VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE

Veštacka inteligencija

Seminarski rad

Predikcija sentimenta u tekstu korišćenjem nadgledanog učenja: analiza recenzija filmova

Predmetni nastavnik:

Dr. Aleksandar Simović

Studenti:

Sara Jovanović 262-24

Viktor Unković 261-24

Sadržaj

Rezime....3
Ključne reči 3
1. Uvod u analizu sentimenta 4
1.1 Definicija analize sentimenta 4
1.2 Primeri upotrebe analize sentimenta 4
1.3 Zašto je analiza sentimenta važna 4
2. Nadgledano učenje i algoritmi za klasifikaciju 5
2.1 Nadgledano učenje 5
2.2 Algoritmi za klasifikaciju 5
2.3 Osnovne faze obučavanja modela 6
3. Tehnike obrade prirodnog jezika (NLP) 7
3.1 Tehnike obrade teksta 7
3.1.1 Tokenizacija 7
3.1.2 Uklanjanje stop-reči 7
3.1.3 TF-IDF (Term Frequency - Inverse Document Frequency) 7
3.2 Prednosti i izazovi NLP-a u analizi sentimenta 8
4. Primena analize sentimenta u industriji filmova 9
4.1 Analiza recenzija filmova 9
4.2 Primene analize sentimenta u industriji filmova 9
4.2.1 Predviđanje uspeha filma 9
4.2.2 Prepoznavanje problema u filmu 10
4.2.3 Poboljšanje angažovanja publike 10
5. Izazovi u analizi sentimenta 11
5.1 Kontekstualni problemi 11
5.2 Polisemija 11
5.3 Povezanost reči i struktura rečenica 12
6. Primena teme kroz zadatak 13
7. Zaključak 26
Literatura 27

Rezime

Tema "Predikcija sentimenta u tekstu korišćenjem nadgledanog učenja: analiza recenzija filmova" istražuje korišćenje tehnika nadgledanog učenja za klasifikaciju sentimenta u tekstualnim podacima, konkretno u recenzijama filmova. Sentiment analiza je proces identifikovanja i klasifikovanja emocionalnog tona u tekstu, kao što su pozitivne i negativne recenzije, što je ključna komponenta u mnogim aplikacijama poput analize mišljenja, sistema za preporuke i praćenja tržišnog mišljenja.

U okviru nadgledanog učenja, model se trenira na označenim podacima koji sadrže recenzije filmova i njihove odgovarajuće sentiment oznake. TF-IDF (Term Frequency-Inverse Document Frequency) metoda se često koristi za vektorizaciju teksta, pretvarajući reči u numeričke vrednosti koje model može koristiti. Zatim, različiti modeli mašinskog učenja, poput logističke regresije, SVM (Support Vector Machines) ili neuronskih mreža, se primenjuju kako bi predvideli sentiment na temelju naučenih obrazaca u podacima.

Prednosti ove tehnike uključuju mogućnost obrade velikih količina podataka i automatsku analizu sentimenta bez potrebe za ručnim pregledom recenzija. Takođe, predikcija sentimenta može se koristiti za poboljšanje sistema preporuka, marketinga i analize korisničkog mišljenja.

Ovaj pristup pruža okvir za razvoj modela koji prepoznaje i klasifikuje emocije u tekstu, što ima široku primenu u industrijama kao što su zabava, e-trgovina, političke analize i društvene mreže.

Ključne reči

Predikcija sentimenta, Nadgledano učenje, Sentiment analiza, Klasifikacija teksta, TF-IDF, Mašinsko učenje, Logistička regresija, Recenzije filmova

1. Uvod u analizu sentimenta

Analiza sentimenta, takođe poznata kao analiza mišljenja, predstavlja važnu oblast obrade prirodnog jezika (NLP) koja se bavi identifikovanjem i klasifikovanjem emocija i stavova izraženih u tekstu. S obzirom na to da digitalni svet svakodnevno generiše ogromnu količinu tekstualnih podataka, proces prepoznavanja sentimenta u tim podacima postao je ključan za razumevanje mišljenja i emocija ljudi. Osnovna svrha analize sentimenta je da automatski odredi da li je stav izražen u tekstu pozitivan, negativan ili neutralan, što je korisno u različitim industrijama, kao što su marketing, politika, film i mnoge druge.

1.1 Definicija analize sentimenta

Analiza sentimenta se može definisati kao proces primene algoritama mašinskog učenja i tehnika obrade prirodnog jezika na tekstualne podatke kako bi se identifikovale emocionalne reakcije ili stavovi autora prema određenoj temi. Algoritmi za analizu sentimenta koriste lingvističke i statističke pristupe kako bi odredili ton i emocije koje se nalaze u tekstu. Na osnovu ovog procesa, sentiment se obično klasifikuje u tri osnovne kategorije: pozitivan, negativan i neutralan. Osim toga, analiza sentimenta može obuhvatiti složenije pristupe kao što su prepoznavanje jačine sentimenta (npr. "veoma pozitivan" ili "malo negativan") ili prepoznavanje specifičnih emocija, kao što su bes, tuga ili radost.

1.2 Primeri upotrebe analize sentimenta

Analiza sentimenta našla je široku primenu u različitim domenima. Jedan od najpoznatijih primera je u marketingu, gde kompanije koriste ovu tehnologiju za praćenje mišljenja svojih kupaca. Analizom recenzija proizvoda i komentara na društvenim mrežama, organizacije mogu uvideti stavove korisnika, što im pomaže u poboljšanju proizvoda, usluga ili marketinških kampanja. Takođe, u političkom kontekstu, analiza sentimenta omogućava političkim analitičarima da prate reakcije javnosti na određene izjave, odluke ili događaje, što im pomaže da bolje razumeju trenutne tendencije i prilagode svoje strategije.

U filmskoj industriji, analiza sentimenta takođe ima značajnu ulogu. Analizom recenzija filmova, filmske kompanije mogu shvatiti kako je određeni film prihvaćen od strane publike, što može imati značajan uticaj na marketing i distribuciju filma. Ako su recenzije uglavnom negativne, to može signalizirati potrebu za promenama u promociji, dok pozitivne recenzije mogu doprineti većem interesovanju za film. Korišćenje recenzija filmova kao podataka za analizu sentimenta omogućava bolje razumevanje reakcija gledalaca, što može doprineti uspehu ili neuspehu filma na tržištu.

1.3 Zašto je analiza sentimenta važna

Značaj analize sentimenta ogleda se u njenoj sposobnosti da automatski i efikasno obrađuje velike količine tekstualnih podataka. S obzirom na to da u današnjem digitalnom dobu postoji ogromna količina recenzija, postova i komentara koji se generišu u realnom vremenu, ručna analiza takvih podataka bila bi vremenski zahtevna i nepraktična. Automatski procesi omogućavaju da se brzo i precizno klasifikuju sentimenti u različitim vrstama tekstova, što je od velikog značaja za organizacije koje žele da reaguju na mišljenja korisnika u realnom vremenu.

Analiza sentimenta takođe pomaže u razumevanju dubljih emocija i stavova korisnika, što može biti korisno za prilagođavanje proizvoda, usluga ili kampanja. U industrijama kao što su film, e-trgovina, i politika, uvidi koje pruža analiza sentimenta mogu imati direktan uticaj na strategije i odluke. Na primer, filmske industrije mogu koristiti analizu sentimenta za pravovremeno prepoznavanje negativnih reakcija na filmove i prilagođavanje marketinških strategija kako bi poboljšali performanse filmova na tržištu.

2. Nadgledano učenje i algoritmi za klasifikaciju

Nadgledano učenje predstavlja jedan od ključnih pravaca mašinskog učenja koji se koristi za rešavanje problema klasifikacije, uključujući analizu sentimenta u tekstu. Ovaj pristup omogućava kreiranje modela koji može da prepozna specifične obrasce u podacima i na osnovu tih obrazaca da izvrši klasifikaciju novih, nepoznatih podataka. U kontekstu analize sentimenta, nadgledano učenje se koristi kako bi model naučio da klasifikuje tekstove prema tome da li oni izražavaju pozitivan, negativan ili neutralan sentiment.

2.1 Nadgledano učenje

Nadgledano učenje (supervised learning) je podvrsta mašinskog učenja koja podrazumeva treniranje modela na označenim podacima. Ovi podaci se sastoje od ulaznih vrednosti (koje mogu biti bilo koji podaci, kao što su tekstovi, slike ili vremenski podaci) i odgovarajućih izlaznih vrednosti (tj. oznaka ili klasa). U slučaju analize sentimenta, izlazne vrednosti mogu biti kategorije kao što su "pozitivan", "negativan" ili "neutralan", dok su ulazne vrednosti tekstualni podaci kao što su recenzije filmova, komentari ili tweetovi.

Proces nadgledanog učenja uključuje korišćenje prethodno označenih podataka (tj. podaci koji već imaju poznatu klasifikaciju) za obučavanje modela. Model se zatim "uči" da prepozna pravilnosti u podacima, kako bi mogao da primeni iste te pravilnosti na nove, nepoznate podatke. Na primer, ako imamo dataset sa recenzijama filmova koje su već označene kao pozitivne ili negativne, model može naučiti da prepozna koje reči, fraze ili obrasci u tekstu signaliziraju pozitivan ili negativan sentiment.

2.2 Algoritmi za klasifikaciju

Za obavljanje zadatka klasifikacije sentimenta, koriste se različiti algoritmi mašinskog učenja. Jedan od najjednostavnijih i najefikasnijih algoritama za klasifikaciju je **logistička regresija**. Iako je naziv "regresija" pomalo zbunjujući, ovaj algoritam se koristi za binarne klasifikacione probleme, gde je cilj da se klasifikuje ulaz u dve kategorije. U analizi sentimenta, logistička regresija može da se koristi za predviđanje da li je sentiment u recenziji pozitivan ili negativan (binarni klasifikacioni problem), na osnovu prethodno obučavajućih podataka.

Logistička regresija funkcioniše tako što koristi logističku funkciju (ili sigmoidnu funkciju) koja mapira izlaz modela na vrednosti između 0 i 1. Na osnovu tih vrednosti, model zatim odlučuje kojoj klasi pripada određeni ulazni podatak (na primer, ako je vrednost veća od 0.5, klasifikovaćemo je kao "pozitivan sentiment", dok ćemo u suprotnom slučaju klasifikovati kao "negativan sentiment").

Pored logističke regresije, postoje i drugi algoritmi za klasifikaciju koji se koriste u analizi sentimenta, kao što su:

- **Naivni Bajesov klasifikator (Naive Bayes)**, koji je zasnovan na teoriji verovatnoće i često se koristi za tekstualne klasifikacije.
- **Support Vector Machines (SVM)**, koji koristi geometrijsku podelu podataka kako bi pronašao granicu koja najbolje razdvaja različite klase.
- **K-nearest neighbors (K-NN)**, koji klasifikuje podatke na osnovu sličnosti sa najbližim podacima u trening setu.

Međutim, logistička regresija je često prva opcija zbog svoje jednostavnosti, brzine treniranja i efektivnosti u rešavanju binarnih klasifikacionih problema.

2.3 Osnovne faze obučavanja modela

Obučavanje modela u nadgledanom učenju obuhvata tri ključne faze: **priprema podataka**, **obučavanje modela** i **evaluacija modela**.

1. Priprema podataka: Prvi korak u procesu obučavanja modela je prikupljanje i priprema podataka. U slučaju analize sentimenta, to znači prikupljanje velikog broja recenzija ili komentara koji su već označeni sa odgovarajućim sentimentom (pozitivno, negativno ili neutralno). Ovaj dataset se mora očistiti i pripremiti za dalje obučavanje. To uključuje procese kao što su uklanjanje nepotrebnih karaktera (npr. interpunkcija, brojevi), pretvaranje svih reči u mala slova, uklanjanje stop reči (rečima koje se često pojavljuju, ali nemaju specifično značenje u kontekstu analize sentimenta, poput "i", "ili", "the"), kao i tokenizaciju (deljenje teksta na manje delove, poput reči ili fraza).

2. Obučavanje modela: Nakon što su podaci pripremljeni, sledeći korak je obučavanje modela. Tokom ove faze, odabrani algoritam (npr. logistička regresija) koristi trening set, koji se sastoji od označenih podataka, da nauči pravilnosti i obrasce koji se javljaju u vezi sa sentimentom u tekstu. Model "uči" kako da mapira ulazne podatke (recenzije) na odgovarajuće izlazne vrednosti (klase sentimenta). Na primer, model će naučiti da su reči kao što su "odličan", "fantastičan", "izvanredno" povezane sa pozitivnim sentimentom, dok su reči kao što su "loš", "razočaran", "gubitak vremena" povezane sa negativnim sentimentom.

3. Evaluacija modela: Nakon obučavanja, model se testira na **test setu**, koji je odvojen od trening seta i sadrži podatke koji nisu korišćeni tokom obučavanja. Evaluacija modela ima za cilj da proveri koliko tačno model može da klasifikuje sentiment novih, nepoznatih podataka. Tipične metrike za evaluaciju uključuju tačnost (accuracy), preciznost (precision), odziv (recall) i F1-score. Na osnovu rezultata evaluacije, model se može dalje optimizovati ili prilagoditi, na primer, podešavanjem hiperparametara, kako bi se postigla bolja tačnost klasifikacije.

3. Tehnike obrade prirodnog jezika (NLP)

Obrada prirodnog jezika (NLP) je oblast mašinskog učenja i lingvistike koja se bavi razumevanjem, interpretacijom i generisanjem ljudskog jezika pomoću računara. NLP omogućava računarima da interaguju sa ljudima na način koji je prirodan za ljude, prepoznajući i analizirajući tekstualne podatke. U kontekstu analize sentimenta, NLP se koristi za prepoznavanje emotivnih tonova u tekstovima, kao što su recenzije filmova, postovi na društvenim mrežama, komentari i drugi oblici pisanog izraza.

Kroz korišćenje različitih tehnika obrade prirodnog jezika, računari mogu "razumeti" tekst i pretvoriti ga u numeričke podatke koje mašinski algoritmi mogu dalje obraditi. Za uspešno prepoznavanje sentimenta, ove tehnike omogućavaju da se prepoznaju ključne reči, fraze i obrasci koji ukazuju na emocionalni ton, bilo da je pozitivan, negativan ili neutralan.

3.1 Tehnike obrade teksta

3.1.1 Tokenizacija

Tokenizacija je osnovna tehnika u NLP-u koja se koristi za razbijanje teksta na manje jedinice koje se nazivaju **tokeni**. Tokeni mogu biti reči, fraze ili čak pojedinačni karakteri, zavisno od cilja analize. U analizi sentimenta, najčešće se koristi **tokenizacija na reči**, što znači da se tekst razdvaja na pojedinačne reči. Na primer, recenzija "Ovaj film je fantastičan!" bi se tokenizovala u sledeće reči: ["Ovaj", "film", "je", "fantastičan"]. Tokenizacija omogućava dalju analizu svake reči, kao što je njena važnost za sentiment, njena povezanost sa drugim rečima ili njena frekvencija u tekstu.

Tokenizacija je prvi korak u mnogim NLP zadacima, jer omogućava da se analiziraju pojedinačni delovi teksta koji mogu nositi značajnu informaciju o sentimentu.

3.1.2 Uklanjanje stop-reči

Stop-reči su najčešće reči u jeziku koje ne dodaju značajnu vrednost u analizi teksta. To su reči poput "i", "je", "na", "to", "samo", "od", "ili", koje se često pojavljuju u svim vrstama teksta, ali obično ne utiču na razumevanje osnovnog značenja ili sentimenta. Zbog toga je uklanjanje stop-reči važan korak u obradi teksta, jer pomaže da se fokusiramo na ključne reči koje zaista mogu uticati na klasifikaciju sentimenta.

Na primer, u recenziji "Ovaj film je zaista odličan i zabavan!", reči "je" i "i" bi se uklonile, a preostale reči ("Ovaj", "film", "zaista", "odličan", "zabavan") bi se koristile za dalju analizu. Uklanjanje stop-reči smanjuje dimenzionalnost podataka, čineći proces analize bržim i efikasnijim.

3.1.3 TF-IDF (Term Frequency - Inverse Document Frequency)

Jedna od ključnih tehnika za pretvaranje teksta u numerički format je **TF-IDF**. Ova metoda omogućava da se identifikuje važnost svake reči u kontekstu čitavog skupa podataka. TF-IDF se sastoji od dva osnovna faktora:

- **Term Frequency (TF)**: Merenje učestalosti pojavljivanja reči u određenom dokumentu. Veća učestalost reči u dokumentu može značiti da je ta reč važnija za taj dokument.
- **Inverse Document Frequency (IDF)**: Merenje koliko je retka ili uobičajena reč u celokupnom skupu podataka. Reči koje se često pojavljuju u mnogim dokumentima (kao što su "je", "u", "na") imaju nisku IDF vrednost, dok reči koje se retko pojavljuju u svim dokumentima imaju veću IDF vrednost.

Kombinovanjem TF i IDF vrednosti, dobijamo broj koji pokazuje značaj određene reči u odnosu na ceo skup podataka. U analizi sentimenta, reči koje imaju visok TF-IDF skor često su ključne za određivanje sentimenta, jer se pojavljuju specifično u dokumentima koji izražavaju određeni stav. Na primer, reči poput "predivan", "očajan" ili "fantastičan" će imati visoke TF-IDF vrednosti u dokumentima sa izrazito pozitivnim ili negativnim sentimentom, jer su specifične za određeni ton.

3.2 Prednosti i izazovi NLP-a u analizi sentimenta

Prednosti:

- **Automatska obrada velikih količina podataka**: Jedna od najvećih prednosti primene NLP tehnika u analizi sentimenta je sposobnost automatske obrade ogromnih količina tekstualnih podataka. Na internetu se svakodnevno generišu milioni recenzija, komentara, postova na društvenim mrežama i drugih tekstova. Ručna analiza takvih podataka bila bi vremenski zahtevna i nepraktična, dok NLP omogućava brzo i efikasno prepoznavanje sentimenta u realnom vremenu.

- **Skalabilnost**: NLP tehnike su skalabilne, što znači da mogu da se primene na velike skupove podataka, uključujući društvene mreže, e-trgovinu, forume i mnoge druge izvore. Analiza sentimenta može pomoći organizacijama da identifikuju trendove, prate javno mnjenje ili prate reakcije na proizvode i usluge.

- **Poboljšanje korisničkog iskustva**: Korišćenjem NLP-a, organizacije mogu bolje razumeti emocije korisnika, što omogućava bolje prilagođavanje proizvoda i usluga potrebama korisnika, kao i brže rešavanje negativnih iskustava pre nego što eskaliraju.

Izazovi:

- **Razumevanje konteksta**: Iako NLP omogućava analizu velikih količina podataka, jedan od najvećih izazova je razumevanje konteksta u kojem su reči korišćene. Na primer, reč "super" može imati pozitivan sentiment u kontekstu filma, dok može imati negativnu konotaciju u kontekstu proizvoda koji je "super loš". NLP tehnike moraju biti dovoljno sofisticirane da prepoznaju ove razlike u kontekstu, što je često izazov.

- **Nijanse jezika**: Ljudi koriste mnoge nijanse u jeziku, kao što su ironične ili sarkastične izjave, koje je teško prepoznati za računare. Takođe, emotivni ton može biti suptilan i teško ga je kvantifikovati u jednostavne kategorije kao što su "pozitivan" ili "negativan". Ovaj problem može biti još složeniji kada se uzmu u obzir regionalne varijante jezika ili specifični sleng.

- **Ambigvitet reči**: Mnoge reči mogu imati višestruka značenja u zavisnosti od konteksta u kojem se koriste. Na primer, reč "light" može označavati nešto "lagano" ili "svetlo". U analizi sentimenta, pravilno razumevanje značenja ovih reči u kontekstu može biti izazov, što zahteva napredne NLP tehnike kao što su semantička analiza ili duboko učenje.

4. Primena analize sentimenta u industriji filmova

Industrija filma je jedno od područja koje značajno koristi analizu sentimenta kako bi bolje razumela reakcije publike, optimizovala strategije promocije i unapredila poslovne odluke. S obzirom na to da je filmska industrija zasnovana na velikim investicijama i visokim očekivanjima, važno je što pre prepoznati kako publika reaguje na određene filmove, što može imati značajan uticaj na marketinške i distribucijske strategije. Korišćenjem analize sentimenta, filmski studiji mogu brzo dobiti uvid u to kakav je opšti utisak publike, kao i u specifične aspekte filma koji su izazvali pozitivne ili negativne reakcije.

4.1 Analiza recenzija filmova

Filmovi i serije često generišu ogromnu količinu tekstualnih podataka u obliku recenzija na različitim platformama kao što su IMDb, Rotten Tomatoes, Metacritic, kao i postova i komentara na društvenim mrežama poput Twittera, Facebooka i YouTube-a. Na ovim platformama, gledaoci izražavaju svoje mišljenje o filmovima koristeći recenzije, ocene i komentare. Ove povratne informacije su izuzetno vredne za filmske studije, ali ručno analiziranje velikog broja recenzija i komentara može biti vremenski i resursno zahtevno. U ovom kontekstu, analiza sentimenta postaje ključna alatka koja omogućava da se u kratkom vremenu dođe do korisnih informacija.

Analiza sentimenta omogućava filmskoj industriji da brzo proceni kako publika reaguje na nove filmove i serije. Prepoznavanjem pozitivnih, negativnih ili neutralnih reakcija, studio može dobiti uvid u to da li film ima potencijal za uspeh ili ako postoje negativni trendovi koje treba hitno adresirati. Ovaj proces ne samo da pomaže u donošenju poslovnih odluka, već i pruža filmskim stvoriteljima informacije o tome šta publika voli ili ne voli, što može direktno uticati na dalji razvoj projekta.

4.2 Primene analize sentimenta u industriji filmova

4.2.1 Predviđanje uspeha filma

Jedna od glavnih primena analize sentimenta u industriji filma je **predviđanje uspeha filma**. Filmski studiji mogu koristiti analizu sentimenta kako bi procenili reakcije na trejler filma, promotivne kampanje ili prve verzije filma pre nego što se zvanično objavi na tržištu. Na osnovu reakcija publike (pozitivnih, negativnih ili neutralnih), studio može bolje razumeti da li će film biti prihvaćen od strane široke publike ili će se suočiti sa negativnim kritikama koje mogu uticati na njegov komercijalni uspeh.

Na primer, ako analiza sentimenta na društvenim mrežama i recenzijama pre izlaska filma ukazuje na negativne reakcije, filmski studio može da preispita strategiju promocije, izvrši izmene u filmu (kao što su dodatni reshoot-ovi ili promocija određenih aspekata filma) ili se fokusira na preformulisanje marketinške strategije. Takođe, studio može proceniti da li film treba da bude premijerno prikazan u širokom distribucijskom krugu ili bi bilo bolje da se fokusira na specifične geografske ili demografske segmente.

Pomoću analize sentimenta, studio može prepoznati ključne reči i fraze koje se ponavljaju u recenzijama, kao što su "sjajan efekat specijalnih efekata", "zabavna gluma", ili "loša režija", što im omogućava da usmere dalju promociju i strategiju distribucije.

4.2.2 Prepoznavanje problema u filmu

Drugi važan način na koji se analiza sentimenta koristi u filmskoj industriji je **prepoznavanje problema u filmu**. Kada se recenzije i komentari u velikoj meri usmere na negativne aspekte filma, kao što su loša gluma, neuvjerljiva produkcija, nedovoljno razvijen zaplet ili loši specijalni efekti, studio može brzo reagovati i prilagoditi svoju strategiju.

Na primer, ako se veliki broj recenzija usmeri na loše performanse glavnog glumca ili glumice, studio može doneti odluku da usmeri dalju promociju na druge aspekte filma, kao što su specijalni efekti, muzika ili scenarista. Ako se recenzije odnose na problem sa zapletom ili režijom, studio može odlučiti da pokrene dodatne marketing kampanje koje prepoznaju i ispravljaju ove negativne aspekte. Takođe, povratne informacije mogu biti korisne u budućem razvoju filma, jer se mogu koristiti za unapređenje sledećih projekata.

Nadalje, analiza sentimenta može pomoći u prepoznavanju specifičnih problema koji mogu biti previđeni tokom proizvodnje filma. Na primer, ako recenzije u velikoj meri spominju negativan ton prema određenim dijalozima ili karakterima, studio može reagovati i pokušati da izmeni ili optimizuje sledeće projekte, čineći ih bližima željama i potrebama publike.

4.2.3 Poboljšanje angažovanja publike

Kroz analizu sentimenta, filmska industrija može dobiti uvid u to kako gledaoci reaguje na određene scene, likove ili tematske pristupe. To omogućava studioima da bolje angažuju svoju publiku, stvore dublje emocionalne veze sa filmom i poboljšaju svoje marketinške strategije.

Na primer, ako je analiza sentimenta pokazala da publika posebno voli lik nekog od glavnih junaka, studio može odlučiti da se više fokusira na te likove u promociji filma, stvarajući dodatne video materijale, interaktivne sadržaje ili merchandise koji je vezan za te likove. Slično tome, ako se otkrije da je određena scena izazvala negativne reakcije zbog neskladnosti u dijalogu ili loših vizuelnih efekata, studio može pravovremeno uputiti korekcije kako bi izbegao negativne utiske na široj publici.

5. Izazovi u analizi sentimenta

Iako analiza sentimenta predstavlja moćan alat za automatsku procenu reakcija publike, postoje brojni izazovi u njenoj primeni, naročito kada se koristi za analizu filmova. Razumevanje emocija i stavova izraženih u tekstu nije trivijalan zadatak, jer ljudski jezik je kompleksan i često se koristi u kontekstu koji može biti dvosmislen ili čak protivan doslovnom značenju reči. U ovom delu ćemo razmotriti najvažnije izazove u analizi sentimenta, sa posebnim naglaskom na specifičnosti vezane za analizu filmova.

5.1 Kontekstualni problemi

Jedan od najvećih izazova u analizi sentimenta je razumevanje **konteksta** u kojem su reči i fraze upotrebljene. Jezik je bogat nijansama, i mnoge reči ili izrazi mogu značiti različite stvari u različitim situacijama. **Sarkazam**, **humor** i **ironija** predstavljaju ozbiljne prepreke za pravilno prepoznavanje sentimenta. Na primer, rečenica "Ovaj film je definitivno najgori koji sam ikada video, neverovatno loš!" može se činiti negativnom, ali ako je izrečena u sarkastičnom tonu, u stvarnosti može biti izjava koja implicira suprotan sentiment (da je film zapravo dobar).

Računarski modeli, čak i oni zasnovani na naprednim tehnološkim rešenjima poput dubokog učenja, često imaju problema da prepoznaju sarkazam jer se on ne oslanja samo na semantičko značenje reči, već i na ton glasa, intonaciju i neverbalne signale, koji su ključni za ljudsku interpretaciju. U tekstualnim recenzijama filmova, humor može dodatno zakomplikovati analizu, jer se često koristi kao način izražavanja oduševljenja ili kritike na indirektan način.

Na primer, recenzija poput "Film je bio takav haos, savršen!" može biti dvosmislena. Ako algoritam ne prepozna da se koristi u kontekstu pozitivnog oduševljenja, mogao bi klasifikovati recenziju kao negativnu. Razumevanje konteksta je ključno za ispravnu procenu sentimenta, ali to predstavlja izazov za mašinsko učenje, jer algoritmi moraju da nauče kako da razlikuju različite tonove i intonacije u tekstu.

5.2 Polisemija

Polisemija, ili višeznačnost reči, još je jedan značajan izazov u analizi sentimenta. Mnoge reči u jeziku imaju više značenja koja mogu biti veoma različita, zavisno od konteksta u kojem se koriste. Ovo je naročito problematično u analizi filmova, gde se isti termin može koristiti na različite načine u različitim kontekstima. Na primer, reč "cool" može imati različita značenja u recenzijama filmova. U jednoj recenziji može značiti da je nešto "zanimljivo" ili "fantastično", dok u drugoj može značiti "hladno" ili "neprijatno", što bi imalo negativnu konotaciju.

Polisemija se javlja kada reč menja značenje u zavisnosti od konteksta rečenice. Na primer, reč "sharp" u kontekstu filma može značiti "oštar" u fizičkom smislu, ali može se koristiti i da označi "inteligentan" ili "brz" u smislu analize karaktera ili dijaloga. Bez pravilnog razumevanja tog konteksta, algoritmi za analizu sentimenta mogu doneti netačne zaključke.

S obzirom na to da mnoge reči imaju više značenja, algoritmi moraju da prepoznaju koji je pravi kontekst za svaku reč, što je često izazov čak i za napredne modele dubokog učenja, koji moraju uzeti u obzir ne samo značenje reči, već i širi kontekst cele rečenice, paragrafa i čak celokupnog teksta.

5.3 Povezanost reči i struktura rečenica

Iako su pojedinačne reči važne za analizu sentimenta, **povezanost između reči** i **struktura rečenica** takođe igraju ključnu ulogu u tačnoj klasifikaciji sentimenta. Razumevanje kako se reči međusobno povezuju u okviru rečenice omogućava bolju interpretaciju stava autora. Na primer, rečenica "Film je, iako je imao nekoliko loših scena, uglavnom bio zabavan" sadrži i pozitivne i negativne elemente, pa je teško doneti jednostavan zaključak. Bez pravilne analize povezivanja između reči i njihovog konteksta, algoritmi bi mogli nespretno klasifikovati takav tekst kao negativan ili pozitivan, što bi bilo netačno.

Struktura rečenica je od presudnog značaja u prepoznavanju sentimenta, jer raspored reči može značajno promeniti značenje. Na primer, rečenica "Film je bio strašan, ali zabavan" nosi potpuno suprotan sentiment od rečenice "Film je bio zabavan, ali strašan", iako se u oba slučaja pojavljuje ista reč ("strašan"). Razumevanje složenosti jezičke strukture, kao što su kontrasti, negacije i modifikatori, zahteva napredne tehnike obrade jezika i dubokog učenja.

Većina tradicionalnih algoritama za analizu sentimenta, kao što su bag-of-words ili jednostavni modeli zasnovani na frekvenciji reči, ne uzima u obzir ove povezane strukture i semantičke veze između reči, što može dovesti do površinskih ili netačnih rezultata. Da bi se ove veze pravilno interpretirale, neophodno je koristiti napredne tehnike kao što su rekurentne neuronske mreže (RNN), dugoročne memorije (LSTM) ili transformatori, koji mogu bolje obraditi i razumeti sekvencijalnu prirodu jezika.

6. Primena teme kroz zadatak

Ovaj zadatak se bavi analizom sentimenata u recenzijama filmova korišćenjem nadgledanog učenja. Cilj je izgraditi model koji može da prepozna da li je recenzija pozitivna ili negativna na osnovu njenog teksta. Za postizanje ovog cilja koristićemo skup podataka sa IMDb recenzijama filmova, koji se sastoji od teksta recenzija i njihovih pripadajućih sentimenta.

U prvom koraku, učitaćemo podatke i očistiti tekst tako da uklonimo nepotrebne stop-reči i specijalne karaktere. Zatim, pomoću tehnike **TF-IDF** (Term Frequency-Inverse Document Frequency), konvertovaćemo tekstualne podatke u numerički oblik koji može da se koristi za obučavanje modela.

Za modeliranje ćemo koristiti **logističku regresiju**, koja je efikasan model za binarnu klasifikaciju (u ovom slučaju, predikcija da li je recenzija pozitivna ili negativna). Nakon što obučimo model, testiraćemo njegovu tačnost i evaluiraćemo ga koristeći metrike poput **tačnosti** i **izveštaja o klasifikaciji**.

Konačno, model će biti sačuvan zajedno sa TF-IDF vektorizatorom, kako bi mogao da se koristi za predikciju sentimenta novih recenzija. Na kraju, biće implementirana funkcija koja omogućava korisnicima da unesu novu recenziju i dobiju predikciju o njenom sentimentu (pozitivnom ili negativnom).

Ovaj pristup omogućava efikasnu analizu velikih količina tekstualnih podataka i automatsku klasifikaciju recenzija filmova, što može biti korisno u raznim aplikacijama kao što su preporučivači filmova ili analiza korisničkog zadovoljstva.

[1]:
```
import pandas as pd

# Učitavanje fajla sa zaglavljem
file_path = 'C:/Users/Windows HD/Desktop/Imdb/IMDb_Movie_Reviews.csv'
```

Ovaj deo koda je početak učitavanja podataka iz CSV fajla koji sadrži recenzije filmova.

import pandas as pd: Ovim se importuje biblioteka pandas, koja je jedan od najpopularnijih alata za rad sa podacima u Pythonu. Koristi se za rad sa tabelama (dataframe-ovima), omogućavajući lako manipulaciju i analizu podataka.

file_path = 'C:/Users/Windows HD/Desktop/Imdb/IMDb_Movie_Reviews.csv': Ova linija definiše putanju do fajla koji sadrži recenzije filmova., i ovde se koristi fajl sa imenom IMDb_Movie_Reviews.csv. Ovaj fajl se nalazi na desktopu u folderu Imdb. U sledećem delu koda se koristi ova putanja da bi se učitali podaci iz CSV fajla u pandas DataFrame.

[2]:
```
# Učitavanje CSV fajla sa ručno dodeljenim kolonama
df = pd.read_csv(file_path, header=None, names=['review', 'sentiment'])
```

Ovaj deo koda se koristi za učitavanje podataka iz CSV fajla u pandas DataFrame sa dodatnim podešavanjima za nazive kolona

pd.read_csv(file_path): Ovo je funkcija koja se koristi za učitavanje podataka iz CSV fajla (putanja fajla je prethodno definisana kao file_path). Ova funkcija učitava podatke u pandas DataFrame, što je struktura podataka u obliku tabela (slično kao Excel ili SQL tabele).

header=None: Ova opcija govori pandas-u da CSV fajl ne sadrži zaglavlje (imena kolona) u prvoj liniji. To znači da prva linija u CSV fajlu sadrži stvarne podatke, a ne nazive kolona.

names=['review', 'sentiment']: Pošto fajl nema zaglavlje, ovaj argument omogućava da ručno dodelimo imena kolonama. Kolone u CSV fajlu će biti označene kao review (za tekst recenzije) i sentiment (za sentiment koji označava da li je recenzija pozitivna ili negativna, verovatno sa vrednostima 0 ili 1). Na kraju, rezultati će biti smešteni u promenljivu df, koja je pandas DataFrame sa dve kolone: review: Sadrži tekstove recenzija filmova. sentiment: Sadrži sentiment (npr. 1 za pozitivnu recenziju, 0 za negativnu recenziju). Ovaj DataFrame je sada spreman za dalje procesiranje i analizu, kao što su čišćenje teksta i priprema za modeliranje.

[3]:
```
# Prikazivanje prvih nekoliko redova i imena kolona
print(df.head())
print(df.columns)
import nltk
from nltk.corpus import stopwords
import re
```

```
                                              review  sentiment
0                                                  0          1
1  I did not enjoy the film Eraser whatsoever. It...          0
2  Be very afraid of anyone who likes this film. ...          0
3  The 3rd and last big screen spin off from the ...          0
4  Barely three and a half years after just scrap...          1
Index(['review', 'sentiment'], dtype='object')
```

Ovaj deo koda ima dva ključna zadatka: prvo prikazuje prve redove učitanog DataFrame-a, a zatim učitava biblioteke koje će se koristiti za obradu teksta.

print(df.head()): df.head() prikazuje prvih 5 redova DataFrame-a. Ova funkcija je korisna kada želimo da vidimo kako podaci izgledaju u početnoj fazi obrade. Ispisaće sadržaj prvih nekoliko recenzija zajedno sa njihovim pripadajućim sentimentima

print(df.columns): df.columns prikazuje imena kolona DataFrame-a. Ova funkcija se koristi za potvrdu da li su kolone pravilno imenovane. U ovom slučaju, očekuje se da će biti prikazano nešto poput: Index(['review', 'sentiment'], dtype='object')

import nltk: Ova linija učitava biblioteku NLTK (Natural Language Toolkit), koja je jedna od najpopularnijih biblioteka za obradu prirodnog jezika u Pythonu. Pruža mnoge alate za analizu i rad sa tekstom, uključujući tokenizaciju, lematizaciju, prepoznavanje entiteta, rad sa stop-rečima i drugo.

from nltk.corpus import stopwords: stopwords je skup često korišćenih reči u jeziku (kao što su "i", "a", "the" u engleskom jeziku) koje obično nemaju značajnu informaciju za analize. Ovaj import omogućava korišćenje stop-reči sa NLTK bibliotekom.

import re: re je Python modul koji se koristi za rad sa regularnim izrazima. Regularni izrazi omogućavaju efikasno prepoznavanje i manipulaciju tekstom, kao što je uklanjanje specijalnih karaktera, brojeva, interpunkcija i slično. Ovaj modul će biti koristan u procesu čišćenja teksta.

[5]:
```
# Preuzimanje stop-words
import nltk
from nltk.data import find
from nltk.corpus import stopwords

# Provera da li je stopwords već preuzet
try:
    find('corpora/stopwords.zip')  # Traži stopwords
except LookupError:
    nltk.download('stopwords')  # Ako nije pronađen, preuzmi ga
```

Ovaj deo koda se bavi preuzimanjem stop-reči sa NLTK biblioteke, ako one već nisu preuzete.

import nltk: Kao i ranije, ovo učitava biblioteku NLTK koja se koristi za rad sa prirodnim jezikom.

from nltk.data import find: Ova linija omogućava da koristimo find funkciju iz nltk.data modula, koja je korisna za pretragu podataka u NLTK biblioteci, kao što su predtrenirani modeli, korpusi, stop-reči itd.

from nltk.corpus import stopwords: Ovo omogućava korišćenje stopwords skupa reči iz NLTK biblioteke. Stop-reči su reči koje se obično uklanjaju iz teksta jer ne sadrže značajnu semantičku vrednost, kao što su "i", "the", "on" i druge, koje se često pojavljuju u svakodnevnom govoru.

try: find('corpora/stopwords.zip'): Ova linija pokušava da pronađe paket stopwords na vašem računaru. Funkcija find traži datoteku sa stop-rečima na predefinisanom putu ('corpora/stopwords.zip'), koji je uobičajeni direktorijum gde NLTK skladišti podatke.

except LookupError:: Ako find funkcija ne može da pronađe paket stop-reči (što znači da nije preuzet), pokreće se except blok.

nltk.download('stopwords'): Ako stop-reči nisu preuzete, ova linija preuzima paket stop-reči sa interneta. Ovaj proces preuzimanja može da potraje neko vreme, ali jednom kada su preuzete, moći ćete da ih koristite u svim budućim analizama. Kao rezultat, nakon što se ovaj kod izvrši, biće vam omogućeno da koristite skup stop-reči na engleskom jeziku, koji se koristi za filtriranje manje značajnih reči prilikom obrade teksta, kao što su u ovom zadatku recenzije filmova. Sledeći korak je primena funkcije za čišćenje teksta.

[6]:
```
# Lista stop-words na engleskom
stop_words = set(stopwords.words('english'))
```

Ova linija koda preuzima listu stop-reči na engleskom jeziku iz NLTK biblioteke i smešta je u promenljivu stop_words.

stop_words = set(stopwords.words('english')):

stopwords.words('english'): Ova funkcija preuzima listu stop-reči na engleskom jeziku sa NLTK biblioteke. Ove reči su često korišćene reči u jeziku koje obično nemaju značajnu informaciju za analize teksta. Primeri stop-reči u engleskom jeziku uključuju "i", "the", "on", "at", "is" itd. set(...): Ovo koristi Python-ov set tip podataka kako bi pretvorio listu stop-reči u skup (set). Skup je struktura podataka koja garantuje da su svi elementi unikatni (ne može biti duplih vrednosti). Koristeći skup, osiguravamo da se svaka stop-reč pojavljuje samo jednom, što može ubrzati kasnije operacije filtriranja i obrade. Nakon što se ovaj kod izvrši, stop_words sadrži

skup svih stop-reči na engleskom jeziku. Ovaj skup se kasnije koristi da se uklone manje značajne reči iz recenzija tokom procesa čišćenja teksta. Ako želite, možemo preći na sledeći deo koda koji implementira funkciju za čišćenje teksta.

[7]:
```python
# Funkcija za čišćenje teksta
def clean_text(text):
    # Pretvaranje u mala slova
    text = text.lower()

    # Uklanjanje specijalnih karaktera, brojeva i interpunkcija
    text = re.sub(r'[^a-z\s]', '', text)

    # Razbijanje teksta na reči i uklanjanje stop-reči
    words = text.split()
    words = [word for word in words if word not in stop_words]

    # Vraćanje čistog teksta kao string
    return ' '.join(words)
```

Ova funkcija clean_text je ključna za pripremu teksta pre nego što se koristi za obučavanje modela za predikciju sentimenata. Njena svrha je da očisti tekstualne podatke uklanjanjem nepotrebnih delova, kao što su stop-reči, specijalni karakteri i brojevi, čineći ga pogodnijim za analizu.

1. Pretvaranje u mala slova:

python Copy code text = text.lower() Ova linija konvertuje ceo tekst u mala slova. Na taj način se eliminiše razlika između velikih i malih slova, što pomaže u uniformnosti teksta (npr. "Film" i "film" biće tretirani kao ista reč). Ovo je važan korak u analizi teksta.

2. Uklanjanje specijalnih karaktera, brojeva i interpunkcija:

python Copy code text = re.sub(r'[^a-z\s]', '', text) re.sub(r'[^a-z\s]', '', text) koristi regularni izraz (regex) da bi uklonio sve karaktere koji nisu slova (a-z) i razmake (\s). To znači da se uklanjaju brojevi, interpunkcija i svi specijalni simboli. [^a-z\s]: Ovaj regularni izraz znači "sve što nije slovo ili razmak". Na primer, tekst "This is a 5-star movie!" će postati "this is a star movie".

3. Razbijanje teksta na reči i uklanjanje stop-reči:

python Copy code words = text.split() words = [word for word in words if word not in stop_words] text.split(): Ova funkcija razdvaja tekst na listu reči koristeći razmake kao separator. Na primer, tekst "this is a great movie" biće podeljen na listu reči: ["this", "is", "a", "great", "movie"].

[word for word in words if word not in stop_words]: Ova linija filtrira reči tako da se uklone stop-reči. Koristi prethodno definisani skup stop_words koji sadrži često korišćene reči poput "the", "is", "and", itd. Samim tim, reči koje nisu informativne za analizu (poput "is" ili "a") se brišu iz liste reči. Na primer, lista reči ["this", "is", "a", "great", "movie"] postaje ["great", "movie"] nakon uklanjanja stop-reči.

4. Vraćanje čistog teksta kao string:

python Copy code return ' '.join(words) ' '.join(words) spaja listu reči u jedan string, vraćajući rezultat u formi čistog teksta, gde su reči odvojene razmakom. Na primer, lista ["great", "movie"] biće pretvorena u string "great movie".

Rezime: Funkcija clean_text čisti svaki tekstualni input tako što: Pretvara sve reči u mala slova. Uklanja brojeve, specijalne karaktere i interpunkciju. Uklanja stop-reči koje nemaju značajnu informaciju. Vraća tekst koji je sada spreman za dalju analizu, kao što je vektorizacija ili modeliranje. Ova funkcija će biti primenjena na sve recenzije u skupu podataka pre nego što se koriste za treniranje modela.

[8]:
```
# Primena funkcije na sve recenzije
df['cleaned_text'] = df['review'].apply(clean_text)  # 'review' je ime kolone sa recenzijama
```

Ova linija koda primenjuje funkciju clean_text na sve recenzije u DataFrame-u, što znači da svaka recenzija iz kolone review prolazi kroz proces čišćenja definisan u prethodnoj funkciji.

Detaljno objašnjenje:

df['cleaned_text'] = df['review'].apply(clean_text): df['review']: Ovo označava kolonu review u DataFrame-u df, koja sadrži originalne tekstove recenzija.

apply(clean_text): apply() je metoda koja omogućava primenu funkcije na svaki element određene kolone ili reda u DataFrame-u. U ovom slučaju, apply(clean_text) znači da će se funkcija clean_text primeniti na svaku recenziju u koloni review.

df['cleaned_text']: Ova nova kolona u DataFrame-u će sadržati očistite (prečišćene) verzije originalnih recenzija. Dakle, umesto originalnog teksta, biće sačuvane verzije sa malim slovima, bez specijalnih karaktera i stop-reči.

Sledeći korak uključuje vektorizaciju teksta, što znači pretvaranje očišćenih recenzija u numerički format koji može biti upotrebljen za obučavanje modela.

[9]:
```
# Ispisivanje nekoliko čistih recenzija
print(df['cleaned_text'].head())
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
```

Ovaj deo koda ima dva osnovna cilja: prvi je ispisivanje nekoliko očišćenih recenzija, dok je drugi uvođenje biblioteka potrebnih za dalje korake u obradi podataka, kao što su vektorizacija i podela skupa podataka na obučavajući i testirajući skup.

1. Ispisivanje nekoliko čistih recenzija:

print(df['cleaned_text'].head())

df['cleaned_text']: Ovo se odnosi na novu kolonu u DataFrame-u, koja sadrži očišćene recenzije (tj. recenzije koje su prošle kroz funkciju clean_text).

.head(): Ovo je metoda Pandas biblioteke koja prikazuje prvih nekoliko (podrazumevano 5) redova iz odabrane kolone ili DataFrame-a. U ovom slučaju, prikazivaće prvih pet očigledno očišćenih recenzija.

2. Uvoz biblioteka za vektorizaciju i podelu podataka:

from sklearn.feature_extraction.text import TfidfVectorizer: Ova linija uvozi TfidfVectorizer iz sklearn biblioteke, što je alat za pretvaranje tekstualnih podataka (poput recenzija) u numerički format.

TF-IDF (Term Frequency-Inverse Document Frequency) je tehnika za vektorizaciju teksta koja daje težinu svakom terminu (reči) u dokumentu u zavisnosti od toga koliko često se ta reč pojavljuje u dokumentu u odnosu na sve ostale dokumente

u skupu. To pomaže da se da veća važnost rečima koje su specifične za određeni dokument, a manja važnost onima koje su česte u većini dokumenata.

from sklearn.model_selection import train_test_split: Ova linija uvozi funkciju train_test_split iz sklearn.model_selection, koja omogućava jednostavno podelu podataka na dva skupa: jedan za obučavanje modela (trening skup) i drugi za testiranje modela (test skup). To je osnovni korak u većini mašinskog učenja, jer vam omogućava da procenite performanse modela na podacima koji nisu korišćeni u obuci, što pomaže da se izbegne pretreniranje (overfitting).

[10]:
```
# Inicijalizacija TF-IDF vektorizatora
vectorizer = TfidfVectorizer(max_features=5000)  # Koristi samo 5000 najvažnijih reči
```

Ova linija koda inicijalizuje TF-IDF vektorizator koji se koristi za pretvaranje tekstualnih podataka (recenzija) u numerički format. Ovde se koristi TfidfVectorizer iz sklearn biblioteke, sa dodatkom parametra max_features=5000.

TfidfVectorizer: Ovaj vektorizator konvertuje tekstualne podatke u numerički oblik tako što koristi TF-IDF (Term Frequency-Inverse Document Frequency) tehniku. Term Frequency (TF) meri koliko često se reč pojavljuje u određenom dokumentu (u ovom slučaju, recenziji). Inverse Document Frequency (IDF) meri koliko je reč specifična za određeni dokument u odnosu na sve dokumente u skupu podataka. Reči koje se često pojavljuju u mnogim recenzijama dobijaju manju težinu, dok reči koje se retko pojavljuju u svim recenzijama (ali su specifične za određeni dokument) dobijaju veću težinu. Korišćenjem ove tehnike, svaka recenzija se transformiše u numerički vektor koji odražava važnost svake reči u tom dokumentu.

max_features=5000: Ovaj parametar postavlja maksimalni broj karakteristika (reči) koje će se koristiti u vektorizaciji. max_features=5000 znači da će vektorizator koristiti samo 5000 najvažnijih reči, tj. reči koje imaju najveću TF-IDF vrednost u skupu podataka. Ovo je korisno jer može smanjiti dimenzionalnost podataka i poboljšati efikasnost obrade, posebno u velikim skupovima podataka.

[11]:
```
# Primena vektorizacije na obučavajući skup
X = df['cleaned_text']  # Ulazni podaci (čiste recenzije)
y = df['sentiment']     # Izlazne vrednosti (sentiment - 0 ili 1)
```

Ova linija koda priprema ulazne (features) i izlazne (labels) podatke koji će se koristiti za obučavanje modela mašinskog učenja.

X = df['cleaned_text']: X označava ulazne podatke (ili features) koje model koristi za obučavanje. df['cleaned_text'] predstavlja kolonu cleaned_text u DataFrame-u df, koja sadrži očišćene recenzije (tekstove) koji su prethodno prošli kroz funkciju clean_text. Ove recenzije će biti korišćene kao ulaz u model za predikciju sentimenta (pozitivna ili negativna recenzija). Kada se vektorizacija primeni, svaka recenzija će biti pretvorena u numerički vektor, koji će predstavljati sve reči u recenziji sa njihovim odgovarajućim TF-IDF vrednostima.

y = df['sentiment']: y označava izlazne podatke (ili labels) koje model treba da predvidi.

df['sentiment'] predstavlja kolonu sentiment u DataFrame-u, koja sadrži oznake za svaku recenziju (0 za negativan sentiment, 1 za pozitivan sentiment). Ove vrednosti su ono što model treba da nauči da predviđa na osnovu ulaznih podataka (očistjenih recenzija).

[12]:
```
# Podela podataka na obučavajući i testirajući skup (70% obučavanje, 30% testiranje)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Ova linija koda koristi funkciju train_test_split da podeli podatke na obučavajući (training) i testirajući (test) skup. Ova podela je ključna u procesu mašinskog učenja, jer omogućava modelu da se obuči na jednom skupu podataka, a zatim se testira na drugom skupu koji nije korišćen tokom obuke. Na taj način se procenjuju performanse modela i izbegava pretreniranje (overfitting).

Ovaj skup sadrži 70% podataka, jer je test_size=0.3 (30% podataka će biti u testirajućem skupu).

random_state=42: Ovaj parametar omogućava da podela podataka bude reproduktivna. Kada koristite isti random_state, dobićete istu podelu podataka svaki put kada pokrenete kod, što je korisno za reprodukciju rezultata. Broj "42" je proizvoljan, ali ga često koriste ljudi u programiranju kao "magijski broj".

[13]:
```
# Vektorizacija (pretvaranje recenzija u numerički format)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Ovaj deo koda primenjuje TF-IDF vektorizaciju na obučavajući i testirajući skup podataka. Vektorizacija je ključni korak u pripremi tekstualnih podataka za modeliranje u mašinskom učenju, jer transformiše recenzije, koje su tekstualne prirode, u numerički format koji mašinski modeli mogu da obrade.

[14]:
```
# Ispisivanje oblika matrica (dimenzije)
print("Shape of X_train_tfidf:", X_train_tfidf.shape)
print("Shape of X_test_tfidf:", X_test_tfidf.shape)
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

```
Shape of X_train_tfidf: (17500, 5000)
Shape of X_test_tfidf: (7501, 5000)
```

Ova linija koda je korisna za proveru dimenzionalnosti (oblika) podataka koji su prošli kroz TF-IDF vektorizaciju. Na osnovu ovih dimenzija možemo videti broj recenzija i broj karakteristika (reči) koje su korišćene u vektorizaciji.

Dimenzija matrice je u formatu (broj uzoraka, broj karakteristika): Broj uzoraka (prva dimenzija) je broj recenzija u skupu podataka (70% za obučavanje i 30% za testiranje). Broj karakteristika (druga dimenzija) je broj najvažnijih reči koje je TF-IDF vektorizator izabrao da koristi, u ovom slučaju 5000 reči, prema prethodnoj postavci max_features=5000. Značaj ovog ispisa: Ovi brojevi pomažu da proverimo da li je broj uzoraka i broj karakteristika u skladu sa vašim očekivanjima. Na primer, ako smo koristili 5000 karakteristika, očekujete da broj kolona bude 5000. Ukoliko je broj redova manji ili veći od broja recenzija u skupu, može biti korisno proveriti podatke.

```
[15]: # Kreiranje i treniranje modela
      model = LogisticRegression(max_iter=1000)
      model.fit(X_train_tfidf, y_train)
```

```
[15]: LogisticRegression
      LogisticRegression(max_iter=1000)
```

Ovaj deo koda kreira i trenira logističku regresiju kao model za predikciju sentimenta na osnovu TF-IDF vektorizovanih podataka. Logistička regresija je popularni model za binarne klasifikacije, kao što je predviđanje pozitivnog ili negativnog sentimenta na osnovu recenzija.

Ova funkcija kreira objekat modela logističke regresije. max_iter=1000: Parametar max_iter određuje maksimalni broj iteracija koje model može napraviti tokom procesa optimizacije. U ovom slučaju, model će pokušati da konvergira unutar 1000 iteracija. Ako model ne može da pronađe optimalna rešenja u ovom broju iteracija, proces obuke se može zaustaviti. Povećanje ovog broja može biti korisno ako model ima problema sa konvergencijom, ali previše iteracija može biti prekomerno.

Trening modela: Ova metoda trenira model na osnovu podataka. Model koristi X_train_tfidf (TF-IDF vektorizovane recenzije iz obučavajućeg skupa) kao ulazne podatke, i y_train (oznaku sentimenta) kao ciljnu promenljivu (target variable). Tokom ovog koraka, model uči kako da poveže karakteristike (TF-IDF vrednosti) sa oznakama sentimenta (pozitivno ili negativno), koristeći tehniku optimizacije koja minimizuje grešku modela.

```
[16]: # Predviđanje na test skupu
      y_pred = model.predict(X_test_tfidf)
```

Ova linija koda koristi trenirani model da predvidi sentiment za recenzije u testirajućem skupu. Na osnovu prethodnih podataka i treniranja, model će pokušati da odredi da li je sentiment recenzije pozitivan (1) ili negativan (0).

[17]:
```
# Evaluacija modela
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.8880149313424877
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.87      0.89      3761
           1       0.88      0.90      0.89      3740

    accuracy                           0.89      7501
   macro avg       0.89      0.89      0.89      7501
weighted avg       0.89      0.89      0.89      7501
```

Ovaj deo koda koristi funkcije iz biblioteke sklearn.metrics za evaluaciju performansi modela, tačnije za izračunavanje tačnosti i detaljnog izveštaja o klasifikaciji.

Ova funkcija izračunava tačnost modela. Tačnost je jednostavna metrike koja predstavlja udeo tačno klasifikovanih uzoraka u odnosu na sve uzorke u testirajućem skupu.

Ova funkcija generiše detaljan izveštaj o klasifikaciji koji uključuje nekoliko važnih metrika:

Preciznost (Precision): Ovo je udeo tačno klasifikovanih pozitivnih predikcija u odnosu na sve predikcije koje su model označene kao pozitivne. Preciznost odgovara na pitanje: "Koliko je od mojih predviđenih pozitivnih recenzija zaista bilo pozitivnih?

Odziv (Recall): Ovo je udeo tačno klasifikovanih pozitivnih predikcija u odnosu na sve stvarne pozitivne uzorke. Odziv odgovara na pitanje: "Koliko je od stvarnih pozitivnih recenzija model uspešno predvideo?"

Podrška (Support): Ovo je broj uzoraka u svakoj klasi (pozitivnoj ili negativnoj) u testirajućem skupu. Izveštaj o klasifikaciji je vrlo koristan za ocenjivanje performansi modela po klasama, naročito kada su klase neuravnotežene (ako ima mnogo više negativnih recenzija nego pozitivnih, ili obrnuto).

Šta možemo da očekujete: Tačnost (Accuracy) će biti broj između 0 i 1, gde veće vrednosti označavaju bolji model. Izveštaj o klasifikaciji daje detaljan pregled performansi modela za svaku klasu (pozitivnu i negativnu). To uključuje metrike kao što su preciznost, odziv i F1-skor, kao i podršku.

Iz rezultata koje smo dobili možemo analizirati performanse modela na sledeći način:

Tačnost (Accuracy): Accuracy: 0.888 (oko 88.8%) Tačnost modela je 88.8%, što znači da je model ispravno klasifikovao oko 89% testnih recenzija. Ovo je solidan rezultat i pokazuje da model dobro funkcioniše.

Izveštaj o klasifikaciji: Za klasu 0 (Negativna recenzija):

Preciznost (Precision): 0.90 Preciznost od 90% znači da je 90% od svih predviđenih negativnih recenzija zaista bilo negativnih.

Odziv (Recall): 0.87 Odziv od 87% znači da je model uspeo da prepozna 87% svih stvarnih negativnih recenzija. F1-skor (F1-score): 0.89 F1-skor od 0.89 znači da postoji dobar balans između preciznosti i odziva za negativne recenzije.

Za klasu 1 (Pozitivna recenzija):

Preciznost (Precision): 0.88 Preciznost od 88% znači da je 88% od svih predviđenih pozitivnih recenzija zaista bilo pozitivnih.

Odziv (Recall): 0.90 Odziv od 90% znači da je model uspeo da prepozna 90% svih stvarnih pozitivnih recenzija. F1-skor (F1-score): 0.89 F1-skor od 0.89 pokazuje da model dobro balansira između preciznosti i odziva za pozitivne recenzije.

Makro i ponderisani prosek: Makro prosek (Macro avg) i ponderisani prosek (Weighted avg) za preciznost, odziv i F1-skor su svi približno 0.89, što znači da model ima veoma izbalansirane performanse između pozitivnih i negativnih recenzija.

Makro prosek računa prosek metrika bez obzira na broj uzoraka u svakoj klasi, dok ponderisani prosek uzima u obzir broj uzoraka u svakoj klasi (tako da klase sa više uzoraka imaju veći uticaj na ukupnu metriku).

Zaključak: Model ima vrlo dobar učinak sa tačnošću od 88.8%, što znači da je u stanju da predvidi sentiment (pozitivnu ili negativnu recenziju) sa velikom preciznošću. Preciznost, odziv i F1-skor za obe klase (pozitivnu i negativnu recenziju) su dobro izbalansirani i visoki, što znači da model nije sklonn prekomernim greškama u klasifikaciji. Sve u svemu, model je postigao dobar rezultat.

[18]:
```
import joblib

# Sačuvaj TF-IDF vektorizator
joblib.dump(vectorizer, 'tfidf_vectorizer.pkl')

# Sačuvaj model (logistic regression)
joblib.dump(model, 'sentiment_model.pkl')

print("Model i TF-IDF vektorizator su sačuvani!")
```

```
Model i TF-IDF vektorizator su sačuvani!
```

joblib.dump(vectorizer, 'tfidf_vectorizer.pkl'): Ova linija koda čuva TF-IDF vektorizator (koji je prethodno obučen na podacima) u fajl pod imenom tfidf_vectorizer.pkl. TF-IDF vektorizator je alat koji je preveo tekstualne recenzije u numeričke vektore (TF-IDF vrednosti) koji se koriste za modeliranje. Ovaj vektorizator će biti potreban za transformaciju novih recenzija u istom formatu, kako bi model mogao da ih analizira i predviđa njihov sentiment.

joblib.dump(model, 'sentiment_model.pkl'): Ova linija koda čuva trenirani model logističke regresije u fajl pod imenom sentiment_model.pkl. Model sadrži naučene parametre koji omogućavaju predviđanje sentimenta na osnovu numeričkih vektora koji su dobijeni pomoću TF-IDF vektorizatora.

print("Model i TF-IDF vektorizator su sačuvani!"): Ova linija samo ispisuje potvrdu da su oba objekta uspešno sačuvana.

Zašto je ovo korisno? Sačuvane verzije modela i vektorizatora omogućavaju da kasnije ponovo koristimo oba objekta bez potrebe da ponovo učimo model i vektorizator. Na taj način možemo učitati sačuvane modele i primeniti ih na nove recenzije za predviđanje sentimenta.

[19]:
```
import joblib

# Učitaj TF-IDF vektorizator i model
vectorizer = joblib.load('tfidf_vectorizer.pkl')
model = joblib.load('sentiment_model.pkl')
```

joblib.load('tfidf_vectorizer.pkl'): Ova linija učitava sačuvani TF-IDF vektorizator sa diska. On je prethodno obučen na recenzijama i sačuvan u fajl sa imenom tfidf_vectorizer.pkl. Ovaj vektorizator će biti korišćen da transformiše nove tekstove (recenzije) u numeričke vrednosti koje model može da koristi za predviđanje.
joblib.load('sentiment_model.pkl'): Ova linija učitava sačuvani model (logističku regresiju) koji je obučen na podacima i sačuvan u fajl sa imenom sentiment_model.pkl. Model sadrži parametre koji omogućavaju klasifikaciju recenzija na pozitivne ili negativne na osnovu prethodno obučenih podataka.

[20]:
```
# Funkcija za čišćenje novog teksta pre predikcije
def clean_new_text(text):
    # Pretvaranje u mala slova
    text = text.lower()

    # Uklanjanje specijalnih karaktera, brojeva i interpunkcija
    text = re.sub(r'[^a-z\s]', '', text)

    # Razbijanje teksta na reči i uklanjanje stop-reči
    words = text.split()
    words = [word for word in words if word not in stop_words]

    # Vraćanje čistog teksta kao string
    return ' '.join(words)
```

Ova funkcija clean_new_text() se koristi za čišćenje novog teksta pre nego što ga prosledite modelu za predviđanje sentimenta. Cilj je da obezbedimo da tekst bude u odgovarajućem formatu, sličnom onom na kojem je model treniran.

[21]:
```python
# Funkcija za predviđanje sentimenta
def predict_sentiment(new_review):
    # Čišćenje novog komentara
    cleaned_review = clean_new_text(new_review)

    # Vektorizacija nove recenzije pomoću iste TF-IDF transformacije
    review_tfidf = vectorizer.transform([cleaned_review])

    # Predviđanje sentimenta (0 ili 1) pomoću modela
    sentiment = model.predict(review_tfidf)

    # Vraćanje rezultata kao "Pozitivna recenzija" ili "Negativna recenzija"
    return "Pozitivna recenzija" if sentiment[0] == 1 else "Negativna recenzija"
```

Ova funkcija predict_sentiment() koristi prethodno očišćen i vektorizovani tekst da bi predvidela sentiment (pozitivan ili negativan) nove recenzije, koristeći prethodno trenirani model i TF-IDF vektorizator.

Čišćenje novog komentara: Prvo, funkcija poziva prethodno definisanu funkciju clean_new_text() kako bi očistila novi tekst (recenziju) koji želimo da analiziramo. Ovo uklanja sve specijalne karaktere, brojeve, i stop-reči, čineći tekst spremnim za dalje procesiranje.

Vektorizacija: Nakon što je tekst očišćen, koristi se TF-IDF vektorizator da bi se transformisala recenzija u numerički format. Ovaj korak pretvara reči u numeričke vektore, koji omogućavaju modelu da ih razume i analizira. Ovaj korak koristi isti vektorizator koji je korišćen pri obuci modela (učitava se iz fajla). Rezultat ove transformacije je sparse matrica koja sadrži numeričke vrednosti koje predstavljaju učestalost reči u recenziji.

Predviđanje sentimenta: Model koji je treniran na prethodnim podacima koristi TF-IDF transformisane podatke da predvidi sentiment nove recenzije. Predviđeni rezultat će biti 0 (negativna recenzija) ili 1 (pozitivna recenzija).

Vraćanje rezultata: Na osnovu predviđanja, funkcija vraća string "Pozitivna recenzija" ako je predviđeni sentiment 1 (pozitivno), ili "Negativna recenzija" ako je predviđeni sentiment 0 (negativno).

Kako funkcioniše predviđanje: clean_new_text(): Očišćava novu recenziju tako što uklanja nepotrebne reči, specijalne znakove i brojeve. vectorizer.transform(): Pretvara očišćenu recenziju u numerički oblik pomoću TF-IDF vektorizatora. model.predict(): Model zatim koristi te numeričke vrednosti da predvidi sentiment na osnovu onoga što je naučio iz prethodnih podataka.

Primer: "I absolutely loved this movie! It was amazing and exciting." Funkcija bi prvo očistila tekst da bi izgledao ovako: "absolutely loved movie amazing exciting" Zatim, uz pomoć TF-IDF vektorizatora, recenzija bi bila transformisana u numerički oblik, koji model koristi da predvidi sentiment. Ako model predvidi 1 (pozitivno), funkcija bi vratila:"Pozitivna recenzija".

```
[22]: # Testiranje sa novim komentarima
new_review = "I absolutely loved this movie! It was so exciting and fun to watch."
predicted_sentiment = predict_sentiment(new_review)
print(predicted_sentiment)  # Očekivani rezultat: "Pozitivna recenzija"

Pozitivna recenzija
```

U ovom delu koda, testiramo funkcionalnost predviđanja sentimenta koristeći funkciju predict_sentiment().

new_review = "I absolutely loved this movie! It was so exciting and fun to watch." Ova recenzija sadrži vrlo pozitivne reči kao što su "loved", "exciting", "fun". Očekuje se da model prepozna pozitivan sentiment. Pozivanje funkcije za predviđanje sentimenta: predicted_sentiment = predict_sentiment(new_review)

Funkcija predict_sentiment() prvo čisti recenziju, zatim koristi TF-IDF vektorizator da je transformiše u numerički oblik, a zatim koristi trenirani model za predviđanje sentimenta.

Ispis rezultata: print(predicted_sentiment) Na kraju, funkcija će vratiti rezultat "Pozitivna recenzija", jer je recenzija vrlo pozitivna, a model bi trebao to prepoznati na osnovu naučenih obrazaca. Očekivani izlaz: Ako je model tačno treniran, izlaz bi trebao biti: Pozitivna recenzija

Ovaj izlaz potvrđuje da model ispravno prepoznaje sentiment u tekstu.

7. Zaključak

U današnjem digitalnom dobu, analiza sentimenta postaje sve važniji alat u različitim industrijama, a posebno u industriji filmova. Korišćenje nadgledanog učenja za analizu sentimenta omogućava precizno razumevanje stavova i emocija izraženih u recenzijama filmova, što je od suštinskog značaja za filmske studije, distributere i marketinške stručnjake. Ovaj proces pomaže u donošenju informisanih poslovnih odluka, predviđanju uspeha filmova, kao i prepoznavanju mogućih problema u proizvodnji i promociji.

Tehnička dostignuća u oblasti obrade prirodnog jezika (NLP) i mašinskog učenja značajno su unapredila sposobnost analize sentimenta. Korišćenje algoritama kao što su logistička regresija, kao i naprednih modela dubokog učenja poput LSTM i transformatora, omogućava tačnu klasifikaciju recenzija, čak i kada su u pitanju složeni izrazi i jezičke nijanse. Algoritmi za analizu sentimenta mogu prepoznati i razvrstati tekstualne podatke prema emocionalnom tonu, što je ključno za filmsku industriju u procesu kreiranja i promovisanja filmova.

Međutim, kao što smo videli, izazovi u analizi sentimenta, poput kontekstualnih problema, polisemije i složenosti jezičkih struktura, i dalje postoje. Ponekad, prepoznavanje sentimenta u filmovima zahteva dublje razumevanje specifičnih kulturnih i jezičkih konteksta, kao i sposobnost da se pravilno interpretiraju sarkazam, humor i ironične izjave, koje su uobičajene u recenzijama filmova. Uprkos ovim izazovima, napredak u NLP-u i tehnologijama mašinskog učenja pokazuje veliki potencijal za dalje poboljšanje tačnosti analize sentimenta.

Budućnost analize sentimenta u industriji filmova i drugim industrijama izgleda vrlo obećavajuće. Sa stalnim napretkom u razvoju modela za duboko učenje i obradom složenih jezičkih struktura, analiza sentimenta će postati još preciznija i široko primenjiva. Filmski studiji će moći bolje da predviđaju tržišne reakcije, optimizuju marketinške kampanje i brže reaguju na povratne informacije publike. Takođe, ove tehnologije će se širiti na druge oblasti, kao što su politika, trgovina i obrazovanje, gde se analiza sentimenta može koristiti za bolje razumevanje stava i mišljenja velikih grupa ljudi.

U zaključku, analiza sentimenta predstavlja ne samo tehnički napredak u obradi prirodnog jezika, već i značajan korak ka boljoj interakciji između industrija i njihovih potrošača. U filmskoj industriji, kao i u mnogim drugim, koristi od ove tehnologije će nastaviti da rastu, omogućavajući brže, tačnije i efikasnije donošenje odluka na osnovu podataka.

Literatura

1. **Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.**

2. **Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. Foundations and Trends in Information Retrieval, 2(1-2), 1-135.**

3. **Sebastiani, F. (2002). *Machine Learning in Automated Text Categorization*. ACM Computing Surveys (CSUR), 34(1), 1-47.**

4. **Bo Pang, Lillian Lee. (2004). *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*. Proceedings of the ACL 2004.**

5. **Scikit-learn dokumentacija (https://scikit-learn.org)**

6. **NLTK dokumentacija (https://www.nltk.org)**