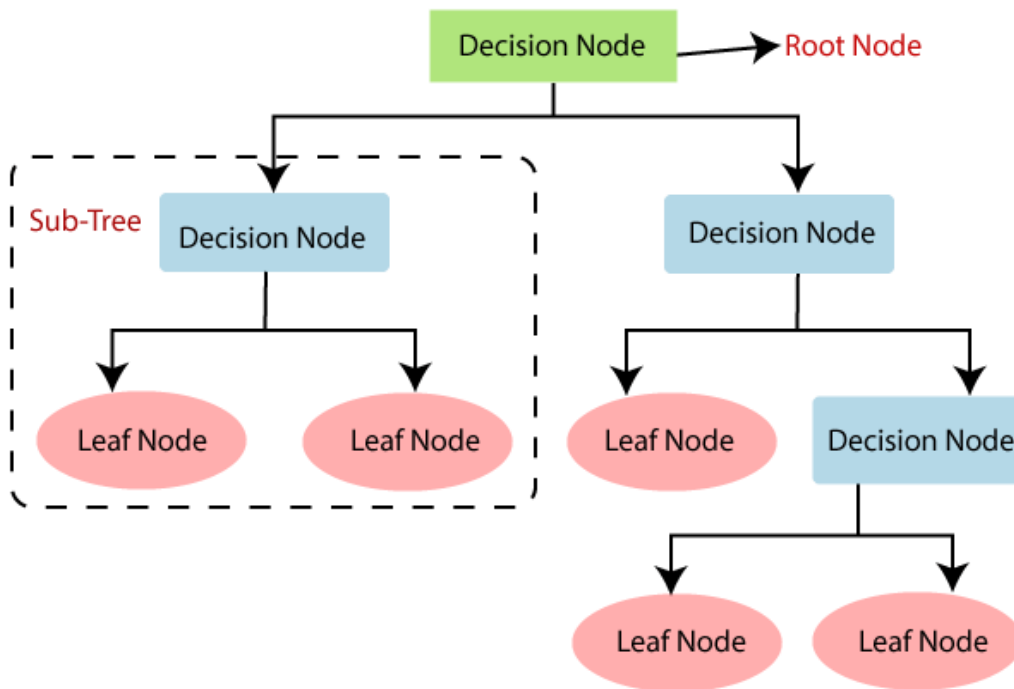
(Izvor: [41])

Na gornjem grafu, najniža točka na paraboli javlja se pri $x = 1$. Cilj algoritma gradijentnog silaska je pronaći vrijednost "x" tako da je "y" minimalan. Vrijednost "y" se ovdje označava kao ciljna funkcija na kojoj djeluje algoritam gradijentnog silaska, kako bi se spustila na najnižu točku. Gradijentni silazak stoga je iterativni algoritam koji započinje od slučajne točke na funkciji i u koracima putuje niz njezin nagib dok ne dosegne najnižu točku te funkcije" [41]. Ovaj je algoritam koristan u slučajevima kada se optimalne točke ne mogu pronaći izjednačavanjem nagiba funkcije na 0. U slučaju linearne regresije zbroj preostalih ostataka možemo mentalno mapirati kao funkciju "y", a vektor težine kao "x" u gornjoj paraboli.

„Stohastički gradijentni silazak“ je vrsta algoritma gradijentnog silaska koji slučajno odabire jednu podatkovnu točku iz cijelog skupa podataka pri svakoj iteraciji kako bi uvelike smanjio potrebne računalne kalkulacije. Također je uobičajeno uzorkovanje malog broja podatkovnih točaka umjesto samo jedne točke u svakom koraku, a to se naziva gradijentni spust ili "mini-batch". „Mini batch“ pokušava uspostaviti ravnotežu između dobrog uspona i brzine stohastičkog gradijentnog silazaka.

2.3.1.3.Stablo odlučivanja

Algoritam stabla odlučivanja pripada obitelji nadziranih algoritama učenja. Za razliku od ostalih nadziranih algoritama učenja, algoritam stabla odlučivanja može se koristiti i za rješavanje problema regresije, kao i za rješavanje problema klasifikacije. Algoritam je strukturiran po stablu, gdje unutarnji čvorovi predstavljaju značajke skupa podataka, grane predstavljaju pravila odluke i svaki čvor lista predstavlja rezultat [42]. U stablu odlučivanja postoje dva čvora, a to su čvor odlučivanja i čvor lista. Čvorovi odlučivanja koriste se za donošenje bilo koje odluke i imaju više grana, dok su čvorovi lista izlaz tih odluka i ne sadrže daljnje grane. Naziv je dobio zato jer slično stablu, započinje korijenskim čvorom koji se širi na daljnje grane i gradi strukturu nalik stablu. Cilj korištenja stabla odlučivanja je stvoriti model za treniranje koji se može koristiti za predviđanje klase ili vrijednosti ciljne varijable učenjem jednostavnih pravila odlučivanja zaključenih iz podataka koji su se koristili za treniranje. Za predviđanje oznake razreda za zapis započinjemo od korijena stabla, te se zatim uspoređuju vrijednosti korijenskog atributa s atributom zapisa. Na temelju usporedbe pratimo granu koja odgovara toj vrijednosti i prelazimo na sljedeći čvor. Slika ispod prikazuje generalnu strukturu stabla odlučivanja:

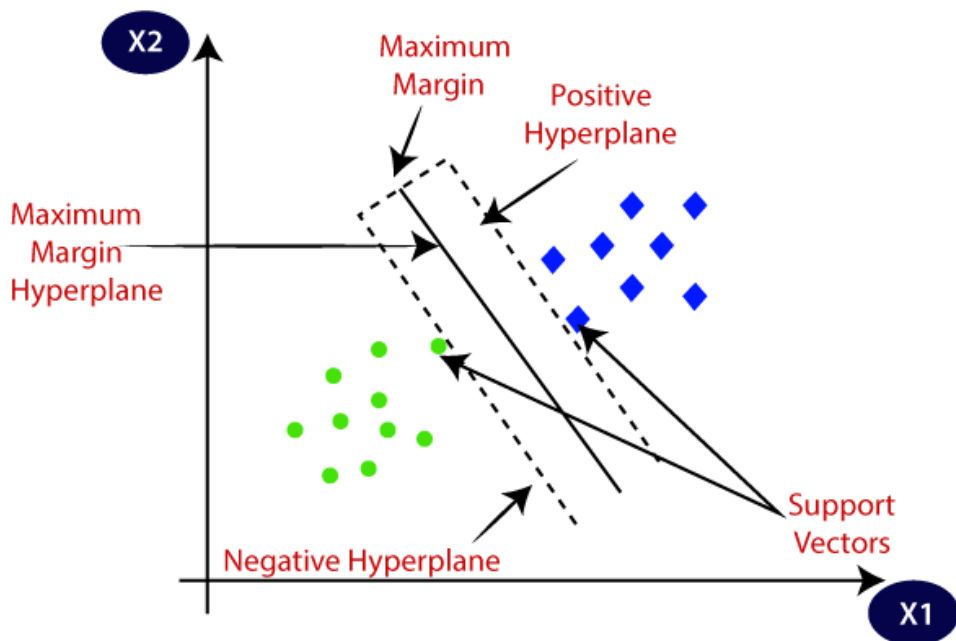


Slika 7. Struktura Stabla odlučivanja

(Izvor: [42])

2.3.1.4.SVM(Support Vector Machine)

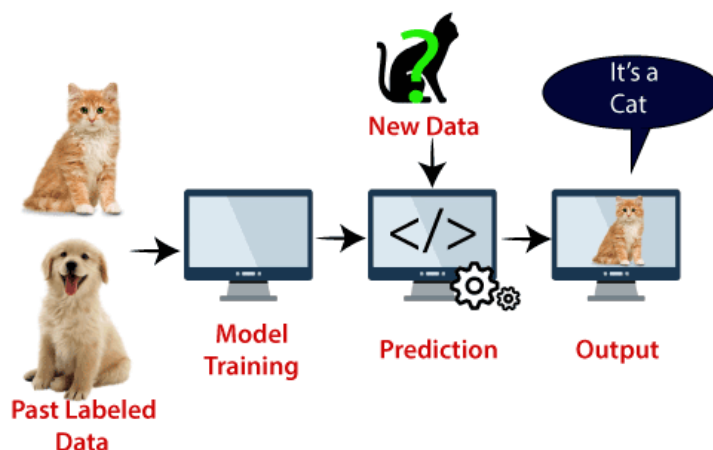
Support Vector Machine odnosno SVM je jedan od najpopularnijih algoritama nadziranog učenja, koji se koristi za klasifikaciju i probleme regresije. Međutim, prvenstveno se koristi za klasifikacijske probleme u strojnom učenju. Cilj SVM algoritma je stvoriti najbolju liniju ili granicu odluke koja može razdvojiti n-dimenzionalni prostor u klase kako bismo novu podatkovnu točku mogli lako staviti u ispravnu kategoriju u budućnosti [43]. Ova granica najbolje odluke naziva se „hyperplane“ ili hiper-ravnina. SVM odabire krajnje točke/vektore koji pomažu u stvaranju hiper-ravnine. Ovi ekstremni slučajevi nazivaju se „support vectors“, pa se stoga algoritam naziva Support Vector Machine. Na donjoj slici prikazan je model algoritma:



Slika 8. Model SVM-a

(Izvor: [43])

Kao primjer možemo navesti situaciju u kojoj smo vidjeli čudnu mačku koja također ima neke značajke psa, ako želimo model koji može točno odrediti je li to mačka ili pas, takav model možemo napraviti pomoću SVM algoritma. Prvo se taj model trenira s puno slika mačaka i pasa kako bi mogao naučiti o različitim značajkama mačaka i pasa, a zatim ćemo ga testirati na željenom psu/mački. Dakle, kako „support vector“ stvara granicu odluke između ova dva podataka (mačka i pas) i odabire ekstremne slučajeve, dobit ćemo krajnji rezultat, odnosno odgovor na pitanje dali gledamo u sliku mačke ili psa. Opisani postupak je prikazan na slici ispod:



Slika 9. Konkretna primjer korištenja SVM-a

(Izvor: [43])

2.3.1.5. Naivni-Bayes

Naivni Bayesov algoritam je algoritam nadziranog učenja koji se temelji na Bayesovom teoremu i koristi se za rješavanje klasifikacijskih problema [44]. Uglavnom se koristi u klasifikaciji teksta koja uključuje višedimenzionalni skup podataka za treniranje. Naivni Bayesov klasifikator jedan je od jednostavnih i najučinkovitijih algoritama klasifikacije koji pomaže u izgradnji brzih modela strojnog učenja koji mogu raditi brze predikcije. Spada u kategoriju klasifikatora vjerojatnosti, što znači da predviđa na temelju vjerojatnosti objekta. Neki od popularnih primjera primjene Naivnog Bayesova algoritma su filtriranje neželjene pošte, sentimentalna analiza i klasifikacija članaka. Na primjer, voće se može smatrati jabukom ako je crveno, okruglo i promjera oko 3 centimetra. Čak i ako ove značajke ovise jedna o drugoj ili o postojanju ostalih značajki, naivni Bayesov klasifikator smatrao bi da sva ta svojstva neovisno doprinose vjerojatnosti da je ovo voće jabuka. Naivni Bayesov model lako je izraditi, a posebno je koristan za vrlo velike skupove podataka. Pored svoje jednostavnosti, poznato je i da Naivni Bayes nadmašuje čak i vrlo sofisticirane metode klasifikacije. Bayesov teorem je prikazan na slici ispod:

The diagram shows the Bayes' theorem equation: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Arrows point from the terms to their labels: $P(x | c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', $P(c | x)$ is labeled 'Posterior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'. Below the equation is the expanded form: $P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$.

Slika 10. Bayesov teorem

(Izvor: [44])

2.3.1.6. KNN (K-Najbližih Susjeda)

KNN je jedan od najjednostavnijih algoritama strojnog učenja koji se temelji na tehnici nadziranog učenja. KNN algoritam pretpostavlja sličnost između novog slučaja/podataka i dostupnih slučajeva te novi slučaj svrstava u kategoriju koja je najbližnja dostupnim kategorijama [45]. KNN algoritam pohranjuje sve dostupne podatke i klasificira novu točku podataka na temelju sličnosti. To znači da se, kad se pojave novi podaci, pomoću KNN algoritma oni lako klasificiraju u ispravnu kategoriju. KNN algoritam može se koristiti za regresiju kao i za klasifikaciju, ali uglavnom se koristi za probleme s klasifikacijom. KNN je ne-parametarski algoritam, što znači da ne donosi nikakve pretpostavke o temeljnim podacima. Naziva se i algoritmom „lijenog učenika“ jer ne uči odmah iz skupa za treniranje, već pohranjuje skup podataka i u vrijeme klasifikacije izvodi radnju na skupu podataka. KNN algoritam u fazi treninga samo sprema skup podataka i kad dobije nove podatke, klasificira te podatke u kategoriju koja je najbližnja novim podacima. Za primjer možemo koristiti onaj koji smo koristili kod opisa SVM-a, zamislimo da imamo sliku bića koje izgleda slično mački

i psu, ali želimo znati je li to mačka ili pas. Dakle, za ovu identifikaciju možemo koristiti KNN algoritam, jer radi na mjeri sličnosti. Naš KNN model pronaći će slične značajke nove slike, prema skupu podataka koji sadrži slike mačaka i pasa, a na temelju najbližijih značajki staviti će ga u kategoriju mačaka ili pasa. Na slici ispod možemo vidjeti jednostavni model prethodno opisanog primjera:



Slika 11. Prikaz jednostavnog KNN modela

(Izvor: [45])

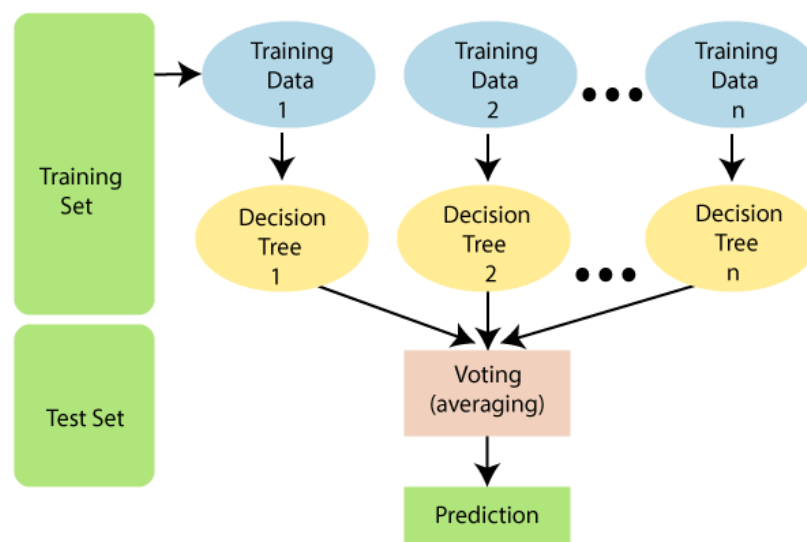
2.3.1.7. Slučajna šuma

Slučajna šuma popularan je algoritam strojnog učenja koji pripada nadziranoj tehnici učenja. Može se koristiti i za probleme klasifikacije i regresije u strojnom učenju. Temelji se na konceptu ansambl učenja, što je postupak kombiniranja više klasifikatora za rješavanje složenog problema i poboljšanje izvedbe modela. Kao što i samo ime govori, „Slučajna šuma“ je klasifikator koji sadrži niz stabala odlučivanja o različitim pod-skupovima datog skupa podataka i uzima prosjek za poboljšanje točnosti predviđanja tog skupa podataka [46]. Umjesto da se oslanja na jedno stablo odluke, slučajna šuma uzima predviđanje sa svakog stabla na temelju većine glasova predviđanja i na temelju toga predviđa konačni rezultat. Što je veći broj stabala u šumi to je veća točnost modela te se sprječava problem „overfittanja“. „Overfitting“ je izraz koji se koristi u statistici, a odnosi se na grešku modeliranja koja se dešava kada funkcija previše odgovara određenom skupu podataka [47].

Svako se stablo „sadi i uzgaja“ na sljedeći:

- Ako je broj slučajeva u skupu za treniranje N , tada se slučajno uzima uzorak od N slučajeva, ali s zamjenom. Ovaj uzorak bit će set za „uzgoj“ stabla.
- Ako postoji M ulaznih varijabli, naveden je broj $m \ll M$ takav da se na svakom čvoru slučajno odabire m varijabli iz M i koristi se najbolji „split“ na tih m za razdvajanje čvora. Vrijednost m održava se konstantnom tijekom „uzgoja“ šume.
- Svako se stablo uzgaja u najvećoj mogućoj mjeri.

Na slici ispod prikazan je dijagram rada algoritma slučajne šume:



Slika 12. Dijagram rada algoritma Slučajne šume

(Izvor: [46])

2.3.1.8. AdaBoost algoritam

Danas je strojno učenje temelj svih velikih inovacija, te se tvrtke nadaju da će im i u budućnosti strojno učenje omogućiti donošenje najboljih poslovnih odluka putem točnih predviđanja. Ali što se događa kad je osjetljivost algoritama strojnog učenja na pogreške velika i neobjašnjiva? Tada na scenu stupa koncept poznat kao „Ensemble Learning“. Ansambl učenje kombinira nekoliko osnovnih algoritama u jedan optimizirani algoritam predviđanja [48]. AdaBoost je metoda učenja ansambla (poznata i kao "meta-učenje") koja je u početku stvorena kako bi povećala učinkovitost binarnih klasifikatora. AdaBoost koristi iterativni pristup kako bi učio na pogreškama slabih klasifikatora i pretvorio ih u jake [49]. Pojedinačni klasifikator možda neće moći točno predvidjeti klasu objekta, ali kada grupiramo više slabih klasifikatora, pri čemu svaki progresivno uči na pogrešno klasificiranim objektima drugih, možemo izgraditi jedan jaki model. Ovdje treba spomenuti da klasifikator može biti bilo koji od osnovnih klasifikatora. Ovdje se postavlja pitanje, što je to "slabi" klasifikator? Slabi klasifikator je onaj koji se izvodi bolje od slučajnog pogađanja, ali i dalje je slab u određivanju klasa objektima. Umjesto da bude sam po sebi model, AdaBoost se može primijeniti na bilo koji klasifikator kako bi naučio iz njegovih nedostataka i predložio precizniji model. Iz tog se razloga obično naziva "najboljim klasifikatorom izvan okvira". AdaBoost pri radu koristi nešto što nazivamo „Decision Stumps“. Decision Stumps („panjevi odluke“) su poput drveća u slučajnoj šumi, ali nisu "potpuno odrasli". Imaju jedan čvor i dva lista. AdaBoost koristi šumu takvih panjeva, a ne drveće. Panjevi sami po sebi nisu dobar način za donošenje odluka. Puno uzgojeno stablo kombinira odluke svih varijabli za predviđanje ciljne vrijednosti. Panj, s druge strane, može koristiti samo jednu varijablu za donošenje odluke.

Princip rada AdaBoost algoritma:

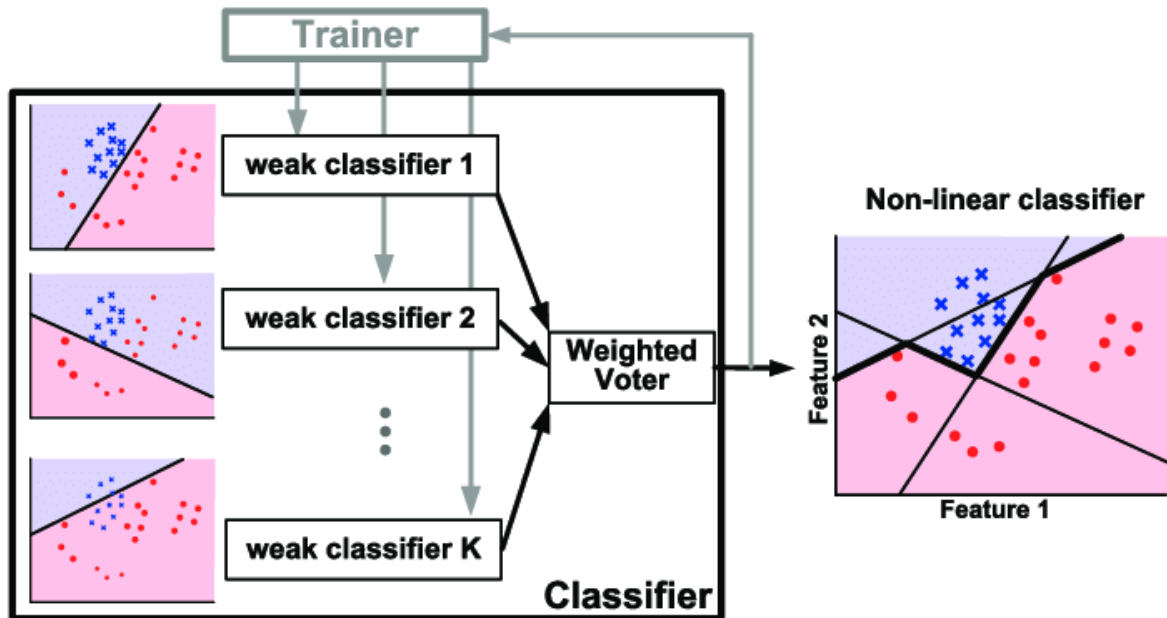
Korak 1: Slabi klasifikator (npr. panj odluke) izrađuje se povrh podataka za treniranje na temelju ponderiranih uzoraka. Ovdje težine svakog uzorka pokazuju koliko je važno pravilno klasificirati. U početku, za prvi panj, dajemo svim uzorcima jednake težine.

Korak 2: Stvaramo panj odluke za svaku varijablu i vidimo koliko dobro svaki panj klasificira uzorke u svoje ciljne klase.

Korak 3: Pogrešno klasificiranim uzorcima dodjeljuje se veća težina, tako da su ispravno klasificirani u sljedećem panju odluke. Težina se također dodjeljuje svakom klasifikatoru na temelju točnosti klasifikatora, što znači velika točnost = velika težina!

Korak 4: Ponavlja se postupak iz koraka 2 dok sve točke podataka nisu pravilno klasificirane ili dok se ne dosegne maksimalna razina ponavljanja.

Na sljedećoj slici može se vidjeti princip rada AdaBoost algoritma:



Slika 13. Princip rada AdaBoost algoritma

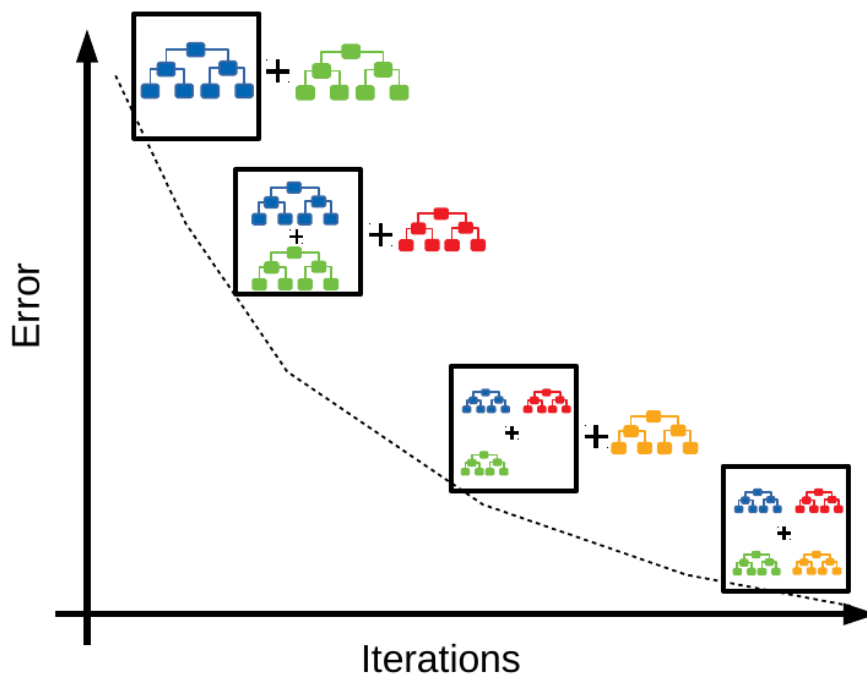
(Izvor: [48])

2.3.1.9. Algoritam podizanja gradijenta (GBM)

Algoritam podizanja gradijenta je algoritam koji se koristi kada imamo posla s velikom količinom podataka za predviđanje, a želimo postići veliku snagu predviđanja. Podizanje je zapravo skup algoritama učenja koji kombiniraju predviđanje nekoliko osnovnih prediktora kako bi se poboljšala robusnost u odnosu na korištenje jednog prediktora [50]. Algoritam kombinira više slabih ili prosječnih prediktora s jakim prediktorom. Ovi algoritmi za pojačavanje uvijek dobro rade na natjecanjima iz područja „date science“ kao što su Kaggle, AV Hackathon, CrowdAnalytix.

GBM (Gradient Boosting) je najlakše objasniti uvođenjem algoritma AdaBoost. Algoritam AdaBoost započinje treniranjem stabla odlučivanja u kojem je svakom promatranju dodijeljena jednaka težina. Nakon procjene prvog stabla, povećavamo težine onih promatranja koja je teško klasificirati, a smanjujemo težine onih koje je lako klasificirati. Stoga se na tim podacima uzgaja drugo stablo. Ovdje je ideja poboljšati predviđanja prvog stabla. Naš je novi model stoga Stablo 1 + Stablo 2. Zatim izračunavamo klasifikacijsku pogrešku iz ovog novog modela ansambla od 2 stabla i uzgajamo treće stablo kako bismo predvidjeli revidirane

ostatke. Ponavljamo ovaj postupak za određeni broj ponavljanja. Naknadna stabla pomažu nam u klasificiranju opažanja koja prethodna stabla nisu dobro klasificirala. Predviđanja konačnog modela cjeline stoga su zbroj predviđanja iz prethodnih modela stabla. GBM trenira modele postupno, aditivno i uzastopno. Glavna razlika između AdaBoost i GBM-a je u tome kako dva algoritma prepoznaju nedostatke slabih prediktora. Iako model AdaBoost identificira nedostatke korištenjem točaka podataka s velikom težinom, GBM to radi koristeći gradijente u funkciji gubitka ($y = ax + b + e$, e treba posebno spomenuti jer je to izraz pogreške) [51]. Funkcija gubitka mjeri je koja pokazuje koliko su dobri koeficijenti modela u uklapanju temeljnih podataka. Na primjer, ako pokušavamo predvidjeti prodajne cijene pomoću GBM-a, tada bi se funkcija gubitka temeljila na pogrešci između stvarnih i predviđenih cijena kuća. Jedna od najvećih prednosti korištenja GBM-a je ta što omogućuje optimizaciju funkcije gubitka koju određuje korisnik. Prethodno opisani način rada algoritma možemo vidjeti na slici ispod:



Slika 14. Model rada GBM-a

(Izvor: [50])

2.3.1.10. Algoritam Podizanja Gradijenta (GBM) koji se temelji na histogramu

Glavni problem algoritma podizanja gradijenta je što ima veoma sporo vrijeme treniranja modela. To je osobito problem kada se model koristi na velikim skupovima podataka s desecima tisuća podataka. Obuka stabala koja se dodaju može se dramatično ubrzati diskretiziranjem (spajanjem) kontinuiranih ulaznih varijabli u nekoliko stotina jedinstvenih vrijednosti. Algoritmi pojačavanja gradijenta koji implementiraju ovu tehniku i prilagođavaju algoritam treniranja prema ulaznim varijablama, nazivaju se algoritmima za pojačavanje gradijenta temeljenim na histogramu [52]. Ova metoda omogućuje stablu odluka da djeluje na rednom segmentu (cijeli broj) umjesto specifičnih vrijednosti u skupu podataka za treniranje. Uz to, učinkovite strukture podataka mogu se koristiti za predstavljanje spajanja ulaznih podataka, na primjer, mogu se koristiti histogrami, a algoritam izrade stabla može se dalje prilagoditi za učinkovitu upotrebu histograma u konstrukciji svakog stabla. Ove su tehnike izvorno razvijene krajem 1990-ih za efikasno razvijanje stabala pojedinačnih odluka na velikim skupovima podataka, ali se mogu koristiti u cjelinama stabala odlučivanja, poput podizanja gradijenta.

2.3.2.Mjere vrednovanja klasifikacije

U ovome poglavlju opisane su mjere koje će se koristiti za vrednovanje dobivenih modela strojnoga učenja: Točnost/Accuracy, Preciznost/Precision, Odziv/Recall i F1.

Za svaki model strojnog učenja znamo da je postizanje dobrog uklapanja(eng. „good fit“) izuzetno važno. Matrica konfuzije tablica je koja se često koristi za opisivanje izvedbe klasifikacijskog modela na skupu testnih podataka za koje su poznate prave vrijednosti. Matrica konfuzije je najintuitivnija vizualizacija za tumačenje izvedbe statističkog klasifikacijskog modela. To je tablica dimenzije 2x2 u kojoj svaki redak predstavlja instancu u predviđenoj klasi, dok stupci predstavljaju instance stvarne klase [55].

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Slika 15. Nacrt matrice konfuzije

(Izvor: [55])

Sa prethodne slike možemo vidjeti kako je matrica konfuzije podijeljena u četiri kvadranta. . Prvi kvadrant sadrži broj točno klasificiranih pozitivnih slučajeva (TP), drugi sadrži broj lažno klasificiranih pozitivnih slučajeva (FP), treći sadrži broj lažno klasificiranih negativnih slučajeva (FN), a četvrti sadrži broj točno klasificiranih negativnih slučajeva (TN). Kada imamo matricu konfuzije, jednostavno možemo iz nje iščitati koliko je predviđenih vrijednosti istinito, a koliko je predviđenih vrijednosti lažno.

2.3.2.1. Točnost/Accuracy

Točnost je najintuitivnija mjera vrednovanja i ona se definira kao omjer točno predviđenih opažanja i broja ukupnih opažanja [56].

$$\text{Točnost} = \frac{\text{Broj točno predviđenih opažanja}}{\text{Ukupni broj opažanja}} \quad (1)$$

Ali kada se bavimo problemom klasifikacije, pokušavamo predvidjeti binarni ishod. Je li to prijevara ili nije? Hoće li ta osoba platiti najam ili ne? Dakle, ono što nas zanima pored ovog ukupnog omjera jesu i predviđanja koja su lažno klasificirana kao pozitivna i lažno klasificirana kao negativna, posebno s obzirom na kontekst onoga što pokušavamo predvidjeti. Stopa točnosti od 99% mogla bi biti prilično dobra ako pokušavamo predvidjeti nešto poput prijave s kreditnom karticom, ali što ako lažni negativ predstavlja nekoga tko ima ozbiljan virus koji se može brzo širiti? Ili osobu koja ima rak? Formulu za točnost možemo zapisati i na sljedeći način:

$$\text{Točnost} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Gdje je TP broj točno klasificiranih pozitivnih slučajeva, TN = broj točno klasificiranih negativnih slučajeva, FP = broj lažno klasificiranih pozitivnih slučajeva i FN = broj lažno klasificiranih negativnih slučajeva.

Netko može pomisliti da je njihov model najbolji ako ima visoku točnost. Da, točnost je važna mjera, ali samo ako imamo simetrične skupove podataka gdje su vrijednosti lažnih pozitiva i lažnih negativa gotovo iste. Stoga moramo promatrati druge parametre kako bismo procijenili izvedbu modela.

2.3.2.2. Preciznost/Precision

Prije nego što možemo objasniti mjere preciznost i odziv, potrebno je objasniti vrste grešaka pod nazivom: „Pogreška tipa 1“ i „Pogreška tipa 2“. Ovi pojmovi nisu jedinstveni za probleme klasifikacije u strojnom učenju, već su također izuzetno važni kada je riječ o ispitivanju statističkih hipoteza.

Pogreška tipa I: lažan pozitiv (odbacivanje istinitih nultih hipoteza)

Pogreška tipa II: lažan negativ (ne-odbacivanje lažnih nultih hipoteza)

Preciznost je omjer točno predviđenih pozitivnih opažanja i ukupnih predviđenih pozitivnih opažanja. Visoka preciznost odnosi se na nisku stopu lažno pozitivnih rezultata [57].

$$\text{Preciznost} = \frac{TP}{TP + FP} \quad (3)$$

2.3.2.3. Odziv/Recall

Odziv je omjer ispravno predviđenih pozitivnih opažanja prema svim opažanjima u stvarnoj klasi [58].

$$\text{Odziv} = \frac{TP}{TP + FN} \quad (4)$$

2.3.2.4. F1

F1 mjera je harmonijska sredina između preciznosti i odziva [59]. Stoga ovaj rezultat uzima u obzir i lažne pozitivne i lažne negativne podatke. Intuitivno ga nije lako razumjeti kao točnost, ali F1 je obično korisniji od točnosti, pogotovo ako imamo neravnomjernu raspodjelu klasa. Točnost najbolje funkcionira ako lažni pozitivni i lažni negativni podaci imaju slične troškove. Ako su troškovi lažnih pozitivnih i lažnih negativnih rezultata vrlo različiti, bolje je pogledati i preciznost i odziv. Prethodno smo spomenuli da lažni pozitivni i lažni negativni mogu biti apsolutno presudni za neku studiju, dok istiniti negativni često manje utječu na problem koji se pokušava riješiti, posebno u poslovnom okruženju. Rezultat F1 pokušava to uzeti u obzir, dajući veću težinu lažnim negativima i lažnim pozitivima, a ne dopuštajući da veliki broj istinitih negativa utječe na krajnji rezultat.

$$F1 = 2 * \frac{\text{preciznost} * \text{odziv}}{\text{preciznost} + \text{odziv}} \quad (5)$$

2.3.3.Alati

Pri izradi programskog dijela ovoga diplomskog rada koristit će se editor za pisanje koda pod nazivom „Jupyter Notebook“, dok je programski jezik koji se koristi „Python“, točnije verzija „Python 3“.

2.3.3.1.Jupyter Notebook

„Jupyter Notebook“ je web aplikacija otvorenog koda koja korisnicima omogućuje stvaranje i dijeljenje dokumenata koji sadrže programski kod, jednadžbe, vizualizacije i običan tekst koji opisuje tijek rada [53]. Neka od područja u kojima se „Jupyter Notebook“ koristi uključuju: čišćenje i transformaciju podataka, numeričku simulaciju, statističko modeliranje, vizualizaciju podataka te strojno učenje.

2.3.3.2.Python

Python je programski jezik opće namjene na visokoj razini. Pythonova filozofija dizajna naglašava čitljivost koda uporabom značajnih udubljenja(indentation-a). Njegove jezične konstrukcije kao i objektno orijentirani pristup imaju za cilj pomoći programerima da napišu jasan, logičan kod za male i velike projekte. Podržava više programskih paradigmi, uključujući strukturirano(posebno proceduralno), objektno orijentirano i funkcionalno programiranje. Python se često opisuje kao jezik s „uključenim baterijama“ zbog toga što sadržava veliki broj standardnih biblioteka. Guido van Rossum počeo je raditi na Pythonu krajem 1980-ih, kao nasljedniku programskog jezika ABC, a prvi put ga je objavio 1991. godine kao Python 0.9.0. Python 2.0 objavljen je 2000. godine i predstavio je nove značajke. Python 3.0 objavljen je 2008. godine i predstavljao je veliku reviziju jezika koji nije u potpunosti kompatibilan s prethodnim verzijama, pa stoga veliki dio Python 2 koda ne radi na Python 3 verziji, stoga je potrebno napraviti neke izmjene. Python 2 ukinut je s verzijom 2.7.18 u 2020. godini. Python se zadnjih godina konstantno navodi kao jedan od najpopularnijih programskih jezika [54].

2.3.4. Biblioteke koje se koriste

U ovome poglavlju dat je kratak opis svake od biblioteka koja se koristi pri izradi praktičnog dijela diplomskog rada.

2.3.4.1. pandas

Pandas je softverska biblioteka napisana za programski jezik Python koja se koristi za manipulaciju i analizu podataka. Nudi strukture podataka i operacije za manipulaciju nad numeričkim tablicama. Riječ je o besplatnom softveru objavljenom pod tročlanom BSD licencom.

2.3.4.2. matplotlib.pyplot

Matplotlib.pyplot je sučelje za matplotlib koje pruža način crtanja nalik MATLAB-u.

2.3.4.3. collections

Collections je standardna biblioteka Robot Framework-a koja nudi skup ključnih riječi za rukovanje Python listama i rječnicima.

2.3.4.4. worldcloud

Worldcloud je biblioteka koja se koristi za generiranje oblaka riječi, također u sebi sadrži i stop riječi.

2.3.4.5. nltk

NLTK je vodeća platforma za izgradnju Python programa za rad s podacima o ljudskom jeziku. Pruža sučelje koje je lako za korištenje s preko 50 korpusa i leksičke resurse kao što je WordNet, zajedno s paketom biblioteka za obradu teksta za klasifikaciju, tokenizaciju, stemming, označavanje, parsiranje i semantičko obrazloženje.

2.3.4.6. re

Ovaj modul pruža mogućnost korištenja regularnih izraza slične onima koje pruža programski jezik Perl.

2.3.4.7. sklearn

Sklearn-scikit je besplatna softverska biblioteka strojnog učenja za programski jezik Python. Sadrži razne algoritme klasifikacije, regresije i klasterizacije, uključujući vektorske strojeve, slučajne šume, podizanje gradijenta, k-means i DBSCAN, a dizajniran je za interakciju s Pythonovim numeričkim i znanstvenim knjižnicama NumPy i SciPy. Ova biblioteka je temelj za izradu praktičnog dijela zadatka jer se iz nje importaju funkcije kao što su: „CountVectorizer“, „TfidfTransformer“, „classification_report“, „confusion_matrix“, „accuracy_score“, „f1_score“, „precision_score“, „recall_score“, „LinearSVC“, „Pipeline“, „SGDClassifier“, „LogisticRegression“, itd..., a koje su ključne za definiranje modela strojnoga učenja, računanje mjera vrednovanja klasifikacije i crtanje matrica konfuzije.

2.3.4.8. joblib

Joblib je skup alata za pružanje jednostavnog korištenja pipelina u Pythonu. Joblib je optimiziran da bude brz i robustan posebno na velikim podacima te ima specifične optimizacije za numpy nizove. Ima BSD-licencu.

2.3.4.9.numpy

NumPy je biblioteka za programski jezik Python, koja dodaje podršku za velike, višedimenzionalne nizove i matrice, zajedno s velikom kolekcijom matematičkih funkcija na visokoj razini za rad s tim nizovima.

2.3.4.10.itertools

Itertools je modul koji implementira brojne gradivne blokove iteratora nadahnute konstrukcijama iz APL-a, Haskell-a i SML-a koji zajedno čine 'iteratorsku algebru' što omogućava sažetu izradu specijaliziranih alata u Pythonu.

3. Praktični dio

3.1. Statistika skupa podataka

U ovome diplomskom radu za praktični dio koriste se prethodno dostupni skupovi podataka koji su vezani uz COVID-19 i lažne vijesti. Skupovi podataka su sljedeći: „test_data.xlsx“, „train_data.xlsx“, „validation_data.xlsx“ te „test_data_with_labels.xlsx“ [65] [66]. Svaki od tih skupova podataka je dio jednog te istog originalnog dataseta koji je vezan uz objave koje su objavljene na Twitteru, a vezane su uz COVID-19. Skupovi podataka „train_data.xlsx“, „validation_data.xlsx“ te „test_data_with_labels.xlsx“ sastoje se od tri stupca, a to su „id“, „tweet“ i „label“, dok skup „train_data.xlsx“ sadrži samo stupce „id“ i „tweet“. Stupac „id“ je broj koji jedinstveno označuje svaku vijest(tweet), stupac „tweet“ sadrži vijest koja je objavljena na Twitteru, a koja je vezana uz COVID-19, dok stupac „label“ može poprimiti oznaku true/false, ovisno o tome jeli taj tweet istinit/lažan. U dijelu praktičnog rada koji se odnosi na statistiku skupa podataka koriste se sva tri skupa, odnosno skupovi za treniranje, testiranje i validaciju, dok se za drugi dio praktičnog zadatka, koji se odnosi na izgradnju modela strojnog učenja i usporedbu istih, koriste samo skupovi za treniranje i validaciju.

3.1.1. Instalacija i pokretanje jupyter notebook-a

Prva stvar koju je potrebno napraviti je instalirati „Jupyter notebook“ te ga pokrenuti korištenjem sljedećih naredbi u „command promptu“:

```
PS C:\Users\ArijanJasarevic> pip install notebook
```

Slika 16. Instalacija "Jupyter Notebooka"

Nakon što se instalacija uspješno izvršila potrebno je pokrenuti sljedeću naredbu koja otvara „Jupyter Notebook“ u web pregledniku „Mozilla Firefox“:

```
PS C:\Users\ArijanJasarevic> jupyter-notebook --browser=firefox
```

Slika 17. Otvaranje Notebook-a u Mozilli

Potom se napravi novi folder pod nazivom „diplomski“ i unutar njega se prvo kreira novi „python notebook“ pod nazivom „main.ipynb“, nakon toga u folderu „diplomski“, napravi se novi folder pod imenom „podaci“ te se u njega prebace skupovi podataka koje koristimo.

3.1.2.Statistika

Prvo se učitaju biblioteke koje su potrebne kako bi se prikazala statistika skupova podataka, zatim se podaci učitaju u varijable da se može raditi s njima, i nakon toga se ispisuju informacije za svaki od skupova.

STATISTIKA SKUPA PODATAKA KOJI SE KORISTI U DIPLOMSKOM RADU

```
# Importanje potrebnih biblioteka

import pandas as pd
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from collections import Counter
stopwords = set(STOPWORDS)

# Učitavanje skupova podataka u varijable

trening_podaci = pd.read_excel('podaci/train_data.xlsx', engine='openpyxl')
test_podaci = pd.read_excel('podaci/test_data.xlsx', engine='openpyxl')
validation_podaci = pd.read_excel('podaci/validation_data.xlsx', engine='openpyxl')

# Informacije o skupu "train_data.xlsx"
# Možemo vidjeti kako skup za testiranje čine stupci "id", "tweet" i "label"
# id = broj koji jedinstveno definira svaku vijest
# tweet = vijest odnosno tweet koji je objavljen na Twitteru
# label = oznaka dali je određeni tweet istinit ili lažan

trening_podaci.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6420 entries, 0 to 6419
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0    id      6420 non-null     int64
1    tweet   6420 non-null     object
2    label   6420 non-null     object
dtypes: int64(1), object(2)
memory usage: 150.6+ KB
```

Slika 18. Učitavanje biblioteka i podataka u varijable, ispis informacija za skup "train_data.xlsx"

Sa prethodne slike možemo vidjeti kako skup podataka „train_data.xlsx“ sadrži 6420 redaka, odnosno ima 6420 podataka u sebi.

```
# Informacije o skupu "test_data.xlsx"
# Možemo vidjeti kako skup za testiranje čine stupci "id" i "tweet", dakle nemamo stupca "label"
# To je zbog toga jer su ti podaci ta testiranje, pa stupac label nije potreban
# id = broj koji jedinstveno definira svaku vijest
# tweet = vijest odnosno tweet koji je objavljen na Twitteru
```

```
test_podaci.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2140 entries, 0 to 2139
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   id       2140 non-null    int64
1   tweet   2140 non-null    object
2   label   2140 non-null    object
dtypes: int64(1), object(2)
memory usage: 50.3+ KB
```

```
# Informacije o skupu "validation_data.xlsx"
# Možemo vidjeti kako skup za testiranje čine stupci "id", "tweet" i "label"
# id = broj koji jedinstveno definira svaku vijest
# tweet = vijest odnosno tweet koji je objavljen na Twitteru
# label = oznaka dali je određeni tweet istinit ili lažan
```

```
validation_podaci.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2140 entries, 0 to 2139
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   id       2140 non-null    int64
1   tweet   2140 non-null    object
2   label   2140 non-null    object
dtypes: int64(1), object(2)
memory usage: 50.3+ KB
```

Slika 19. Informacije za skupove "test_data_with_labels.xlsx" i " validation_data.xlsx"

Sa prethodne slike možemo vidjeti kako oba skupa podataka sadrže 2140 redaka, odnosno svaki od skupova „test_data.xlsx“ i „validation_data.xlsx“ sadrži 2140 redaka, to jest ima 2140 podataka u sebi.

Zatim se skupovi povezuju u zajednički originalni skup te se također ispisuju informacije o njemu, možemo vidjeti da cjelokupni povezani skup podataka sadrži 10700 redaka, odnosno podataka.

```
# Povezivanje skupova u jedan veliki (originalni) skup koji sadrži sve podatke
# Taj skup je originalni skup, kojeg su zatim podijelili u skupove za trening, testiranje i validaciju
# Objedinjeni skup sadrži 10700 records

cijeli_podaci = pd.concat([trening_podaci, test_podaci, validation_podaci])
cijeli_podaci.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 10700 entries, 0 to 2139
Data columns (total 3 columns):
#   Column  Non-Null Count  Dtype
---  -
0   id      10700 non-null   int64
1   tweet  10700 non-null   object
2   label  10700 non-null   object
dtypes: int64(1), object(2)
memory usage: 334.4+ KB
```

Slika 20. Informacije objedinjenog skupa

Zatim se ispisuje statistika koja pokazuje koliko svaki od skupova ima lažnih odnosno istinitih tweetova/vijesti/objava, sa iduće slike se može vidjeti kako je situacija sljedeća: skup za treniranje „train_data.xlsx“ sadrži 3360 istinitih i 3060 lažnih objava, skup za testiranje „test_data_with_labels.xlsx“ sadrži 1120 istinitih objava i 1020 lažnih objava, skup za validaciju „validation_data.xlsx“ sadrži također 1120 istinitih objava i 1020 lažnih objava, dok objedinjeni skup pod nazivom „cijeli_podaci“ sadrži 5600 istinitih objava i 5100 lažnih objava:

```
# Ispis statistike istinitih/lažnih tweetova po skupovima

print(trening_podaci['label'].value_counts())
print(test_podaci['label'].value_counts())
print(validation_podaci['label'].value_counts())
print(cijeli_podaci['label'].value_counts())

real      3360
fake      3060
Name: label, dtype: int64
real      1120
fake      1020
Name: label, dtype: int64
real      1120
fake      1020
Name: label, dtype: int64
real      5600
fake      5100
Name: label, dtype: int64
```

Slika 21. Statistika istinitih/lažnih tweetova

Nakon toga uzimamo tweetove iz objedinjenog skupa i svrstavamo ih u varijable: svi_tweetovi (svi tweetovi koji su uzeti iz objedinjenog skupa), svi_lažni_tweetovi(svi lažni tweetovi koji su uzeti iz objedinjenog skupa) i svi_istiniti_tweetovi(svi istiniti tweetovi koji su uzeti iz objedinjenog skupa). Potom se radi funkcija pod nazivom „konkatenacija“ koja će konkatenirati sve tweetove ovisno o tome koje podatke provideamo kao input, i spremiti tokene u jednu od troje varijabli: „tokeni“, „lažni_tokeni“ i „istiniti_tokeni“:

```
# Uzimanje tweetova iz objedinjenog skupa i svrstavanje u varijable
#svi_tweetovi = svi tweetovi koji su uzeti iz objedinjenog skupa
#svi_lažni_tweetovi = svi lažni tweetovi koji su uzeti iz objedinjenog skupa
#svi_istiniti_tweetovi = svi istiniti tweetovi koji su uzeti iz objedinjenog skupa

svi_tweetovi = cijeli_podaci['tweet']
svi_lažni_tweetovi = cijeli_podaci[cijeli_podaci['label']=='fake']['tweet']
svi_istiniti_tweetovi = cijeli_podaci[cijeli_podaci['label']=='real']['tweet']

# Radimo funkciju "konkatenacija" koje će konkatenirati sve tweetove ovisno o tome koje podatke provideamo kao input
# Funkcija vraća sve tokene za date podatke

def konkatenacija(tweetovi):
    tokeni = ''
    for tekst in tweetovi:
        tgt = str(tekst).lower().strip()
        tgt = tgt.replace('\n', ' ')
        tgt = tgt.replace('&', ' ')
        tgt = tgt.replace('#', '')
        tgt = tgt.replace('-', '')
        tgt = tgt.replace(';', '')
        tgt = " ".join(tgt.split())
        tokeni += tgt + " "
    return tokeni

# Postavljamo tokene za skupove
# tokeni = tokeni za sve tweetove
# lažni_tokeni = tokeni za lažne tweetove
# istiniti_tokeni = tokeni za istinite tweetove

tokeni = konkatenacija(svi_tweetovi)
lažni_tokeni = konkatenacija(svi_lažni_tweetovi)
istiniti_tokeni = konkatenacija(svi_istiniti_tweetovi)
```

Slika 22. Isječak koda u kojem se definiraju varijable za tokene i funkcija konkatenacija

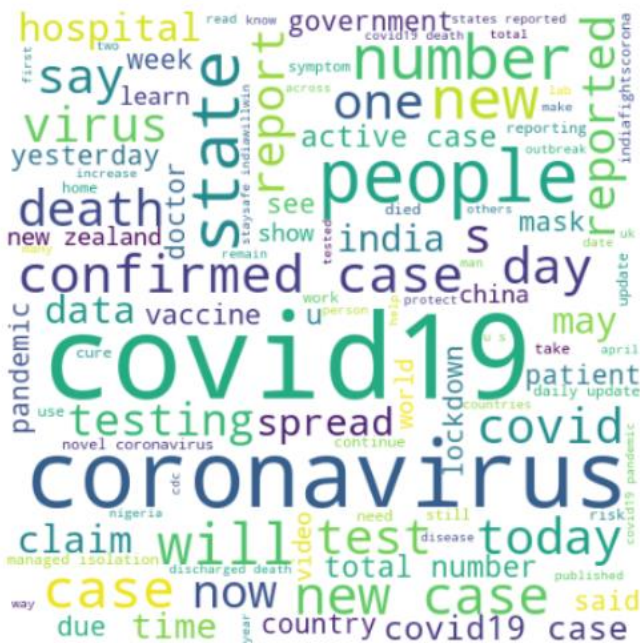
U nastavku je prikazano generiranje „oblaka riječi“ za prethodno definirane i navedene skupine tweetova: „tokeni“ (svi tweetovi), „lažni_tokeni“ (lažni tweetovi) i „istiniti_tokeni“ (istiniti tweetovi).

```
# Crtanje "oblaka riječi" za sve tokene(tweetove)

wordcloud = WordCloud(width = 500, height = 500,
                      background_color = 'white',
                      stopwords = stopwords.union({'https', 't', 'co'}), min_font_size = 10).generate(tokeni)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Slika 23. Generiranje wordclouda za sve tweetove

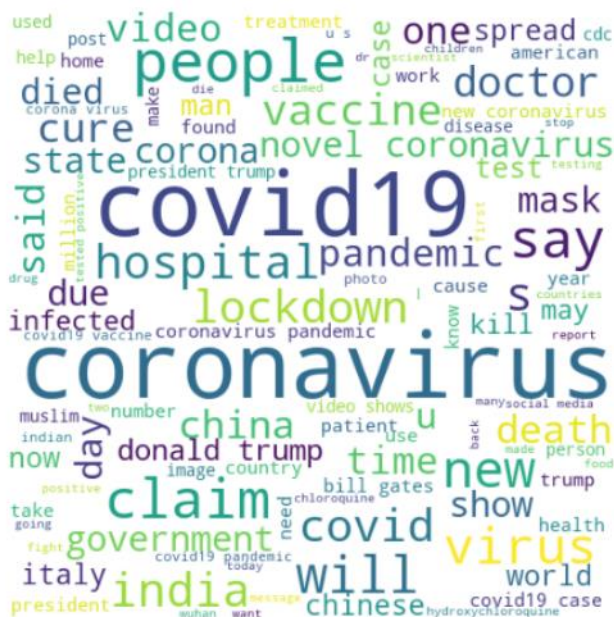
Sa prethodne slike možemo zaključiti kako su za skup svih tweetova, odnosno skup koji sadrži i lažne i istinite tweetove, riječi koje se najčešće pojavljuju: „covid 19“, „coronavirus“, „people“, „number“, „state“, „confirmed“, „death“, „case“, „new case“, „covid“ itd.

```
# Crtanje "oblaka riječi" za sve lažne tweetove

wordcloud = WordCloud(width = 500, height = 500,
                      background_color = 'white',
                      stopwords = stopwords.union({'https', 't', 'co'}), min_font_size = 10).generate(lažni_tokeni)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Slika 24. Generiranje wordclouda za lažne tweetove

Sa prethodne slike možemo vidjeti kako su za skup svih lažnih tweetova, riječi koje se najčešće pojavljuju, a nisu baš uobičajene u prošlom objedinjenom skupu: „india“, „donald trump“, „bill gates“, „italy“, „hydroxychloroquine“, „video“, „muslim“, „video shows“, „china“ itd. To je veoma zanimljivo jer se osobno sjećam vijesti da su Donald Trump i Bill Gates sudionici u zavjeri vezanoj uz cjepivo, isto tako kako je pijenje klora lijek za koronavirus, te različitih videozapisa koji tvrde da su vezani uz COVID-19 te kako je njihov sadržaj istinit.

```
# Crtanje "oblaka riječi" za sve istinite tokene(tweetove)
```

```
wordcloud = WordCloud(width = 500, height = 500,
                      background_color = 'white',
                      stopwords = stopwords.union({'https', 't', 'co'}), min_font_size = 10).generate(istiniti_tokeni)

plt.figure(figsize = (6, 6), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Slika 25. Generiranje wordclouda za istinite tweetove

Sa prethodne slike možemo vidjeti kako su za skup svih istinitih tweetova, riječi koje se najčešće pojavljuju: „covid 19“, „confirmed case“, „people“, „today“, „number“, „state“, „new case“, „reported“, „test“ itd. Takve riječi generirane su najčešće u dnevnim objavama koje su vezane za broj novoizaraženih po državama.

Potom se definiraju varijable: „svi_tweetovi“, „lazni_tweetovi“ i „istiniti_tweetovi“ u koje se sprema lista „splitanih“ riječi ovisno iz kojeg skupa tokena dolaze, što se može vidjeti na sljedećoj slici:

```
# Statistika za top 25 riječi u svakoj skupini tweetova
```

```
svi_tweetovi = [r for r in tokeni.split() if r not in stopwords]
lazni_tweetovi = [r for r in lažni_tokeni.split() if r not in stopwords]
istiniti_tweetovi = [r for r in istiniti_tokeni.split() if r not in stopwords]
```

Slika 26. Podjela tweetova u željene skupine

Te se skupine potom koriste za ispis top 25 riječi što se može vidjeti u tablici ispod:

	svi_tweetovi	lazni_tweetovi	istiniti_tweetovi
1.	covid19, 4236	coronavirus, 1672	covid19, 2723
2.	cases, 2471	covid19, 1513	cases, 2252
3.	coronavirus, 2101	people, 455	new, 1442
4.	new , 1785	will, 394	tests, 941
5.	people, 1182	new, 343	number, 789
6.	tests, 1025	coronavirus., 335	total, 775
7.	number, 858	trump, 333	people, 727
8.	will, 813	says, 320	reported, 719
9.	deaths, 804	video, 307	confirmed, 712
10.	total, 802	covid19, 300	states, 668
11.	confirmed, 782	vaccine, 286	deaths, 668
12.	reported, 774	virus, 270	testing, 613
13.	states, 753	president, 243	now, 485
14.	testing, 701	covid, 233	health, 447
15.	covid19., 661	cases, 219	rt, 437
16.	health, 631	shows, 219	coronavirus, 429
17.	covid, 623	pandemic, 217	will, 419
18.	now, 593	hospital, 217	report, 415
19.	india, 537	india, 211	state, 395
20.	one, 535	claims, 197	covid, 390
21.	state, 501	corona, 196	one, 375
22.	says, 480	said, 189	daily, 369
23.	rt, 471	health, 184	today, 364
24.	report, 443	lockdown, 183	update, 363
25.	case, 424	claim, 177	covid19., 361

Tablica 1. Top 25 riječi ovisno o skupini

3.2. Algoritmi strojnog učenja

3.2.1. Funkcija za crtanje matrice konfuzije

Prvo je potrebno importati/učitati potrebne biblioteke i funkcije što možemo vidjeti na slici ispod:

ALGORITMI STROJNOG UČENJA

```
# Importanje potrebnih biblioteka
```

```
import pandas as pd
import nltk
import re
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
from sklearn.svm import LinearSVC
from sklearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier, LogisticRegression
from sklearn.base import TransformerMixin
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from joblib import dump, load
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.experimental import enable_hist_gradient_boosting
from sklearn.ensemble import HistGradientBoostingClassifier
from sklearn import tree
import numpy as np
import itertools
```

Slika 27. Importanje potrebnih paketa i funkcija

Prije samoga definiranja prethodno navedenih modela strojnog učenja potrebno je definirati funkciju koja će crtati graf zadane matrice konfuzije. Funkciju nazivamo kao „matrica_konfuzije“, pritom su ulazni parametri: „matrica“, „target_leveli“, „title“ koji poprima vrijednost „Matrica konfuzije“ ako pri pozivanju funkcije ne provideamo taj parametar, „cmap“ i „normalize“ koji može poprimiti vrijednost True/False. Varijabla „matrica“ je matrica konfuzije dobivena pomoću „sklearn.metrics.confusion_matrix“ funkcije, „target_leveli“ poprima vrijednost „high“, „medium“, „low“ ovisno o inputu, „cmap“ je gradijent vrijednosti prikazan iz „matplotlib.pyplot.cm“ funkcije, dok je „normalize“ logička vrijednost koja može biti True ili False. Na slici ispod možemo vidjeti prethodno opisanu funkciju:

```

# Definiranje funkcije za crtanje matrice konfuzije

def matrica_konfuzije(matrica, target_leveli, title='Matrica konfuzije', cmap=plt.cm.Greens, normalize=True):

    accuracy = np.trace(matrica) / float(np.sum(matrica))
    misclass = 1 - accuracy

    if cmap is None:
        cmap = plt.get_cmap('Blues')

    plt.figure(figsize=(5, 4))
    plt.imshow(matrica, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()

    if target_leveli is not None:
        tick_marks = np.arange(len(target_leveli))
        plt.xticks(tick_marks, target_leveli, rotation=45)
        plt.yticks(tick_marks, target_leveli)

    if normalize:
        matrica = matrica.astype('float') / matrica.sum(axis=1)[:, np.newaxis]

    thresh = matrica.max() / 1.5 if normalize else matrica.max() / 2
    for i, j in itertools.product(range(matrica.shape[0]), range(matrica.shape[1])):
        if normalize:
            plt.text(j, i, "{:0.4f}".format(matrica[i, j]),
                    horizontalalignment="center",
                    color="white" if matrica[i, j] > thresh else "black")
        else:
            plt.text(j, i, "{:,}".format(matrica[i, j]),
                    horizontalalignment="center",
                    color="white" if matrica[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()

```

Slika 28. Funkcija za crtanje matrice konfuzije

Također je potrebno definirati funkciju koja će ukloniti stop riječi iz skupova podataka. U programiranju, stop ili zaustavne riječi su riječi koje se obično filtriraju prije obrade podataka prirodnog jezika (teksta). Iako se stop riječi obično odnose na najčešće riječi u jeziku, ne postoji jedinstveni univerzalni popis stop riječi koje koriste svi alati za obradu prirodnog jezika [60].

```

# Definiranje funkcije za uklanjanje stop riječi

stop_rijeci = set(stopwords.words("english"))
def cleantext(string):
    tekst = string.lower().split()
    tekst = " ".join(tekst)
    tekst = re.sub(r"http(\S)+", ' ', tekst)
    tekst = re.sub(r"www(\S)+", ' ', tekst)
    tekst = re.sub(r"&", ' and ', tekst)
    tekst = tekst.replace('&', ' ')
    tekst = re.sub(r"^[0-9a-zA-Z]+", ' ', tekst)
    tekst = tekst.split()
    tekst = [rijec for rijec in tekst if not rijec in stop_rijeci]
    tekst = " ".join(tekst)
    return tekst

```

Slika 29. Funkcija za uklanjanje stop riječi

Zatim se definira funkcija koja ispisuje prethodno navedene mjere vrednovanja klasifikacije: „Točnost/Accuracy“, „Preciznost/Precision“, „Odziv/Recall“ te „F1“.

```
# Definiranje funkcije koja ispisuje mjere vrednovanja klasifikacije
```

```
def mjere_vrednovanja_klasifikacije(predikcija, istinita_vrijednost):  
    print(confusion_matrix(istinita_vrijednost, predikcija))  
    print(classification_report(istinita_vrijednost, predikcija,))  
    print("Točnost/Accuracy : ", accuracy_score(predikcija, istinita_vrijednost))  
    print("Preciznost/Precision : ", precision_score(predikcija, istinita_vrijednost, average = 'weighted'))  
    print("Opoziv/Recall : ", recall_score(predikcija, istinita_vrijednost, average = 'weighted'))  
    print("F1 : ", f1_score(predikcija, istinita_vrijednost, average = 'weighted'))
```

Slika 30. Mjere vrednovanja klasifikacije

Naposljetku napokon možemo učitati podatke za treniranje i validaciju u željene varijable koje se potom koriste za definiranje željenih modela strojnog učenja:

```
# Učitavanje skupova podataka koji su potrebni za definiranje modela
```

```
train_podaci = pd.read_excel('podaci/train_data.xlsx', engine='openpyxl')  
val_podaci = pd.read_excel('podaci/validation_data.xlsx', engine='openpyxl')
```

```
train_podaci['tweet'] = trening_podaci['tweet'].map(lambda x: cleantext(x))  
val_podaci['tweet'] = validation_podaci['tweet'].map(lambda x: cleantext(x))
```

Slika 31. Učitavanje skupova podataka u varijable

3.2.2. Logistička regresija

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela logističke regresije, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
logisticka_regresija = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', LogisticRegression())
])
logisticka_regresija_fit = logisticka_regresija.fit(train_podaci['tweet'], train_podaci['label'])
print('Logistička regresija/Logistic Regression')
print('Vrijednost:')
predikcija = logisticka_regresija.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
                  normalize = False, title = 'Matrica konfuzije za model Logističke regresije')
```

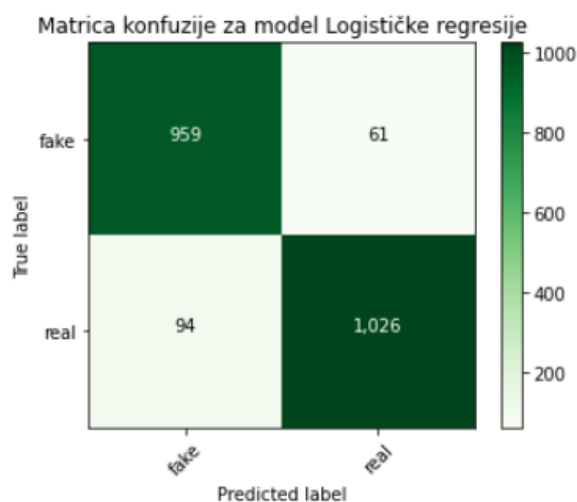
Slika 32. Funkcija logističke regresije

```
Logistička regresija/Logistic Regression
Vrijednost:
[[ 959   61]
 [   94 1026]]
           precision    recall  f1-score   support

      fake         0.91     0.94     0.93     1020
      real         0.94     0.92     0.93     1120

   accuracy                   0.93     2140
  macro avg         0.93     0.93     0.93     2140
 weighted avg         0.93     0.93     0.93     2140

Točnost/Accuracy : 0.927570093457944
Preciznost/Precision : 0.9279421090866252
Opoziv/Recall : 0.927570093457944
F1 : 0.9275350904769529
```



Slika 33. Rezultati za logističku regresiju

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9275, 0.9279, 0.9275, 0.9275.

3.2.3. Stohastički gradijentni silazak

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela stohastičkog gradijentnog silaska, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
sgd = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', SGDClassifier())
])
sgd_fit = sgd.fit(train_podaci['tweet'], train_podaci['label'])
print('Stohastički gradijentni silazak/Stochastic Gradient Descent')
print('Vrijednost:')
predikcija = sgd.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
                  normalize = False, title = 'Matrica konfuzije za model Stohastički gradijentni silazak')
```

Slika 34. Funkcija SGD-a

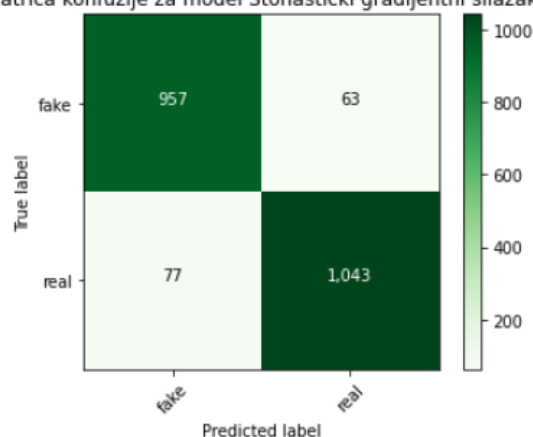
```
Stohastički gradijentni silazak/Stochastic Gradient Descent
Vrijednost:
[[ 957  63]
 [ 77 1043]]
      precision    recall  f1-score   support

   fake      0.93      0.94      0.93      1020
   real      0.94      0.93      0.94      1120

   accuracy                0.93      2140
  macro avg      0.93      0.93      0.93      2140
 weighted avg      0.93      0.93      0.93      2140

Točnost/Accuracy : 0.9345794392523364
Preciznost/Precision : 0.9346251374381528
Opoziv/Recall : 0.9345794392523364
F1 : 0.9345622120343164
```

Matrica konfuzije za model Stohastički gradijentni silazak



Slika 35. Rezultati za SGD

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9345, 0.9346, 0.9345, 0.9345.

3.2.4. Stablo odlučivanja

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela stabla odlučivanja, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
stablo_odlucivanja = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', tree.DecisionTreeClassifier())
])
stablo_odlucivanja_fit = stablo_odlucivanja.fit(train_podaci['tweet'], train_podaci['label'])
print('Stablo odlučivnaja/Decision Tree')
print('Vrijednost:')
predikcija = stablo_odlucivanja.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
                  normalize = False, title = 'Matrica konfuzije za model Stabla odlučivanja')
```

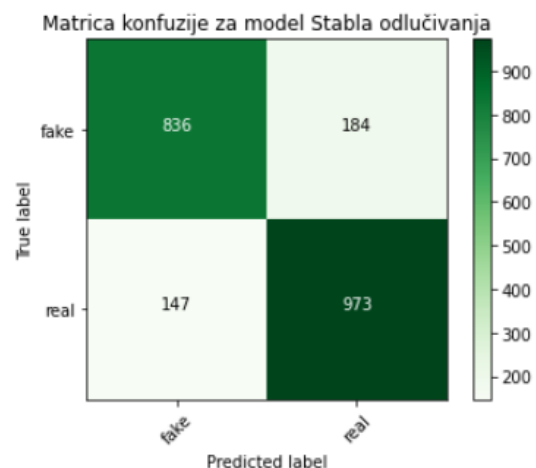
Slika 36. Funkcija stabla odlučivanja

```
Stablo odlučivnaja/Decision Tree
Vrijednost:
[[836 184]
 [147 973]]
      precision    recall  f1-score   support

   fake      0.85     0.82     0.83     1020
   real      0.84     0.87     0.85     1120

 accuracy                0.85     2140
 macro avg              0.85     0.84     0.84     2140
 weighted avg          0.85     0.85     0.85     2140

Točnost/Accuracy : 0.8453271028037384
Preciznost/Precision : 0.8461767569177203
Opoziv/Recall : 0.8453271028037384
F1 : 0.8454990093961746
```



Slika 37. Rezultati za stablo odlučivanja

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.8453, 0.8461, 0.8453, 0.8454.

3.2.5.SVM

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela SVM-a, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
svm = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', LinearSVC())
])
svm_fit = svm.fit(train_podaci['tweet'], train_podaci['label'])
print('SVM/Support Vector Machine')
print ('Vrijednost:')
predikcija = svm.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'],predikcija),target_leveli=['fake','real'],
    normalize = False, title = 'Matrica konfuzije za model SVM')
```

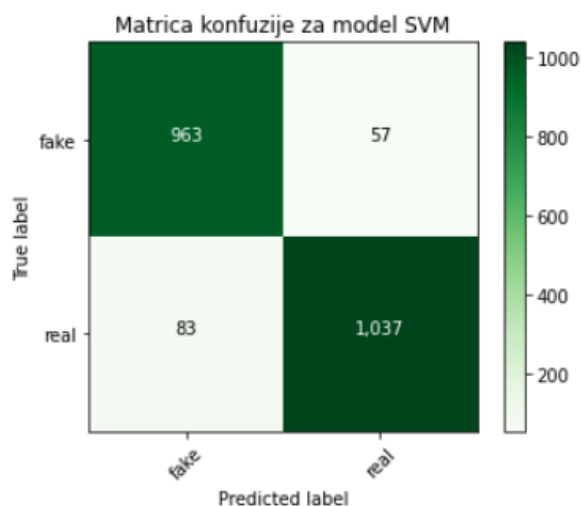
Slika 38. Funkcija za SVM

```
SVM/Support Vector Machine
Vrijednost:
[[ 963   57]
 [  83 1037]]
      precision    recall  f1-score   support

   fake      0.92     0.94     0.93     1020
   real      0.95     0.93     0.94     1120

 accuracy                0.93     2140
 macro avg              0.93     0.94     0.93     2140
 weighted avg          0.93     0.93     0.93     2140

Točnost/Accuracy : 0.9345794392523364
Preciznost/Precision : 0.9348008619335585
Opoziv/Recall : 0.9345794392523364
F1 : 0.9345519215989282
```



Slika 39. Rezultati za SVM

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9345, 0.9348, 0.9345, 0.9345.

3.2.6.Naivni-Bayes

Prije samoga definiranja modela Naive-Bayes, potrebno je definirati klasu „DenseTransformer“ koja će transformirati podatke u „dense“ matricu jer inače dolazi do greške u izvođenju koda:

```
class DenseTransformer(TransformerMixin):  
  
    def fit(self, X, y=None, **fit_params):  
        return self  
  
    def transform(self, X, y=None, **fit_params):  
        return X.todense()
```

Slika 40. Klasa DenseTransformer

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela Naive-Bayes, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
naivni_bayes = Pipeline([  
    ('vectorizer', CountVectorizer()),  
    ('statistical measure', TfidfTransformer()),  
    ('to_dense', DenseTransformer()),  
    ('classifier', GaussianNB())  
)  
  
naivni_bayes_fit = naivni_bayes.fit(train_podaci['tweet'], train_podaci['label'])  
print('Naivni-Bayes/Naive-Bayes')  
print ('Vrijednost:')  
predikcija = naivni_bayes.predict(val_podaci['tweet'])  
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])  
matrica_konfuzije(confusion_matrix(val_podaci['label'],predikcija),target_leveli=['fake','real'],  
    normalize = False, title = 'Matrica konfuzije za model Naive-Bayes')
```

Slika 41. Funkcija Naive-Bayes

Sa prethodne slike možemo vidjeti kako je u pipeline prije koraka 'classifier' dodan korak 'to_dense'.

```

Naivni-Bayes/Naive-Bayes
Vrijednost:
[[ 804 216]
 [ 97 1023]]

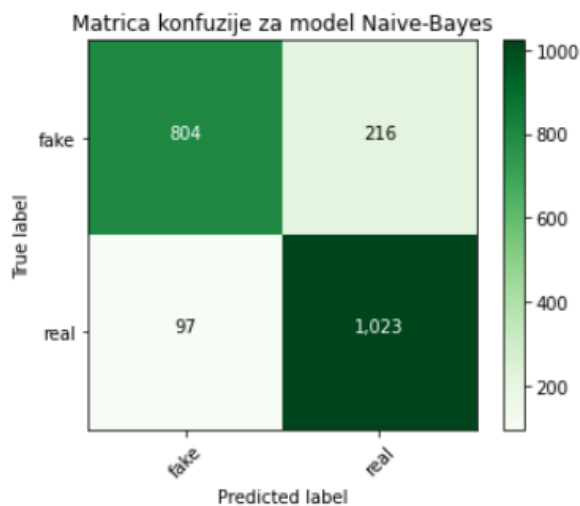
```

	precision	recall	f1-score	support
fake	0.89	0.79	0.84	1020
real	0.83	0.91	0.87	1120
accuracy			0.85	2140
macro avg	0.86	0.85	0.85	2140
weighted avg	0.86	0.85	0.85	2140

```

Točnost/Accuracy : 0.8537383177570094
Preciznost/Precision : 0.8606980140186915
Opoziv/Recall : 0.8537383177570094
F1 : 0.854579453977466

```



Slika 42. Rezultati za Naive-Bayes

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.8537, 0.8606, 0.8537, 0.8545.

3.2.7.KNN

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela KNN, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
knn = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', KNeighborsClassifier())
])
knn_fit = knn.fit(train_podaci['tweet'], train_podaci['label'])
print('KNN/K-Najbližih-Susjeda')
print ('Vrijednost:')
predikcija = knn.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'],predikcija),target_leveli=['fake','real'],
    normalize = False, title = 'Matrica konfuzije za model KNN')
```

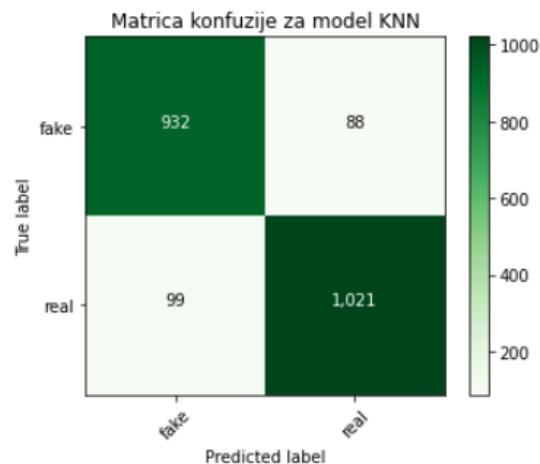
Slika 43. Funkcija KNN

```
KNN/K-Najbližih-Susjeda
Vrijednost:
[[ 932  88]
 [ 99 1021]]
           precision    recall  f1-score   support

      fake       0.90      0.91      0.91      1020
      real       0.92      0.91      0.92      1120

   accuracy                0.91      2140
  macro avg       0.91      0.91      0.91      2140
 weighted avg       0.91      0.91      0.91      2140

Točnost/Accuracy : 0.9126168224299065
Preciznost/Precision : 0.9126277111311814
Opoziv/Recall : 0.9126168224299065
F1 : 0.9125981098013863
```



Slika 44. Rezultati za KNN

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9126, 0.9126, 0.9126, 0.9125.

3.2.8.Slučajna šuma

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela Slučajne šume, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
slucajna_suma = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', RandomForestClassifier())
])
slucajna_suma_fit = slucajna_suma.fit(train_podaci['tweet'], train_podaci['label'])
print('Slučajna šuma/Random Forest')
print('Vrijednost:')
predikcija = slucajna_suma.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
    normalize = False, title = 'Matrica konfuzije za model Random Forest')
```

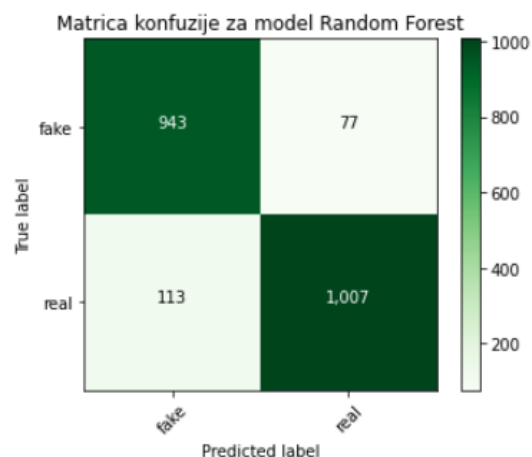
Slika 45. Funkcija Random Forest-a

```
Slučajna šuma/Random Forest
Vrijednost:
[[ 943  77]
 [ 113 1007]]
           precision    recall  f1-score   support

      fake      0.89      0.92      0.91      1020
      real      0.93      0.90      0.91      1120

   accuracy              0.91      2140
  macro avg              0.91      2140
 weighted avg              0.91      2140

Točnost/Accuracy : 0.9112149532710281
Preciznost/Precision : 0.911642287756224
Opoziv/Recall : 0.9112149532710281
F1 : 0.9111702454644861
```



Slika 46. Rezultati za Random Forest

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9112, 0.9116, 0.9112, 0.9111.

3.2.9. Adaboost

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela Adaboost, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
adaboost = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', AdaBoostClassifier())
])
adaboost_fit = adaboost.fit(train_podaci['tweet'], train_podaci['label'])
print('AdaBoost/Adaptive Boosting')
print('Vrijednost:')
predikcija = adaboost.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
    normalize = False, title = 'Matrica konfuzije za model AdaBoost')
```

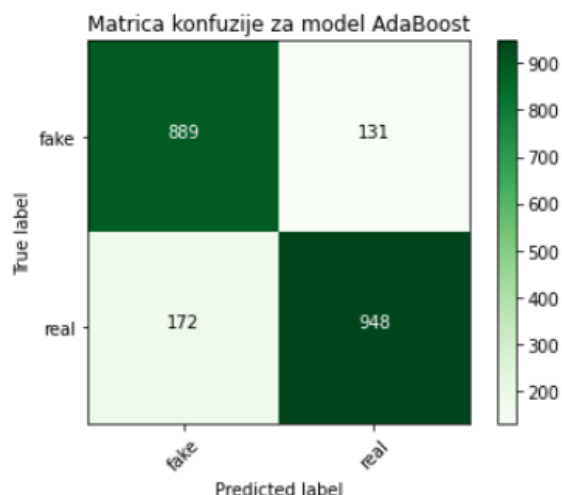
Slika 47. Funkcija Adaboost

```
AdaBoost/Adaptive Boosting
Vrijednost:
[[889 131]
 [172 948]]
      precision    recall  f1-score   support

   fake      0.84      0.87      0.85     1020
   real      0.88      0.85      0.86     1120

 accuracy                0.86     2140
 macro avg              0.86      0.86      0.86     2140
 weighted avg          0.86      0.86      0.86     2140

Točnost/Accuracy : 0.858411214953271
Preciznost/Precision : 0.8588928702322051
Opoziv/Recall : 0.858411214953271
F1 : 0.8583363691568597
```



Slika 48. Rezultati za Adaboost

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.8584, 0.8588, 0.8584, 0.8583.

3.2.10. Podizanje gradijenata

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela Gradient Boosting, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
gbm = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('classifier', GradientBoostingClassifier())
])
gbm_fit = gbm.fit(train_podaci['tweet'], train_podaci['label'])
print('Podizanje gradijenata/Gradient boosting')
print ('Vrijednost:')
predikcija = gbm.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
    normalize = False, title = 'Matrica konfuzije za model GBM')
```

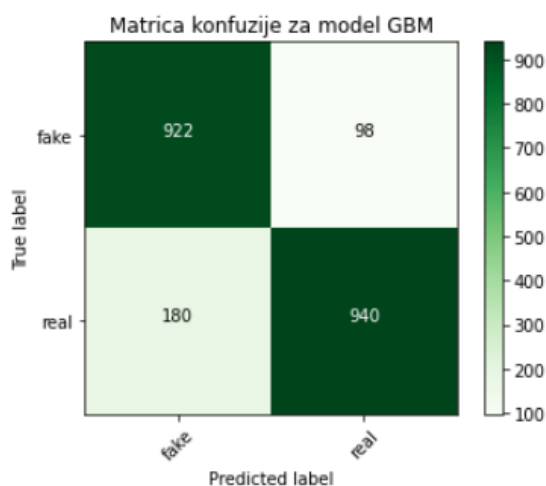
Slika 49. Funkcija Gradient Boosting

```
Podizanje gradijenata/Gradient boosting
Vrijednost:
[[922  98]
 [180 940]]
      precision    recall  f1-score   support

   fake      0.84      0.90      0.87      1020
   real      0.91      0.84      0.87      1120

 accuracy          0.87      2140
 macro avg          0.87      2140
weighted avg          0.87      2140

Točnost/Accuracy : 0.8700934579439252
Preciznost/Precision : 0.8725701589046834
Opoziv/Recall : 0.8700934579439252
F1 : 0.8700515862471604
```



Slika 50. Rezultati za Gradient Boosting

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.8700, 0.8725, 0.8700, 0.8700.

3.2.11. Podizanje gradijenata temeljeno na histogramu

Na slici ispod možemo vidjeti dio koda koji se odnosi na kreiranje modela „Histogram based Gradient Boosting“, te crtanje matrice konfuzije i ispis pripadnih mjera:

```
hist_gb = Pipeline([
    ('vectorizer', CountVectorizer()),
    ('statistical measure', TfidfTransformer()),
    ('to_dense', DenseTransformer()),
    ('classifier', HistGradientBoostingClassifier())
])
hist_gb_fit = hist_gb.fit(train_podaci['tweet'], train_podaci['label'])
print('Alogritam podizanja gradijenta koji se temelji na histogramu')
print ('Vrijednost:')
predikcija = hist_gb.predict(val_podaci['tweet'])
mjere_vrednovanja_klasifikacije(predikcija, val_podaci['label'])
matrica_konfuzije(confusion_matrix(val_podaci['label'], predikcija), target_leveli=['fake', 'real'],
    normalize = False, title = 'Matrica konfuzije za model Hist-GB')
```

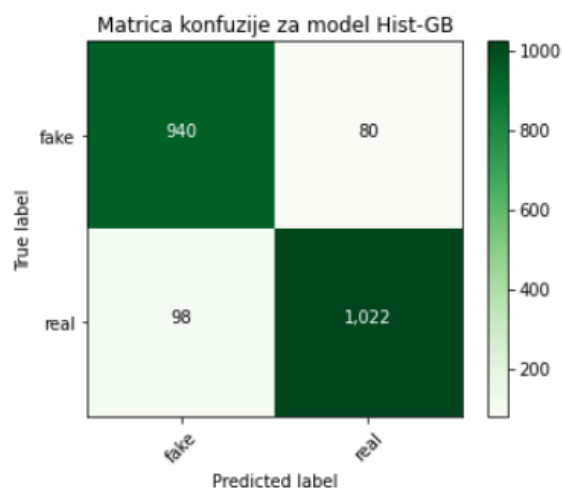
Slika 51. Funkcija Histogram based Gradient Boosting

```
Alogritam podizanja gradijenta koji se temelji na histogramu
Vrijednost:
[[ 940   80]
 [  98 1022]]
      precision    recall  f1-score   support

   fake      0.91      0.92      0.91      1020
   real      0.93      0.91      0.92      1120

 accuracy                0.92      2140
 macro avg              0.92      0.92      0.92      2140
 weighted avg          0.92      0.92      0.92      2140

Točnost/Accuracy : 0.9168224299065421
Preciznost/Precision : 0.9168987080813633
Opoziv/Recall : 0.9168224299065421
F1 : 0.9167955824491743
```



Slika 52. Rezultati za Histogram based Gradient Boosting

Na prethodnoj slici možemo vidjeti mjere vrednovanja accuracy, precision, recall i F1 koje redom iznose: 0.9168, 0.9168, 0.9168, 0.9167.

4.Rezultati

Mjere vrednovanja modela strojnoga učenja koje su u ovom radu korištene već smo prethodno naveli i opisali, a to su: „Matrica konfuzije/Confusion matrix“, „Točnost/Accuracy“, „Preciznost/Precision“, „Odziv/Recall“ te F1.

Točnost je u ovome diplomskom radu dobar pokazatelj izvedbe modela zato jer su skupovi podataka izbalansirani, odnosno imamo približno isti broj podataka koji se nalaze u klasi koja je označena kao „Real“, te u klasi koja je označena kao „Fake“. U slučaju da to nije vrijedilo, odnosno da je podjela na klase bila nebalansirana, trebali bi također promatrati i mjere kao što su „preciznost“, „odziv“ i „F1“. Međutim, bez obzira na to što je podjela bila balansirana, u ovome se radu koriste i te mjere, jer je to dobra praksa. Mjera „F1“ rezultat je harmonijske sredine između „preciznosti“ i „odziva“. Raspon za „F1“ rezultat, kao i za ostale mjere, je [0, 1], a on nam u slučaju „F1“ govori koliko je precizan naš klasifikator/model(koliko instanci ispravno klasificira), kao i koliko je robustan(ne propušta značajan broj instanci). Visoka „preciznost“ i niži „odziv“, daju vrlo visoki rezultat za „mjeru točnosti“, ali tada se propušta velik broj slučajeva koje je teško klasificirati. Stoga možemo zaključiti što je veći rezultat mjere „F1“, to su bolje performanse našeg modela.

Logistička regresija

Kada promatramo „matricu konfuzije“ za model logističke regresije možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 959, što znači da je 959 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 61, što znači da je 61 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1026, što znači da je 1026 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 94, što znači da su 94 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1985, dok je broj pogrešno klasificiranih vijesti 155. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, „preciznosti“, „odziva“ i „F1“.

Mjere vrednovanja točnost, preciznost, recall i F1 za model logističke regresije iznose: 0.9275, 0.9279, 0.9275, 0.9275. Možemo vidjeti kako su te mjere vrlo blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], pa možemo zaključiti kako je model logističke regresije veoma pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19.

Stohastički gradijentni silazak

Kada promatramo „matricu konfuzije“ za model stohastičkog gradijentnog silaska možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 957, što znači da je 957 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 63, što znači da je 63 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1043, što znači da je 1043 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 77, što znači da su 77 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 2000, dok je broj pogrešno klasificiranih vijesti 140. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model stohastičkog gradijentnog silaska iznose: 0.9345, 0.9346, 0.9345, 0.9345. Možemo vidjet kako su te mjere vrlo blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], čak su nešto bolje nego kod modela logističke regresije, pa možemo zaključiti kako je model stohastičkog gradijentnog silaska veoma pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19.

Stablo odlučivanja

Kada promatramo „matricu konfuzije“ za model stabla odlučivanja možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 836, što znači da je 836 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 184, što znači da je 184 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 973, što znači da je 973 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 147, što znači da su 174 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1809, dok je broj pogrešno klasificiranih vijesti 331. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model stabla odlučivanja iznose: 0.8453, 0.8461, 0.8453, 0.8454. Možemo vidjet kako su te mjere dosta blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela logističke regresije i stohastičkog gradijentnog silaska. Ipak možemo zaključiti kako je model stabla odlučivanja dosta pouzdan model na ovome skupu podataka te je dosta pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega.

SVM

Kada promatramo „matricu konfuzije“ za model SVM možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 963, što znači da je 963 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 57, što znači da je 57 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1037, što znači da je 1037 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 83, što znači da su 83 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 2000, dok je broj pogrešno klasificiranih vijesti 140. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model SVM iznose: 0.9345, 0.9348, 0.9345, 0.9345. Možemo vidjeti kako su te mjere vrlo blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], čak su nešto bolje nego kod dosada najbolje rangiranog modela stohastičkog gradijentnog silaska, pa možemo zaključiti kako je model SVM veoma pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, te trenutno rangira kao najbolje ocijenjeni algoritam/model koji se koristio na odabranom skupu podataka.

Naivni-Bayes

Kada promatramo „matricu konfuzije“ za model Naivni-Bayes možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 804, što znači da je 804 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 216, što znači da je 216 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1023, što znači da je 1023 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 97, što znači da su 97 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1827, dok je broj pogrešno klasificiranih vijesti 313. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model Naivni-Bayes iznose: 0.8537, 0.8606, 0.8537, 0.8545. Možemo vidjeti kako su te mjere dosta blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela SVM-a, stohastičkog gradijentnog silaska i logističke regresije, dok su nešto bolje nego kod modela stabla odlučivanja. Ipak možemo zaključiti kako je model Naivni-Bayes dosta pouzdan model na ovome skupu podataka te je dosta pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega.

KNN

Kada promatramo „matricu konfuzije“ za model KNN možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 932, što znači da je 932 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 88, što znači da je 88 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1021, što znači da je 1021 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 99, što znači da su 99 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1953, dok je broj pogrešno klasificiranih vijesti 187. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model KNN iznose: 0.9126, 0.9126, 0.9126, 0.9125. Možemo vidjeti kako su te mjere veoma blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela SVM-a, stohastičkog gradijentnog silaska i logističke regresije, dok su nešto bolje nego kod modela Naivni-Bayes i stabla odlučivanja, pa možemo zaključiti kako je model KNN visoko pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega.

Slučajna šuma

Kada promatramo „matricu konfuzije“ za model slučajne šume možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 943, što znači da je 943 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 77, što znači da je 77 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1007, što znači da je 1007 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 113, što znači da su 113 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1950, dok je broj pogrešno klasificiranih vijesti 190. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model KNN iznose: 0.9112, 0.9116, 0.9112, 0.9111. Možemo vidjeti kako su te mjere veoma blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela SVM-a, stohastičkog gradijentnog silaska, logističke regresije i KNN-a, dok su nešto bolje nego kod modela Naivni-Bayes i stabla odlučivanja, pa možemo zaključiti kako je model slučajne šume visoko pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega. Ovdje se također može potvrditi prethodna tvrdnja da je algoritam slučajne šume bolji od algoritma stabla odlučivanja, jer slučajna šuma uzima u obzir više stabla odlučivanja, a ne samo jedan.

AdaBoost

Kada promatramo „matricu konfuzije“ za model AdaBoost možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 889, što znači da je 889 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 131, što znači da je 131 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 948, što znači da je 948 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 172, što znači da su 172 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1837, dok je broj pogrešno klasificiranih vijesti 303. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model KNN iznose: 0.8584, 0.8588, 0.8584, 0.8583. Možemo vidjeti kako su te mjere dosta blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela SVM-a, stohastičkog gradijentnog silaska, logističke regresije, KNN-a i slučajne šume, dok su nešto bolje nego kod modela Naivni-Bayes i stabla odlučivanja, pa možemo zaključiti kako je model AdaBoost dosta pouzdan model na ovome skupu podataka te je dosta pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega.

Podizanje gradijenata

Kada promatramo „matricu konfuzije“ za model podizanja gradijenata možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 922, što znači da je 922 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 98, što znači da je 98 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 940, što znači da je 940 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 180, što znači da su 180 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1837, dok je broj pogrešno klasificiranih vijesti 303. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model KNN iznose: 0.8700, 0.8725, 0.8700, 0.8700. Možemo vidjeti kako su te mjere dosta blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela SVM-a, stohastičkog gradijentnog silaska, logističke regresije, KNN-a i slučajne šume, dok su nešto bolje nego kod modela AdaBoost, Naivni-Bayes i stabla odlučivanja, pa možemo zaključiti kako je model podizanja gradijenata dosta pouzdan model na ovome skupu podataka te je dosta pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega.

Podizanje gradijenata temeljeno na histogramu

Kada promatramo „matricu konfuzije“ za model podizanja gradijenata temeljenog na histogramu možemo iščitati sljedeće. Vidimo da je broj istinitih pozitiva 940, što znači da je 940 vijesti ispravno detektirano kao lažne vijesti. Također vidimo da je broj lažnih pozitiva 80, što znači da je 80 vijest detektirana kao lažna, a zapravo je istinita. Broj istinitih negativna je 1022, što znači da je 1022 vijesti ispravno detektirano kao istinite vijesti, dok je broj lažnih negativna 98, što znači da su 98 vijesti detektirane kao istinite, a zapravo su lažne. Po svemu tomu možemo zaključiti kako je ukupan broj dobro klasificiranih vijesti 1962, dok je broj pogrešno klasificiranih vijesti 178. Iz matrice konfuzije se ubiti očitavaju podaci koji su potrebni za računanje mjera „točnosti“, „preciznosti“, „odziva“ i „F1.

Mjere vrednovanja točnost, preciznost, recall i F1 za model KNN iznose: 0.9168, 0. 9168, 0. 9168, 0. 9167. Možemo vidjet kako su te mjere vrlo blizu gornjem dijelu raspona kojeg mogu poprimiti [0, 1], međutim rangiraju nešto niže nego kod modela: SVM-a, stohastičkog gradijentnog silaska i logističke regresije, dok su nešto bolje nego kod modela: KNN, slučajne šume, podizanja gradijenta, AdaBoost, Naivni-Bayes i stabla odlučivanja, pa možemo zaključiti kako je model podizanja gradijenta temeljenog na histogramu visoko pouzdan model na ovome skupu podataka te je visoko pouzdan u detekciji lažnih/istinitih vijesti vezanih uz COVID-19, međutim postoje i bolji algoritmi, odnosno modeli od njega. Ovdje možemo zaključiti kako se potvrdilo da algoritam podizanja gradijenata ima bolje performanse kada koristi histograme, kao što je i prethodno bilo navedeno.

5.Tablica rezultata

	Točnost	Preciznost	Odziv	F1
Logistička regresija	0.9275	0.9279	0.9275	0.9275
Stohastički gradijentni silazak	0.9345	0.9346	0.9345	0.9345
Stablo odlučivanja	0.8453	0.8461	0.8453	0.8454
SVM	0.9345	0.9348	0.9345	0.9345
Naivni-Bayes	0.8537	0.8606	0.8537	0.8545
KNN	0.9126	0.9126	0.9126	0.9125
Slučajna šuma	0.9112	0.9116	0.9112	0.9111
AdaBoost	0.8584	0.8588	0.8584	0.8583
Podizanje gradijenta	0.8700	0.8725	0.8700	0.8700
Podizanje gradijenta temeljeno na histogramu	0.9168	0.9168	0.9168	0.9167

Tablica 2. Mjere za testirane algoritme strojnog učenja

Iz prethodne tablice možemo vidjeti kako ja za najvažniju mjeru, odnosno za mjeru F1 rangiranje sljedeće: 1)SVM, 2)SGC, 3)Logistic Regression, 4)Histogram Based Gradinet Boosting, 5)KNN, 6)Random Forests, 7)Gradient Boosting, 8)AdaBoost, 9)Naive-Bayes te 10)Decision Tree.

5. Zaključak

Iako je od pojave prvog slučaja virusa COVID-19 prošlo više od godinu i pol dana, svijet još uvijek prolazi kroz smrtonosnu pandemiju uzrokovanu tim virusom, a lažne vijesti o bolesti, njezinim lijekovima, prevenciji i uzrocima još uvijek se s namjerom plasiraju širom svijeta milijunima ljudi.

Kroz ovaj rad dobili smo uvid u to kako širenje tih lažnih vijesti tijekom ovakvog nesigurnog vremena može imati ozbiljne posljedice što dovodi do raširene panike i pojačavanja prijetnje pandemije, pa čak i do kobnih posljedica kao što su gubitci ljudskih života. Baš zbog toga je od presudne važnosti ograničiti širenje lažnih vijesti i osigurati da se točne informacije prenose javnosti. Upravo ovdje važnu ulogu imaju algoritmi strojnog učenja za detekciju širenja lažnih vijesti koji pomažu u detekciji i sprečavanju širenja lažnih vijesti vezanih uz COVID-19. Pregledavši dosadašnju literaturu na ovu temu da se lako zaključiti kako su se mnogi prethodni radovi bazirali na izgradnji sustava za otkrivanje lažnih vijesti. Većina od tih radova se oslanjala na ručno označene podatke za postupak učenja modela u svrhu automatskog otkrivanja lažnih vijesti, kao što je slučaj i u ovome diplomskom radu. U ovome radu na odabranom skupu podataka testirano je deset algoritama strojnog učenja: Logistic Regression(logistička regresija), SGC(stohastički gradijenti silazak), Decision Tree(stablo odlučivanja), SVM(Support Vector Machine), Naive-Bayes(Naivni Bayes), KNN(K-Najbližih-Susjeda), Random Forest(slučajna šuma), AdaBoost, Gradient Boosting(podizanje gradijenata), te Histogram Based Gradient Boosting(podizanje gradijenata temeljeno na histogramu).

Nakon proučavanja rezultata za svaki od navedenih algoritama napravljeno je rangiranje od najboljeg prema najlošijem algoritmu koje redom izgleda ovako: 1)SVM, 2)SGC, 3)Logistic Regression, 4)Histogram Based Gradient Boosting, 5)KNN, 6)Random Forests, 7)Gradient Boosting, 8)AdaBoost, 9)Naive-Bayes te 10)Decision Tree. Prema tim rezultatima možemo vidjeti kako je najbolji algoritam koji je bio testiran u ovome radu SVM, odnosno „Support Vector Machine“, što je veoma zanimljivo jer je i jedan od radova koji je referenciran prethodno, a koji je također testirao mjeru preciznosti na različitim algoritmima strojnog učenja, također postigao najbolje rezultate za algoritam SVM-a. Isto tako po rezultatima možemo vidjeti da je najlošiji algoritam „Decision Tree“.

Također treba napomenuti da su proučena dva rada koja su se temeljila na istom datasetu pa ćemo spomenuti i njihove rezultate. Prvi rad se bavio isprobavanjem BERT-like i sličnih modela kojima je zajedničko da se baziraju na transformerima, te je u tome radu postignuta čak i veća točnost nego ona koja je dobivena u ovome radu, ona iznosi preko 98% za F1 mjeru pa možemo zaključiti kako klasični algoritmi strojnog učenja koji su isprobavani u ovome radu malo zaostaju u odnosu na te novije algoritme [65].

Drugi rad se bavio isprobavanjem klasičnih algoritama strojnog učenja kao što je slučaj i u ovome diplomskom radu, međutim isprobana su samo četiri algoritma, a to su: „SVM“, „Logistička regresija“, „Decision Tree“ i „Gradient boosting“. Algoritam koji je u tome radu postigao najveću točnost je SVM što je slučaj i u ovome radu [67].

S obzirom na to da su rezultati za svaki od dobivenih modela veoma dobri, potvrđuje se premisa rada kako su algoritmi strojnog učenja veoma korisni alati za detekciju i sprečavanje širenja lažnih vijesti vezanih uz COVID-19. To posebno vrijedi u današnjem svijetu, kada se podaci mjere u ogromnim količinama, te ručna provjera vijesti koje ti skupovi podataka sadrže nije moguća, a ni isplativa.

6.Literatura

- [1] Della Vedova, Marco & Tacchini, Eugenio & Moret, Stefano & Ballarin, (2018). Automatic Online Fake News Detection Combining Content and Social Signals. Preuzeto: Svibanj. 02, 2021. [Online]. Link: https://www.researchgate.net/publication/325735377_Automatic_Online_Fake_News_Detection_Combining_Content_and_Social_Signals
- [2] Frontiers in Artificial Intelligence, Case Report: Utilazing AI and NLP to Assist with Healthcare and Rehabilitation During the COVID-19 Pandemic. Preuzeto: Svibanj. 02, 2021. [Online]. Link: <https://www.frontiersin.org/articles/10.3389/frai.2021.613637/full>
- [3] Edu, The Now, What is fake news?. Preuzeto: Svibanj. 02, 2021. [Online]. Link: <https://edu.gcfglobal.org/en/thenow/what-is-fake-news/1/>
- [4] Reuters Institute, Dr. J. Scott Brennan, Felix Simon, Dr Philip N. Howard, Prof. Rasmus Kleis Nielsen. Types, sources, and claims of COVID-19 misinformation. Preuzeto: Svibanj. 02, 2021. [Online]. Link: http://www.primaonline.it/wp-content/uploads/2020/04/COVID-19_reuters.pdf
- [5] World Health Organization Official Website. Preuzeto: Svibanj. 02, 2021.[Online]. Link: <https://covid19.who.int/>
- [6] Coronavirus disease (COVID-19) Weekly Epidemiological Update and Weekly Operational Update, WHO. Svibanj. 03, 2021. [Online]. Link: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports>
- [7] WorldOMeter, Coronavirus, COVID-19 coronavirus pandemic. Preuzeto: Svibanj. 03, 2021. [Online]. Link: <https://www.worldometers.info/coronavirus/>
- [8] [9] Hua J., Shaw R. Corona virus (covid-19) infodemic and emerging issues through a data lens: the case of china. Preuzeto: Svibanj. 03, 2021. [Online]. Link: <http://dx.doi.org/10.3390/ijerph17072309>
- [10] Eysenbach G. Infodemiology: The epidemiology of (mis) information. The American journal of medicine, vol. 113, no. 9, pp. 763–765; 2002. Preuzeto: Svibanj. 03, 2021. [Online]. Link: [https://doi.org/10.1016/S0002-9343\(02\)01473-0](https://doi.org/10.1016/S0002-9343(02)01473-0)
- [11] Eysenbach G. Infodemiology and infoveillance: framework for an emerging set of public health informatics methods to analyze search, communication and publication behavior on the Internet. Journal of medical Internet research, vol. 11, no. 1, p. e11; 2009. Preuzeto: Svibanj. 03, 2021. [Online]. Link: [10.2196/jmir.1157](https://doi.org/10.2196/jmir.1157)
- [12] Eysenbach G. Infodemiology and infoveillance: tracking online health information and cyberbehavior for public health. American journal of preventive medicine, vol. 40, no. 5, pp. S154–S158; 2011. Preuzeto: Svibanj. 03, 2021. [Online]. Link: <https://doi.org/10.1016/j.amepre.2011.02.006>
- [13] [14] [15] [34] Sear R.F., Velásquez N., Leahy R., Restrepo N.J., Oud S.E., Gabriel N., Lupu Y., Johnson N.F. Quantifying covid-19 content in the online health opinion war using machine learning. Preuzeto: Svibanj. 03, 2021. [Online]. Link: <https://ieeexplore.ieee.org/abstract/document/9091126>
- [16] Centric, Machine Learning: A Quick Introduction and Five Core Steps. Preuzeto: Svibanj. 03, 2021. [Online]. Link: <https://centricconsulting.com/blog/machine-learning-a-quick-introduction-and-five-core-steps/>
- [17] . Kumar and R. Verma, “KDD techniques: A survey,” Int. J. Electron.Comput, Preuzeto: Svibanj. 05 2021. [Online]. Link: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.348.5838&rep=rep1&type=pdf>

- [18] R. K. Kaliyar and N. Singh, "Misinformation Detection on Online Social Media-A Survey," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Preuzeto: Svibanj. 10, 2021. [Online]. Link: <https://ieeexplore.ieee.org/document/8944587>
- [19] ResearchGate, Xichen Zhang, Ali.A.Ghorbani, Characterization, detection, and discussion Preuzeto: Svibanj.10,2021.[Online].Link:https://www.researchgate.net/publication/331913505_An_overview_of_online_fake_news_Characterization_detection_and_discussion
- [20] Collins, Botambu & Hoang, Dinh Tuyen & Nguyen, Ngoc Thanh & Hwang, Dosam. (2020). Trends in combating fake news on social media – a survey. Preuzeto: Svibanj. 11, 2021. [Online]. Link: https://www.researchgate.net/publication/347218658_Trends_in_combating_fake_news_on_social_media_-_a_survey
- [21] B. Al Asaad and M. Erascu, "A Tool for Fake News Detection," *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. Preuzeto: Svibanj. 11, 2021. [Online]. Link:https://www.researchgate.net/publication/347218658_Trends_in_combating_fake_news_on_social_media
- [22] ACL Anthology, Kai Nakamura, Sharon Levy, A New Multimodal Benchmark Dataset for Fake News Detection. Preuzeto: Svibanj. 11, 2021. [Online]. Link: <https://www.aclweb.org/anthology/2020.lrec-1.755/>
- [23] ResearchGate, Ibrishimova, Kin Fun. (2020). A Machine Learning Approach to Fake News Detection Using Knowledge Verification and Natural Language Processing. Preuzeto: Svibanj. 12, 2021. [Online]. Link: https://www.researchgate.net/publication/335191041_A_Machine_Learning_Approach_to_Fake_News_Detection_Using_Knowledge_Verification_and_Natural_Language_Processing/citation/download
- [24] Semantic Sgholar, J. Posadas-Durán, Helena Gómez-Adorno, Jesús Jaime Moreno Escobar, Detection of fake news in Spanish Language. Preuzeto: Svibanj. 12, 2021. [Online]. Link: <https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs179034>
- [25] Y. Wang, “‘liar, liar pants on Fire’: A new benchmark dataset for fakenews detection, Preuzeto: Svibanj. 13, 2021. [Online]. Link: <https://arxiv.org/abs/1705.00648>
- [26] [27] Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, “FEVER: A large-scale dataset for fact extraction and verification,” Preuzeto: Svibanj. 13, 2021. [Online]. Link: <https://www.aclweb.org/anthology/N18-1074/>
- [28] Brennen J., Simon F., Howard P., Nielsen R. 2020. Types, Sources, and Claims of COVID-19 Misinformation. Preuzeto: Svibanj. 14, 2021. [Online]. Link: <https://reutersinstitute.politics.ox.ac.uk/types-sources-and-claims-covid-19-misinformation>
- [29] Kouzy R., Jaoude J.A., Kraitem A. Coronavirus goes viral Preuzeto: Svibanj. 14, 2021. [Online]. Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7152572/>
- [30] Jmir Publications, Jia Xue, Junxiang Chen, Ran Hu, Chen Chen, Chengda Zheng, Yue Su, Tingshao Zhu. Twitter Discussions and Emotions About the COVID-19 Pandemic: Machine Learning Approach. Preuzeto: Svibanj. 15, 2021. [Online]. Link: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7690968/>
- [31] Pennycook G., McPhetres J., Zhang Y. Fighting covid-19 misinformation on social media. Preuzeto: Svibanj. 15, 2021. [Online]. Link: <https://journals.sagepub.com/doi/full/10.1177/0956797620939054>
- [32] R. Pandey, V. Gautam. A machine learning application for raising wash awareness in the times of Covid-19. Preuzeto: Svibanj. 15, 2021. [Online]. Link: <https://www.mdpi.com>
- [33] Leahy R., Goldberg B., Johnson N.F. Hate Multiverse Spreads Malicious Covid-19 Content Online Beyond Individual Platform Preuzeto: Svibanj. 16, 2021. [Online]. Link: <https://arxiv.org/abs/2004.00673>

- [35] Bruns A., Harrington S., Hurcombe E. corona? 5 g? or both?'. Preuzeto: Svibanj. 22, 2021. [Online]. Link: <https://eprints.qut.edu.au/203086/>
- [36] Mejova Y., Kalimeri K. Covid-19 on facebook ads: competing agendas around a public health crisis. Preuzeto: Svibanj. 16, 2021. [Online]. Link: https://www.researchgate.net/publication/342613149_COVID-19_on_Facebook_Ads_Competing_Agendas_around_a_Public_Health_Crisis
- [37] Cornell University, Shahan Ali Memon, Kathleen M. Carley, Characterizing COVID-19 Misinformation Communities Using a Novel Twitter Dataset. Preuzeto: Svibanj. 16, 2021. [Online]. Link: <https://arxiv.org/abs/2008.00791>
- [38] First Monday, Emilio Ferrara, What thypes of COVID-19 conspiracies, are populated by Twitter bots? Preuzeto: Svibanj. 17, 2021. [Online]. Link: <https://firstmonday.org/ojs/index.php/fm/article/view/10633/9548>
- [39] Springer Link, Lisa Singh, Understanding high- and low-quality URL Sharing on COVID-19 Twitter streams. Preuzeto: Svibanj. 17, 2021. [Online]. Link: <https://firstmonday.org/ojs/index.php/fm/article/view/10633/9548>
- [40] Towards Data Science, Saishruthi Sqaminathan, Logistic Regression – Detailed Overvie. Preuzeto: Svibanj. 19, 2021. [Online]. Link: <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>
- [41] Sebastian Ruder, An overview of gradient descnet optimization algorithms. Preuzeto: Svibanj. 20, 2021. [Online]. Link: <https://ruder.io/optimizing-gradient-descent/>
- [42] Lucidchart, What is a Decision Tree Diagram. Preuzeto: Svibanj. 20, 2021. [Online]. Link: <https://www.lucidchart.com/pages/decision-tree>
- [43] Analytics Vidhya, Understanding Support Vector Machine(SVM) algorithm from examples(along with code). Preuzeto: Svibanj. 20, 2021. [Online]. Link: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- [44] Analytics Vidhya, Sunik Ray, 6 Easy Steps to Lern Naive Bayes Algorithm with codes in Python and R. Preuzeto: Svibanj. 20, 2021. [Online]. Link: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>
- [45] Towards Dana Science, Onel Harrison, Machine Learning Basics with the K-Nearest Neighbors Algorithm. Preuzeto: Svibanj. 20, 2021. [Online]. .Link: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
- [46] Berkley.edu, Leo Breiman, Adele Cutler, Random Forests. Preuzeto: Svibanj. 21, 2021. [Online]. Link: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- [47] Investopedia, Alexandra Twin, Overfitting, Preuzeto: Svibanj. 21, 2021. [Online]. Link: <https://www.investopedia.com/terms/o/overfitting.asp>
- [48] TechTalks, Ben Dickson, What is ensemble learning?. Preuzeto: Svibanj. 21, 2021. [Online]. Link: <https://bdtechtalks.com/2020/11/12/what-is-ensemble-learning/>
- [49] ScienceDirect, Rajdeep Chatterjee, D.K. Sanyal, Machine Learning in Bio-Signal and Diagnostic Imaging. Preuzeto: Svibanj. 22, 2021. [Online]. Link: <https://www.sciencedirect.com/topics/engineering/adaboost>
- [50] Analytics Vidhya, Sunil Ray, Quick Introduction to Boosting Algorithms in Machine Learning. Preuzeto: Svibanj. 27, 2021. [Online]. Link: <https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/>

- [51] Machine Learning Mastery, Jason Brownlee, A gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. Preuzeto: Svibanj. 22, 2021. [Online]. Link: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>
- [52] Machine Learning Mastery, Jason Brownlee, Histogram-Based Gradient Boosting Ensembles Python. Preuzeto: Svibanj. 22, 2021. [Online]. Link: <https://machinelearningmastery.com/histogram-based-gradient-boosting-ensembles/>
- [53] Jupyter, Preuzeto: Svibanj. 17, 2021. [Online]. Link: <https://jupyter.org/>
- [54] Python, What is Python? Executive summary, Preuzeto: Svibanj. 19, 2021. [Online]. Link: <https://www.python.org/doc/essays/blurb/>
- [55] ScienceDirect, Ajay Kulkarni, Confusion Matrix. Preuzeto: Svibanj. 22, 2021. [Online]. Link: <https://www.sciencedirect.com/topics/engineering/confusion-matrix>
- [56] [56] [58] [59] Exilio Solutions, Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. Preuzeto: Svibanj. 23, 2021. [Online]. Link: <https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/>
- [60] Wikipedia, Indeksiranje. Preuzeto: Svibanj. 23, 2021. [Online]. Link: <https://hr.wikipedia.org/wiki/Indeksiranje>
- [61] Wikimedia Commons, Kako prepoznati lažne vijesti. Preuzeto: Svibanj. 03, 2021. [Online]. Link: [https://commons.wikimedia.org/wiki/File:Kako_prepoznati_la%C5%BEn_e_vijesti_\(How_To_Spot_Fake_News\).jpg](https://commons.wikimedia.org/wiki/File:Kako_prepoznati_la%C5%BEn_e_vijesti_(How_To_Spot_Fake_News).jpg)
- [62] zzjz-zz.hr, Preporuke i upute vezane za sigurnost stanovništva, Preuzeto: Svibanj. 03, 2021. [Online]. Link: <https://www.zzzjz-zz.hr/covid-19/>
- [63] Abc Net Au, As the coronavirus spread, an experiment showed Facebook was struggling to keep up with fake news, Preuzeto: Svibanj. 04, 2021. [Online]. Link: <https://www.abc.net.au/news/2020-04-24/facebook-approves-ads-with-covid-19-misinformation/12172168>
- [64] Twitter, S&D Group, Warning: Watch out for #COVID19 fake news. Preuzeto: Svibanj. 04, 2021. [Online]. Link: <https://twitter.com/theprogressives/status/1240693648750022658>
- [65] GitHub, diptamath/covid_fake_news/data/, S&D Group, Warning: Watch out for #COVID19 fake news. Preuzeto: Svibanj. 01, 2021. [Online]. Link: https://github.com/diptamath/covid_fake_news/tree/main/data
- [66] CodaLab, Constraint@AAAI2021 - COVID19 Fake News Detection in English. Preuzeto: Svibanj. 01, 2021. [Online]. Link: <https://competitions.codalab.org/competitions/26655>
- [67] Arxiv, Fighting an Infodemic: COVID-19, Fake News Dataset, Parth Patwa, Shivam Sharma, Srinivas. Preuzeto: Svibanj. 23, 2021. [Online]. Link: <https://arxiv.org/ftp/arxiv/papers/2011/2011.03327.pdf>

7. Popis slika

Slika 1. Na koje sve načine možemo prepoznati lažne vijesti?	9
Slika 2. Upute u slučaju zaraze koronavirusom	11
Slika 3. Lažna vijest vezana uz COVID-19 koja se širila Facebookom	15
Slika 4. Neke od lažnih vijesti koje su se širile putem Twittera	16
Slika 5. Model Logističke regresije	18
Slika 6. Funkcije parabole s parametrima x i y	18
Slika 7. Struktura Stabla odlučivanja	20
Slika 8. Model SVM-a	21
Slika 9. Konkretni primjer korištenja SVM-a	21
Slika 10. Bayesov teorem	22
Slika 11. Prikaz jednostavnog KNN modela	23
Slika 12. Dijagram rada algoritma Slučajne šume	24
Slika 13. Princip rada AdaBoost algoritma	26
Slika 14. Model rada GBM-a	27
Slika 15. Nacrt matrice konfuzije	29
Slika 16. Instalacija "Jupyter Notebooka"	36
Slika 17. Otvaranje Notebook-a u Mozilli	36
Slika 18. Učitavanje biblioteka i podataka u varijable, ispis informacija za skup "train_data.xlsx"	37
Slika 19. Informacije za skupove "test_data_with_labels.xlsx" i "validation_data.xlsx"	38
Slika 20. Informacije objedinjenog skupa	39
Slika 21. Statistika istinitih/lažnih tweetova	39
Slika 22. Isječak koda u kojem se definiraju varijable za tokene i funkcija konkatencija	40
Slika 23. Generiranje wordclouda za sve tweetove	41
Slika 24. Generiranje wordclouda za lažne tweetove	42
Slika 25. Generiranje wordclouda za istinite tweetove	43
Slika 26. Podjela tweetova u željene skupine	44
Slika 27. Importiranje potrebnih paketa i funkcija	45
Slika 28. Funkcija za crtanje matrice konfuzije	46
Slika 29. Funkcija za uklanjanje stop riječi	46
Slika 30. Mjere vrednovanja klasifikacije	47
Slika 31. Učitavanje skupova podataka u varijable	47
Slika 32. Funkcija logističke regresije	48
Slika 33. Rezultati za logističku regresiju	48
Slika 34. Funkcija SGD-a	49
Slika 35. Rezultati za SGD	49
Slika 36. Funkcija stabla odlučivanja	50
Slika 37. Rezultati za stablo odlučivanja	50
Slika 38. Funkcija za SVM	51
Slika 39. Rezultati za SVM	51
Slika 40. Klasa DenseTransformer	52
Slika 41. Funkcija Naive-Bayes	52

Slika 42. Rezultati za Naive-Bayes	53
Slika 43. Funkcija KNN	54
Slika 44. Rezultati za KNN	54
Slika 45. Funkcija Random Forest-a	55
Slika 46. Rezultati za Random Forest.....	55
Slika 47. Funkcija Adaboost.....	56
Slika 48. Rezultati za Adaboost.....	56
Slika 49. Funkcija Gradient Boosting.....	57
Slika 50. Rezultati za Gradient Boosting.....	57
Slika 51. Funkcija Histogram based Gradient Boosting.....	58
Slika 52. Rezultati za Histogram based Gradient Boosting.....	58

8. Popis tablica

Tablica 1. Top 25 riječi ovisno o skupini.....	44
Tablica 2. Mjere za testirane algoritme strojnog učenja	65