Analiza sentimenta novinskih članaka vezanih za tematiku koronavirusa

Ilić, Anton

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://urn.nsk.hr/urn:nbn:hr:195:789102

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

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

Sveučilište u Rijeci
Fakultet informatike
i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

zir.nsk.hr

DIGITALNA KNJIŽNICA

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Anton Ilić

Analiza sentimenta novinskih članaka vezanih za tematiku koronavirusa

Diplomski rad

Mentor: *izv. prof. dr. sc.* Ana Meštrović

Komentor: *dr. sc.* Slobodan Beliga

Rijeka, rujan 2021.

ODJEL ZA INFORMATIKU
SVEUČILIŠTE U RIJECI

Rijeka, 13.4.2021.

Zadatak za diplomski rad

Pristupnik: Anton Ilić

Naziv diplomskog rada: Analiza sentimenta novinskih članaka vezanih za tematiku koronavirusa
Naziv diplomskog rada na eng. jeziku: Sentiment Analysis of Coronavirus Related News Articles

Sadržaj zadatka:
Cilj diplomskog rada je prikupiti tekstualne podatke s odabranog novinskog (internetskog) portala za određeno vremensko razdoblje. Prikupljene tekstualne podatke potrebno je organizirati i pohraniti u oblik prikladnom za obradu nestrukturiranih podataka. Prilikom odabira novinskih objava bitno je odabrati one koje imaju označene reakcije čitatelja u obliku emocija (tzv. *emoji* reakcija). Potrebno je opisati postupak prikupljanja podataka te mjerama deskriptivne statistike izračunati udjele novinskih članaka vezane za tematiku koronavirusa. Osim toga, potrebno je načiniti podskup novinskih članaka vezanih za tematiku koronavirusa prema definiranom COVID-19 tezaurusu koji će se koristiti u zadatku analize sentimenta korona objava. Podatkovni skup vezan za korona objave potrebno je podijeliti u skupove za strojno učenje i testiranje. U postupku pripreme podataka treba zadržati samo one članke koji imaju označene emocije od najmanje 3 čitatelja (najmanje 3 *emoji* reakcije čitatelja). Model za predikciju sentimenta klasificirati će svaki novinski članak u pozitivnu, neutralnu ili negativnu klasu. Za analizu sentimenta na svim tekstovima je u inicijalnom koraku potrebno odraditi standardne korake predobrade i normalizacije teksta te definirati skale na kojima će se mjeriti sentiment (vrijednosti iz intervala od -1 do 1) i način izračuna vrijednosti sentimenta koji uzima u obzir više različitih reakcija (npr. ljutnja, oduševljenje, sreća i sl.). Prvi model za predikciju sentimenta konstruirati nenadziranim pristupom (npr. koristeći tezaurus sentimenta), a drugi inducirati nadziranim pristupom koristeći standardne modele strojnog učenja (npr. SVM). Dobivene rezultate koji predviđaju sentiment potrebno je evaluirati standardnim mjerama.

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

Voditeljica za diplomske radove:
izv. prof. dr. sc. Ana Meštrović

Komentor:
dr. sc. Slobodan Beliga

Zadatak preuzet:

(potpis pristupnika)

Sažetak

Početkom 2020. godine, pandemija novog koronavirusa *SARS-Cov-2* je zahvatila cijeli svijet te svojim utjecajem promijenila naše načine života u društvenom i digitalnom smislu. Cilj ovog rada je utvrditi utjecaj pandemije na medijsko izvještavanje te izmjeriti emocionalni utjecaj tog izvještavanja na čitatelje programskim putem metodama analize sentimenta. U tu svrhu je prikazan cijeli postupak rudarenja mišljenja od faze automatiziranog prikupljanja i strukturiranja podataka, do njihove obrade i analize. U fazi analize su prikazane osnove statističkih opažanja iz prikupljenih podataka te njihova interpretacija u kontekstu pandemije i naravi ljudskog mišljenja u negativnom i ne-negativnom smislu. Nadalje u radu je demonstriran rad i provedena usporedba efikasnosti korištenja konvencionalnih lingvističkih metoda s modernijim pristupima baziranih na umjetnoj inteligenciji.

Ključne riječi: analiza sentimenta, COVID-19, novinski portal, online vijesti, Python, strojno učenje, nenadzirano učenje

Abstract

In early 2020, a pandemic of the novel coronavirus *SARS-Cov-2* spread to the entire world and its impact changed our ways of life in a social and digital sense. The aim of this paper is to determine the impact of the pandemic on media reporting and to measure the emotional impact of this reporting on readers through programmatic methods of sentiment analysis. For this purpose, the whole process of opinion mining is presented which includes the phase of automated data collection and structuring, through to their processing and analysis. The analysis phase presents the basics of statistical observations from the collected data and their interpretation in the context of the pandemic and the nature of human opinion in a negative and non-negative sense. Furthermore, the paper demonstrates the work and compares the efficiency of using conventional linguistic methods with more modern approaches based on artificial intelligence.

Tablica sadržaja

1. Uvod 1

2. Analiza sentimenta 3

2.1. Algoritmi nenadziranog učenja 5

2.1.1. Pristup baziran na rječniku 6

2.1.2. Pristup baziran na korpusu 6

2.2. Algoritmi nadziranog učenja 7

2.2.1. Naivni Bayes 7

2.2.2. Metoda potpornih vektora 8

2.2.3. Algoritam slučajnih šuma 9

3. Izvor podataka 11

4. Prikupljanje podataka 14

4.1. Prikupljanje poveznica 14

4.2. Prikupljanje sadržaja članaka 18

5. Obrada podataka 20

5.1. Identifikacija COVID-19 članaka 20

5.2. Svođenje riječi na osnovni oblik 22

6. Analiza podataka 23

6.1. Skala pozitivnosti prema reakcijama korisnika 23

6.2. Metrika korištena u analizi 26

6.2.1. Točnost 27

6.2.2. Preciznost 27

6.2.3. Odziv 28

6.2.4. F1 mjera 28

6.3. Statistička obilježja skupa podataka 29

6.4. Klasifikacija prema polaritetu sentimenta 33
6.4.1. Analiza sentimenta na razini tekstualnog elementa 33
6.4.2. Analiza sentimenta na razini rečenice 36
6.4.3. Analiza sentimenta na razini rečenice bibliotekom VADER 38
6.4.4. Grafički prikaz toka polariteta kod nenadziranog učenja 41
6.5. Klasifikacija algoritmima nadziranog učenja 43
6.5.1. Balansiranje skupa podataka 43
6.5.2. Vektorizacija i podjela podataka na skupove za treniranje i testiranje 45
6.5.3. K-Fold unakrsna validacija 45
6.5.4. Naivni Bayesov klasifikator 46
6.5.5. Klasifikator metode potpornih vektora 48
6.5.6. Klasifikator algoritma slučajnih šuma 50
7. Usporedba rezultata 52
8. Zaključak 55
9. Popis literature 57
Popis slika 62
10. Popis tablica 63
11. Popis priloga 64

1. Uvod

Razvojem interaktivnog 2.0 Web-a te olakšanom dostupnošću informacijske tehnologije, čovječanstvu je omogućen novi način komunikacije, a distribucija slobodnog vremena je zauvijek promijenjena. Suvremena mrežna mjesta su postale platforme na kojima korisnik može stvarati vlastiti sadržaj koji se nerijetko sastoji od mišljenja i stavova. Neki od primjera modernih web stranica uključuju *online* forume, portale s vijestima te razne forme koje omogućuju pisanje recenzija, komentara i odgovora drugim korisnicima tih usluga.

Razvoj Weba je u neprestanom zamahu, od samih njegovih početaka, a novonastala situacija uzrokovana pandemijom bolesti COVID-19 je korištenje Interneta dodatno intenzivirala. COVID-19 je infektivna respiratorna bolest koja se je pojavila krajem 2019. godine, a uzrokovana je novootkrivenim virusom iz porodice koronavirusa. Simptomi bolesti su kod većine oboljelih slabog do srednjeg intenziteta, a uključuju visoku temperaturu, respiratorne smetnje, gubitak mirisa i okusa [1]. Virus se je brzo nakon otkrića u kineskom gradu Wuhanu, zbog velikog potencijala zaraznosti, proširio po cijelom svijetu, nakon čega je proglašena globalna pandemija. Refleksni odgovor svih država na nepoznatu i potencijalnu prijetnju se je sastojao od preventivnih socijalnih mjera [2]. Neke od tih mjera su ograničavanje socijalnog kontakta, rad i školovanje od kuće, ograničavanje i zabrana rada te policijski sat. Zbog same prirode tih mjera, pandemija je postala središnja tema svih prenositelja vijesti, a ljudska komunikacija izvan kućanstva je postala potpuno ovisna o internetskim uslugama.

Ukoliko Internet promatramo kao nestrukturiranu bazu podataka, prebacivanjem cjelokupnog društvenog života *online*, količina podataka je nemjerljivo porasla. Nestrukturirani podaci su neuređeni tip podataka najčešće tekstualnog oblika koji nemaju definiranu strukturu pohrane kao tradicionalne baze podataka koje podatke spremaju u stupce [3]. Korisnici su korištenjem web usluga predali velike količine podataka koristeći se značajkama komentiranja, objava i ostalih komunikacijskih kanala. Nadalje, web portali su osim izvještavanja o svakodnevnim vijestima veliki dio svojih resursa uložili u dnevno izvještavanje o tijeku pandemije i valovima u kojima se ona manifestira. Na taj način je u domeni znanosti o podacima otvorena prilika za obradu i analizu tih nestrukturiranih podataka čime je moguće steći nova znanja na temu pandemije, ljudskog ponašanja i analize mišljenja, što je i glavna tema ovog rada. Rad je podijeljen

na više dijelova, a obuhvaća cijeli proces predstavljanja problema, prikupljanja, obrade i analize podataka. Glavni cilj je provesti analizu sentimenta, odnosno analizirati mišljenja i stavove korisnika (čitatelja) medijskog portala *dalmacijadanas.hr*. Na taj način će biti analizirano uzrokuju li pročitani tekstovi koji se odnose na tematiku pandemije pozitivne ili negativne emocije i dojmove kod čitatelja.

Prvi dio rada predstavlja uvid u područje dubinske analize mišljenja (sentimenta), a u njemu su objašnjenje najčešće vrste i pristupi koji su korišteni u daljnjoj analizi. Osim toga, poglavlje sadrži i pregled algoritama strojnog učenja koji će dalje biti korišteni. U drugom dijelu je predstavljen nestrukturirani izvor podataka te proces na koji će ti podaci biti prikupljeni. Prikupljanje i strukturiranje podataka s portala je realizirano programskim putem korištenjem programskog jezika *Python 3* metodom web struganja. Web struganje (*Eng.* Web Scraping) je metoda za automatizirano prikupljanje podataka s web stranica te spremanje u strukturiran i uređen skup podataka [4]. Treći dio se odnosi na obradu i standardizaciju prikupljenog skupa podataka što uključuje postupak nalaženja članaka u željenom području interesa – pandemiji koronavirusa. Nadalje, u ovom dijelu su opisani i postupci računalne analize jezika kao što su svođenje riječi na osnovni, tj. kanonski oblik radi suočavanja s različitim morfološkim oblicima riječi (npr. različite izvedenice).

Sljedeći dio predstavlja korak analize prikupljenih i obrađenih podataka. U ovom dijelu je prvo predstavljena deskriptivna statistika prikupljenog skupa podataka. Glavni predmet ove analize je kvantitativni pregled članaka, distribucija po kategorijama te statistika reakcija korisnika izražena putem *emojija*. Nadalje, opisana je metrika za evaluaciju analize mišljenja kojom će algoritmi za analizu mišljenja biti objektivno uspoređeni prema raznim dimenzijama točnosti. Nakon toga slijedi demonstracija rada i evaluacija konvencionalnih metoda za računanje sentimenta korištenjem algoritama vlastite izrade i algoritama koji su bazirani na umjetnoj inteligenciji. U zadnjem dijelu rada je prikazana usporedba efikasnosti svih metoda koje su korištene za analizu sentimenta nad podacima prikupljenim s promatranog portala.

2. Analiza sentimenta

Analizu sentimenta (*Eng.* Sentiment Analysis, Opinion Mining) definiramo kao skupinu postupaka koji se bave analizom ljudskih mišljenja, stavova, evaluacija i preporuka. Subjekti na koje se sentiment odnosi su proizvodi, usluge, organizacije, pojedinci, događaji te ostale pojave u ljudskoj svakodnevici [5]. Cilj analize je dobiti uvid u emocionalnu perspektivu jednog ili više pojedinaca, a velikom količinom takvih uvida, moguće je formirati objektivna opažanja o nekoj pojavi. U analizi sentimenta, promatrane rečenice koje su predmet analize svrstavamo u dvije vrste: činjenice (objektivne) i mišljenja (subjektivne). Činjenične pritom možemo definirati kao neutralni pogled opisan dokazima uz najblaži oblik emocija. Mišljenje definiramo kao izražavanje osobnih vjerovanja, emocija i stavova, te kao takvo može biti predmet preispitivanja [6].

Primjer objektivne rečenice glasi „Ova mačka je crna.“, dok subjektivna inačica na navedenu temu može glasiti „Mačke su predivne životinje!“. Iz primjera je očito da subjektivni primjer rečenice donosi traženu informacijsku vrijednost za određivanje mišljenja. Međutim, postoje slučajevi u kojima i objektivni tip rečenice iskazuje emotivnu vrijednost. Na primjer, rečenica „Ne volim mačke, alergičan sam.“ ujedno nosi i objektivnu i subjektivnu konotaciju. Zadatak analize sentimenta je iskoristiti takve značajke i pridodati im osjećajni kontekst te u odnosu na njih klasificirati rečenice u određene grupe, ovisno o opažanjima koja smo stekli kroz analizu. U kontekstu rudarenja podataka, klasifikaciju definiramo kao tehniku za grupiranje podataka pronalazeći njihove zajedničke značajke [7]. Ta svojstva se mogu odnositi na samu pozitivnost teksta u kategorijama koje predstavljaju pozitivnost, negativnost, neutralnost, objektivnost *i sl*.

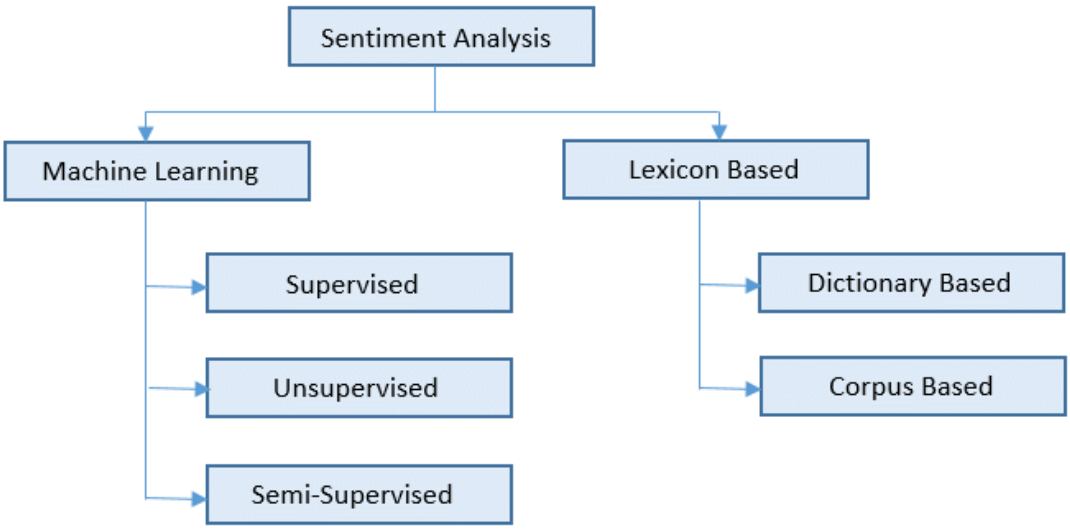
Trenutno korištene metode u tehnikama analize sentimenta su zbog šire dostupnosti računalnih sustava i informacijske tehnologije početkom 21. stoljeća doživjele veliku prekretnicu. Dovoljna informacija koja utvrđuje taj podatak je činjenica da je 99 % znanstvenih članaka u spomenutom području nastalo nakon 2004. godine [8]. Unatoč tome, tehnike za utvrđivanje mišljenja su postojale i prije spomenutog razdoblja, a većinom se je mišljenje „mjerilo“ manualnim putem od strane istražitelja. Neke od tih metoda uključuju ručno traženje riječi za koje je poznato da prenose pozitivni i negativni kontekst te daljnje operacije nad takvim riječima. Nasuprot tome,

ovom radu je primarni cilj prikazati automatizirane pristupe analizi sentimenta primjenom računalne obrade prirodnog jezika i strojnog učenja, a temelji takve analize slijede u nastavku.

Računalna obrada prirodnog jezika (*Eng.* Natural Language Processing - NLP) je disciplina koja se bavi pretvaranjem lingvističkih podataka (*npr.* tekstova ili dokumenata) u strojno čitljivu reprezentaciju korištenjem računalnih metoda [9]. Strojno čitljiva reprezentacija omogućuje izvođenje matematičkih i statističkih operacija nad formiranim podacima te kao takva čini revolucionarnu promjenu u brzini i kvaliteti analize tekstualnih skupova podataka.

Nestrukturirani podaci koji su predmet analize mišljenja dolaze u raznim formama. Analiza se može odnositi na jednu rečenicu ili nekoliko njih. Nerijetko se kroz algoritme za analizu obrađuju i veći tekstovi, kao što su novinski članci. Takve razlike u veličini promatranih ulaznih podataka dovode do uvođenja analize sentimenta na više razina. Prva razina se odnosi na klasifikaciju teksta na razini dokumenta. U tom pristupu se cijeli promatrani tekst promatra kao cjelina, neovisno o broju rečenica pri čemu je cilj dohvatiti ukupan dojam o tekstu koji može biti pozitivan i negativan. Drugi pristup se naziva klasifikacija na razini rečenice. U njoj je cilj tekst razdvojiti na rečenice te svaku od njih klasificirati prvo u objektivne i subjektivne, a zatim pozitivne i negativne te izračunati prosječnu vrijednost. Treći pristup se odnosi na promatranje značajke u kojem se predmet na koji se sentiment odnosi raščlanjuje na svojstva prema kojima se određuje pozitivnost i negativnost [10]. Spomenuti pristupi su temelj za sve algoritme te svaki od njih nalazi svoje područje primjene, ovisno o temi i količini teksta koji se obrađuje.

Algoritmi koji se koriste kod analize sentimenta obrađeni u ovom radu imaju cilj klasificirati orijentaciju sentimenta. Klasifikacija orijentacije sentimenta je postupak grupiranja promatranog teksta ili njegovog segmenta u kojemu je cilj odrediti je li promatrani tekst pozitivan, neutralan ili negativan u odnosu na svojstva nad kojima se algoritam provodi [6]. Te algoritme svrstavamo u dvije skupine, a najčešće inačice u odnosu na poznate podjele su prikazani na Slika 1. Prva skupina se odnosi na nenadzirani pristup koji se koristi tradicionalnijim korištenjem skupova riječi i pravila koje se odnose na skup riječi. Drugi pristup se koristi u nadziranim pristupima, korištenjem algoritama strojnog učenja koji su empirijskim metodama pokazali dobre rezultate za klasifikaciju teksta u vidu analize sentimenta [11]. Kao što je vidljivo na slici, algoritme strojnog učenja je moguće kombinirati s nenadziranim učenjem za dostizanje boljih rezultata. U sljedećem potpoglavljima je opisan generalni način rada tih algoritama.



Slika 1: Podjela algoritama nenadziranog i nadziranog učenja (Preuzeto: https://www.researchgate.net/figure/Sentiment-Analysis-techniques_fig1_343712784)

2.1. Algoritmi nenadziranog učenja

Iako pojam nenadziranog učenja pronalazimo i u terminologiji algoritama strojnog učenja, u kontekstu analize sentimenta se najčešće radi o algoritmima baziranima na korištenju leksikona. Ti leksikoni sadrže riječi za koje je utvrđeno da nose negativnu ili pozitivnu konotaciju uz naznačen koeficijent te notacije, a oni se dijele na one opće primjene i specifične primjene, ovisno o temi nad kojom se koriste. Cilj korištenja tih leksikona je pronalaženje funkcije koja ukazuje na indikatore pozitivnosti i negativnosti te računanje ukupne pozitivnosti ovisno o tim indikatorima [12]. Algoritmi nadziranog učenja se dijele na one bazirane na rječniku, te one bazirane na korpusu, a oni su opisani u nastavku.

2.1.1. Pristup baziran na rječniku

Pristup baziran na korištenju rječnika je osnovan na konceptu identificiranja riječi kojima se iskazuje sentiment. Dalje se individualnom procjenom osoba koje provode istraživanje tim riječima daje ocjena pozitivnosti. Sljedeći korak je pronaći potencijalne sinonime i antonime tih riječi, odnosno riječi koje nose isto značenje, a također se mogu pojavljivati u istom kontekstu [12]. Korištenjem nekih riječi u tekstu se nedvojbeno pokušava iskazati *npr.* pozitivni kontekst, zbog čega riječi „Odlično“ sigurno možemo pridodati pozitivan doprinos, kao što riječ „Mrzim“ daje negativni doprinos. Na taj način se pokušava doći do riječi koje nose slično značenje, na primjeru riječi „Odlično“, to mogu biti riječi „Izvrsno“ ili „Izvanredno“. Tim se riječima pridodaje isti koeficijent pozitivnosti.

Kreiranje ovakvog rječnika je relativno jednostavan postupak kojim se može doći do primitivne analize sentimenta čak i manualnim putem bez korištenja računalne tehnologije. Međutim, korištenjem *online* rječnika poput *WordNet-a* koji pruža pregled relacija između riječi proces može biti znatno ubrzan. Uzimanjem korijenske riječi moguće je putem spomenutih relacija doći do većeg broja sinonima i antonima. Zatim se izvedeni antonimi i sinonimi također koriste kao referenca za traženje u ostalim rječnicima čime se kreira puno veći leksikon [13]. Kreirani leksikon se kasnije može primijeniti u računalnoj analizi korištenjem određenog seta pravila. Jedan od tih seta pravila je kreiranje funkcije koja računa sumu pozitivnosti ukoliko je riječ iz leksikona pronađena u promatranom tekstu.

2.1.2. Pristup baziran na korpusu

Pristup baziran na korpusu možemo smatrati modifikacijom pristupa baziranog na rječniku korištenjem statističkih mjera i sintaktičkih uzoraka. Osnovna ideja ovakvog pristupa je analiza pojavljivanja riječi uz onu riječ za koju je poznato da pridonosi određivanju polariteta sentimenta rečenice [14].

Konačni polaritet se formira raznim metodama kao što su spoznaja da dva pridjeva korištena u istoj rečenici obično imaju jednak polaritet rečenice. Nadalje, korištenje riječi „ali“ ili „ne“ pridonose negativnom polaritetu [5]. U ovoj metodi se korištenjem takvih struktura određenim riječima mijenja osnovni koeficijent polarnosti sentimenta koji je isprve identificiran pristupom baziranim na rječniku. Dobiveni korpus riječi se zatim može koristiti za računalnu analizu u kojoj je cilj odrediti orijentaciju sentimenta.

Na primjeru pozitivne rečenice „Mačke su smiješne i umiljate!“ možemo uočiti da riječi smiješne i umiljate u kontekstu govora o mačkama donose pozitivnu poruku, zbog čega bi u ovom pristupu takvim riječima dali veću pozitivnu vrijednost sentimenta. Međutim, na primjeru rečenice „Ne budi smiješan...“ vidimo da riječ smiješan nosi duboko negativnu poruku u kombinaciji s riječi „budi“ što je pojačano riječi negacijom. Iz tog primjera je vidljivo da riječ ovisno o kontekstu rečenice nosi potpuno drugi smisao. Zbog takvih slučajeva pristup baziran na korpusu nije univerzalan za svaki analizirani tekst, već ga je poželjno koristiti samo na temama za koje je predviđen.

2.2. Algoritmi nadziranog učenja

Nadzirano učenje je pristup strojnog učenja u kojem model učenjem pokušava kreirati funkciju za predikciju unaprijed poznatih klasa i rezultata nad danim ulaznim vrijednostima [1]. Drugim riječima, algoritmi koji koriste nadzirani pristup pokušavaju pronaći ponovljive uzorke u ulaznim podacima koje mogu jedinstveno identificirati pripadnost nekoj grupi, odnosno klasi. Njihov rad je omogućen statističkim, matematičkim, grafičkim i programskim metodama, a oni koji se koriste u projektu su opisani u nastavku.

2.2.1. Naivni Bayes

Naivni Bayesov algoritam (*Eng.* Naive Bayes - NB) je algoritam nadziranog strojnog učenja baziran na Bayesovom teoremu o vjerojatnosti događaja. Zadatak algoritma je identificirati klasu promatrane značajke u odnosu na njezine vrijednosti [15].

Algoritam računa vjerojatnost $P(B|A)$, tako da vrijednosti ulazne A značajke pripadaju nekoj od predefiniranih klasa B. Pritom $p(A|B)$ predstavlja kombinaciju značajki A (*Npr.* Riječi tekstualnog elementa) i vjerojatnosti B da se te vrijednosti odnose na promatranu kombinaciju značajki. Broj u nazivniku $p(A)$ služi za normalizaciju podataka i predstavlja skup značajki. Algoritam pretpostavlja da su sve ulazne značajke nepovezane te da promjena jedne od vrijednosti neće utjecati na konačni rezultat [16].

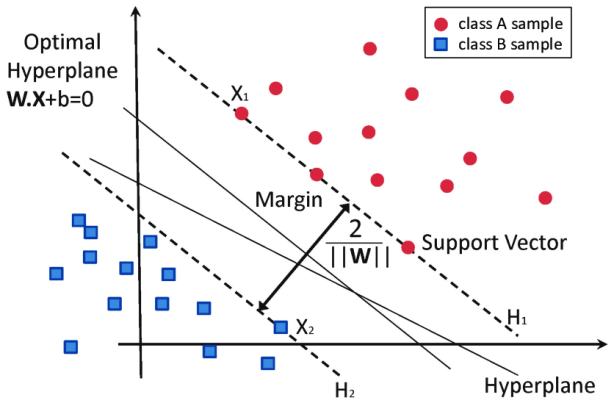
$$P(B|A) = \frac{p(A|B)p(B)}{p(A)}$$

Radi se o jednostavnom algoritmu koji je poznat po velikim brzinama izvođenja kalkulacija, zbog čega je njegova primjena moguća za rješavanje raznih problema klasifikacije. Neki od tih problema uključuju trenutne potrebe za klasifikacijom, kao što je identifikacija neželjene pošte kod primjera klasifikacije teksta te kao takav može biti korišten i u ovom projektu.

2.2.2. Metoda potpornih vektora

Metoda potpornih vektora (*Eng.* Support Vector Machine - SVM) je metoda nadziranog učenja koja je primarno korištena za rješavanje problema binarne klasifikacije. Kao i kod ostalih algoritama nadziranog učenja, osnova SVM algoritma su klase koje predstavljaju kategorizaciju promatranih ulaznih podataka.

Algoritam podatke svrstava u geometrijski sustav tako da se svakom podatku dodijeli određena pozicija. Zatim se u fazi treniranja modela za promatrani skup podataka traži razgraničavajući pravac (*Eng.* Hyperplane) koji najbolje opisuje granicu između promatranih klasa. Pozicija pravca je određena računanjem najmanje udaljenosti od pravca do prvih pojavljivanja klasa koje se njime razgraničavaju. Pritom se najbliži primjerci klasa nazivaju potpornim vektorima, a udaljenosti do njih se nazivaju marginama [17].



Slika 2: Vizualni prikaz metode potpornih vektora (preuzeto: https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323)

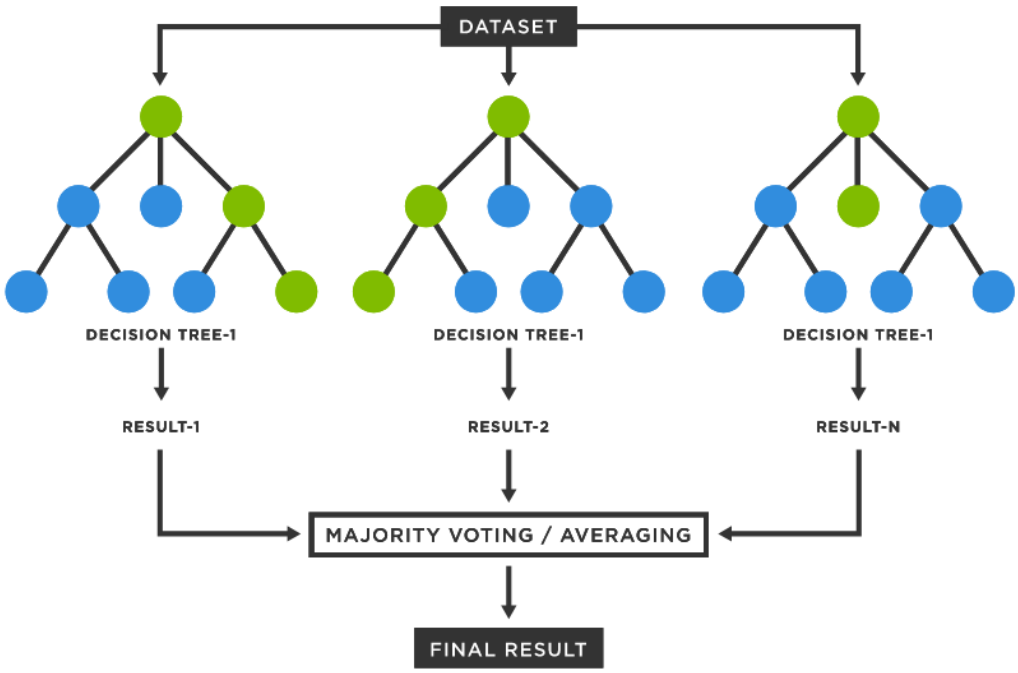
U fazi testiranja dobivenog modela, promatra se distribucija uzoraka klase iz skupa za testiranje te njezina udaljenost od prije definiranih margina. Ovisno o poziciji promatranog uzorka odnosu na razgraničavajući pravac, uzorak se svrstava u jednu od klasa [18].

2.2.3. Algoritam slučajnih šuma

Algoritam slučajnih šuma (*Eng.* Random Forest - RF) je algoritam nadziranog učenja čija je primjena moguća u slučajevima klasifikacije i regresije. Zbog širokog spektra namjene, lakoće implementacije te generalno dobrih rezultata, RF algoritam je jedan od najčešćih metoda za dobivanje uvida u podatke.

Riječ „šuma“ je prisutna zbog činjenice da algoritam kao pogon generira stabla odluke najčešće bazirana na *Bagging* algoritmu pri čemu svako stablo služi kao individualan klasifikator. Svako od tih stabala uzima nasumično odabrani uzorak (značajku) u obliku ulaznih podataka skupa za treniranje. Za svaku od tih značajki se koristi proces grananja koji se pita donosi li promatrana značajka neki informacijsku dobit te postoji li njen značaj za određivanje konačne klase. Ukoliko grana prosudi da informacijske dobiti nema, grananje u tom smjeru se obustavlja, dok se grana s istinitom vrijednosti nastavlja granati. Postupak se ponavlja za sve uzorke nasumično [19].

Nakon izgradnje stabala svih stabala, kao rezultat se uzima vrijednost stabla s najvećim brojem grananja. Analizom svake značajke respektivno, algoritam pruža i mogućnost procjene važnosti svake značajke čime je moguće dobiti dublji uvid u promatrani slučaj.



Slika 3: Grafički prikaz rada RF algoritma.
(Preuzeto: https://www.tibco.com/reference-center/what-is-a-random-forest)

3. Izvor podataka

Izvor podataka nad kojim je proveden projekt prikupljanja, obrade i analize podataka je lokalni hrvatski portal Dalmacija Danas (*dalmacijadanas.hr*). Radi se o regionalnom portalu koji je primarno fokusiran na prijenos i pisanje vijesti s područja Dalmacije, s naglaskom na Split, obalne gradove i Dalmatinsku Zagoru. Misija portala je prijenos vijesti na razini Hrvatske, kao i prijenos priča o „običnim“ ljudima što uključuje gospodarstvenike, studente, udruge, *itd*. [20]

Prema istraživanjima Gamius Rating agencije [21], portal je rangiran na šesnaestom mjestu (u odnosu na ostale prenositelje vijesti) po prosječnom broju dnevnih korisnika, a taj broj iznosi 71.883 korisnika dnevno. Također, još jedan bitan podatak je da doseg portala na internetu iznosi 18,26%. Zbog velikog broja članaka i širokog obujma područja koje Dalmacija Danas pokriva, portal služi kao odlična baza za prikupljanje informacija u raznim domenama.

Portal pokriva šest kategorija koje se granaju na preko četrdeset potkategorija, a možemo ih podijeliti na one „ozbiljne“ i „svakodnevne“ prirode. Naprimjer, kategorije Dalmacija, Vijesti i Kolumne pokrivaju lokalne događaje i važne vijesti. S druge strane, kategorije Sport, Relax i Specijali pokrivaju vijesti zabavnog karaktera. Mješavina „ozbiljnih“ i „svakodnevnih“ tema pritom može biti korištena u raznim domenama obrade prirodnog jezika, strojnog učenja te antropološkim istraživanjima i zaključcima. Dizajn i elementi portala su prikazani na slikama u nastavku.

Slika 4: Naslovnica portala Dalmacija Danas (Preuzeto: https://www.dalmacijadanas.hr/)

Slika 5: Primjer članka na portalu Dalmacija Danas (Preuzeto: https://www.dalmacijadanas.hr/video-skakljanja-raze-postao-svjetski-hit-strucnjak-od-uboda-raze-mozete-umrijeti/)

4. Prikupljanje podataka

U poglavlju je opisan proces izrade skupa podataka portala Dalmacija Danas. Cilj ovog dijela projekta je prikupiti sve dostupne meta-podatke svakog pojedinog članka u svrhu provođenja daljnje analize.

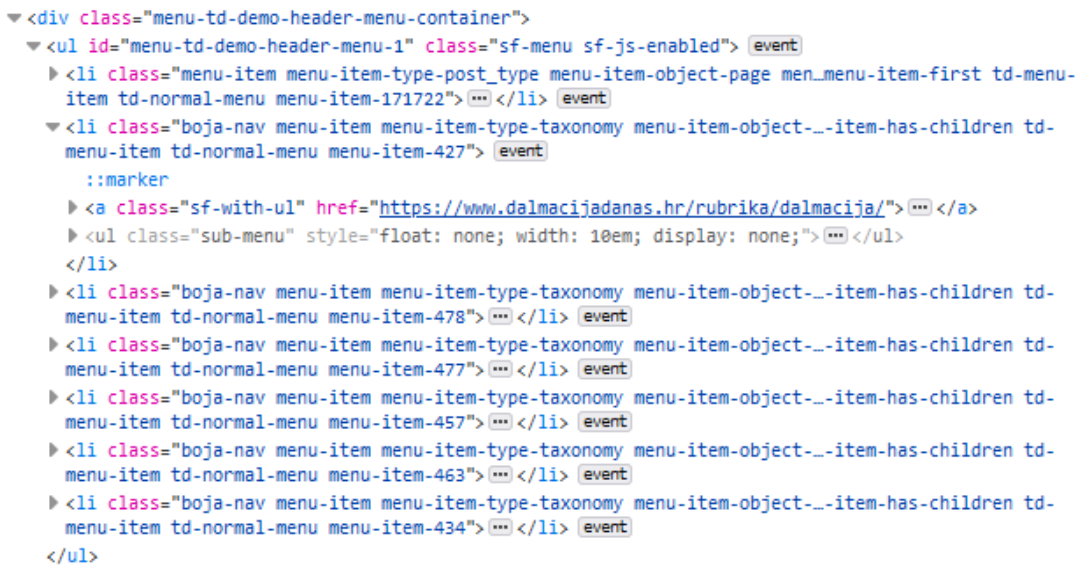
Prikupljanje poveznica i podataka sadržanih u člancima je provedeno respektivno radi mogućnosti ponovnog prikupljanja podataka i što manjeg broja pristupa portalu. Navedeni dio projekta je realiziran postupkom web struganja (*Eng. Web scraping*) omogućenu odgovarajućom programskom podrškom.

Programska podrška korištena za prikupljanje podataka uključuje programski jezik *Python 3* uz odgovarajuće biblioteke. Za identifikaciju elemenata koji sadrže tražene podatke korištena je biblioteka *BeautifulSoup4* za obradu HTML jezika. Radi se o skupu klasa koji služi za navigaciju, pretraživanje i modifikaciju HTML stabla [22].

4.1. Prikupljanje poveznica

Portal je strukturiran kroz šest kategorija koje sadrže preko četrdeset potkategorija zbog čega se prikupljanje u početku može činiti izazovnim. Članci se mogu identificirati po kategorijama i potkategorijama neovisno zbog čega je proces olakšan, a svaki članak je dohvatljiv pomoću glavnih kategorija.

Glavne kategorije su redom: Dalmacija, Vijesti, Sport, Relax, Specijali i Kolumne. Za operacije nad kategorijama i člancima koje one sadrže, zadužena je skripta *article_url_scraper.py*. Skripta je koncipirana na način da identificira skup HTML elemenata pomoću *BeautifulSoup4* modula koji sadrži poveznicu na svaku pojedinu kategoriju (Slika *6*) kojom se zatim programski pristupa.



Slika 6: HTML elementi s poveznicama na kategoriju (Preuzeto: https://www.dalmacijadanas.hr/)

Otvaranjem stranice kategorije dolazimo do liste koja sadrži sliku, tekst i kratki opis članka. Navedenu listu članaka korisnik širi korištenjem klizača miša u smjeru dna stranice, što u jeziku Python nije moguće provoditi bez određenih međuovisnosti. Takve operacije traže radnje koje možemo očekivati samo od korisnika servisa, a njihovo je izvođenje moguće imitirati drugim programskim jezicima kao što je *JavaScript*. Iz tog razloga je korištena dodatna podrška u obliku modula *Selenium Web Driver*. *Selenium* je *framework* otvorenog koda za testiranje web aplikacija koji omogućuje automatizirano upravljanje web preglednikom, kao i korištenje „ljudskih“ radnji korištenjem skripti kao što je korištenje klizača miša [23].

Framework je pritom korišten u *tzv.* „*headless*“ načinu rada u kojemu se operacije u pretraživaču izvode bez grafičkog sučelja. Funkcija klizača miša *scroll_category()* je implementirana zasebno za svaku kategoriju uz proizvoljan broj klizanja, ovisno o opsežnosti promatrane kategorije. Rezultat spomenutog klizanja je prošireni HTML dokument (Slika *7* i Slika *8*) koji okvirno sadrži sve poveznice na članke u razdoblju od 1.1.2020 do 1.5.2021 za svaku kategoriju.

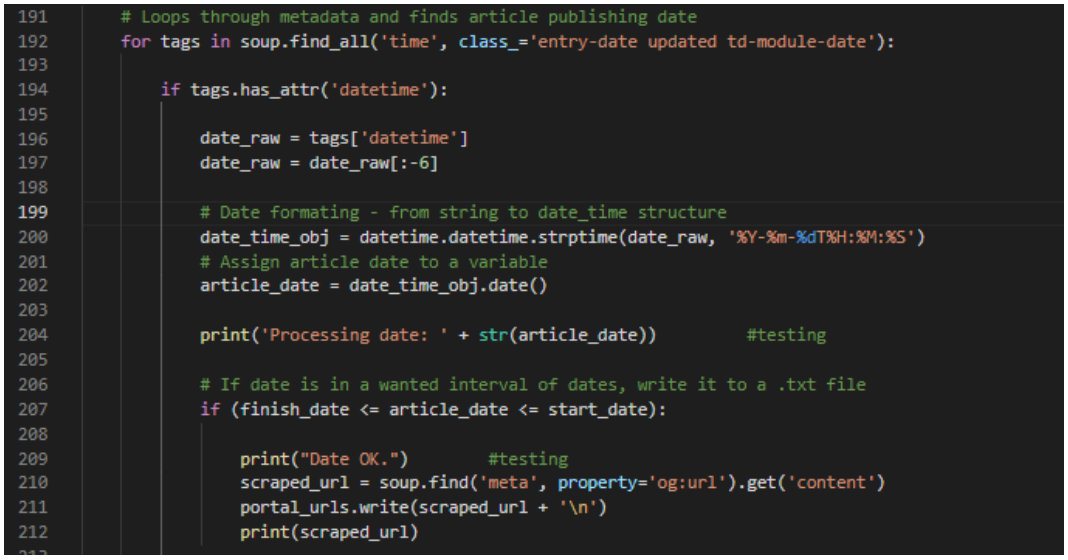
```
<!--module-->
<div class="td_module_10 td_module_wrap td-animation-stack td-meta-info-hide">…</div>
<div class="td_module_10 td_module_wrap td-animation-stack td-meta-info-hide">…</div>
<div class="td_module_10 td_module_wrap td-animation-stack td-meta-info-hide">…</div>
<div class="td_module_10 td_module_wrap td-animation-stack td-meta-info-hide">
  <div class="td-module-thumb">…</div>
  <div class="item-details">
    <h3 class="entry-title td-module-title">
      <a href="https://www.dalmacijadanas.hr/mnogi-su-ostali-u-cudu-testira…covid-ali-vas-je-neugodno-iznenadio-racun-evo-pravih-cijena/" rel="bookmark" title="MNOGI SU OSTALI U ČUDU Testirali ste se na covid, ali vas je neugodno iznenadio račun. Evo pravih cijena">…</a>
    </h3>
    <div class="td-module-meta-info">…</div>
    <div class="td-excerpt">
      Svi koji putuju sa splitskog aerodroma moraju napravit brzi antigenski test, no zanimljivo je da taj test - ne košta za sve jednako. Tako će...
    </div>
  </div>
</div>
```

Slika 7: Primjer kartica s vijestima u kategoriji – HTML (Preuzeto: https://www.dalmacijadanas.hr/rubrika/dalmacija/)

Slika 8: Izgled kartica s vijestima u kategoriji Dalmacija (Preuzeto: https://www.dalmacijadanas.hr/rubrika/dalmacija/)

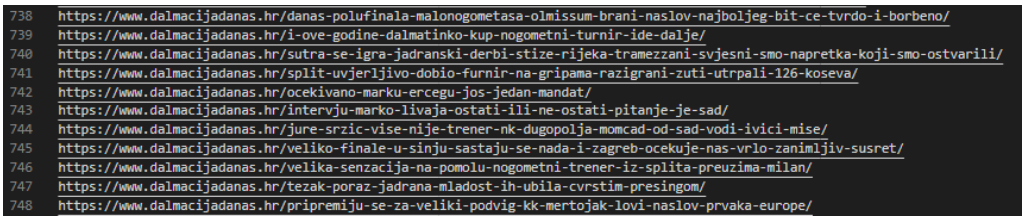
Kod nekih portala je uobičajeno da svaka kartica koja predstavlja svaki pojedini članak ima prikaz datuma objave, što na promatranom portalu nije slučaj. Da bi skripta mogla pronaći početni i krajnji uvjet unosa u bazu podataka (u našem slučaju datum), potrebno je pronaći datum članka. Jedini način za pronalazak navedenog podatka je otvaranje svakog članka posebno te iz dostupnih meta-podataka identificirati element koji sadrži datum objave i učiniti ga strojno čitljivim.

Kao što je vidljivo na slici, u HTML stablu se prvo pronalazi objekt koji sadrži tekstualni (*string*) zapis vremena oblika **2020-06-15T15:36:05+02:00** koji se pretvara u strojno čitljiv oblik bibliotekom *datetime()* na liniji 200 (Slika *9*). U kodu su pritom zadani rubni datumi spomenuti u prethodnom paragrafu, te ukoliko je datum u zadanom intervalu, poveznica za promatrani članak će biti spremljena u datoteku *portal_urls.txt*.



Slika 9: Skripta za dohvaćanje i spremanje poveznice članka

Skripta se ponavlja za svaku glavnu kategoriju, sve dok sve ispravne poveznice u željenom intervalu nisu prikupljene. Rezultat ovog dijela projekta je lista od 34332 poveznice koje su prikupljene u više navrata. Primjer izgleda i formata prikupljenih poveznica je prikazan u nastavku (Slika *10*).



Slika 10: Uzorak prikupljenih poveznica na članke portala

4.2. Prikupljanje sadržaja članaka

Nakon što su prikupljene sve dostupne poveznice, slijedi faza struganja dostupnih meta-podataka članaka portala. Za struganje članaka implementirana je skripta *article_scraper.py* čiji je zadatak otvoriti svaki članak iz prikupljene liste poveznica te pronaći HTML elemente koji nose željene informacijske vrijednosti.

Neki od značajnijih podataka koji su prikupljeni su naslov, podnaslov, tekst članka te vrijeme objave. Također, kod dohvaćanja tekstualnih elemenata potrebno je uvesti dodatne uvjete kojima ih čistimo od određenih pojava koje se u njima nalaze. Neke od tih pojava uključuju reklame, slike i znakove koji u daljnjoj analizi mogu dovesti do krnjih redaka s podacima. Problem koji se u ovoj radnji dogodio uključuje pojavu naslova paragrafa u posebnim podebljanim poljima koje analizator (*Eng*. Parser) nije uspio prepoznati, što je otklonjeno korištenjem regularnih izraza.

Zadnja vrsta struganih elemenata je ujedno i temelj izvedbe cijelog projekta, a odnosi se na sedam animiranih *emojija* pomoću kojih čitatelj može iskazati svoje mišljenje o predstavljenoj temi. Radi se o dinamičnim programiranim elementima koji sadrže vrijednost *data-count* koja predstavlja broj reakcija korisnika, a naš cilj je dohvatiti te vrijednosti. Zbog određenih varijacija u spomenutim elementima također su uvedeni regularni izrazi radi mogućnosti postojanja vrijednosti mogu slučajno promaknuti.

Podaci se zatim zapisuju u *portal_articles.csv* datoteku dimenzija 17 stupaca (Tablica *1*) i 34.332 retka, odnosno članka. Zbog dugotrajnog izvođenja skripte koje traje više od 15 sati i mogućnosti da vlasnici portala brišu pojedine članke, implementiran je i sustav iznimki za nošenje s navedenim problemima i potencijalnim greškama u obradi. Nadalje, treba napomenuti da je prikupljanje podataka odrađeno u dvije faze između kojih je Dalmacija Danas mijenjala predložak svoje web stranice. Pritom su izbačeni određeni meta-podaci zbog čega u drugoj fazi neki od njih nisu dostupni, kao *npr.* Modified_time.

Podatak	Tip podatka	Opis podatka
ID	*Integer*	Jedinstveni identifikator članka
Title	*String*	Naslov članka
Subtitle	*String*	Podnaslov članka
URL	*String*	Poveznica na članak
Section	*String*	Kategorija članka
Article_text	*String*	Cijeli tekst članka
Published_time	*String*	Datum objave članka
Modified_time	*String*	Datum zadnje izmjene članka
Author	*String*	Autor članka/fotografije
Comments	*Integer*	Broj komentara
Reaction_love	*Integer*	Broj korisnika s reakcijom 1
Reaction_laugh	*Integer*	Broj korisnika s reakcijom 2
Reaction_hug	*Integer*	Broj korisnika s reakcijom 3
Reaction_ponder	*Integer*	Broj korisnika s reakcijom 4
Reaction_sad	*Integer*	Broj korisnika s reakcijom 5
Reaction_mad	*Integer*	Broj korisnika s reakcijom 6
Reaction_mind_blown	*Integer*	Broj korisnika s reakcijom 7

Tablica 1: Stupci datoteke portal_articles.csv

5. Obrada podataka

Kod rudarenja podataka, jedan od najdugotrajnijih procesa predstavlja upravo obrada izvora podataka nad kojim se provodi analiza. Sirovi skup podataka može sadržavati nerelevantne, redundantne i nepotpune podatke koje je potrebno programski izmijeniti i prilagoditi individualnim kriterijima. U ovom projektu izazov predstavlja izolacija teksta s promatranom temom – uzročnikom bolesti COVID-19 te relativno kompleksna gramatika hrvatskog jezika čime se ovo poglavlje i bavi.

5.1. Identifikacija COVID-19 članaka

U prošlom poglavlju prikupljena su 34.332 članka portala Dalmacija Danas što ujedno čini godinu i pet mjeseci prikupljenih članaka. Zbog toga, iz ovog skupa podataka moguće je sintetizirati informacije koje čine znanja iz raznih područja u svrhu provođenja različitih vrsta analiza. U ovom slučaju cilj je izolirati članke čija je tematika ekskluzivno vezana uz novi koronavirus koji zadovoljavaju određene uvjete.

Intuitivni pristup za određivanje članaka vezanih uz aktualni virus je prepoznavanje terminologije koja se koristi u slučajevima kada se o njemu piše. Pretpostavka je da takvi članci koriste senzacionalne naslove te da kroz tekst članka ponavljaju korištenje određenih izraza. U tu svrhu je razvijena lista od 160 COVID-19 termina koji su proizvoljno određeni te zatim spremljeni u datoteku *covid_dictionary.txt*. Neke od tih riječi su *pcr*, *samoizolacija*, *covid-19*, *zaražen*, *pandemija*, *cijepljenje*, *stožer*, *koronavirus* te brojne izvedenice navedenih riječi.

Za identifikaciju COVID-19 članaka implementirana je skripta *article_filter.py*. Skripta kao ulaz koristi skup podataka *portal_articles.csv* pritom izgrađujući novi skup podataka *portal_articles_covid.*csv. Skripta u ulaznoj datoteci prolazi kroz stupce naslova, podnaslova i teksta članka pritom pregledavajući je li u tekstualnim elementima pronađena riječ iz datoteke *covid_dictionary.txt*. Osim toga, radi daljnje analize sentimenta koji je određen reakcijama čitatelja članka, u razmatranje se uzimaju samo članci na koje je pomoću jednog od sedam *emojija* reagiralo minimalno 3 čitatelja. Implementacija koraka filtriranja članaka je prikazana na slici (Slika *11*).

```
65          # Emoji value total sum used for article filtering
66          emoji_sum =  reaction_love + reaction_laugh + reaction_blushy + \
67          reaction_ponder + reaction_sad + reaction_mad + reaction_mind_blown
68
69
70          # Identifies covid articles, based on a list of words
71          # in covid_dictionary.txt
72          if (any(map(title.__contains__, covid_dict))
73              or any(map(subtitle.__contains__, covid_dict))
74                  or any(map(article_text.__contains__, covid_dict))):
75
76                  if(emoji_sum >= 3):
77
78                      csv_writer.writerow([row[0], row[1], row[2], row[3], row[4], row[5], row[6],
79                      row[7], row[8], row[9], row[10], row[11], row[12], row[13], row[14], row[15],
80                      row[16]])
81
82                      covid_counter += 1
```

Slika 11: Skripta article_filter.py za izoliranje COVID-19 članaka

Navedenim koracima je prikupljeno 12.717 članaka koji se odnose na pandemiju koronavirusa, ali nažalost manji broj tih članaka je iskoristiv u našoj analizi. Drugim riječima, od spomenutih 12.717 članaka, njih 3.392 ima više od tri reakcije čitatelja ostvarenih pomoću *emojija* (Tablica *2*). Daljnji koraci projekta se odnosi samo na skup članaka koji zadovoljava prije spomenute kriterije.

Ukupan broj članaka	34.332
Ukupan broj COVID članaka	12.717
Ukupan broj iskoristivih članaka	3.392

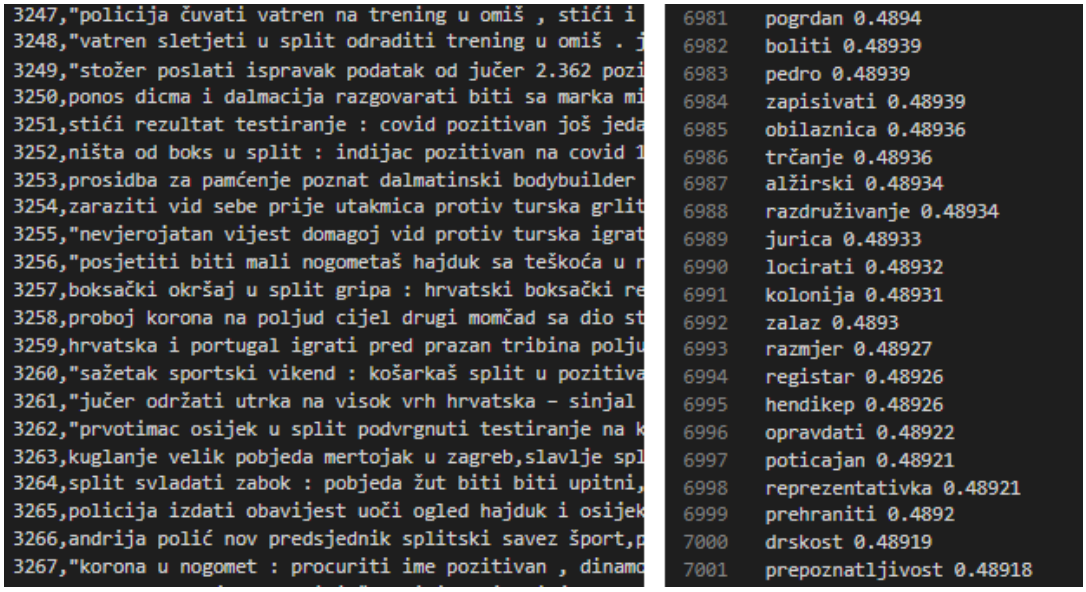
Tablica 2: Kvantifikacija prikupljenih članaka

5.2. Svođenje riječi na osnovni oblik

Algoritmi koji su korišteni kroz analizu podataka u pravilu nisu dizajnirani za prepoznavanje gramatičkih pravila, što se posebno odnosi na promjenjive vrste riječi, kao i izvođenje pojedinih riječi. Ti algoritmi funkcioniraju na način da uspoređuju riječi teksta sa skupom pre-definiranih pravila, zbog čega svaka izvedenica poznate riječi predstavlja šum u podacima. Naime, ukoliko jedna riječ ima više izvedenica, svaka od tih izvedenica će biti tretirana kao zasebna riječ iako je značenje riječi identično. Iz tog razloga, riječi je potrebno standardizirati što je u ovom slučaju realizirano svođenjem riječi na njezin kanonski oblik.

Osnovni ili kanonski oblik riječi je forma u kojoj riječi prikazujemo u njihovom izvornom obliku. Na primjeru imenica, osnovni oblik je zapis riječi u nominativu jednine, kod glagola je to infinitiv te kod pridjeva nominativ jednine muškog roda [24]. Za obradu prirodnog engleskog jezika dostupno je mnoštvo alata od kojih možemo istaknuti biblioteku *Stanza* razvijenu na sveučilištu Stanford. Tim alatom je moguće izvoditi razne obrade u vidu analize riječi, razdvajanja rečenica u smislene cjeline, svođenje riječi na osnovni oblik, *itd*. Jedna od inačica spomenute biblioteke imena *Classla* je razvijena posebno za specifične slavenske jezike kao što su slovenski, hrvatski, srpski, bugarski i makedonski [25].

Za svođenje riječi na osnovni oblik korištene su skripte *lemmatization_article.py* i *lemmatization_words_list.py*. Pomoću prve spomenute skripte obrađeni su stupci koji nose tekstualne informacijske vrijednosti, a to su naslov i podnaslov te tekst članka. Drugom skriptom su obrađeni leksikoni s koeficijentima za određivanje polariteta sentimenta, a primjere rezultata obaju transformacija možemo vidjeti na slici u nastavku (Slika *12*). Spomenutim koracima je smanjen šum u podacima, a fokus se stavlja na broj instanci svake korištene riječi, a ne na način izražavanja koji može biti specifičan od autora do autora.

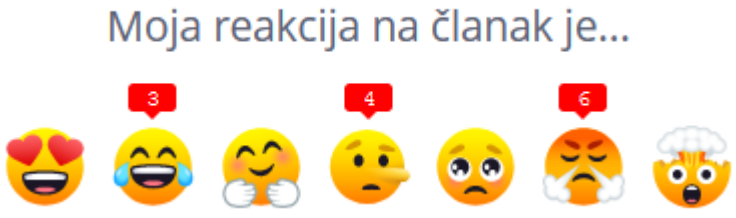


Slika 12: Rezultati svođenja riječi na osnovni oblik

6. Analiza podataka

6.1. Skala pozitivnosti prema reakcijama korisnika

Za evaluaciju rezultata analize potrebno je odrediti temelj u obliku ciljne varijable kojom ćemo pridodati ukupni koeficijent pozitivnosti svakog pojedinog članka u odnosu na faktor ljudske procjene. S obzirom da je sentiment oblik ljudskog mišljenja o određenoj temi, za vjerodostojnu procjenu pozitivnosti možemo koristiti reakcije korisnika određene pomoću 7 *emojija* koji su dostupni na svakom članku. Primjer takvih reakcija je prikazan na slici (Slika *13*).



Slika 13: Primjer reakcija putem *emojija* na članku portala (Preuzeto: https://www.dalmacijadanas.hr/direktor-prometa-priznao-nas-vozac-nije-smio-uzeti-taj-autobus-jer-je-bio-neispravan/)

Kao što je vidljivo na prethodnoj slici, pomoću spomenutih reakcija moguće je intuitivno iskazati emocije koje je članak ostavio na čitateljima. Da bi kvalitetnije odredili koji *emojiji* uzrokuju određenu emociju, provedena je anketa nad manjim brojem ispitanika (>10). U anketi se od ispitanika traži da odrede proizvoljni koeficijent pozitivne, neutralne i negativne emocije u intervalu (-1, 1) koji s bi osobno pridodali svakom pojedinom *emojiu*.

Rezultati ankete, odnosno finalni koeficijenti svakog pojedinog *emojija* su prikazani u tablici (Tablica *3*). Analizom je utvrđeno da je korištenje konkretnih *emojija* visoko polarizirano te da ne postoji *emoji* kojim je moguće iskazati neutralnost nad pročitanim člankom. Također, može se zaključiti da čitatelji koje je članak ostavio neutralnima nemaju potrebu iskazivati svoju emociju, zbog čega oni služe samo za iskazivanje emocija sarkazma, ljutnje, ogorčenosti, ali i sreće i radosti. Na taj način možemo uočiti da reakcije pod brojem 1, 3, 5 i 6 nose jednoznačni oblik pozitivne ili negativne emocije. S druge strane, reakcija 2 može služiti kao sreća, ali i sarkazam zbog čega je dodijeljen manji koeficijent, kao i kod reakcije pod brojem 7 koja može služiti kao sarkastičan, ali i neutralan pogled na temu.

Rbr.	***Emoji***	**Koeficijent pozitivnosti**	**Interpretacija**
1	Reaction love	1	Strogo pozitivan
2	Reaction laugh	0,25	Blago pozitivan
3	Reaction hug	1	Strogo pozitivan
4	Reaction ponder	-0,25	Blago negativan
5	Reaction sad	-1	Izričito negativan
6	Reaction mad	-1	Izričito negativan
7	Reaction mind blown	-0,50	Srednje negativan

Tablica 3: Koeficijenti pozitivnosti emocije *emojija*

Dobivene vrijednosti su korištene za računanje finalnog normaliziranog koeficijenta u intervalu (-1, 1). Broj reakcija je pritom pomnožen sa svakim od sedam *emojija*, a konačni rezultat je dijeljen s ukupnim brojem reakcija na članak, što je realizirano formulom u nastavku. Rezultati su spremljeni u stupac nestrukturiranog skupa podataka za svaki članak respektivno.

$$postivity = \big(reaction\ love \times 1 + reaction\ laugh \times 0.25 + reaction\ hug\ \times 1 + reaction\ ponder \times (-0.25) + reaction\ sad \times (-1) + reaction\ mad \times (-1) \times reaction\ mind\ blown\ \times (-0.50)\big) \div reaction\ count$$

Koeficijenti su dobiveni u numeričkom obliku te ih je u daljnjem koraku potrebno transformirati u standardizirani oblik. Određivanje pozitivnosti je problem klasifikacije u kojem je rezultat prikazan u obliku kategorija koje označavaju činjenicu je li članak pozitivan, negativan ili neutralan zbog čega vrijednost transformiramo u kategoričku varijablu. Kategorička varijabla je varijabla koja se sastoji od dvije ili više vrijednosti bez zadanog poretka između njih [26]. Korištenjem takvih varijabli dobivamo oblik metrike kojim možemo uspoređivati rad algoritama u nastavku.

Kategorizacija rezultata je dobivena uvođenjem intervala u kojem vrijednost možemo svrstati u negativnu ili ne-negativnu klasu. Ukoliko je vrijednost prije utvrđene pozitivnosti veća ili jednaka nuli, varijablu označavamo kao NONNEGATIVE klasu te ukoliko je manja od nule, svrstavamo ju u NEGATIVE klasu. Pritom je skup koji sadrži pozitivnu i neutralnu vrijednost ujedinjen zbog male količine članaka koje možemo smatrati neutralnima, zbog čega može biti narušena kvaliteta klasifikatora.

Tablica u nastavku (Tablica *4*) prikazuje konačan broj negativnih i ne-negativnih članaka. Kao što je na tablici vidljivo, na promatranom uzorku postotak negativnih članaka iznosi 66,95% te je njihov broj znatno veći od ne-negativnih.

Svojstvo	**Broj negativnih članaka**	**Broj ne-negativnih članaka**
Broj članaka	2.271	1.121
Interval koeficijenta pozitivnosti	< 0	≥ 0

Tablica 4: Zastupljenost klasa po pozitivnosti

6.2. Metrika korištena u analizi

Potpoglavlje u nastavku opisuje mjere koje su korištene radi unificirane usporedbe svih algoritama kroz analizu sentimenata. Radi vjerodostojnosti eksperimenta potrebno je dobiti jednak format i zapis svih rezultata, a u tu svrhu su korištene mjere ukupne točnosti, preciznosti, odaziva i F1 mjere. Navedene mjere ukazuju na količinu, odnosno postotak krivo i točno klasificiranih podataka, a njihovim korištenjem možemo doći do detaljnijeg uvida u točnost korištenih algoritama.

Za razumijevanje navedenih formula, potrebno je uvesti notaciju kojom označavamo istinitost, odnosno lažnost dobivenih vrijednosti, što je najlakše prikazati pomoću matrice konfuzije. U području strojnog učenja i statističkih opažanja, matrica konfuzije je dvodimenzionalna tablica koja prikazuje performanse korištenog algoritma kroz prikaz točno i krivo klasificiranih podataka [27]. Kao što je vidljivo iz tablice (Tablica *5*), u matrici je vidljiva kvantifikacija točno klasificiranih pozitivnih (**TP**) i negativnih vrijednosti (**TF**), kao lažno klasificiranih pozitivnih (**FP**) te lažno klasificiranih negativnih (**FN**) vrijednosti.

		Predviđena klasa	
Stvarna klasa		**Istinita vrijednost**	**Lažna vrijednost**
	Istinita vrijednost	**TP** (Istinita pozitivna vrijednost)	**FN** (Lažna negativna vrijednost)
	Lažna vrijednost	**FP** (Lažna istinita vrijednost)	**TN** (Istinita negativna vrijednost)

Tablica 5: Matrica konfuzije

6.2.1. Točnost

Točnost predikcije (*Eng.* Accuracy score) je najjednostavnija mjera koja u obzir uzima omjer točno previđenih opservacija u odnosu na ukupni broj opservacija [28]. Kao što je prikazano na formuli u nastavku, u brojniku se razmatraju samo točno klasificirani slučajevi (Točni pozitivni i Točni negativni).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Ova mjera u našem slučaju ne nosi potrebnu informacijsku vrijednost zbog činjenice da pomoću nje ne dobivamo uvid u odnos lažno pozitivnih (**FP**) i istinito pozitivnih (**TP**) opservacija, a isto vrijedi i za negativne slučajeve. Također, u slučaju nebalansiranog skupa podataka, točnost može dovesti do pogrešnih zaključaka. Zbog toga je kod kreiranja konačnog mišljenja ukupnu točnost potrebno kombinirati s ostalim metrikama.

6.2.2. Preciznost

Preciznost (*Eng*. Precision) je mjera kojom kod algoritma ispitujemo sposobnost da klasifikator klasificira istinito točne opservacije u odnosu na sve pretpostavljeno točne opservacije [29]. Drugim riječima, ovom metrikom dobivamo uvid u broj netočno klasificiranih točnih opservacija. Na primjer analize sentimenta, postavlja se pitanje: Od svih istinito klasificiranih pozitivnih članaka (**TP**), koliko njih je zaista istinito klasificirano (**TP + FP**), odnosno koliki je omjer lažno pozitivno klasificiranih članaka među istinito klasificiranim člancima [28].

$$Precision = \frac{TP}{TP + FP}$$

6.2.3. Odziv

Odziv ili osjetljivost (*Eng.* Recall) je mjera kojom kod algoritma ispitujemo sposobnost da klasifikator identificira sve istinito pozitivne opservacije u odnosu na sve opservacije stvarne klase [30]. Na primjeru analize sentimenata, postavlja se pitanje: Od svih dohvaćenih istinito pozitivnih članaka (**TP**), koliko njih je zaista istinito pozitivno (**TP + FN**) [28].

$$Recall = \frac{TP}{TP + FN}$$

6.2.4. F1 mjera

F1 mjera (*Eng*. F1 score) je težinski prosjek mjera preciznosti i odaziva koji kod kalkulacije u obzir uzima broj lažno pozitivnih i lažno negativnih opservacija. Služi kao nadopuna mjeri točnosti, a najčešće se koristi kod nebalansiranih skupova podataka [28] [31].

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

6.3. Statistička obilježja skupa podataka

Potpoglavlje opisuje neka od svojstava deskriptivne statistike skupa podataka u kojem članci zadovoljavaju uvjet sadržavanja korona tematike te minimalno tri reakcije korisnika. Neka od tih svojstava su već opisana u prijašnjim tablicama (Tablica *2* i Tablica *3*).

Tablica *6* prikazuje silazno sortiranu distribuciju članaka po kategorijama i potkategorijama portala Dalmacija Danas. Vidljivo je da glavnina vijesti koje promatramo spadaju u kategorije Vijesti i Dalmacija iz čega možemo zaključiti da je portal primarno fokusiran na objavu vijesti svakodnevnih događanja. Promatrane članke svrstavamo u članke korona tematike, a oni se najčešće prikazuju u obliku svakodnevnih lokalnih i globalnih vijesti zbog čega je ovakva distribucija očekivana. Također, za očekivati je da će korisnici portala reagirati na svakodnevne vijesti komentarima i ocjenama putem *emojija* te zbog toga brojne kategorije koje nisu relevantne za korona tematiku nisu u promatranoj tablici.

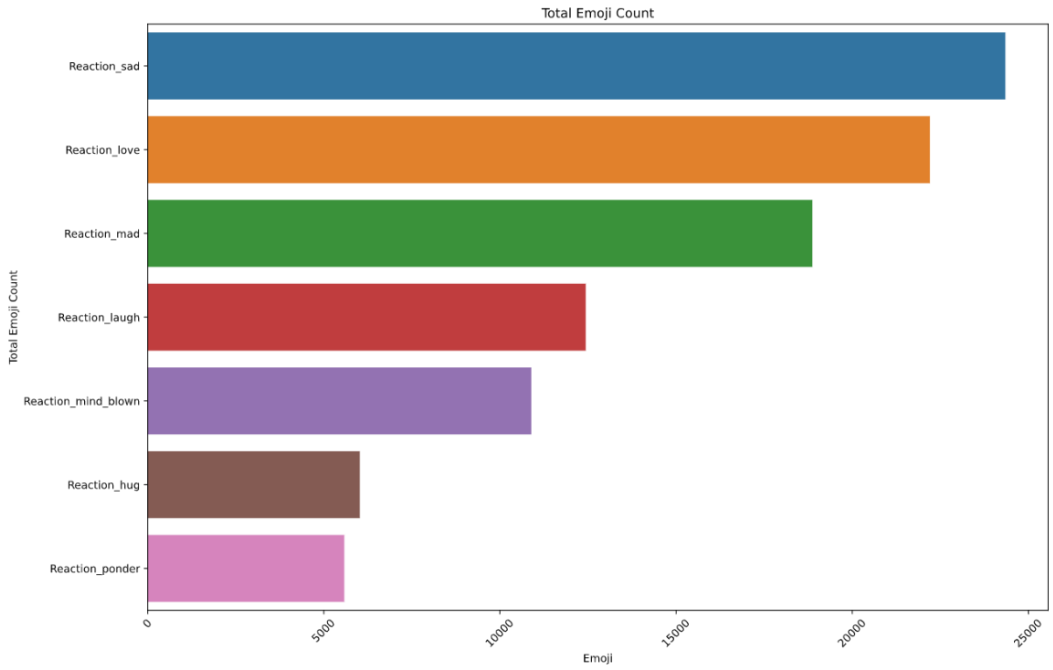
Kategorija	Broj članaka
Vijesti	1.564
Dalmacija	1.286
Relax	288
Sport	189
Hrvatska	19
Kolumne	19
Izbori 2021	9
Specijali	8
Split	4
Crna Kronika	2
Svijet	2
Obala	1
Otoci	1

Tablica 6: Distribucija članaka po kategorijama

Tablica *7* prikazuje ukupan broj reakcija putem *emojija* koji je izražen od strane korisnika. Na portalu je u promatranom vremenskom intervalu najviše korisnika reagiralo reakcijom kojom iskazuju žalost (**sad**), ali je broj takvih reakcija za dvije tisuće manji od pozitivne reakcije **love**. Zatim treće mjesto zauzima najveći broj ljutih reakcija, a na četvrtom mjestu stoji reakcija **laugh** koja može nositi sarkastičan kontekst, ali u nekim slučajevima i veseo. Svakako možemo zaključiti da veći broj reakcija korisnika upućuje na negativan kontekst jer manji udio reakcija čine pozitivne reakcije (**love** i **hug**). u nastavku prikazuje grafički prikaz spomenutih rezultata. Iz navedenog možemo uočiti da pandemija utječe na čitatelje u negativnom kontekstu do razine u kojoj moraju izraziti svoje nezadovoljstvo.

Emoji	**Broj reakcija**
Reaction sad	24.346
Reaction love	22.205
Reaction mad	18.868
Reaction laugh	12.434
Reaction mind blown	10.890
Reaction hug	6.023
Reaction ponder	5.581

Tablica 7: Distribucija reakcija putem *emojija* u člancima



Slika 14: Graf distribucija reakcija putem *emojija* u člancima

Tablica 8 daje uvid u distribuciju svih reakcija izraženih pomoću sedam *emojija* po kategorijama, respektivno. Najveći broj reakcija je prisutan u kategorijama **Dalmacija** i **Vijesti**, a zatim slijede kategorije **Relax** i **Sport** što nadopunjuje zaključke iz Tablica 6.

Najviše pozitivnih i negativnih reakcija pronalazimo u kategoriji **Dalmacija**, iako između njih postoji određena ravnoteža. Sljedeća kategorija s najviše reakcija je **Vijesti** u kojoj su najzastupljenije negativne emocije. Kod kategorija **Relax**, **Specijali**, **Sport** i **Kolumne** nailazimo na veći udio pozitivnih reakcija što ima smisla jer navedene kategorije možemo svrstati pod one „zabavnog" karaktera. U zabavnim kategorijama se koronavirus vjerojatno spominje usputno kao okolnost, a ne glavna tema.

Iako se radi o jednoj od manje zastupljenih, kategorija **Obala** pretežito ima negativne reakcije ljutnje što ukazuje da su čitatelji ogorčeni lokalnim odlukama. S druge strane, promatrajući kategoriju **Svijet** možemo zaključiti da vijesti iz svijeta na portalu Dalmacija Danas nisu popularne. Kategorija **Crna kronika** pretežito sadrži reakciju tužnog *emojija* (**sad**) zbog čega ovaj *emoji* možemo kategorizirati u negativnu, ali empatičnu reakciju.

Kategorija	**Reaction love**	**Reaction laugh**	**Reaction hug**	**Reaction ponder**	**Reaction sad**	**Reaction mad**	**Reaction mind blown**
Crna-kronika	2	5	2	4	**192**	15	11
Dalmacija	14.280	5.158	3.330	2.803	**14.343**	10.652	6.178
Hrvatska	117	50	38	39	128	**295**	80
Izbori-2021	**91**	74	10	37	11	21	31
Kolumne	**243**	35	59	17	35	15	30
Obala	4	35	2	43	48	**138**	31
Otoci	**4**	3	1	2	1	2	0
Relax	2.358	**2.634**	701	445	960	811	837
Specijali	**141**	8	43	8	7	13	18
Split	10	12	5	2	7	**31**	9
Sport	**1.018**	264	243	112	372	228	185
Svijet	1	**5**	0	0	2	1	1
Vijesti	3936	4.151	1.589	2.069	**8.240**	6.646	3.479

Tablica 8: Suma reakcija korisnika putem *emojija* po kategorijama

Tablica 9 prikazuje distribuciju *emojija* po negativnoj i ne-negativnoj klasi. Ovim putem možemo ocijeniti naše proizvoljno zadane koeficijente pozitivnosti (Tablica *3*) svakog pojedinog *emojija* ovisno o pripadnosti pojedinoj klasi.

Iz rezultata zaključujemo da **Laugh** *emoji* sigurno predstavlja sarkastičan kontekst zbog pretežite uporabe u NEGATIVE člancima, što vrijedi i za **Mind Blown** *emoji*. Naj neutralniji *emoji*i u ovom kontekstu su **Ponder** i **Hug**, ali i oni blago naginju prema negativnom i pozitivnom kontekstu što je još jedan dodatan argument za ujedinjenje pozitivne i neutralne klase.

Nadalje, *emoji*i koji pripadaju NONNEGATIVE klasi a nose pozitivan kontekst se slažu s procjenom iznesenom u Tablica *3*. **Love** i **Hug** su zasigurno *emojiji* koji nose pozitivan kontekst zbog čega slijedi da su svi *emojiji* dobro ocijenjeni te da su koeficijenti uspješno validirani.

	Love	**Laugh**	**Hug**	**Ponder**	**Sad**	**Mad**	**Mind Blown**	**Total**
NEGATIVE	4.568	10.167	2.342	4.205	2.3283	17.077	9.123	61.642
NONNEGATIVE	17.637	2.267	3.681	1.376	1.063	1.791	1.767	27.815

Tablica 9: Distribucija *emojija* prema klasama

6.4. Klasifikacija prema polaritetu sentimenta

Prvi pristup za rješavanje problema klasifikacije korištenjem triju algoritama od kojih dva su izrađena specifično za potrebe ovog projekta, dok treći služi za evaluaciju rezultata. Izvedba vlastitih algoritama je provedena skriptom *article_positivity.py* koja nad promatranim tekstom računa polaritet pozitivnosti nenadziranim pristupom korištenjem leksikona. Tim pristupima pokušavamo pronaći ponovljive uzorke koji predstavljaju pozitivnost korištenih riječi u polarnom pozitivnom i negativnom intervalu.

Ručna ocjena pozitivnosti velikih količina riječi bi zbog subjektivnosti ocjenjivača i potrebnog vremena bila nemoguć pothvat. Zbog toga, za određivanje polariteta kod prva dva algoritma korišten je leksikon *CroSentiLex* [32] koji sadrži 37 tisuća hrvatskih riječi koje su grupirane u dvije tekstualne datoteke prema pozitivnoj i negativnoj skali u intervalu [0,1], odnosno [-1,0]. Za ocjenu svake pojedine riječi, *CroSentiLex* kreiran pomoću *PageRank* algoritma kojim je mjerena frekvencija učestalog ponavljanja i važnosti riječi. Treći algoritam za razliku od onih vlastite izrade koristi razrađene metode i gotove leksikone pomoću kojih možemo dobiti detaljan uvid u razliku između jednostavnog i kompleksnog načina za rješavanje problema polariteta sentimenta

Usporedba rezultata je realizirana korištenjem mjera točnosti, preciznosti, odaziva i F1 mjerom. Te mjere su opisane u poglavlju 6.2, a služe za dobivanje vjerodostojnih i standardiziranih podataka o točno i netočno klasificiranim uzorcima koji se u projektu koriste. U nastavku slijedi opis i interpretacija svakog pojedinog algoritma i dobivenih rezultata.

6.4.1. Analiza sentimenta na razini tekstualnog elementa

Prvi algoritam za izračun sentimenta predstavlja pokušaj korištenja najintuitivnijeg pristupa za rješavanje problema izračuna sentimenta korištenjem ukupne sume pozitivnosti na razini promatranog tekstualnog elementa.

Koraci korišteni kroz implementaciju su opisani u nastavku:

- U funkciju se iz *.csv* datoteke učitava lista koja sadrži element naslova/podnaslova/teksta članka u kojem je svaka riječ prethodno svedena na kanonski oblik procesom lematizacije.
- Lista se razlama na riječi, pri čemu su kao separatori korišteni interpunkcijski znakovi i razmak između riječi. Tim se postupkom dobiva lista u kojoj svaki element predstavlja jedna riječ.
- Za svaku riječ u listi iz prošlog koraka prolazimo kroz liste pozitivnih i negativnih riječi iz *CroSentiLex* rječnika. Ukoliko je promatrana riječ pronađena, brojčani koeficijent uz riječ se pridodaje ukupnoj sumi.
- Dobivena suma koeficijenata se dijeli s brojem riječi tekstualnog elementa.
- Koeficijenti se spremaju u *.csv* datoteku za daljnju obradu.
- Podaci s izračunatim koeficijentom iz svih dobivenih redaka se normaliziraju u interval [-1,1]. Treba napomenuti da u ovom koraku poništavamo vrijednosti koje višestruko odstupaju od ostalih te se one tretiraju kao greške u računanju i kao takve su izbačene iz razmatranja.
- U daljnjem koraku se algoritam evaluira računanjem ukupnog postotka točnosti klasifikacije, te mjere preciznosti, odaziva i F1 mjere.
- Ukoliko je dobivena vrijednost veća ili jednaka nuli, tekstualnu vrijednost pridodajemo klasi NONNEGATIVE, ukoliko je manja od nule, pridodajemo joj vrijednost NEGATIVE.

Analizom polariteta sentimenta po **naslovima** (Slika referenca) članaka ukupna dobivena točnost iznosi 53,39% Vidljiva je velika oscilacija u točnosti između NEGATIVE i NONNEGATIVE klase u vidu mjera preciznosti, odaziva i F- mjere. Taj problem možemo pripisati i činjenici da klasa NEGATIVE sadrži puno veći broj uzoraka. To znači da čak i ako algoritam ima veći postotak točnosti, toj točnosti ne treba vjerovati jer će algoritam u većini slučajeva „pogoditi“ da se radi o NEGATIVE klasi. To je posebno vidljivo na mjerama preciznosti i odaziva. Rezultat je gori od očekivanog jer se naslov sastoji od manjeg broja riječi, pa ima i manje prostora za anomalije u radu algoritma.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NEGATIVE': 1482, 'NONNEGATIVE': 743})

              precision    recall  f1-score   support

    NEGATIVE       0.64      0.65      0.65      1453
 NONNEGATIVE       0.32      0.31      0.32       772

    accuracy                           0.53      2225
   macro avg       0.48      0.48      0.48      2225
weighted avg       0.53      0.53      0.53      2225

Accuracy Score:  53.3932584269663
```

Slika 15: Rezultati algoritma provedenih nad cijelim naslovom članka

Rezultati analize polariteta sentimenta prema tekstu članka su lošiji od onih dobivenih računanjem nad naslovom članka te točnost pritom iznosi 52,81%. U rezultatima je ponovo vidljiva prije spomenuta oscilacija u rezultatima. Iz samog pristupa je vidljivo da računanje sentimenta prema cijelom skupu riječi ne može dati dobre rezultate jer će pozitivni i negativni pol uvijek utjecati jedan na drugoga ne uzimajući segmente rečenica u obzir. Korištenje ovakvog pristupa je poželjno samo u slučajevima kad evaluiramo kratke tekstualne elemente. U takvim će elementima pokazivati slične performanse kao algoritam koji analizira svaku rečenicu respektivno pod pretpostavkom da korišteni leksikon koristi skup riječi promatrane tematike.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NEGATIVE': 1369, 'NONNEGATIVE': 856})

              precision    recall  f1-score   support

    NEGATIVE       0.65      0.61      0.63      1453
 NONNEGATIVE       0.34      0.37      0.36       772

    accuracy                           0.53      2225
   macro avg       0.49      0.49      0.49      2225
weighted avg       0.54      0.53      0.53      2225

Accuracy Score:  52.80898876404494
```

Slika 16: Rezultati algoritma provedenih nad cijelim tekstom članka

6.4.2. Analiza sentimenta na razini rečenice

Računanje polariteta sentimenta nad ukupnim tekstom zanemaruje činjenicu da svaka rečenica nosi svoj doprinos u ukupno formiranoj misli, pa i naravi kojom ju autor pokušava iskazati. Zbog detaljnijeg uvida u sentiment svakog pojedinog teksta, implementiran je drugi algoritam čiji je cilj sentiment analizirati na razini svake pojedine rečenice.

Analiza rečenice je kompleksan proces u kojem je moguće promatrati gramatičke pojave koje bi se kroz daljnje istraživanje mogle pokazati kao česta pojava u pozitivnim ili negativnim sentimentima. U ovom slučaju se radi o jednostavnom algoritmu koji na razini rečenice pronalazi poznate riječi, uzimajući u obzir samo pojavu negacije riječi kao negativnog fenomena.

Način rada je opisan u nastavku:

- Učitavanje skupa podataka koristi istu metodu kao i algoritam opisan u poglavlju 6.4.1.
- Priprema skupa podataka koristi istu metodu kao i algoritam u poglavlju 6.4.1.
- Svaki ulazni tekstualni element se razlama na rečenice čije se riječi uspoređuju s listama pozitivnih i negativnih riječi iz *CroSentiLex* rječnika.
- Ukoliko je promatrana riječ u rečenici pronađena u rječniku, vrijednost pridodana toj riječi se dodaje ili oduzima od ukupne sume za tu riječ.
- Algoritam ima jednostavni sustav za detekciju negacije u rečenici. Ukoliko kroz obradu prođe kroz riječ „ne“, u ostatku rečenice svaki koeficijent sljedećih riječi poprima negativni predznak.
- Ukupna dobivena suma rečenice se dijeli s ukupnim brojem riječi, a rezultat čini lista vrijednosti sentimenta za svaku od njih.
- Konačna vrijednost sentimenta se dobiva računanjem prosječne vrijednosti dobivene iz liste izračunatih koeficijenata za svaku pojedinu rečenicu.
- Vrijednosti su normalizirane, a zatim su pridodane pripadajućim NONNEGATIVE i NEGATIVE klasama na isti način kao u algoritmu poglavlja 6.4.1.

Implemetacijom ovog algoritma nad **naslovom članka** očekivana je povećana točnost od one dobivene algoritmom koji sentiment određuje na razini tekstualnog elementa. To se u eksperimentu nije pokazalo točnim, a ukupna točnost se je pokazala manjom od očekivane, te ona iznosi 48,18%. Ponovno dolazimo do velike razlike u mjeri preciznosti iz čega je vidljivo da je veliki broj negativnih članaka klasificiran pod NONNEGATIVE. Mjera odaziva nosi jednaku vrijednost za obje klase, ponovno uz loš rezultat, dok F1 mjera ima bolji rezultat kod NEGATIVE klase.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NONNEGATIVE': 1125, 'NEGATIVE': 1100})

              precision    recall  f1-score   support

    NEGATIVE       0.64      0.48      0.55      1453
 NONNEGATIVE       0.33      0.48      0.39       772

    accuracy                           0.48      2225
   macro avg       0.48      0.48      0.47      2225
weighted avg       0.53      0.48      0.49      2225

Accuracy Score:  48.17977528089888
```

Slika 17: Rezultati algoritma za računanje sentimenta po rečenici nad naslovom članka

Pokretanjem algoritma nad cijelim **tekstom članka**, ukupna točnost je nešto niža od one nad naslovom članka, te ona iznosi 42,88%. Mjera preciznosti daje gotovo identične rezultate, iako je vidljivo da algoritam točnije prepoznaje klasu NONNEGATIVE. Tu pojavu možemo objasniti činjenicom da je negativnim člancima korištenjem uvedenog pravila za riječ „ne“ drastično smanjena pozitivnost tekstualnog elementa, zbog čega klasifikator lakše prepoznaje NONNEGATIVE klasu. Iz ovog primjera možemo zaključiti da je uvođenje tog pravila pozitivno pridonjelo ukupnoj klasifikaciji.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NONNEGATIVE': 1457, 'NEGATIVE': 768})

              precision    recall  f1-score   support

    NEGATIVE       0.62      0.33      0.43      1453
 NONNEGATIVE       0.33      0.62      0.43       772

    accuracy                           0.43      2225
   macro avg       0.47      0.47      0.43      2225
weighted avg       0.52      0.43      0.43      2225

Accuracy Score:  42.87640449438202
```

Slika 18: Rezultati algoritma za računanje sentimenta po rečenici nad tekstom članka

6.4.3. Analiza sentimenta na razini rečenice bibliotekom VADER

Za razliku od prethodna dva algoritma, u zadnjem eksperimentu nenadziranog pristupa korišten je *„Out of the box“* pristup u obliku specijaliziranog alata VADER za analizu sentimenta. VADER (*Eng.* Valence Aware Dictionary and sEntiment Reasoner).

VADER je alat za analizu sentimenta baziran na korištenju predefiniranog skupa riječi (leksikona) pritom u obzir uzimajući poznate relacije između riječi koje formiraju određeni polaritet sentimenta. Alat uključuje predviđanje za razne lingvističke fenomene kao što su negacija riječi, korištenje raznih vrsta i kombinacija interpunkcijskih znakova, slenga, *emojija* i sličnih pojava. Korpus riječi kojim se alat koristi je specijaliziran za određivanje sentimenta na društvenim mrežama, a riječi su prikupljene iz izvora ko što su Twitter, New York Times, Amazon, *itd*. [33]

Algoritam je implementiran na isti način kao i algoritmi vlastite izrade. Skup podataka nad kojim je polaritet računat je iskorišten u izvornom obliku zbog VADER-ovih opširnih mogućnosti analize teksta. Budući da je alat optimiziran za engleski jezik, svaki tekstualni element je preveden neposredno prije obrade korištenjem *google_trans_new* biblioteke za jezik Python. Rezultat algoritma čini polaritet u intervalu [-5,5] za svaku rečenicu koji je podijeljen s ukupnim brojem riječi. Zatim je od dobivenih koeficijenata izračunata prosječna vrijednost koja predstavlja konačni sentiment za promatrani tekstualni element.

Korištenjem VADER algoritma dobiven je najbolji rezultat kod pristupa nenadziranog učenja. Ukupna točnost računanja polariteta sentimenta nas **naslovima članka** iznosi 62,29%, što je u skladu s očekivanjima. Kroz sva tri algoritma je uočljiva velika razlika u mjeri preciznosti, odnosno kod klasifikacije NONNEGATIVE uzoraka. Potrebno je istaknuti i mjeru odaziva čiji je rezultat sličan prošlom, puno primitivnijem algoritmu, a taj rezultat prelazi 60%.

Iz rezultata svih algoritama nad tekstom članka je vidljivo da nenadzirani klasifikator daje bolje rezultate nad manjom količinom teksta, što u pravilu vrijedi za sve naslove. Naslov u ovom slučaju predstavlja kratko i sažeto objašnjenje teksta, te ukupna količina teksta nikada ne prelazi dvije do tri rečenice. S obzirom da ti naslovi sadrže manji broj riječi i rečenica, može se pretpostaviti da manji broj dobivenih koeficijenata polariteta rečenice pozitivno utječe na ukupni rezultat.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NEGATIVE': 1138, 'NONNEGATIVE': 1087})

              precision    recall  f1-score   support

    NEGATIVE       0.77      0.60      0.68      1453
 NONNEGATIVE       0.47      0.66      0.55       772

    accuracy                           0.62      2225
   macro avg       0.62      0.63      0.61      2225
weighted avg       0.67      0.62      0.63      2225

Accuracy Score:  62.29213483146068
```

Slika 19: Rezultati VADER algoritma po rečenici nad naslovom članka

Kao i u svim prošlim algoritmima koji su korišteni nad **tekstom članka**, korištenjem VADER algoritma dobivena je manja točnost od one dobivene na tekstualnom elementu naslova. Ta točnost iznosi 57,17% što je čini najvišom od svih koje su promatrane nad elementom teksta. Ponovno dolazimo do fenomena velike razlike u mjeri preciznosti i boljeg rezultata odaziva NONNEGATIVE klase.

Iako VADER algoritam rezultira najboljom klasifikacijom, u obzir treba uzeti činjenicu da alat nije optimiziran za hrvatski jezik. Kod evaluacije rezultata i korištenje algoritma nije ocjenjena kvaliteta strojnog prijevoda dobivenog pomoću *Google Translate API*-a. Zbog velike količine tekstova i kratkog roka izrade, kao i potrebe za većim brojem ocjenjivača, nije bilo moguće „ručno“ obraditi takav skup podataka. Također, alat ne koristi korpus riječi vezan za tematiku koronavirusa, zbog čega ove rezultate u odnosu na pristup možemo smatrati zadovoljavajućima.

```
True value count:  Counter({'NEGATIVE': 1453, 'NONNEGATIVE': 772})
Prediction value count:  Counter({'NONNEGATIVE': 1429, 'NEGATIVE': 796})

              precision    recall  f1-score   support

    NEGATIVE       0.81      0.45      0.58      1453
 NONNEGATIVE       0.44      0.81      0.57       772

    accuracy                           0.57      2225
   macro avg       0.63      0.63      0.57      2225
weighted avg       0.68      0.57      0.57      2225

Accuracy Score:  57.168539325842694
```

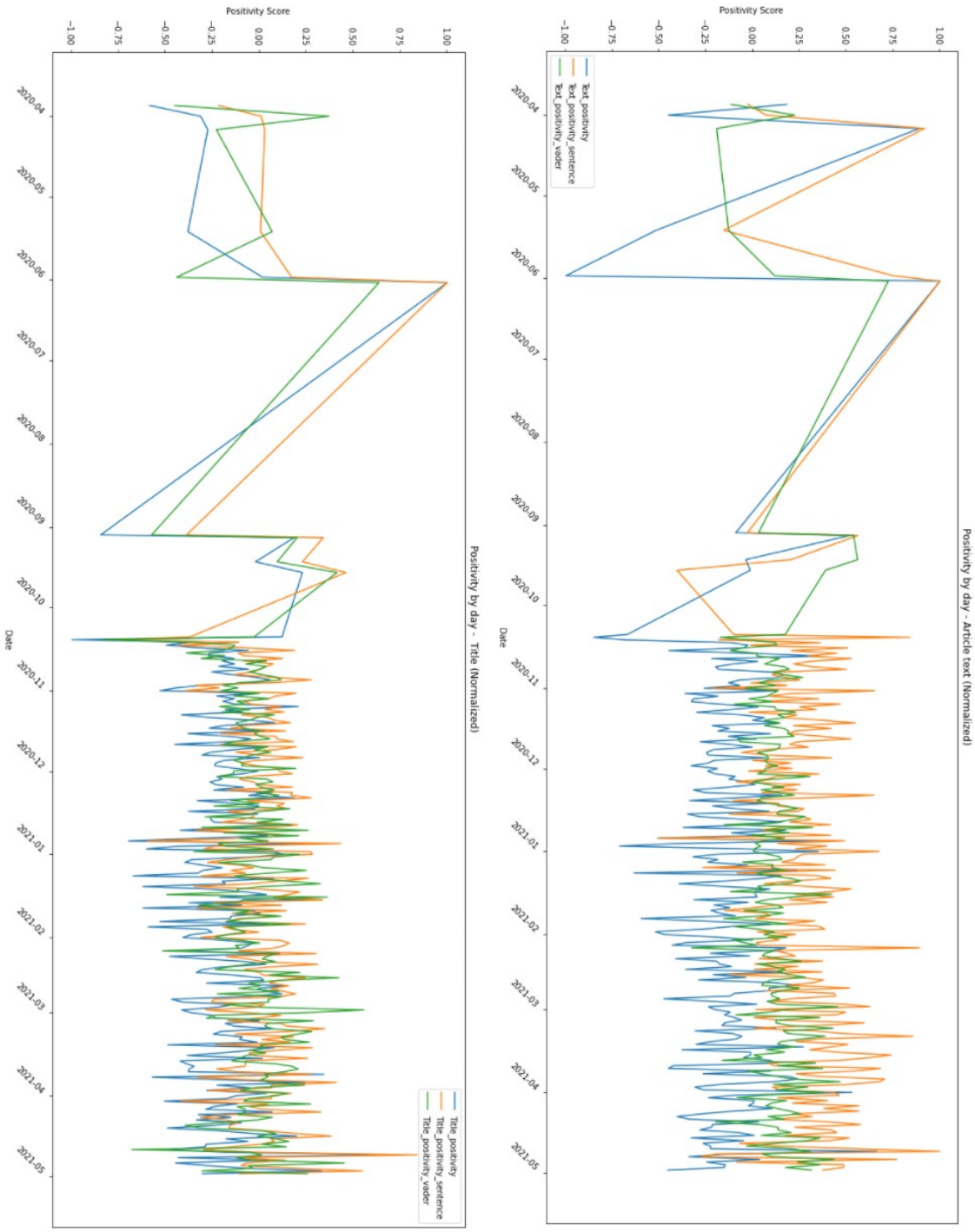
Slika 20: Rezultati VADER algoritma nad tekstom članka

6.4.4. Grafički prikaz toka polariteta kod nenadziranog učenja

Finalne vrijednosti koeficijenta dobivenih računanjem polariteta sentimenta su NEGATIVE i NONNEGATIVE ovisno o uvjetu je li koeficijent u negativnom ili pozitivnom intervalu. Kao takav nam omogućava da pomoću algoritama i obrade koeficijenata koristimo binarnu klasifikaciju nad danim skupom podataka. Međutim, iz tih neobrađenih koeficijenata je moguće doći do određenih opažanja u vidu kontinuiteta broja pozitivnih i negativnih članaka čime se bave grafikoni u nastavku. Grafikoni prikazuju prosječnu dnevnu pozitivnost svih članaka u promatranom razdoblju na koji se skup podataka odnosi. Na taj način je prikazan tok pozitivnosti koji se je na portalu odvijao iz perspektive nenadziranog učenja u razdoblju od travnja 2020. do svibnja 2021 godine.

Prvi detalj koji je potrebno objasniti je velika praznina u broju članaka koja traje do listopada 2020. godine. U analizu sentimenta su uvršteni samo oni članci na kojima postoji tri ili više reakcija korisnika putem *emojija*, dok je ostatak izbačen iz razmatranja. Vizualnom subjektivnom analizom je uočeno da članci u spomenutom razdoblju uopće ne sadrže reakcije korisnika, dok nakon listopada broj reakcija munjevito raste. Razlog tome može biti činjenica da *emojiji* za izražavanje mišljenja dobivaju na popularnosti tek u listopadu, te da prije korisnici nisu imali tendenciju koristiti ih. Druga teorija za manjak reakcija je onemogućavanje korištenja *emojija* ili brisanje broja reakcija iz baze podataka od strane portala Dalmacija Danas.

Iz oba grafikona je vidljiv stalan kontinuitet rasta pozitivnog koeficijenta, kao i pada u negativni pol. Naprimjer, U razdoblju između studenog (11.) 2020. i siječnja(1.) 2021. vidljivo je očit porast pozitivnosti na dnevnoj bazi. Na isti način možemo promatrati razdoblje između ožujka (3.) 2021. i travnja (4.) 2021. u kojemu je pozitivnost također porasla. Što se tiče negativnog osciliranja, najniže (najnegativnije) vrijednosti su zastupljene u siječnju (1.) 2021. Koristeći se promatranim podacima je kroz detaljnije analize moguće odrediti razdoblja u kojima čitatelji osjećaju veće nezadovoljstvo. S obzirom da se radi samo o člancima tematike koronavirusa, iz ovog pristupa je moguće odrediti u kojim razdobljima je virus bio više zastupljen, kada su počinjali novi valovi zaraze i slične opservacije.



Slika 21: Grafički prikaz prosječnih dnevnih vrijednosti koeficijenata nenadziranog učenja

6.5. Klasifikacija algoritmima nadziranog učenja

Za klasifikaciju sentimenata korištenjem strojnog učenja korištena je programska podrška *Jupyter* bilježnice nad skriptom *sentiment_ml.ipynb* koja omogućuje interaktivni rad s Python kodom [34]. Segmente koda moguće izvoditi u logičkim cjelinama pritom zadržavajući rezultate izvan konzole što povećava fleksibilnost testiranja i pregleda rezultata.

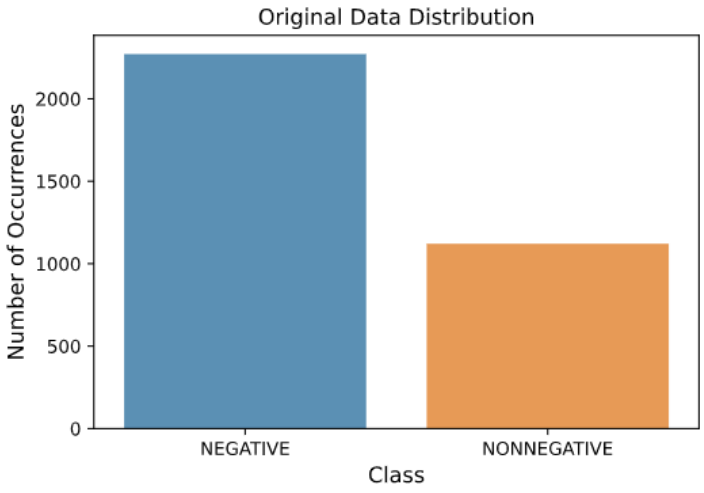
Cijela izvedba strojnog učenja je provedena u *sckit-learn* Python biblioteci. *Sckit-learn* je biblioteka koja objedinjuje veliki dio operacija strojnog učenja u jeziku Python. Pomoću nje su omogućene jednostavne i kompleksne operacije nad podacima kao što su klasifikacija, regresija, grupiranje objekta u klase, pred-obrada podataka i brojne druge tehnike [35].

Skup podataka nad kojim se je provedeno strojno učenje je *portal_articles_classes.csv*. Radi se o skupu podataka u kojem su sve riječi svedene na osnovni oblik, a dijakritički znakovi nisu izbačeni iz razmatranja. Također, broj stupaca iznosi 25, a svi su opisani u tablici (Tablica *1*), uz dodatak koeficijenta pozitivnosti te kategoričke varijable koja označava pozitivnost.

6.5.1. Balansiranje skupa podataka

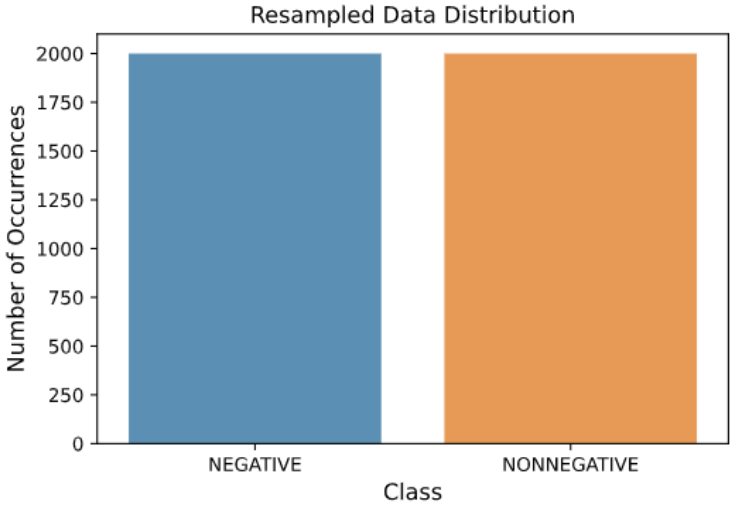
Kao što je uočeno na tablici (Tablica *4*), u korištenom skupu podataka se suočavamo s nebalansiranim skupom podataka što je također vidljivo iz grafa na slici (Slika *23*). Zbog veće zastupljenosti jedne klase u odnosu, klasifikator može postati „prenaučen“ na pojavljivanje jedne klase, dok druga može ostati zanemarena. U ovom slučaju radi se o gotovo dvostrukom pojavljivanju negativne klase (2271) u odnosu na ne-negativnu (1121).

Nebalansiranost podataka je moguće tretirati balansiranjem skupa podataka na način da smanjimo uzrokovanje (*Eng*. Downsample) ili povećamo uzrokovanje (*Eng*. Upsample) problematične klase. Drugim riječima, pomoću *downsample* metode, smanjujemo broj slučajeva, dok ih pomoću *upsample* metode povećavamo. U tu svrhu je implementirana metoda *resample()* iz paketa *sckit-learn* koja omogućuje navedenu operaciju [36].



Slika 22: Graf nebalansirane distribucije klasa

Proizvoljnim eksperimentiranjem je zaključeno da povećanje pojavljivanja ne-negativne klase do izjednačavanja s negativnom dobivamo najbolji balans klasifikacije klasa. Drugim riječima, nad većinskom klasom je smanjen broj pojavljivanja čime je izbjegnuto uvođenje prevelike količine sintetičkih podatka ne-negativne (manjinske) klase čiji je broj povećan. Graf na slici (Slika *23*) u nastavku prikazuje balansirani skup u kojem je broj negativnih pojavljivanja smanjen na 2.000, dok je broj ne-negativnih pojavljivanja povećan na 2.000.



Slika 23: Graf balansirane distribucije klasa

6.5.2. Vektorizacija i podjela podataka na skupove za treniranje i testiranje

Algoritmi umjetne inteligencije nisu zamišljeni za rad s velikim tekstualnim (*string*) elementima, zbog toga je potrebno transformirati podatke nad kojima ćemo raditi predikciju. Promatrane tekstualne stupce je potrebno pretvoriti u numerički oblik, za što je odabrana metoda *TF-IDF* (*Eng.* Term Frequency -Inverse Document Frequency).

TF-IDF je numeričko statistička operacija kojom prikazujemo važnost svake pojedine riječi nekog dokumenta. *TF* predstavlja broj pojavljivanja riječi u dokumentu u odnosu na ukupan broj riječi, dok *IDF* za dodjeljivanje težine riječima ovisno o njihovom broju ponavljanja i važnosti [37, str. 25]. Zbog rada *IDF* značajke nije potrebno očistiti zaustavne riječi jer se njima upravo zbog frekventnosti pridodaje manji značaj. Algoritam *TF-IDF*-a je implementiran metodom *TfidfVectorizer()* iz paketa *sckit-learn* [38].

Nakon što su podaci transformirani u numerički oblik, kreću standardni postupci koji se koriste kod strojnog učenja. U svrhu određivanja pozitivnosti, izrađuju se dva skupa za treniranje, te dva skupa za testiranje modela. U jednom skupu promatramo s kolikim postotkom uspješnosti možemo klasificirati pozitivnost samo prema naslovu članka (*Title*), dok u drugom promatramo je li pozitivnost moguće dobiti koristeći cijeli tekst članka (*Article_text*). Omjer podataka za treniranje i testiranje je postavljen na 70:30, a poredak podataka je nasumično odabran.

6.5.3. K-Fold unakrsna validacija

K-Fold unakrsna validacija (*Eng*. *K-Fold* cross-validation) je metoda za procjenu točnosti predikcije na skupu za treniranje. Algoritmom skup za treniranje dijelimo u *K* odvojenih pod skupova jednake veličine koji ne sadrže iste podatke. Zatim model treniramo na *K-1* pod skupova, a na jednom *K* skupu evaluiramo rezultate treniranja i mjerimo performanse. Postupak se ponavlja za sve pod skupove dok sve kombinacije skupova nisu iskorištene, a konačni rezultat čini prosječna vrijednost rezultata [39].

Navedena metoda na našem skupu podataka služi kao dodatna provjera točnosti treniranja skupa podataka. Kroz sve slučajeve strojnog učenja *K* vrijednost iznosi 5. Radi ubrzavanja procesa unakrsne validacije korišten je grafički procesor.

6.5.4. Naivni Bayesov klasifikator

Iako je očekivano da Naivni Bayesov klasifikator neće rezultirati najboljom klasifikacijom, korišten je zbog brzine računanja i testiranja finih radnji kod pripreme za treniranje modela uz iznenađujuće dobre rezultate. Te radnje uključuju balansiranje i podjelu skupova, podešavanje unakrsne validacije, *itd*.

Treniranjem modela na **naslovima** članaka dobivena je točnost od 81,91%, koja je također unakrsnom validacijom koja iznosi 79,60%. Iz rezultata klasifikacije (*Result values*) i broja testnih podataka (*Test target values*) vidljivo je da klasifikator dodjeljuje veći broj krivih opservacija NEGATIVE klasi (617). Mjere preciznosti, odaziva i F1-mjere nemaju velike oscilacije između NEGATIVE i NONNEGATIVE klase, a njihov rezultat je ~80% točnost. Rezultat F1-mjere je pri jednak za sve klase, te iznosi 0,82 za NEGATIVE i NONNEGATIVE klasu te kroz ove mjere vidimo da algoritam nema problema s prenaučenosti za neku od dvaju klasa.

```
Title:

Naive Bayes:
==================================

Train target values count:  Counter({'NONNEGATIVE': 1400, 'NEGATIVE': 1400})
Test target values count:  Counter({'NONNEGATIVE': 600, 'NEGATIVE': 600})
Result values count:  Counter({'NEGATIVE': 617, 'NONNEGATIVE': 583})

              precision    recall  f1-score   support

    NEGATIVE       0.81      0.83      0.82       600
 NONNEGATIVE       0.83      0.81      0.82       600

    accuracy                           0.82      1200
   macro avg       0.82      0.82      0.82      1200
weighted avg       0.82      0.82      0.82      1200

Accuracy Score:  81.91666666666667
Accuracy(K-Fold): 79.607 (0.027)
```

Slika 24: Rezultati klasifikatora Naive Bayes nad naslovom članka

Kod treniranja modela na **tekstu članka** dobivena je manja točnost (77%) koja je također utvrđena unakrsnom validacijom (75,71%), ali ta točnost je manja nego kod naslova članaka. U ovom modelu također prevladava kriva klasifikacija NEGATIVE kao NONNEGATIVE klase što je vidljivo u mjeri odaziva što upućuje na vidljivu prenaučenost NONNEGATIVE klase. S obzirom na veće količine teksta u odnosu na naslove, razlike između mjera preciznosti, odaziva i F1-mjere su puno veće.

Kada promatramo preciznost NONNEGATIVE klase vidimo da ona iznosi 0,74, u odnosu na 0,80 kod NEGATIVE klase što potvrđuje veći broj krivo klasificiranih istinitih NONNEGATIVE opservacija. S druge strane, kod mjere odaziva dobivamo lošiji rezultat kod NEGATIVE klase (0,71) u odnosu na NONNEGATIVE klasu (0,82). Iz toga možemo zaključiti da je broj istinito klasificiranih jedinki NEGATIVE klase puno manji. Ukupna F1-mjera je približno jednaka, a iznosi 0,76 za NEGATIVE klasu te 0,78 za NONNEGATIVE što ukazuje da klasifikator radi relativno uravnoteženo za obje klase.

```
Article text:

Naive Bayes:
================================

Train target values count:  Counter({'NEGATIVE': 1400, 'NONNEGATIVE': 1400})
Test target values count:  Counter({'NEGATIVE': 600, 'NONNEGATIVE': 600})
Result values count:  Counter({'NONNEGATIVE': 666, 'NEGATIVE': 534})

              precision    recall  f1-score   support

    NEGATIVE       0.80      0.71      0.76       600
 NONNEGATIVE       0.74      0.82      0.78       600

    accuracy                           0.77      1200
   macro avg       0.77      0.77      0.77      1200
weighted avg       0.77      0.77      0.77      1200

Accuracy Score:  77.0
Accuracy(K-Fold): 75.714 (0.008)
```

Slika 25: Rezultati klasifikatora Naive Bayes nad tekstom članka

6.5.5. Klasifikator metode potpornih vektora

Sljedeći algoritam se odnosi na metodu potpornih vektora (SVM), a njegovim korištenjem je očekivana najveća točnost, što se na kraju nije pokazao slučajem. Kod implementacije ovog algoritma su korištene zadane (*Eng*. Default) vrijednosti parametara.

Treniranjem modela na **naslovu članka** ostvarena je točnost od 81,75% uz gotovo identični rezultat dobiven unakrsnom validacijom (81,86%). Za razliku od Naivnog Bayesa, ovaj algoritam NONNEGATIVE klasi u odnosu na NEGATIVE pridodaje veći broj opservacija s razlikom mjera preciznosti i odaziva manjom od 5%. To možemo pripisati povećanom broju NONNEGATIVE klase metodom preuzrokovanja zbog čega vidimo blagu prenaučenost na NONNEGATIVE klasu.

Kod detaljnije analize možemo potvrditi tezu prenaučenosti pomoću mjere odaziva kod NEGATIVE klase u kojoj je vidljivo da je ta vrijednost manja (0,79). Mjera preciznosti pruža sličan rezultat kod NONNEGATIVE i NEGATIVE klase što upućuje da klasifikator dobro klasificira istinito pozitivne uzorke. Mjera F1 je dobro balansirana, a rezultat je prihvatljiv kada u obzir uzmemo činjenicu da je algoritam učen na naslovima članaka koji imaju manji broj riječi od tekstova članaka.

```
Title:

SVM (Default):
==================================

Train target values count:  Counter({'NONNEGATIVE': 1400, 'NEGATIVE': 1400})
Test target values count:  Counter({'NONNEGATIVE': 600, 'NEGATIVE': 600})
Result values count:  Counter({'NONNEGATIVE': 631, 'NEGATIVE': 569})

              precision    recall  f1-score   support

    NEGATIVE       0.83      0.79      0.81       600
 NONNEGATIVE       0.80      0.84      0.82       600

    accuracy                           0.82      1200
   macro avg       0.82      0.82      0.82      1200
weighted avg       0.82      0.82      0.82      1200

Accuracy Score:  81.75
Accuracy(K-Fold): 81.857 (0.024)
```

Slika 26: Rezultati SVM klasifikatora nad naslovom članka

Treniranjem modela na **tekstu članka** finalna točnost iznosi 83,92% što je za 3% više od one postignute unakrsnom validacijom (80,96%). Iz rezultata je vidljivo da je točnost algoritma određena standardnim metrikama za razliku od Naivnog Bayesa i SVM-a nad naslovima veća. To možemo objasniti činjenicom da SVM najbolje radi s većom količinom ulaznih podataka, a to su u našem slučaju riječi teksta članka kojih ima daleko više nego u naslovu.

Za razliku od prijašnjeg SVM modela, sve mjere imaju veću točnost za oko 2 do 4%, ali je kao i kod prijašnjih algoritama vidljiva prenaučenost na NONNEGATIVE klasu što je vidljivo kod odaziva NEGATIVE klase koji iznosi 81%. Performanse za određivanje istinito klasificiranih uzoraka određene mjerom preciznosti su slične modelu koji klasificira prema naslovu uz 2-3% bolje performanse. F1 mjera za obje klase je ponovno dobro balansirana, te niti jedna u konačnom rezultatu ne prevladava te za NEGATIVE klasu iznosi 83%, dok za NON-NEGATIVE klasu iznosi 84%.

```
Article text:

SVM (Default):
==================================

Train target values count:  Counter({'NEGATIVE': 1400, 'NONNEGATIVE': 1400})
Test target values count:  Counter({'NEGATIVE': 600, 'NONNEGATIVE': 600})
Result values count:  Counter({'NONNEGATIVE': 637, 'NEGATIVE': 563})

              precision    recall  f1-score   support

    NEGATIVE       0.86      0.81      0.83       600
 NONNEGATIVE       0.82      0.87      0.84       600

    accuracy                           0.84      1200
   macro avg       0.84      0.84      0.84      1200
weighted avg       0.84      0.84      0.84      1200

Accuracy Score:  83.91666666666666
Accuracy(K-Fold): 80.964 (0.013)
```

Slika 27: Rezultati SVM klasifikatora nad tekstom članka

6.5.6. Klasifikator algoritma slučajnih šuma

Zadnji algoritam strojnog učenja korišten u analizi je algoritam slučajnih šuma. Kroz implementaciju su dobiveni mješoviti rezultati koji su interpretirani u nastavku.

Treniranjem modela na **naslovu članka** ukupna točnost iznosi 83,5%, što je također potvrđeno unakrsnom validacijom (82,4%). Implementacija ovog algoritma služi kao dobar uvid u lošu pouzdanost mjere točnosti. Kao što je vidljivo na primjeru preciznosti, razlika u vrijednosti iznosi gotovo 10% što znači da klasifikator puno gore pronalazi točne NONNEGATIVE uzorke u klasi. Također, vrijednost odaziva je kod NEGATIVE klase čak 24% manja od NONNEGATIVE što balans rješenja čini iznimno nepouzdanim.

Spomenutu pojavu možemo pripisati prirodi rada algoritma slučajnih šuma. S obzirom da naslov članaka sadrži manji broj riječi, algoritam ima manje uzoraka za učenje. Zbog toga dolazimo do situacije u kojoj model od početka krivo interpretira NEGATIVE uzorke zbog čega se grananje nastavlja u krivom smjeru.

```
Title:

Random Forest (Resampled):
==================================

Train target values count:  Counter({'NONNEGATIVE': 1400, 'NEGATIVE': 1400})
Test target values count:  Counter({'NONNEGATIVE': 600, 'NEGATIVE': 600})
Result values count:  Counter({'NONNEGATIVE': 684, 'NEGATIVE': 516})

              precision    recall  f1-score   support

    NEGATIVE       0.89      0.77      0.82       600
 NONNEGATIVE       0.79      0.91      0.85       600

    accuracy                           0.83      1200
   macro avg       0.84      0.83      0.83      1200
weighted avg       0.84      0.83      0.83      1200

Accuracy Score:  83.5
Accuracy(K-Fold): 82.393 (0.014)
```

Slika 28: Rezultati klasifikatora slučajnih šuma nad naslovom članka

Treniranjem modela na tekstu članka ukupna točnost iznosi 89,17% što je djelomično potvrđeno unakrsnom validacijom kojom dobivamo nešto slabiji rezultat od 85%. Za razliku od klasifikacije naslova, u ovom slučaju ne nailazimo na velike oscilacije između metrika točnosti NONNEGATIVE i NEGATIVE klasa. Pritom navedena točnost iznosi skoro pa 90% na gore uz maksimalnu razliku od 5%.

Ovaj model ujedno daje najbolje rezultate od svih dostupnih modela. Uspoređujući rezultate s modelom treniranom na naslovima, možemo zaključiti da veći broj ulaznih riječi kod treniranja algoritmom slučajnih šuma znatno pridonosi kvaliteti klasifikacije. Također, ovaj model pruža najbolju F1 mjeru od 89%

```
Article text:

Random Forest (Resampled):
==================================

Train target values count:  Counter({'NEGATIVE': 1400, 'NONNEGATIVE': 1400})
Test target values count:  Counter({'NEGATIVE': 600, 'NONNEGATIVE': 600})
Result values count:  Counter({'NONNEGATIVE': 628, 'NEGATIVE': 572})

              precision    recall  f1-score   support

    NEGATIVE       0.91      0.87      0.89       600
 NONNEGATIVE       0.87      0.92      0.89       600

    accuracy                           0.89      1200
   macro avg       0.89      0.89      0.89      1200
weighted avg       0.89      0.89      0.89      1200

Accuracy Score:  89.16666666666667
Accuracy(K-Fold): 85.000 (0.012)
```

Slika 29: Rezultati klasifikatora slučajnih šuma nad tekstom članka

7. Usporedba rezultata

Poglavlje u nastavku opisuje rezultate algoritama za analizu sentimenta te interpretaciju podataka u kontekstu pandemije koronavirusa. Prikaz rezultata je pritom odvojen na onaj koji se odnosi na rezultate dobivene analizom tekstualnog elementa naslova i cijelog teksta članka, respektivno.

Tablica 10 u nastavku sadrži usporedni pregled učinkovitosti algoritama nenadziranog i nadziranog učenja nad **naslovom** članaka. U slučaju nenadziranog učenja, VADER algoritam u smislu ukupne točnosti dominira nad jednostavnim algoritmima vlastite izrade s većim postotkom točnosti i do 8 do 12% te ona iznosi 62%. Što se tiče F1 mjere koja služi kao glavna mjera u istraživanju, rezultat VADER algoritma je 2% bolji od algoritma koji analizira tekstualni element kao cjelinu. Malenu razliku između točnosti možemo pripisati činjenici da VADER koristi strojno preveden tekst, a kvaliteta prevođenja pritom nije verificirana.

Algoritmi koji koriste nadzirani pristup putem algoritama umjetne inteligencije pružaju i do 20% bolje rezultate. Najveća točnost je ostvarena korištenjem algoritma slučajnih šuma, te ona iznosi 84%, što ju čini 2% većom od algoritama Naivnog Bayesa i SVM-a. Prosječne vrijednosti mjera preciznosti i odaziva su 1 do 2% manje u slučaju tih algoritama, ali ukupna F1 mjera je za njih lošija 1 do 7%. Iako se radi o generičkim algoritmima koji koriste TF-IDF metodu uz minimalnu manipulaciju parametrima, umjetnom inteligencijom su dobiveni zadovoljavajući rezultati.

	Accuracy	**Precision**	**Recall**	**F1-Score**
Nenadzirani pristup				
Sentiment po cijelom tekstu	0,53	0,53	0,53	0,53
Sentiment po prosjeku rečenice	0,48	0,48	0,48	0,47
VADER	0,62	0,62	0,63	0,55
Nadzirani pristup				
Naive Bayes	0,82	0,82	0,82	0,76
SVM	0,82	0,82	0,82	0,82
Random Forest	0,84	0,84	0,83	0,83

Tablica 10: Usporedba svih algoritama korištenih na naslovu članka

Tablica *11* pruža raznovrsnije rezultate, što možemo pripisati i samo prirodi rada algoritama. U slučaju nenadziranog pristupa VADER algoritam ponovno generira najbolje rezultate, a pritom točnost iznosi 57%, isto kao i F1 mjera. Bitno je napomenuti da su svi rezultati nenadziranog pristupa nad tekstom članka lošiji od onih nad naslovom. Iz toga možemo zaključiti da nenadzirani pristup ne radi u skladu s očekivanja za veće količine teksta.

Nadzirani pristup u slučaju analize nad cijelim tekstom generira rezultat koja je i do 30% veća u slučaju algoritma slučajnih šuma u odnosu na VADER-a, te ona iznosi čak 89%. Ostale metrike (Preciznost, odaziv i F1) također iznose 89%, dok je kod SVM algoritma točnost nešto niža, pri čemu sve mjere iznose 84%. U slučaju algoritma Naivnog Bayesa, F1 mjera ima povećanu točnost od samo 1%, dok je rezultat mjera točnosti, preciznost i odaziva gori nego u slučajevima u kojima obrađuje samo naslov. Razlog toj pojavi može biti da veća količina teksta negativno utječe na Bayesov klasifikator i ukupni rezultat.

	Accuracy	**Precision**	**Recall**	**F1-Score**
Nenadzirani pristup				
Sentiment po cijelom tekstu	0,52	0,54	0,53	0,53
Sentiment po prosjeku rečenice	0,42	0,47	0,47	0,43
VADER	0,57	0,63	0,63	0,57
Nadzirani pristup				
Naive Bayes	0,77	0,77	0,77	0,77
SVM	0,84	0,84	0,84	0,84
Random Forest	0,89	0,89	0,89	0,89

Tablica 11: Usporedba rezultata svih algoritama korištenih na tekstu članka

Iz analize podataka prethodnih tablica možemo zaključiti da algoritmi umjetne inteligencije imaju bolje performanse od nenadziranih primjera. Najbolji rezultat je postignut algoritmom slučajnih šuma u oba slučaja, a vidljivo je da veća količina teksta nad kojom model uči osigurava veći postotak uspješnosti klasifikacije. U razmatranje je potrebno uzeti i brzinu izvođenja algoritma slučajnih šuma kod velikih skupova podataka, što u našem slučaju iznosi nešto više od 9 sekundi. Zbog toga nije nužno da njegovo korištenje može biti smatrano kao ono opće namjene.

Ovisno o dostupnim računalnim resursima, za male količine teksta, istraživanje se može provesti i algoritmom Naivnog Bayesa zbog izričito kratkog vremena izvođenja koje iznosi 0,29 sekundi za cijeli skup podataka. S druge strane, SVM koji pruža solidne performanse za svoj rad zahtijeva 90 sekundi u našem slučaju što ga čini nepovoljnim za ovakav tip istraživanja. Kod algoritama nenadziranog učenja se suočavamo s individualnim slučajevima na svakoj pojedinoj analizi što u konačnici ishodi zahtjevnijoj pripremi i obradi podataka. Proces pripreme i obrade podataka uključuje svođenje riječi na osnovni oblik te veće programske kodove što rezultira nepredviđenim greškama koje dodatno oduzimaju vrijeme istraživanja. Također, količina vremena potrebnog za provođenje nenadzirane analize nerijetko traje i po nekoliko sati po pokretanju programske skripte.

8. Zaključak

U ovom radu je demonstriran proces automatiziranog prikupljanja, obrade i analize podataka novinskog portala Dalmacija Danas te je dan pregled i usporedba rezultata algoritama koji omogućuju analizu sentimenta. Istraživanje je provedeno tijekom godine dana, a veći dio vremena je utrošen na pripremu podataka koji se odnose na sve članke koji su objavljeni u kroz sedamnaest mjeseci. Temeljni cilj rada je bio povezati aktualnu situaciju pandemije bolesti COVID-19 s modernim načelima i metodama za manipulaciju i analizu podataka u domeni znanosti o podacima.

Prvi dio rada prikazuje teorijsku podlogu kojom su prikazana osnovna načela i pristupi koji su korišteni za analizu sentimenta, odnosno rudarenje mišljenja. Na taj način su prikazane vrste analize koje se odnose na stariji nenadzirani pristup te nadzirani pristup koji se je kroz napredak strojnog učenja pokazao bržim, efikasnijim te jednostavnijim za implementaciju. Sljedeći dio se je odnosio na proces prikupljanja podataka automatiziranim putem pomoću metode struganja podataka. Cilj ovog dijela zadatka je bio identificirati relevantne podatke u nestrukturiranom obliku članaka te pomoću programske podrške kreirati „sirovi“ skup podataka pogodan za analizu. Ovaj dio se je pokazao kao jednim od najzahtjevnijih upravo zbog individualnog pristupa svakom problemu prikupljanja tih podataka, a usred istraživanja je web stranica mijenjala dizajn zbog čega je proces morao biti ponovljen više puta. U fazi obrade podataka su identificirani članci relevantni za temu pandemije, a tekstualni elementi naslova i teksta članka su prilagođeni za analizu sentimenta nenadziranim i nadziranim pristupom. Neke od tih prilagodbi uključuju svođenje riječi na osnovni oblik, izbacivanje zaustavnih riječi i strojno prevođenje na engleski jezik. Korištenjem tih tehnika je prikazan osnovni postupak za čišćenje podataka, a pomoću njih je kreirano više skupova podataka od kojih svaki služi za različitu vrstu analize. Neke od otežavajućih okolnosti ovih metoda su vrijeme izvođenja, kao i problem vizualne kontrole kvalitete pretvorbe i prijevoda novonastalog skupa.

Zadnji dio rada se odnosi na analizu kreiranog i prilagođenog skupa podataka. U prvom dijelu je korištena deskriptivna statistika za utvrđivanje ukupnog broja pozitivnih/negativnih članaka pomoću *emojija*, kategorija članaka, *itd*. Tim putem je utvrđeno da veći dio članaka, njih dvije trećine koristi negativno iskazivanje mišljenja. Kod analize nenadziranim pristupom su

dobiveni znatno lošiji rezultati analize od onih koji su dobiveni nadziranim pristupom strojnog učenja. Od algoritama nenadziranog pristupa alat VADER postiže najbolje rezultate s 57% točnosti nad cijelim tekstom članka. Treba napomenuti da su ti rezultati puno lošiji od svih algoritama nadziranog učenja od kojih je algoritam slučajnih šuma postigao točnost iskazanu putem F1 mjere u iznosu od gotovo 89%. Takva točnost osigurana ovim modelom zasigurno može biti korištena u realnim zadacima analize sentimenta. ako su rezultati nenadziranog pristupa daleko lošiji u vidu mjera točnosti klasifikacije, pomoću njih su ipak dobivene neke od korisnih spoznaja. Na primjer, grafičkim prikazom koeficijenta pozitivnosti kod nenadziranog učenja možemo uočiti kronološku oscilaciju krivulje pozitivnosti. Na taj način možemo utvrditi u kojim razdobljima je počeo koji val pandemije.

Potrebno je napomenuti da ovaj zadatak služi kao primjer prikupljanja, obrade i analize podataka u što kraćem roku uz relativno jednostavne algoritme i načine za rješavanje problema. Iako je sama izvedba i dizajn tih algoritama bio izričito jednostavan proces, rezultati su bolji od očekivanih. Algoritmi koji su korišteni kod nenadziranog učenja mogu biti modificirani za veći broj lingvističkih pravila čime bi se njihova točnost mogla povećati. Također, sve metode se mogu kombinirati u svrhu dobivanja dubljeg uvida u podatke. Osim toga, jedno od potencijalnih proširenja analize je korištenje neuronskih mreža i metoda dubokog učenja, primjerice bibliotekom Transformers [40] specijaliziranom za lingvističke analize. Na taj način bi dobili kompletni usporedni pregled rada automatiziranih algoritama.

Postojeća srodna istraživanja u ovom području su vezana za poruke koje se šire na društvenim mrežama i pisane su na hrvatskome jeziku, a odnose se na COVID-19 tematiku, bave se analizom sentimenta poruka [41] i njihovim širenjem u društvenoj mreži [42]. Osim toga, provedena su i istraživanja koja analiziraju vijesti objavljene na mrežnim novinskim portalima [43, 44] u kontekstu korištene terminologije COVID-19 vokabulara, količine COVID-19 objava i modeliranja tema u objavama koje su vezane za COVID-19 tematiku. Analiza sentimenta novinskih objava vezanih za tematiku COVID-19 istraživanja su u [46].

Analiza sentimenta ovim pristupima je relativno nova pojava, a mogućnosti primjene su u znanstvenim i komercijalnim granama. Činjenica da je za dobivanje izričito dobrih rezultata potrebna minimalna ljudska intervencija korištenjem jeftinog hardvera predstavlja revolucionarni pomak u antropološkim, statističkim i ostalim istraživanjima.

9. Popis literature

[1] „Coronavirus“. https://www.who.int/health-topics/coronavirus#tab=tab_1 (pristupljeno ruj. 05, 2021).

[2] „COVID-19 pandemic“, *Wikipedia*. ruj. 05, 2021. Pristupljeno: ruj. 05, 2021. [Na internetu]. Dostupno na: https://en.wikipedia.org/w/index.php?title=COVID-19_pandemic&oldid=1042447454

[3] O.-O. D. Science, „Understanding Unstructured Data With Language Models“, *Medium*, svi. 21, 2019. https://medium.com/@ODSC/understanding-unstructured-data-with-language-models-26c8c6c46bde (pristupljeno ruj. 05, 2021).

[4] O. Castrillo, „Web Scraping: Applications and Tools“, izd. 2015, str. 31, 2015.

[5] B. Liu, „Sentiment Analysis and Opinion Mining“, str. 168.

[6] T. Luo, S. Chen, G. Xu, i J. Zhou, „Sentiment Analysis“, u *Trust-based Collective View Prediction*, New York, NY: Springer New York, 2013, str. 53–68. doi: 10.1007/978-1-4614-7202-5_4.

[7] Department of Computer Science, Allama Iqbal Open University, Islamabad, Pakistan, A. Soofi, i A. Awan, „Classification Techniques in Machine Learning: Applications and Issues“, *J. Basic Appl. Sci.*, sv. 13, str. 459–465, kol. 2017, doi: 10.6000/1927-5129.2017.13.76.

[8] M. V. Mäntylä, D. Graziotin, i M. Kuutila, „The evolution of sentiment analysis—A review of research topics, venues, and top cited papers“, *Computer Science Review*, sv. 27, str. 16–32, velj. 2018, doi: 10.1016/j.cosrev.2017.10.002.

[9] K. Verspoor i K. B. Cohen, „Natural Language Processing“, u *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, K.-H. Cho, i H. Yokota, Ur. New York, NY: Springer New York, 2013, str. 1495–1498. doi: 10.1007/978-1-4419-9863-7_158.

[10] A. Katrekar, „An Introduction to Sentiment Analysis“, *Big Data Analytics*, str. 7.

[11] C. Musto, G. Semeraro, i M. Polignano, „A comparison of Lexicon-based approaches for Sentiment Analysis of microblog posts“, str. 10.

[12] B. Pang i L. Lee, „Opinion mining and sentiment analysis“, str. 94.

[13] M. Boia, B. Faltings, C.-C. Musat, i P. Pu, „Sentiment Analysis Based on Dictionary Approach“, u *2013 International Conference on Social Computing*, Alexandria, VA, USA, ruj. 2013, str. 345–350. doi: 10.1109/SocialCom.2013.54.

[14] M. Darwich, S. A. Mohd Noah, N. Omar, i N. A. Osman, „Corpus-Based Techniques for Sentiment Lexicon Generation: A Review“, *Journal of Digital Information Management*, sv. 17, izd. 5, str. 296, lis. 2019, doi: 10.6025/jdim/2019/17/5/296-305.

[15] M. Schott, „Naives Bayes Classifiers for Machine Learning“, *Medium*, velj. 27, 2020. https://medium.com/capital-one-tech/naives-bayes-classifiers-for-machine-learning-2e548bfbd4a1 (pristupljeno lip. 14, 2021).

[16] S. Yıldırım, „Naive Bayes Classifier — Explained“, *Medium*, svi. 12, 2020. https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed (pristupljeno kol. 28, 2021).

[17] R. Pupale, „Support Vector Machines(SVM) — An Overview“, *Medium*, velj. 11, 2019. https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 (pristupljeno lip. 14, 2021).

[18] S. Patel, „Chapter 2 : SVM (Support Vector Machine) — Theory“, *Machine Learning 101*, svi. 04, 2017. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 (pristupljeno kol. 28, 2021).

[19] Simplilearn, *Random Forest Algorithm - Random Forest Explained | Random Forest in Machine Learning | Simplilearn*, (ožu. 12, 2018). Pristupljeno: kol. 29, 2021. [Na internetu Video]. Dostupno na: https://www.youtube.com/watch?v=eM4uJ6XGnSM

[20] „DALMACIJA DANAS - obala, otoci, Zagora. Najnovije vijesti iz Dalmacije.“ https://www.dalmacijadanas.hr/ (pristupljeno lip. 23, 2021).

[21] „gemiusRating“. https://rating.gemius.com/hr/tree/8 (pristupljeno lip. 12, 2021).

[22] „Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation“. https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (pristupljeno lip. 15, 2021).

[23] „SeleniumHQ Browser Automation“. https://www.selenium.dev/ (pristupljeno lip. 15, 2021).

[24] „Pojmovnik - Hrvatski mrežni rječnik“. http://ihjj.hr/mreznik/page/pojmovnik/6/ (pristupljeno lip. 18, 2021).

[25] CLARIN.SI, *classla: Adapted Stanford NLP Python Library with improvements for specific languages.* Pristupljeno: lip. 18, 2021. [Na internetu]. Dostupno na: https://github.com/clarinsi/classla-stanfordnlp.git

[26] „What is the difference between categorical, ordinal and interval variables?“ https://stats.idre.ucla.edu/other/mult-pkg/whatstat/what-is-the-difference-between-categorical-ordinal-and-interval-variables/ (pristupljeno lip. 22, 2021).

[27] „Confusion matrix“, *Wikipedia*. svi. 13, 2021. Pristupljeno: lip. 22, 2021. [Na internetu]. Dostupno na: https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=1023000804

[28] „Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures“, *Exsilio Blog*, ruj. 09, 2016. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/ (pristupljeno lip. 22, 2021).

[29] „sklearn.metrics.precision_score — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html (pristupljeno lip. 22, 2021).

[30] „sklearn.metrics.recall_score — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html (pristupljeno lip. 23, 2021).

[31] „sklearn.metrics.f1_score — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (pristupljeno lip. 23, 2021).

[32] „Croatian Sentiment Lexicon – META-SHARE“. http://meta-share.ffzg.hr/repository/browse/croatian-sentiment-lexicon/940fe19e6c6d11e28a985ef2e4e6c59eff8b12d75f284d58aacfa8d732467509/ (pristupljeno kol. 16, 2021).

[33] C. J. Hutto, *cjhutto/vaderSentiment*. 2021. Pristupljeno: kol. 20, 2021. [Na internetu]. Dostupno na: https://github.com/cjhutto/vaderSentiment

[34] „Project Jupyter“. https://www.jupyter.org (pristupljeno lip. 23, 2021).

[35] „scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/ (pristupljeno lip. 23, 2021).

[36] „sklearn.utils.resample — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html (pristupljeno lip. 23, 2021).

[37] S. Qaiser i R. Ali, „Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents“, *IJCA*, sv. 181, izd. 1, str. 25–29, srp. 2018, doi: 10.5120/ijca2018917395.

[38] „sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 0.24.2 documentation“. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (pristupljeno lip. 23, 2021).

[39] D. Berrar, „Cross-Validation“, u *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019, str. 542–545. doi: 10.1016/B978-0-12-809633-8.20349-X.

[40] „Transformers“. https://huggingface.co/transformers/index.html (pristupljeno ruj. 04, 2021).

[41] Babić K., Petrović M., Beliga S., Martinčić-Ipšić S., Jarynowski A., Meštrović A. (2022) COVID-19-Related Communication on Twitter: Analysis of the Croatian and Polish Attitudes. In: Yang XS., Sherratt S., Dey N., Joshi A. (eds) Proceedings of Sixth International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems, vol 216. Springer, Singapore. https://doi.org/10.1007/978-981-16-1781-2_35

[42] K. Babić, M. Petrović, S. Beliga, S. Martinšić-Ipšić, M. Pranjić and A.Meštrović. 2021. Prediction of COVID-19 related information spreading on Twitter. In Proceedings of the IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2021), accepted for publication.

[43] S. Beliga, S. Martinšić-Ipšić, M. Matešić and A. Meštrović. 2022. Natural language processing and statistics: The first six months of the COVID-19 infodemic in Croatia, In The Covid-19 Pandemic as a Challenge for Media and Communication Studies. Routledge, Taylor & Francis Group, Edited By Katarzyna Kopecka-Piech, Bartłomiej Łódzki, accepted for publication.

[44] P. K. Bogović, S. Beliga , A. Meštrović and S. Martinčić-Ipšić. 2021. Topic modelling of Croatian news during COVID-19 pandemic. In Proceedings of the IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2021), accepted for publication.

[45] A. Ilić, S. Beliga.The Polarity of Croatian Online News Related to COVID-19: a FistInsight. In Central European Conference on Information and Intelligent Systems. Faculty of Organization and Informatics, Varaždin, 2021.

[46] M. Buhin Pandur, J. Dobša, S. Beliga, A. Meštrović. Topic modelling and sentiment analysis of COVID-19 related news on Croatian Internet portal. In SiKDD 2021, Ljubljana, Slovenia.

Popis slika

Slika 1: Podjela algoritama nenadziranog i nadziranog učenja. Preuzeto: https://www.researchgate.net/figure/Sentiment-Analysis-techniques_fig1_343712784 5
Slika 2: Vizualni prikaz metode potpornih vektora (preuzeto: https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323) 9
Slika 3: Grafički prikaz rada RF algoritma. Preuzeto: https://www.tibco.com/reference-center/what-is-a-random-forest 10
Slika 4: Naslovnica portala Dalmacija Danas (Preuzeto: https://www.dalmacijadanas.hr/) 12
Slika 5: Primjer članka na portalu Dalmacija Danas (Preuzeto: https://www.dalmacijadanas.hr/video-skakljanja-raze-postao-svjetski-hit-strucnjak-od-uboda-raze-mozete-umrijeti/) 13
Slika 6: HTML elementi s poveznicama na kategoriju (Preuzeto: https://www.dalmacijadanas.hr/) 15
Slika 7: Primjer kartica s vijestima u kategoriji – HTML (Preuzeto: https://www.dalmacijadanas.hr/rubrika/dalmacija/) 16
Slika 8: Izgled kartica s vijestima u kategoriji Dalmacija (Preuzeto: https://www.dalmacijadanas.hr/rubrika/dalmacija/) 16
Slika 9: Skripta za dohvaćanje i spremanje poveznice članka 17
Slika 10: Uzorak prikupljenih poveznica na članke portala 17
Slika 11: Skripta article_filter.py za izoliranje COVID-19 članaka 21
Slika 12: Rezultati svođenja riječi na osnovni oblik 23
Slika 13: Primjer reakcija putem *emojija* na članku portala (Preuzeto: https://www.dalmacijadanas.hr/direktor-prometa-priznao-nas-vozac-nije-smio-uzeti-taj-autobus-jer-je-bio-neispravan/) 23
Slika 14: Graf distribucija reakcija putem *emojija* u člancima 30
Slika 15: Rezultati algoritma provedenih nad cijelim naslovom članka 35
Slika 16: Rezultati algoritma provedenih nad cijelim tekstom članka 35
Slika 17: Rezultati algoritma za računanje sentimenta po rečenici nad naslovom članka 37
Slika 18: Rezultati algoritma za računanje sentimenta po rečenici nad tekstom članka 38

Slika 19: Rezultati VADER algoritma po rečenici nad naslovom članka 39
Slika 20: Rezultati VADER algoritma nad tekstom članka 40
Slika 21: Grafički prikaz prosječnih dnevnih vrijednosti koeficijenata nenadziranog učenja 42
Slika 22: Graf nebalansirane distribucije klasa 44
Slika 23: Graf balansirane distribucije klasa 44
Slika 24: Rezultati klasifikatora Naive Bayes nad naslovom članka 46
Slika 25: Rezultati klasifikatora Naive Bayes nad tekstom članka 47
Slika 26: Rezultati SVM klasifikatora nad naslovom članka 48
Slika 27: Rezultati SVM klasifikatora nad tekstom članka 49
Slika 28: Rezultati klasifikatora slučajnih šuma nad naslovom članka 50
Slika 29: Rezultati klasifikatora slučajnih šuma nad tekstom članka 51

10.Popis tablica

Tablica 1: Stupci datoteke portal_articles.csv 19
Tablica 2: Kvantifikacija prikupljenih članaka 21
Tablica 3: Koeficijenti pozitivnosti emocije *emojija* 24
Tablica 4: Zastupljenost klasa po pozitivnosti 25
Tablica 5: Matrica konfuzije 26
Tablica 6: Distribucija članaka po kategorijama 29
Tablica 7: Distribucija reakcija putem *emojija* u člancima 30
Tablica 8: Suma reakcija korisnika putem *emojija* po kategorijama 31
Tablica 9: Distribucija *emojija* prema klasama 32
Tablica 10: Usporedba svih algoritama korištenih na naslovu članka 52
Tablica 11: Usporedba rezultata svih algoritama korištenih na tekstu članka 53

11.Popis priloga

Medij za pohranu podataka (CD) koji sadrži projektni dio zadatka, rezultate, skupove podataka i pripadnu dokumentaciju.

GitHub poveznica za preuzimanje projekta: https://github.com/ailic96/Masters_Thesis

Struktura projekta je vizualizirana u nastavku:

```
Diplomski_Rad
|   .gitignore
|   geckodriver.log
|   output.doc
|   README.md
|
+---.vscode
|   |   settings.json
|   |
|   \---.ropeproject
|           config.py
|           objectdb
|
+---data
|       portal_articles.csv
|       portal_articles_classes.csv
|       portal_articles_covid.csv
|       portal_articles_covid_binary.csv
|       portal_articles_covid_clear.csv
|       portal_articles_covid_clear_sentences.csv
|       portal_articles_covid_positivity_extended.csv
|       portal_articles_covid_positivity_transformers.csv
|       portal_articles_covid_positivity_vader.csv
|       portal_articles_covid_sentences_lemmatized.csv
|       portal_articles_final.csv
|       portal_articles_final_ml.csv
|       portal_article_logger.txt
|       portal_log.txt
|       portal_urls.txt
|
+---data_backup
|   |   portal_articles.csv
|   |   portal_articles_covid.csv
|   |   portal_articles_covid_binary.csv
|   |   portal_articles_covid_clear.csv
|   |   portal_articles_covid_positivity.csv
|   |   portal_articles_covid_positivity_extended copy.csv
|   |   portal_articles_covid_positivity_extended.csv
|   |   portal_articles_covid_positivity_extended_without_zeros.csv
|   |   portal_articles_covid_positivity_extended_zero_mean.csv
|   |   portal_articles_covid_positivity_vader.csv
|   |   portal_articles_covid_positivity_vader_article_text.csv
|   |   portal_articles_covid_positivity_vader_tit_sub.csv
|   |   portal_articles_covid_sentences_lemmatized copy.csv
|   |   portal_urls.txt
|   |
|   \---word_lists_backup
|           covid_dictionary.txt
|           crosentilex-negatives.txt
|           crosentilex-negatives_lemmatized.txt
|           crosentilex-positives.txt
|           crosentilex-positives_lemmatized.txt
|           stop_words.txt
```

```
|
+---imgs
|       descriptions.png
|       distribution_original.PNG
|       distribution_resampled.PNG
|
+---src
|   |   __init__.py
|   |
|   +---analysis
|   |   |   article_add_classes.py
|   |   |   article_positivity.py
|   |   |   article_positivity_vader.py
|   |   |   article_quantification.py
|   |   |   common.py
|   |   |   graphing.ipynb
|   |   |   sentiment_doc2vec.ipynb
|   |   |   sentiment_ml.ipynb
|   |   |   __init__.py
|   |   |
|   |   \---__pycache__
|   |           common.cpython-36.pyc
|   |           common.cpython-38.pyc
|   |
|   +---processing
|   |   |   article_filter.py
|   |   |   common.py
|   |   |   language_cleaning.py
|   |   |   lemmatization_article.py
|   |   |   lemmatization_words_list.py
|   |   |   __init__.py
|   |   |
|   |   \---__pycache__
|   |           common.cpython-38.pyc
|   |
|   \---scraping
|       |   article_scraper.py
|       |   article_url_scraper.py
|       |   __init__.py
|       |
|       \---__pycache__
|               common.cpython-38.pyc
|               functions.cpython-38.pyc
|
+---tables
|       df_positivity_text_negative.html
|       df_positivity_text_true.html
|       df_positivity_title_negative.html
|       df_positivity_title_true.html
|       df_sentence_positivity_text_negative.html
|       df_sentence_positivity_text_positive.html
|       df_sentence_positivity_title_negative.html
|       df_sentence_positivity_title_positive.html
|       df_vader_positivity_text_negative.html
|       df_vader_positivity_text_positive.html
|       df_vader_positivity_title_negative.html
|       df_vader_positivity_title_positive.html
|       mean_emoji_value_by_category.csv
|       mean_emoji_value_by_category.html
|       portal_articles_category.csv
|       total_emoji_sum_by_category.csv
|       total_emoji_sum_by_category.html
|
\---word_lists
        covid_dictionary.txt
        covid_dictionary_2.txt
        crosentilex-negatives.txt
        crosentilex-negatives_lemmatized.txt
        crosentilex-positives.txt
        crosentilex-positives_lemmatized.txt
        stop_words.txt
```