



Univerzitet u Sarajevu
Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku



Klasifikacija novinskih članaka

Predmet: Vještačka Inteligencija



Odgovorni nastavnik:

Vanr. prof. dr Amila Akagić

Tim:

Amina Kazazović

Daris Mujkić

Odgovorni asistent:

Elvedin Smajić

SADRŽAJ

SADRŽAJ.....	2
1. UVOD U PROBLEM I TEHNOLOGIJE.....	3
1.1. OPIS PROBLEMA I PRIMJENA.....	3
1.2. DEFINICIJA OSNOVNIH POJMOVA.....	3
1.3. KRATKI PREGLED POSTOJEĆIH DATASETOVA.....	4
2. PREGLED STANJA U OBLASTI.....	5
2.1. TRENUTNO STANJE U OBLASTI PROBLEMA KOJI SE RJEŠAVA.....	5
Prvi izvor: Machine Learning : Text Classification of News Articles [4].....	5
Drugi izvor: Vietnamese News Articles Classification Using Neural Networks [6].....	10
Treći izvor: Word2vec convolutional neural networks for classification of news articles and tweets [7].....	16
3. IZBOR, ANALIZA I PRETPROCESIRANJE DATASET-A.....	22
3.1. IZBOR I ANALIZA DATASET-A.....	22
3.2. IDENTIFIKOVANI RIZICI I IZAZOVI.....	23
3.3. METODE I IMPLEMENTACIJA PRETPROCESIRANJA PODATAKA.....	24
3.4. PODJELA DATASETA ZA TRENING, VALIDACIJU I TESTIRANJE.....	26
4. ODABIR, FORMATIRANJE, TRENIRANJE I TESTIRANJE MODELA.....	26
4.1. IZBOR METODE I TEHNOLOGIJE.....	26
4.2. PRIPREMA FORMATA PODATAKA DA ODGOVARA MODELU.....	27
4.3. ARHITEKTURA MODELA I OPTIMIZACIJA HIPERPARAMETARA.....	27
4.4 TRENIRANJE I TESTIRANJE FINALNOG MODELA.....	28
4.5 TESTIRANJE MODELA NA NEPOZNATIM PODACIMA.....	31
5. CJELOKUPNI OSVRT NA PROBLEM I DOBIJENO RJEŠENJE.....	31
5.1 OSVRT NA REZULTATE.....	31
5.2 USPOREDBA SA REFERENTNIM RADOVIMA.....	32
5.3 DISKUSIJA O POTENCIJALNIM POBOLJŠANJIMA.....	33
6. IZRADA WEB APLIKACIJE UZ ANVIL FRAMEWORK.....	33
6.1 FRONTEND.....	34
6.2 BACKEND.....	34
7. REFERENCE.....	37

1. UVOD U PROBLEM I TEHNOLOGIJE

1.1. OPIS PROBLEMA I PRIMJENA

U današnjem dobu, velika količina novinskih članaka se objavljuje svakodnevno na internetu. Prevelika količina ovih informacija je izazov ne samo za čitaoce koji žele brzo da pronađu vijesti koje ih zanimaju, već i za medejske kuće koje žele da što efikasnije organizuju svoj sadržaj. Kada bi se kategorizacija vršila ručno to bi bilo vremenski zahtjevno, subjektivno i podložno greškama. Ovo bi dodatno otežalo pretragu i analizu informacija.

Cilj ovog projekta je izrada modela za automatsku klasifikaciju novinskih članaka, koji na osnovu naslova i teksta predviđa njegovu kategoriju. Na ovaj način se postiže lakša pretraga, poboljšano korisničko iskustvo i efikasnije upravljanje sadržajem.

Za rješavanje ovog problema potencijalno se mogu koristiti metode obrade prirodnog jezika, uključujući tokenizaciju, vektorizaciju teksta i neuronske mreže.

1.2. DEFINICIJA OSNOVNIH POJMOMA

1. *Klasifikacija* - Čest zadatak u algoritmima vještačke inteligencije koji se bavi prepoznavanjem objekata kao pripadnika neke kategorije, odnosno klase. Ovaj proces omogućava razvrstavanje ulaznih instanci podataka u unaprijed definisane diskretne vrijednosti (klase). Klasifikacija je jedan od dva tipa nadziranog učenja.
2. *Tokenizacija* - Proces dijeljenja teksta u riječi ili manje jedinice (tokeni). Na ovaj način se formira indeks ili rječnik koji predstavlja vokabular skupa podataka, pružajući informacije o relativnoj frekvenciji pojavljivanja svake riječi u tekstu koji se tokenizira.
3. *Vektorizacija* - Proces pretvaranja teksta u numerički oblik kako bi mogao biti korišten u algoritmima mašinskog učenja. Cilj ovog koraka je pronaći reprezentativniji numerički oblik teksta, koji omogućava računarskim modelima da analiziraju i procesuiraju podatke na način koji je razumljiv za algoritme.
4. *Neuronske mreže* - Računarski modeli inspirisani ljudskim mozgom, sastavljeni od povezanih "neurona" organiziranih u slojeve. Oni služe za prepoznavanje obrazaca i

rješavanje problema poput klasifikacije i regresije, kroz učenje i prilagođavanje parametara na osnovu podataka.

1.3. KRATKI PREGLED POSTOJEĆIH DATASETOVA

Za klasifikaciju novinskih članaka, dostupno je više javnih datasetova koji se mogu koristiti za treniranje i evaluaciju modela. U nastavku je predstavljen kratak pregled nekoliko datasetova koji su slični našoj tematici:

- **Dutch News Articles Dataset (Kaggle) [1]**

Ovaj dataset sadrži nizozemske novinske članke s atributima: vrijeme objavljivanja članka, naslov, tekst, kategorija i url. Zbog dobro strukturiranih podataka, može poslužiti kao primjer za analizu strukture datasetova i kao dodatni resurs za istraživanje višejezične klasifikacije.

- **News Articles Dataset (Kaggle) [2]**

Ovaj skup podataka uključuje više hiljada članaka koji su prikupljeni s različitih web stranica. Atributi uključuju: tekst, datum objavljivanja, naslov i kategorija. Dataset može poslužiti za dodatnu evaluaciju ili za proširivanje osnovnog skupa podataka.

- **Turkish News Articles Dataset (Kaggle) [3]**

Dataset sastavljen od članaka na turskom jeziku, koji uključuje različite tematske kategorije. Iako nije direktno korišten u ovom projektu, predstavlja koristan primjer datasetova koji podržavaju klasifikaciju tekstova na jezicima osim engleskog. U ovom datasetu imamo sljedeće attribute: datum objavljivanja, naziv autora članka, naslov, link i tekst.

Pregled ovih datasetova pokazuje da postoji značajna količina podataka dostupna za istraživanja u oblasti automatske klasifikacije tekstova, što omogućava razvoj i evaluaciju efikasnih modela vještačke inteligencije u ovom domenu.

2. PREGLED STANJA U OBLASTI

2.1. TRENUTNO STANJE U OBLASTI PROBLEMA KOJI SE RJEŠAVA

Prvi izvor: Machine Learning : Text Classification of News Articles [4]

Prvo ćemo se fokusirati na analizu članka “Machine Learning: Text Classification of News Articles” od autora Hedi Manai na stranici Medium. Ovaj članak pruža praktičan primjer primjene tehnika za klasifikaciju novinskih članaka.

OPSEG PROBLEMA

Autor članka se bavi problemom multi-klasne klasifikacije novinskih članaka. Cilj je bio predvidjeti kojoj od 5 kategorija pripada svaki članak iz **BBC News Classification** dataseta [5].

Dataset Description

File descriptions

- **BBC News Train.csv** - the training set of 1490 records
- **BBC News Test.csv** - the test set of 736 records
- **BBC News Sample Solution.csv** - a sample submission file in the correct format

Data fields

- **ArticleId** - Article id unique # given to the record
- **Article** - text of the header and article
- **Category** - category of the article (tech, business, sport, entertainment, politics)

Slika 1: Opis dataseta na kojem je zasnovan članak

KORIŠTENE METODE VJEŠTAČKE INTELIGENCIJE

1. Predobrada podataka (Data Cleaning & Data Preprocessing) - tehnike obrade prirodnog jezika NLP

- Ovo je ključni korak u mašinskom učenju gdje se vrši transformacija sirovih podataka u razumljiv i upotrebljiv format. Na ovaj način se poboljšava kvalitet podataka i olakšava posao modelima.
- Kako je korišteno u članku?
 - Konverzija kategorija u numeričke indekse: Tekstualne oznake kategorija (npr. "Sports", "Business") su pretvorene u numeričke vrijednosti (0, 1, 2, itd.).

	Category	CategoryId
0	business	0
3	tech	1
5	politics	2
6	sport	3
7	entertainment	4

Slika 2: Konvertovane kategorije u numeričke indekse

- Uklanjanje tagova (HTML) i specijalnih karaktera: Autor zadržava samo alfanumeričke znakove.
- Pretvaranje u mala slova (Lowercasing): Osigurava da se iste riječi napisane različitim slovima tretiraju kao identične (npr. "Vijest" i "vijest").
- Uklanjanje stop-riječi (Stopword Removal): Česte riječi koje ne doprinose klasifikaciji članaka (npr. "the", "a", "is", "in") su uklonjene iz teksta.

```
def remove_stopwords(text):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    return [x for x in words if x not in stop_words]

dataset['Text'] = dataset['Text'].apply(remove_stopwords)
dataset['Text'][1]
```

Slika 3: Funkcija za uklanjanje stop - riječi

- Lematizacija (Lemmatizing the Words): Riječi su svedene na njihov osnovni leksički oblik ili korijen (lemu). Na primjer, "running", "runs", "ran" bi bile grupisane kao jedna riječ "run" i na ovaj način se smanjuje broj jedinstvenih riječi.

2. Vektorizacija teksta (Vectorization) - tehnika ekstrakcije atributa iz NLP-a

- Nakon što je tekst obrađen, vrši se njegovo pretvaranje u format (vektor) koji algoritmi mašinskog učenja razumiju.
- Kako je korišteno u članku?
 - Bag of Words (BoW) model pomoću kojeg se svaki članak predstavi kao vektor frekvencija riječi. Svaka jedinstvena riječ u svim člancima postaje kolona u matrici. Naziv modela nije besmislen, pa samim tim svaki članak može da se zamislji kao "vreća" riječi, gdje je bitno samo koje se riječi nalaze u vreći i koliko puta se svaka riječ pojavljuje, a ne njihov raspored.

3. Priprema podataka za modeliranje: trening i test skup – pristup u mašinskom učenju

- Originalni dataset se dijeli na dva dijela: skup za treniranje (train set) i skup za testiranje (test set). Model se trenira na trening skupu, a zatim se njegova performansa evaluira na testnom skupu.
- Kako je korišteno u članku?

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =
0.3, random_state = 0, shuffle = True)
print(len(x_train))
print(len(x_test))
```

Slika 4: Dijeljenje dataseta na 2 skupa

4. Primjena klasifikacionih algoritama mašinskog učenja

- Klasifikacioni algoritmi mašinskog učenja su metode koje računarski sistem koristi da nauči kako da automatski svrstava podatke u unaprijed definisane kategorije. Na osnovu primjera iz trening podataka, gdje su poznate i karakteristike podataka i njihove klase, ovi algoritmi grade model koji prepoznaže obrasce. Nakon treniranja, performanse modela se mjere na testnom skupu.
- Kako je korišteno u članku?

Nakon pripreme podataka za modeliranje, vrši se treniranje različitih klasifikacionih algoritama. Za rješavanje problema, svaki od ovih klasifikatora je korišten unutar OneVsRestClassifier strategije. Ova strategija trenira po jedan klasifikator za svaku klasu, gdje svaki klasifikator uči da razlikuje "svoju" klasu od svih ostalih klasa zajedno.

Korišteni su sljedeći algoritmi:

1. Logistic Regression (Logistička Regresija): Linearni model koji predviđa vjerovatnoću pripadnosti klasi.
 2. Random Forest (Slučajna Šuma): Skup metoda koji se sastoji od više stabala odlučivanja; rezultat se dobija "glasanjem" svih stabala.
 3. Multinomial Naive Bayes (Multinomijalni Naivni Bayes): Probabilistički klasifikator zasnovan na Bayesovoj teoremi.
 4. Support Vector Classifier (SVC) (Mašina Potpornih Vektora): Algoritam koji pronalazi optimalnu hiperravan koja najbolje razdvaja klase u prostoru atributa.
 5. Decision Tree Classifier (Stablo Odluke): Model koji donosi odluke kroz seriju pravila predstavljenih kao stablo.
 6. K Nearest Neighbour (K Najbližih Susjeda): Algoritam koji klasificiže novi podatak na osnovu većinske klase njegovih K najbližih susjeda u prostoru atributa.
 7. Gaussian Naive Bayes (Gausov Naivni Bayes): Varijanta Naivnog Bayesa koja prepostavlja da atributi prate Gausovu (normalnu) distribuciju.
- Svaki model se trenira na skupu, pravi predikcije i na kraju evaluira.

POSTIGNUTI REZULTATI

	Model	Test Accuracy	Precision	Recall	F1
0	Logistic Regression	97.09	0.97	0.97	0.97
1	Random Forest	97.99	0.98	0.98	0.98
2	Multinomial Naive Bayes	97.09	0.97	0.97	0.97
3	Support Vector Classifier	96.64	0.97	0.97	0.97
4	Decision Tree Classifier	83.22	0.83	0.83	0.83
5	K Nearest Neighbour	73.60	0.74	0.74	0.74
6	Gaussian Naive Bayes	76.06	0.76	0.76	0.76

Slika 5: Postignuti rezultati za svaki model

Autor je postigao najbolje rezultate sa Random Forest algoritmom, dok se najgore pokazao K Nearest Neighbour.

POTENCIJALNI PRAVCI ZA POBOLJŠANJE

1. Naprednija vektorizacija teksta

U članku je korišten CountVectorizer, koji se može zamijeniti sa TfidfVectorizer-om. TF-IDF daje bolje rezultate jer pored brojanja frekvencije riječi (kao CountVectorizer), dodjeljuje težinu riječima na osnovu njihove važnosti za konkretni članak, uzimajući u obzir i koliko su te riječi rijetke ili česte u cijelokupnoj kolekciji članaka. Težina riječi raste proporcionalno broju pojavljivanja te riječi u dokumentu, ali se umanjuje proporcionalno frekvenciji pojavljivanja u svim ostalim člancima. Na ovaj način se veći značaj daje riječima koje su ključne za neki članak i umanjuje značaj opštih riječi koje su u mnogo članaka. Ovo pomaže modelima da bolje razlikuju kategorije.

2. Korištenje N-grama

U članku je vektorizacija bila zasnovana na unigramima (pojedinačnim riječima). Problem u tome je što može doći do gubitka konteksta, jer nam je za to nekad potreban skup riječi. Pomoću N-grama se kao token tretira sekvenca od N riječi. Npr. bigram (N=2) bi kombinovao dvije uzastopne riječi ("mašinsko učenje"). Oni se mogu uključiti u CountVectorizer ili TfidfVectorizer uporedno sa unigramima. Na ovaj način model bolje razumije članak i preciznije vrši njegovu klasifikaciju.

3. Korištenje neuronskih mreža

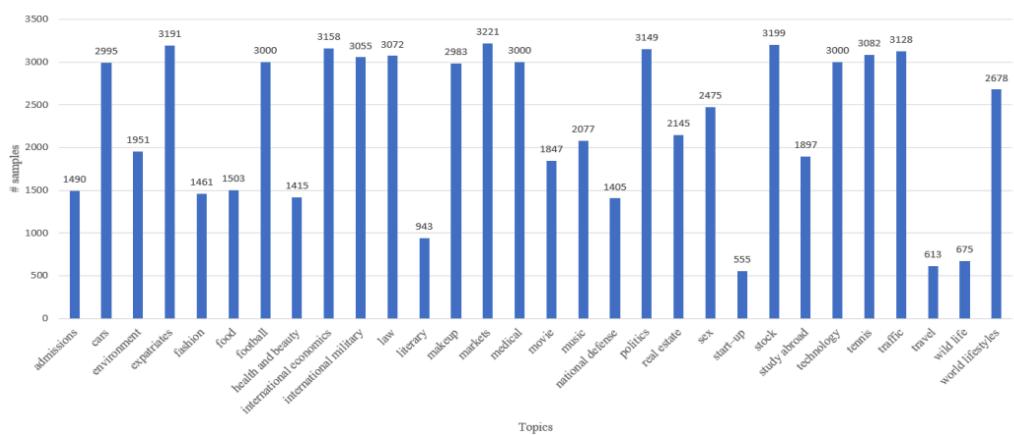
Ovaj članak se više oslanjao na tradicionalne algoritme, koji se mogu unaprijediti primjenom neuronskih mreža. Neuronske mreže mogu automatski učiti složene obrasce i bitne karakteristike direktno iz teksta. Moglo bi se primijeniti različite arhitekture: Obične neuronske mreže, CNN (Konvolucione Neuronske Mreže) i LSTM/GRU (Rekurentne Neuronske Mreže sa Long Short-Term Memory ili Gated Recurrent Unit celijama). Na ovaj način se postižu preciznije klasifikacije s obzirom da mogu da modeliraju kompleksnije jezične strukture i semantičke odnose u tekstu.

Drugi izvor: Vietnamese News Articles Classification Using Neural Networks [6]

Vršimo analizu naučnog rada “Vietnamese News Articles Classification Using Neural Networks” od autora To Nguyen Phuoc Vinh i Ha Hoang Kha.

OPSEG PROBLEMA

Ovaj članak se bavi problemom multi-label klasifikacije online novinskih članaka na vijetnamskom jeziku. Cilj je automatski klasifikovati svaki članak u neke od 30 različitih tema. Dataset koji je korišten je sakupljen od autora članaka sa 5 poznatih vijetnamskih stranica za vijesti, koristeći Python Scrapy framework.



Slika 6: Dataset vijetnamskih online novinskih članaka

KORIŠTENE METODE VJEŠTAČKE INTELIGENCIJE

1. Predobrada teksta (Text Preprocessing) - tehnike obrade prirodnog jezika NLP

- Kao što je rečeno za prošli članak, ovo je fundamentalna faza u svakom NLP projektu. Transformišu se sirovi podaci u čist, konzistentan format koji je pogodan za dalju analizu i primjenu algoritama mašinskog učenja.
- Kako je korišteno u članku?
 - Autori navode da su uklonili brojeve, specijalne karaktere, znakove interpunkcije, te izvršili konverziju u mala slova.
 - Uklanjanje stop-riječi: Napravili su listu od oko 2000 nepotrebnih riječi za klasifikaciju. Na ovaj način su smanjili računarsku složenost.

2. Tokenizacija i kreiranje rječnika - NLP tehnike

- Tokenizacija je proces koji dijeli tekst u niz manjih jedinica (tokeni). Tokeni mogu da karakteri, riječi ili N-grami. Ovim se generiše rječnik u kojem se nalazi svi jedinstveni tokeni iz članaka.
- Kako je korišteno u članku? Koristili su sljedeće modele za tokenizaciju:
 - Bag-of-Words (BoW) model za tokenizaciju
 - N-gram model za tokenizaciju: Autori generišu 3 rječnika pomoću bi-grama, 3-grama i 4-grama.
 - Kombinacija BoW i N-gram modela: Generisali su još 3 rječnika kombinovanjem ova dva modela (BoW i bigram, BoW i 3-gram, BoW i 4-gram).

3. Vektorizacija - tehnika ekstrakcije atributa iz NLP-a

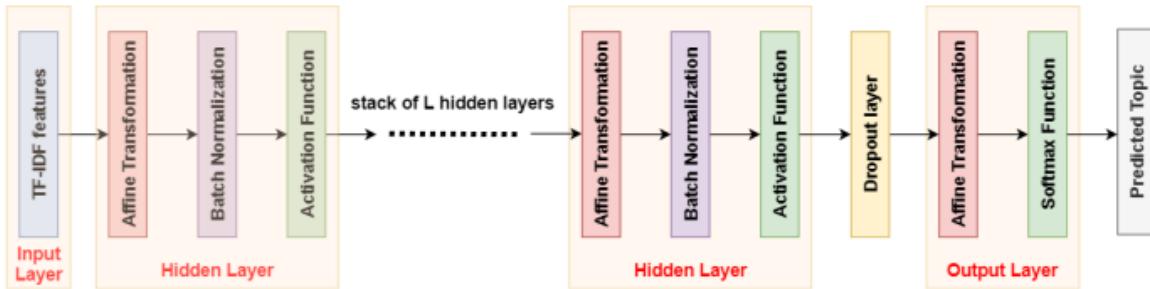
- Nakon tokenizacije i kreiranje rječnika, vrši se pretvaranje svakog članka u vektor fiksne dužine.
- Kako je korišteno u članku? Korištena je TF-IDF tehnika koja svakom tokenu u članku dodjeljuje težinu. Težina je visoka ako se token često pojavljuje u tom članku, a slabo u ostalima. Računali su je putem formule $TF-IDF(w,d) = TF(w,d) * IDF(w)$, pri čemu TF predstavlja frekvenciju tokena u tom članku, a IDF mjera koja pokazuje koliko je riječ rasprostranjena u cijeloj kolekciji članaka (više rasprostranjena - manji IDF). Nakon ovog se svaki članak transformiše u TF-IDF vektor veličine 10000.

4. Odabir značajki (Feature Selection) kao metoda smanjenja dimenzionalnosti

- TF-IDF transformacija kao rezultat daje vektor velike dimenzionalnosti, tako da se vrši tehnika smanjenja dimenzionalnosti.
- Kako je korišteno u članku? Sortiraju se tokeni prema njihovim TF vrijednostima i uzimaju se prvih K s najvišim vrijednostima.

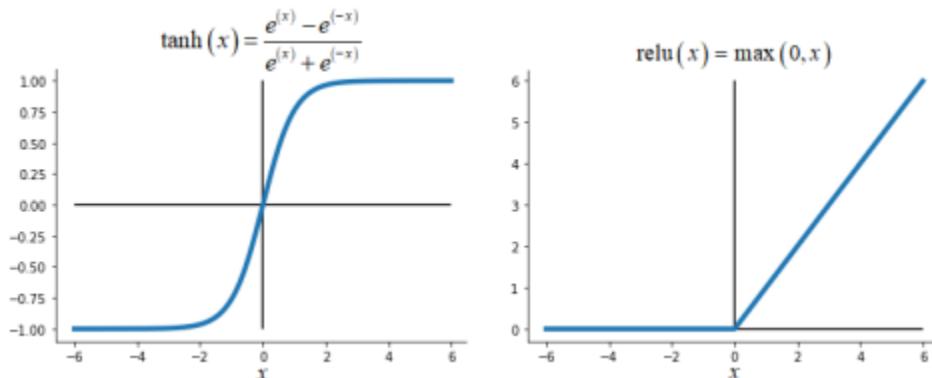
5. Model klasifikacije - Neuronske mreže (Neural Networks)

- Korištene su neuronske mreže za klasifikaciju članaka.
- Kako je korišteno u članku?



Slika 7: Arhitektura neuronske mreže

- 1) **Ulazni sloj:** Prima TF-IDF vektore
- 2) **Skriveni sloj:** Ima L skrivenih slojeva. Svaki sloj uključuje sljedeće linearne transformacije, batch normalizaciju i aktivacijsku funkciju.



Slika 8: Korištene aktivacijske funkcije

- 3) **Dropout sloj:** Korišten samo na posljednjem skrivenom sloju, kako bi se spriječio overfitting. Vjerovatnoća izbacivanja $p=0.3$.
- 4) **Izlazni sloj:** Sastoje se od linearne transformacije i softmax aktivacijske funkcije.

Dataset je podijeljen na trening (75%), validacijski (12.5%) i test (12.5%) skup. Funkcija gubitka je Categorical cross-entropy, optimizator Adam i trening je izvršen kroz 100 epoha sa 128 uzoraka.

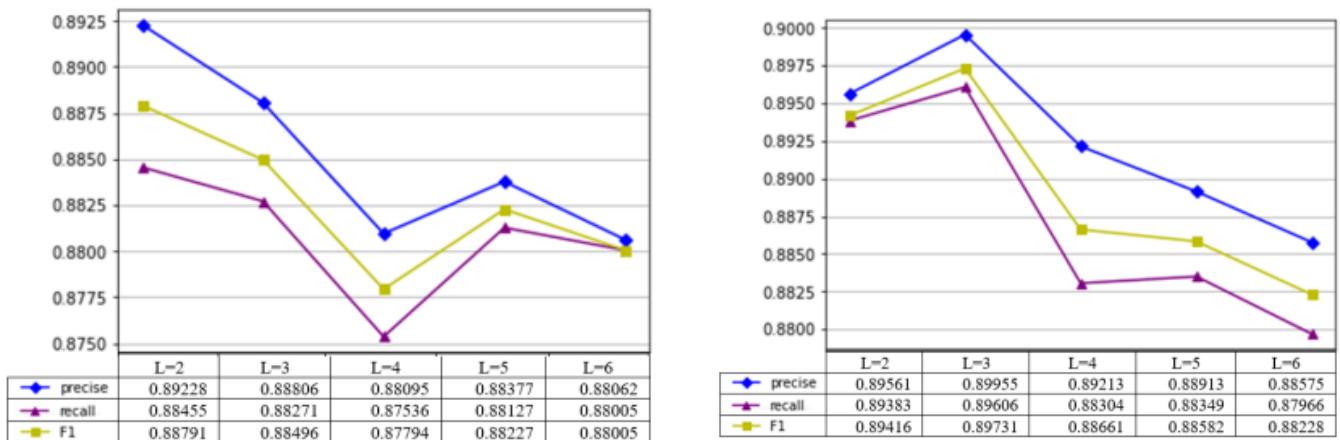
POSTIGNUTI REZULTATI

Izvršena je evaluacija modela koristeći F1-score metriku. Ona predstavlja harmoničnu sredinu između preciznosti (precision) i odziva (recall).

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Slika 9: Korištena formula za F1

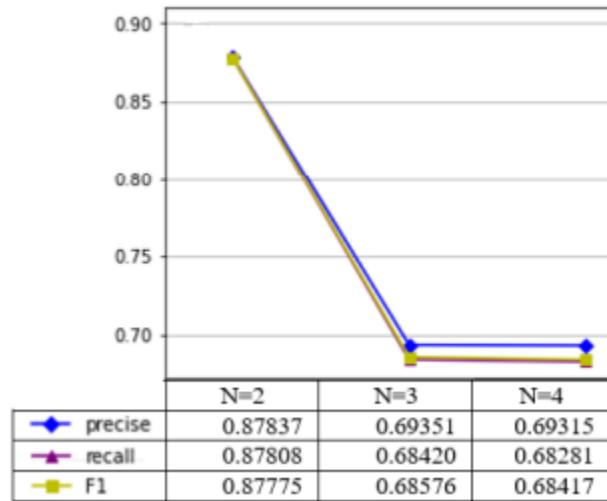
1. Rezultati na osnovu broja skrivenih slojeva i aktivacijskih funkcija



Slika 10: Evaluacija tanh (lijevo) i ReLU (desno) aktivacijske funkcije sa različitim brojem skrivenih slojeva

- Za tanh funkciju su postignuti najbolji rezultati sa 2 skrivena sloja (0.88791)
- Za ReLU funkciju su postignuti najbolji rezultati sa 3 skrivena sloja (0.89731)

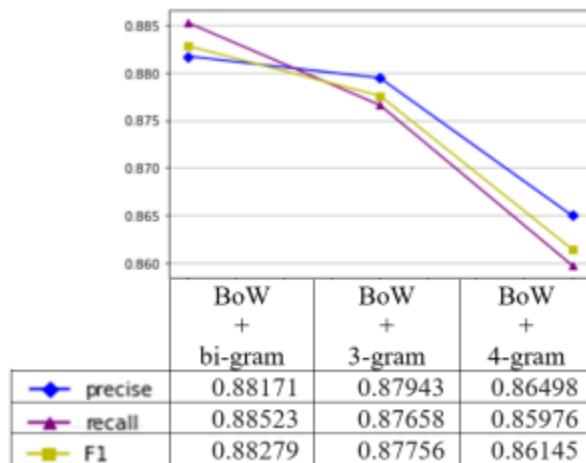
2. Rezultati na osnovu reda N-grama



Slika 11: Evaluacija modela na različitim redovima N-grama

- Korišten je model sa 3 skrivena sloja i ReLU aktivacijskom funkcijom, te je postignut najbolji rezultat sa 1-gramom tjst BoW (0.87775).

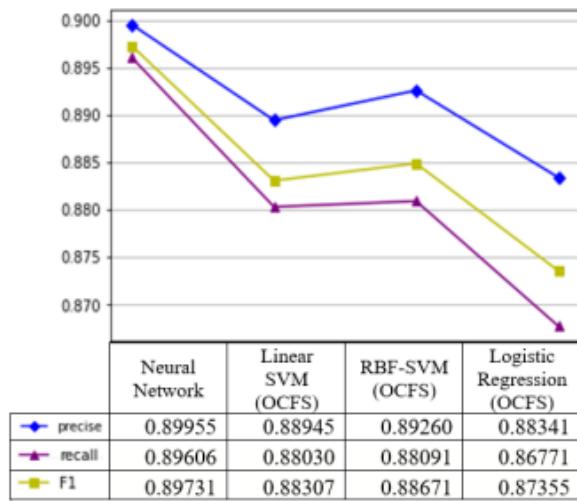
3. Rezultati sa kombinacijom BoW i N-gram modela



Slika 12: Evaluacija modela sa kombinacijom BoW i N-grama

- Najbolji rezultat je postignut sa kombinacijom BoW i bigrama (0.88279).

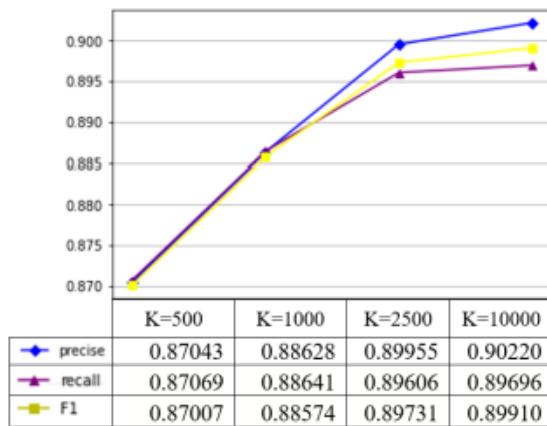
4. Rezultati u odnosu na druge klasifikatore



Slika 13: Komparacija neuronskih mreža sa ostalim klasifikatorima

- Najbolji rezultati su postignuti sa NN modelom autora (0.89731).

5. Rezultati sa različitim veličinama TF-IDF vektora



Slika 14: Komparacija performansi NN modela sa različitim veličinama ulaznog vektora

- Najbolji rezultat je postignut sa punim vektorom K=10000 (0.89910).

POTENCIJALNI PRAVCI ZA POBOLJŠANJE

1. Rješavanje neuravnoteženosti klasa

Na slici 6 možemo da vidimo da postoji varijacija u broju članaka po kategorijama. Kako bi se riješila neuravnoteženost moglo bi se primijeniti ponderiranje funkcije gubitka da bi se uvećala važnost manjih klasa ili resampliranje kako bi se povećao broj primjeraka u manjim klasama. Na ovaj način model bi bolje bio naučen i mogao bi efikasnije i tačnije da klasificuje članke.

2. Automatska optimizacija arhitekture

Uvjeto korištenja fiksne arhitekture i testiranja različitih kombinacija sa brojem slojeva, neurona i slično. Može se koristiti Keras Tuner koji vrši automatsku optimizaciju hiperparametara modela. On vrši testiranje različitih kombinacija i pronađi najbolji model za taj dataset i problem.

Treći izvor: Word2vec convolutional neural networks for classification of news articles and tweets [7]

Vršimo analizu naučnog rada “Word2vec convolutional neural networks for classification of new articles and tweets” od autora Beakcheol Jang, Inhwan Kim i Jong Wook Kim.

Data	Total
Collection Data	March 1, 2018 – May 1, 2018 (62 days)
News articles	122,258
News sequences	2,445,160
News words	160,208,160
Tweets	291,309
Twitter sequences	5,826,160
Twitter words	188,155,940

Slika 15: Prikupljeni podaci za dataset

OPSEG PROBLEMA

Ovaj naučni rad se bavi klasifikacijom online novinskih članaka i tvitova na relevantne i irelevantne. Oni često sadrže bespotrebni tekst kao što su reklame i nepotrebne riječi, a

ovo utiče na model koji uči. Iako je u pitanju binarna klasifikacija, iste tehnike se mogu iskoristiti i na klasifikaciji za više kategorija.

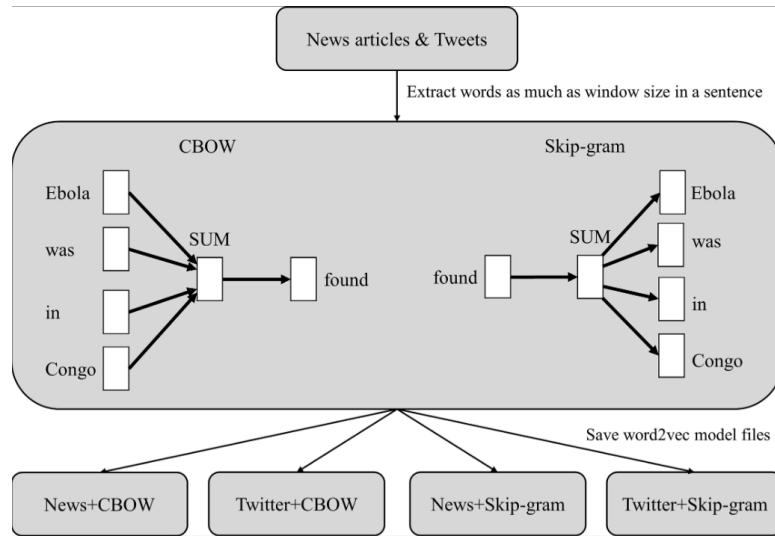
KORIŠTENE METODE VJEŠTAČKE INTELIGENCIJE

1. Predobrada teksta (*Text Preprocessing*) - tehnike obrade prirodnog jezika NLP

- Kako je korišteno u članku? Autori navode da su uklonili URL adrese, HTML tagove. Pored toga korišten je Sift4 algoritam za ekstrakciju jedinstvenih članaka i tvitova. Tekst je podijeljen na individualne riječi koristeći Open Korean Text Processor Java (NLP alat za korejski jezik).

2. Reprezentacija riječi (*Word Embeddings* pomoću *Word2vec*) - NLP Learning tehnika

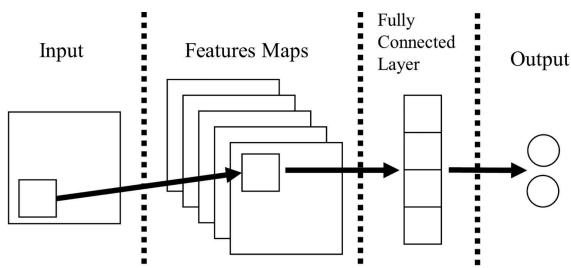
- Word embeddings (ugrađivanje riječi) su tehnike koje mapiraju riječi iz rječnika u vektore realnih brojeva, pri čemu riječi sa sličnim značenjem imaju slične vektore. Word2vec je model koji može da koristi 2 algoritma: Continuous Bag-Of-Words (CBOW) - predviđa trenutnu riječ na osnovu susjednih riječi, Skip-gram - predviđa susjedne riječi na osnovu trenutne.
- Kako je korišteno u članku? Autori su primijenili Word2vec za generisanje vektorskih reprezentacija riječi, koje su onda korištene kao ulazni sloj za Konvolucijsku Neuronsku Mrežu (CNN) zaduženu za klasifikaciju. Trenirana su oba glavna Word2vec algoritma - CBOW i Skip-gram - na datasetu. Na taj način su kreirana 4 seta predtreniranih Word2vec modela, po jedan CBOW i Skip-gram model za svaki tip podataka (članci i tvitovi).



Slika 16: Proces pripreme Word2vec embeddinga, od prikupljanja podataka do generisanja finalnih modela

3. Konvolucionne Neuronske Mreže (CNN) – Deep Learning tehnika

- CNN efikasno klasifikuju tekst tako što u njemu traži ključne obrasce, slično kao što traži oblike na slici. Prvo se svaka riječ sa 'embedding' slojem pretvara u svoj numerički vektor. Zatim, konvolucijski filteri 'klize' preko tih vektora kako bi detektovali važne lokalne fraze. Pooling sloj izdvaja najbitnije prepoznate obrasce i na kraju potpuno povezani slojevi analiziraju te fraze i donose odluku o klasifikaciji. Izlazni sloj daje verovatnoću za svaku kategoriju i omogućava modelu da odabere najverovatniju.
- Kako je korišteno u članku?



Slika 17: Opći CNN model

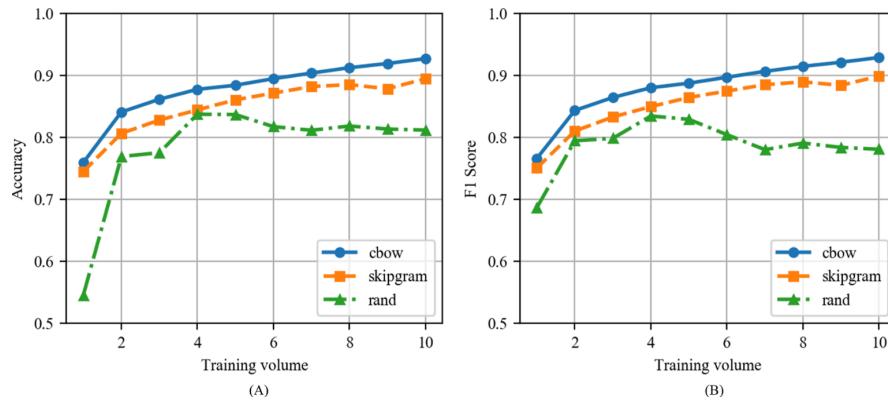
Parameter	Detail / Value
Training volume	1K – 10K
Test volume	20K
Classification	Positive, Negative
Batch size	500
Epochs	1-200
Feature maps	100

Slika 18: Korišteni parametri za CNN

POSTIGNUTI REZULTATI

Autori su proveli eksperimente kako bi ispitali efikasnost CNN u kombinaciji sa različitim Word2vec modelima. Cilj im je bio da utvrde doprinos Word2vec embeddinga i uporede CBOW i Skip-gram algoritam sa nasumično inicijaliziranim vektorima riječi.

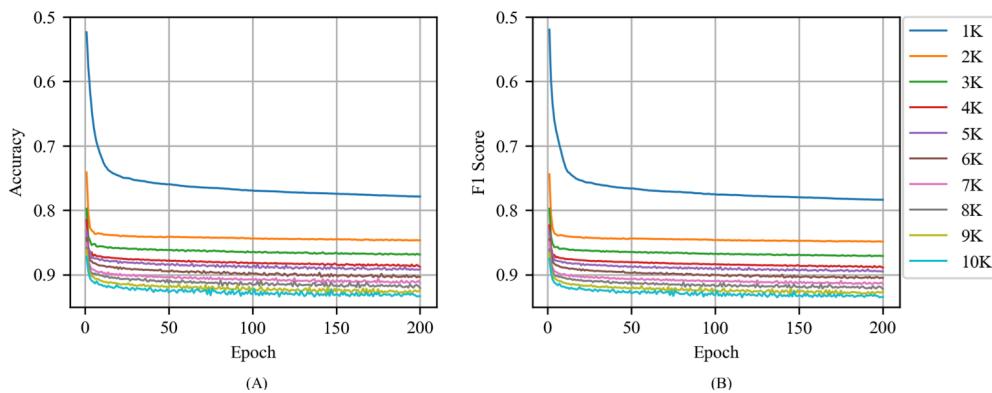
1) Usporedba Word2vec embeddinga sa nasumičnom inicijalizacijom



Slika 19: Tačnost i F1-score sa 3 različita CNN modela - s CBOW, Skip-gram i nasumično inicijaliziranim vektorom

- Korištenje Word2vec embeddinga znatno poboljšava performanse

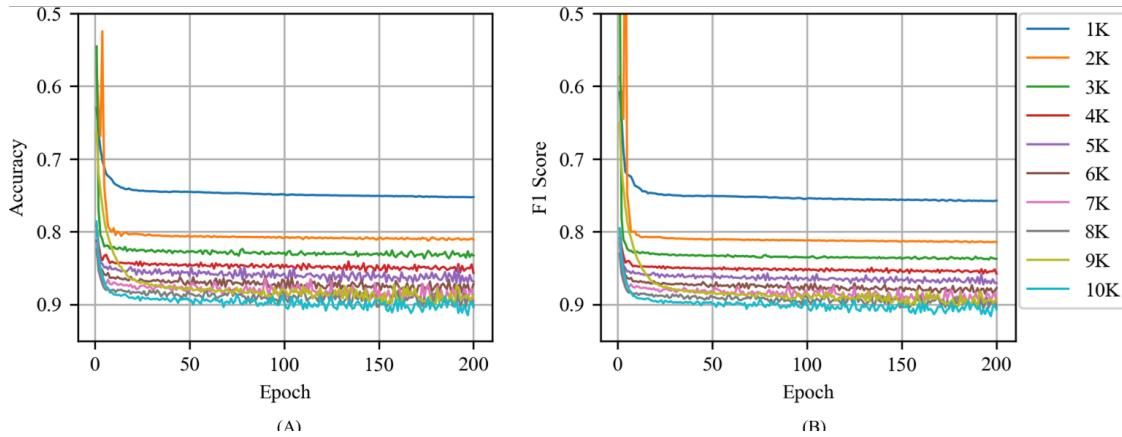
2) Performanse CNN modela sa CBOW embeddingom



Slika 20: Tačnost i F1-score s porastom broja epoha (1-200) za različite veličine trening skupa (1000-10000) uz CBOW

- Uočava se rast performansi s povećanjem broja epoha i veličine trening skupa. Na primjer, s trening skupom od 3000 članaka, model postiže F1-score od približno 0.85 već nakon desetak epoha, dok s većim trening skupovima i većim brojem epoha, F1-score prelazi 0.93.

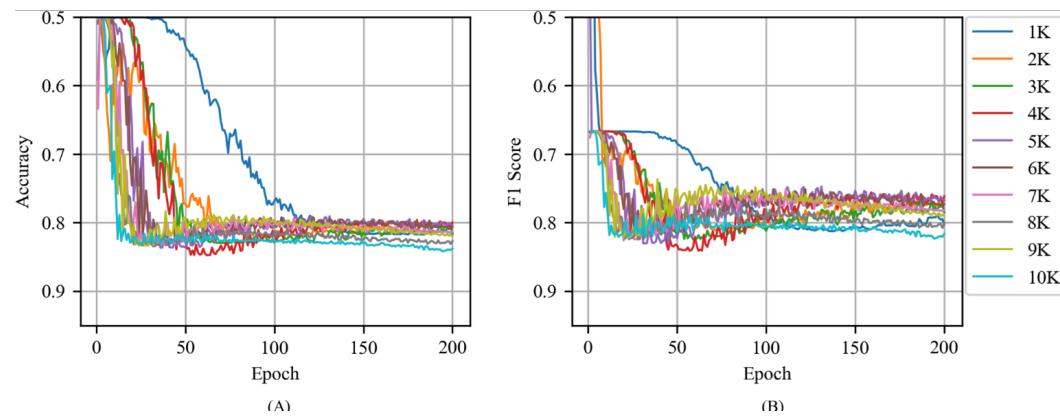
3) Performanse CNN modela sa Skip-gram embeddingom



Slika 21: Tačnost i F1-score s porastom broja epoha (1-200) za različite veličine trening skupa (1000-10000) uz Skip-gram

- I ovaj model pokazuje poboljšanje performansi s povećanjem oba parametra, ali su performanse niže u odnosu na CBOW model. Na primjer, da bi se postigao F1-score od 0.85, Skip-gram model zahtijeva veći trening skup (preko 5000 članaka) ili veći broj epoha. Također, imaju veće fluktuacije u performansama, posebno pri manjim veličinama trening skupa.

4) Performanse CNN modela sa nasumično inicijaliziranim vektorima



Slika 22: Tačnost i F1-score s porastom broja epoha (1-200) za različite veličine trening skupa (1000-10000) uz nasumično inicijalizirane vektore

- Model postiže znatno nižu tačnost i F1-score u odnosu na modele koji koriste Word2vec. Kod ovog modela, nakon određenog broja epoha, dolazi do pada performansi, što može da ukazuje na overfitting.

POTENCIJALNI PRAVCI ZA POBOLJŠANJE

1. Korištenje alternativnih Word embedding modela

Autori su u radu koristili Word2vec model i njegove 2 poznate metode: CBOW i Skip-gram. Mogla bi se ostvariti poboljšanja ukoliko bi se koristili GloVe ili FastText model. GloVe je zasnovan na globalnoj statistici, a FastText uči vektore za n-grame. Glavna prednost FastTexta je što može da generiše vektore i za nepoznate riječi. Na ovaj način bi se moglo dobiti bolje vektorske reprezentacije i time povećati tačnost CNN modela.

2. Proširenje na višekategorisku klasifikaciju

Mogao bi da se proširi model na višekategorisku klasifikaciju, što bi predstavljalo unapređenje i učinilo ovaj model primjenjivijim na širi spektar problema. Ovo bi se moglo izvesti sa modifikacijom izlaznog sloja CNN-a da koristi softmax funkciju s brojem izlaza jednakom broju kategorija.

3. IZBOR, ANALIZA I PRETPROCESIRANJE DATASET-A

3.1. IZBOR I ANALIZA DATASET-A

Za potrebe ovog projekta, odabran je dataset “**News Article Category Dataset**” [8] koji sadrži članke sa portala HuffPost.

Osnovni pregled:

- Izvor skupa podataka: platforma Kaggle
- Format i način preuzimanja: CSV, direktno preuzimanje sa platforme bez pravnih ograničenja
- Broj instanci: 6877 jedinstvenih novinskih članaka
- Atributi: category, title, body
- Klase: Ima 14 klasa - arts and culture, business, comedy, crime, education, entertainment, environment, media, politics, religion, science, sports, tech, women
- Broj instanci po klasama:

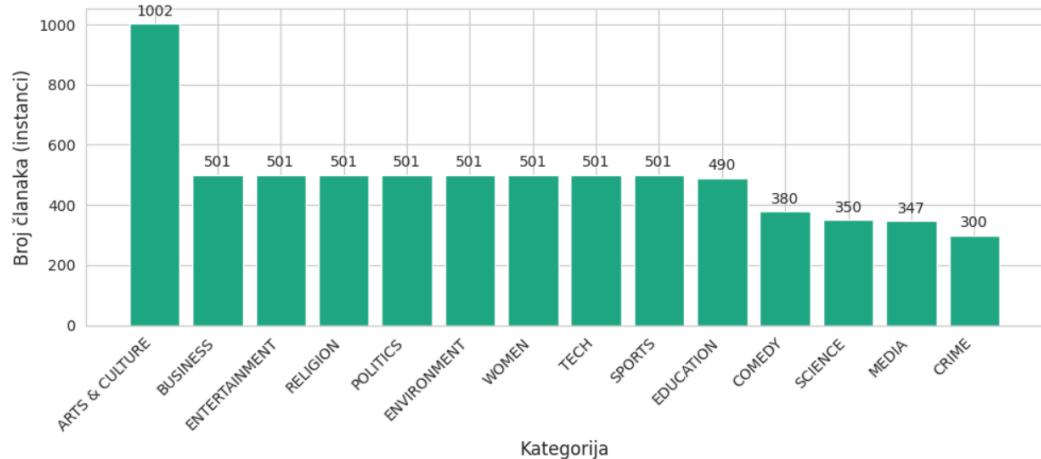
category	Broj instanci	Procenat (%)
ARTS & CULTURE	1002	14.57
BUSINESS	501	7.29
ENTERTAINMENT	501	7.29
RELIGION	501	7.29
POLITICS	501	7.29
ENVIRONMENT	501	7.29
WOMEN	501	7.29
TECH	501	7.29
SPORTS	501	7.29
EDUCATION	490	7.13
COMEDY	380	5.53
SCIENCE	350	5.09
MEDIA	347	5.05
CRIME	300	4.36

Slika 23: Raspodjela instanci po klasama

- Količina podataka (MB): 26.87 MB

3.2. IDENTIFIKOVANI RIZICI I IZAZOVI

1. Imbalans klasa



Slika 24: Distribucija instanci po klasama

- Kao što vidimo na slici 24, dataset je neuravnotežen. Klasa ARTS & CULTURE je gotovo dvostruko zastupljenija od drugih klasa. A klase CRIME, MEDIA, SCIENCE i COMEDY su značajno manje zastupljene. Ovo je rizik jer postoji mogućnost da će biti veća tačnost pri klasifikaciji na zastupljenijoj klasi, a lošija na manjim.

2. Velik broj klasa

- S obzirom na broj klasa, problem je dosta kompleksniji od binarne klasifikacije ili klasifikacije sa par kategorija. Treba nam model koji može da nauči suptilne razlike između velikog broja tema.

3. Semantička sličnost i preklapanje klasa

- Riječi mogu imati različito značenje ovisno o kontekstu i moguće je da postoji preklapanje tema zbog nejasnih granica između kategorija. Npr. članak o politici u sportu može sadržavati rječnik iz obje kategorije (politika, sport). Također, media, entertainment i politics mogu često dijeliti ključne riječi. Ovo može dovesti do pogrešnih klasifikacija.

4. Prisustvo šuma i irelevantnih informacija

	ARTS & CULTURE	BUSINESS	COMEDY	CRIME	EDUCATION	ENTERTAINMENT	ENVIRONMENT	MEDIA	POLITICS	RELIGION	SCIENCE	SPORTS	TECH	WOMEN
0	the	the	the	the	the	the	the	the	the	the	the	the	the	the
1	of	to	a	a	to	to	to	to	to	of	of	to	to	to
2	and	and	to	to	and	a	and	and	of	to	to	and	a	and
3	a	a	of	and	of	and	of	a	a	and	a	a	and	of
4	to	of	s	of	a	of	a	of	and	a	and	of	of	a
5	in	in	and	in	in	in	in	that	in	in	in	in	in	in
6	s	that	in	s	that	s	that	in	s	that	that	s	s	that
7	that	s	on	that	for	i	s	s	that	is	s	that	that	i
8	is	for	that	was	is	that	for	i	for	i	is	for	it	s
9	i	is	i	on	s	it	is	on	on	s	it	on	for	it

Slika 25: Top 10 najčešćih riječi u svakoj kategoriji

- Na slici 25 vidimo da su najčešće riječi stop riječi poput the, a, in, to itd. Ove riječi ne nose nikakve korisne informacije o toj klasi i predstavljaju šum. Pored ovoga članci sa portala mogu da imaju fraze, imena autora i datume unutar teksta koji su irelevantni za klasifikaciju.

3.3. METODE I IMPLEMENTACIJA PRETPROCESIRANJA PODATAKA

Ovo je ključna faza koja transformiše sirove tekstualne podatke u čist i strukturiran format pogodan za treniranje modela mašinskog učenja. Ovim korakom se rješavaju mnogi rizici, koji su spomenuti. Glavni cilj je poboljšavanje kvaliteta podataka, smanjenje dimenzionalnosti i poboljšanje tačnosti modela.

METODE KOJE ĆE BITI KORIŠTENE

1. Spajanje dva polja (title i body)

- Zašto je potrebno? Naslov članka sažima samu suštinu teksta, a tijelo pruža detaljan kontekst. Ako se spoje ova dva polja u jedno polje, model će imati pristup svim bitnim informacijama za klasifikaciju u jednom polju.
- Implementacija: Kreirana je nova kolona full_text.

2. Čišćenje i normalizacija teksta

- Zašto je potrebno? Kao što je već navedeno u rizicima članci sa web portala često imaju elemente koji nemaju nikakvo semantičko značenje i predstavljaju šum. Uklanjanje ovih elemenata će poboljšati model, tako što će se on fokusirati samo na relevantan sadržaj.
- Implementacija:
 - a) Pretvaranje u mala slova: Svi karakteri su pretvoreni u mala slova, kako model ne bi tretirao istu riječ (različito zapisanu) kao dvije različite.
 - b) Uklanjanje specijalnih, interpunkcijskih znakova, URL-ova, email-ova i brojeva: Uklonjeni su svi karakteri koji nisu slova abecede. Na ovaj način se uklanjuju znakovi koji ne donose nikakav kontekst i informaciju.
 - c) Uklanjanje stop-riječi: Uklonjene su česte riječi koje smo vidjeli na slici 25, koje ne doprinose razlikovanju kategorija. Korištena je biblioteka NLTK kao lista engleskih stop-riječi.

3. Lematizacija

- Zašto je potrebno? Na ovaj način svodimo riječi na njihov lingvistički osnovni oblik - lemu. Ovako se grupišu različiti oblici iste riječi, smanjuje rječnik koji će se koristit i pomaže modelu da prepozna isto osnovno značenje riječi.
- Implementacija: Korišten je lematizer iz NLTK biblioteke.

	ARTS & CULTURE	BUSINESS	COMEDY	CRIME	EDUCATION	ENTERTAINMENT	ENVIRONMENT	MEDIA	POLITICS	RELIGION	SCIENCE	SPORTS	TECH	WOMEN
0	art	company	trump	police	school	go	climate	news	trump	people	year	game	company	woman
1	work	make	president	accord	student	make	year	trump	state	god	scientist	team	facebook	sexual
2	make	people	donald	report	education	know	animal	go	president	muslim	science	player	use	make
3	artist	work	check	shoot	teacher	people	change	people	people	church	study	year	make	men
4	people	year	clip	people	child	year	make	say	year	christian	people	sport	people	go
5	year	business	host	charge	make	movie	water	know	make	make	use	go	apple	year
6	woman	take	make	school	year	film	state	make	house	american	make	athlete	year	people
7	go	say	go	victim	state	see	energy	woman	say	religious	research	make	google	take
8	book	go	colbert	year	public	dont	people	time	campaign	trump	space	olympic	user	say
9	take	uber	late	found	learn	take	world	president	law	year	could	take	go	assault

Slika 26: Top 10 najčešćih riječi u svakoj kategoriji nakon preprocesiranja

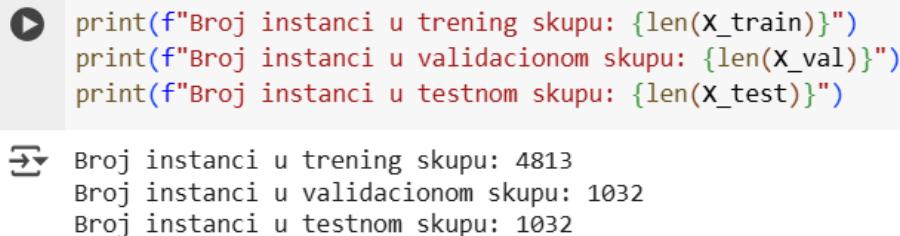
3.4. PODJELA DATASETA ZA TRENING, VALIDACIJU I TESTIRANJE

Dataset se dijeli na 3 dijela:

1. Skup za treniranje (70%) - Na ovom skupu model uči.
2. Skup za validaciju (15%) - Koristi se za podešavanje hiperparametara i praćenja da li dolazi do overfittinga prilikom treniranja.
3. Skup za testiranje (15%) - Na ovom skupu se provjeravaju performanse finalnog modela.

Zašto baš 70-15-15?

Jedan od stručnjaka za AI, Andrew Ng, u svom kursu za “**Machine Learning Design**” [9] navodi omjere poput 60/20/20 ili 70/15/15 za datasetove srednje veličine.



```
print(f"Broj instanci u trening skupu: {len(x_train)}")
print(f"Broj instanci u validacionom skupu: {len(x_val)}")
print(f"Broj instanci u testnom skupu: {len(x_test)}")
```

→ Broj instanci u trening skupu: 4813
Broj instanci u validacionom skupu: 1032
Broj instanci u testnom skupu: 1032

Slika 27: Broj instanci u svakom skupu

4. ODABIR, FORMATIRANJE, TRENING I TESTIRANJE MODELA

4.1. IZBOR METODE I TEHNOLOGIJE

Metoda: Neuronska mreža koja koristi TF-IDF vektorizaciju sa n-gramima.

- Odabran je ovaj pristup kao napredak u odnosu na metode opisane u analiziranim naučnim radovima. Odabrane su neuronske mreže kako bi se odmakli od tradicionalnih algoritama u prvom radu i kako bi model mogao da uči kompleksnije nelinearne obrasce. Vektorizacija je unaprijeđena sa TF-IDF sa bigramima, tako što je dodijeljen veći značaj relevantnim frazama. Iako je u trećem radu pokazana snaga

naprednih CNN arhitektura, pristup sa TF-IDF i jednostavnom neuronskom mrežom nudi bolji balans performansi i računarske zahtjevnosti.

Tehnologije:

1. Programski jezik: Python
2. Pandas: Učitavanje, analiza i manipulacija sa datasetom
3. NLTK (Natural Language Toolkit): Pretprocesiranje - lematizacija i uklanjanje stop riječi
4. Scikit-learn: Podjela dataseta na skupove, TF-IDF vektorizacija i generisanje matrice konfuzije
5. TensorFlow/Keras: Izgradnja, treniranje i evaluacija modela
6. Matplotlib/Seaborn: Vizualizacija podataka

4.2. PRIPREMA FORMATA PODATAKA DA ODGOVARA MODELU

1. Vektorizacija pomoću TF-IDF

Svaki članak je pretvoren u vektor fiksne dužine pomoću TfidfVectorizer. Korišteni su sljedeći parametri: ngram_range (1, 2) i max_features=10000. Prvi parametar omogućava da vektorizator kao karakteristike uzima i pojedinačne riječi i parove susjedih riječi. Drugi parametar ograničava veličinu rječnika na 10000 najvažnijih tokena na osnovu njihove TF-IDF vrijednosti.

2. One-Hot enkodiranje labela

Drugi korak je konvertovanje svake kategorije u binarni vektor čija je dužina jednak ukupnom broju klasa. Sve vrijednosti su 0, osim jedne koja je 1 na indeksu koji odgovara toj kategoriji. Korišten je LabelEncoder iz biblioteke Scikit-learn i to_categorical funkcija kako bi se onda pretvorili u one-hot format. Na ovaj način svaka labela je predstavljena kao vektor dužine 14, gdje je 1 na indeksu koji odgovara toj kategoriji.

4.3. ARHITEKTURA MODELA I OPTIMIZACIJA HIPERPARAMETARA

Za odabir arhitekture modela korišten je Keras Tuner. On automatski pronalazi najbolju kombinaciju hiperparametara koja maksimizira tačnost na validacijskom skupu.

Arhitektura modela za optimizaciju: Definisan je fleksibilni model čiji su parametri podvrgnuti optimizaciji.

1. Dense Layer: Broj neurona u ovom sloju se kretao od 32 do 256
2. Dropout Layer: Stopa regularizacije se kretala od 0.3 do 0.7

3. Optimizator: Testiran je Adam optimizator sa stopama učenja (0.01, 0.001, 0.0001)

Optimizacija: Korištena je Hyperband strategija pretrage, koja uklanja loše performanse u ranim fazama treniranja. Ciljna metrika je tačnost na validacijskom skupu. Pretraga je testirala različite kombinacije hiperparametara i na kraju je identifikovana najbolja konfiguracija za finalni model koji će se koristiti.

```
Trial 30 Complete [00h 00m 18s]
val_accuracy: 0.8149224519729614

Best val_accuracy So Far: 0.8352712988853455
Total elapsed time: 00h 12m 16s
--- Pretraga je završena ---

Pretraga je završena. Optimalni hiperparametri su:
- Broj neurona u gustom sloju: 128
- Dropout stopa: 0.6000000000000001
- Stopa učenja (Learning Rate): 0.01
```

Slika 28: Najbolji hiperparametri za model

Na osnovu ove pretrage formirana je arhitektura modela po ovim parametrima:

1. Input Layer: Ulazni oblik podataka, koji je u ovom slučaju vektor sa 10000 dobijen od TF-IDF.
2. Dense Layer: Ima 128 neurona i relu aktivacijsku funkciju.
3. Dropout Layer: Nasumično gasi 60% neurona iz prethodnog sloja tokom svake iteracije.
4. Output Layer: Sloj sa 14 neurona i softmax aktivacijskom funkcijom.

4.4 TRENIRANJE I TESTIRANJE FINALNOG MODELA

Finalni model je izgrađen na osnovu gore navedenih optimalnih hiperparametara. Treniranje je izvršeno na trening skupu kroz 30 epoha, sa veličinom uzorka 32. Korišten je mehanizam EarlyStopping koji je prekinuo treniranje nakon 5 epoha, jer se performanse na validacijskom skupu nisu dalje poboljšavale. Na ovaj način je spriječen overfitting.

Testiranje performansi je izvršeno na testnom skupu. Korištene su sljedeće metrike:

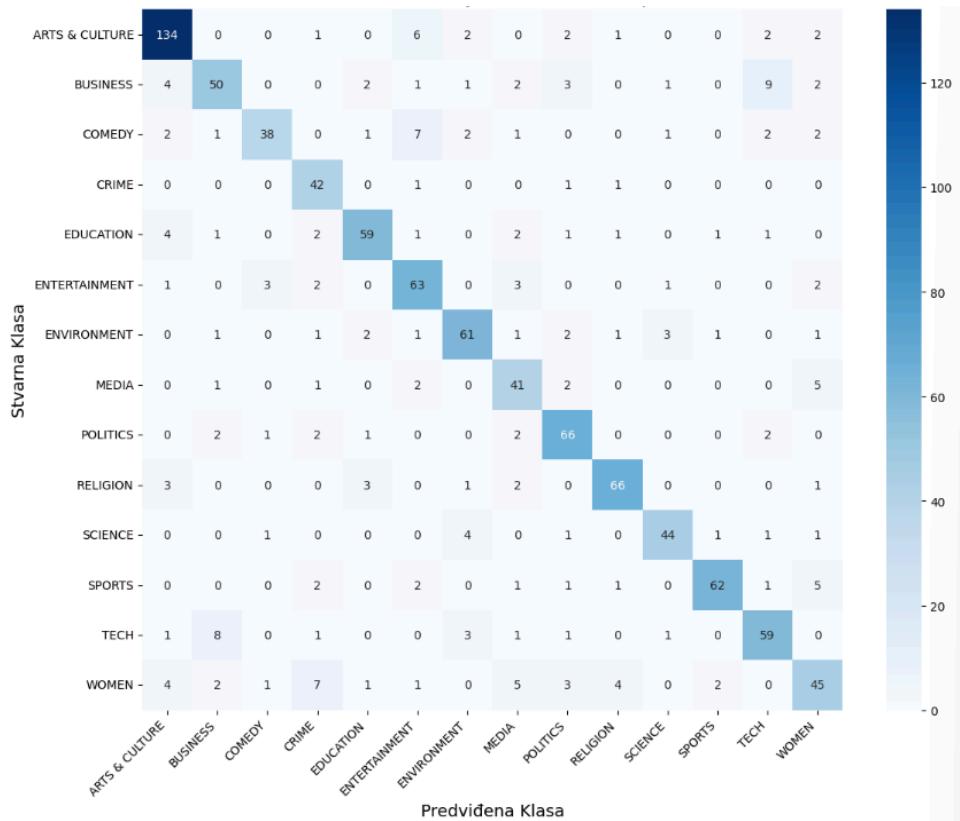
- Preciznost - pouzdanost modela da predvidi klasu (odnos ispravnih pozitivnih predikcija i ukupnog broja pozitivnih predikcija za kategoriju)
- Odziv - koliko dobro model pronalazi instance određene kategorije (odnos ispravnih pozitivnih predikcija i ukupnog broja stvarnih instanci te kategorije)

- F1-score - mjera performansi ((sredina između precisnosti i odziva)
- Tačnost - procenat ispravno klasifikovanih članaka
- Matrica konfuzije - prikaz performansi modela kroz broj tačnih i netačnih predikcija za svaku klasu. Na dijagonali su pravilno klasifikovane instance.

Rezultati evaluacije su prikazani na Slici 29 i matrici konfuzije na Slici 30.

	precision	recall	f1-score	support
ARTS & CULTURE	0.88	0.89	0.88	150
BUSINESS	0.76	0.67	0.71	75
COMEDY	0.86	0.67	0.75	57
CRIME	0.69	0.93	0.79	45
EDUCATION	0.86	0.81	0.83	73
ENTERTAINMENT	0.74	0.84	0.79	75
ENVIRONMENT	0.82	0.81	0.82	75
MEDIA	0.67	0.79	0.73	52
POLITICS	0.80	0.87	0.83	76
RELIGION	0.88	0.87	0.87	76
SCIENCE	0.86	0.83	0.85	53
SPORTS	0.93	0.83	0.87	75
TECH	0.77	0.79	0.78	75
WOMEN	0.68	0.60	0.64	75
accuracy			0.80	1032
macro avg	0.80	0.80	0.80	1032
weighted avg	0.81	0.80	0.80	1032

Slika 29: Izvještaj o klasifikaciji na testnom skupu



Slika 30: Matrica konfuzije na testnom skupu

DISKUSIJA RJEŠENJA I OSVRT NA RIZIKE

Model je postigao ukupnu tačnost od 80%, što je solidan rezultat s obzirom na veliki broj kategorija. Analiza metrike i matrice konfuzije potvrđuje rizike identifikovane u Fazi 3:

- Imbalans klasa:** Imbalans je uticao direktno na performanse. Klase sa najmanjim brojem instanci kao što su WOMEN, BUSINESS i MEDIA imaju F1-score od 0.64, 0.71 i 0.73 respektivno. Najzastupljenija klasa ARTS & CULTURE ima F1-score od 0.88. Zanimljivo je da CRIME klasa ima odziv od 0.93 što znači da uspešno identificiše gotovo sve kriminalističke članke, ali po cijenu niže preciznosti od 0.69 jer se drugi članci pogrešno identificišu kao CRIME. Analizom matrice konfuzije, možemo da uočimo da klasa CRIME ima 7 članaka pogrešno klasifikovanih kao WOMEN. Ovo predstavlja značajnu grešku s obzirom da CRIME ima samo 45 primjera u testnom skupu.
- Semantička sličnost:** Analizom matrice konfuzije možemo da vidimo da postoji dosta sličnosti među klasama. Primjer toga su klase BUSINESS i TECH, gdje je 9 članaka iz BUSINESS pogrešno klasifikovano kao TECH i 8 članaka iz TECH kao BUSINESS. Ovo ukazuje na preklapanje terminologije (npr riječi company, data,

market). Također postoji konfuzija između klasa kao što su COMEDY i ENTERTAINMENT, WOMEN i MEDIA.

4.5 TESTIRANJE MODELA NA NEPOZNATIM PODACIMA

Testirano je 5 članaka sa svjetskih portala BBC, CNN i CBS News. Svaki članak je prošao kroz isti proces preprocesiranja kao i trening podaci. Nakon toga su transformirani u TF-IDF vektor i proslijeđeni finalnom modelu.

Članak	Stvarna kategorija	Predviđena kategorija
Glasovi iz Irana nakon napada SAD	POLITICS	POLITICS
Otpuštanje novinara iz Glasa Amerike	BUSINESS	MEDIA
Navigacija moljaca pomoću zvijezda	SCIENCE	SCIENCE
Muzički video pjevača Benson Boone-a	ENTERTAINMENT	ENTERTAINMENT
Pucnjava na vjenčanju u Francuskoj	CRIME	CRIME

Ispravno je klasifikovano 4 od 5 novih članaka, što pokazuje da je model ispravno naučio ključne karakteristike za većinu kategorija. Jedina netačna kategorija je kod drugog članka. Iako je članak označen kao BUSINESS na platformi, on sadrži dosta riječi vezanih za medije i politiku ("journalists", "news outlet", "Trump's administration"). Model je predvidio da je MEDIA. Ovo je razumljiva greška i primjer semantičkog preklapanja koje je identifikovano kao rizik. Ovo ukazuje na to da za neke kategorije ne postoje jasne granice u stvarnom svijetu. Sveukupno, može se reći da je model robustan i praktično primjenjiv.

5. CJELOKUPNI OSVRT NA PROBLEM I DOBIJENO RJEŠENJE

U nastavku ćemo se osvrnuti na postignute rezultate, uporediti ih sa radovima iz druge faze, te diskutovati o potencijalnim poboljšanjima.

5.1 OSVRT NA REZULTATE

Cilj projekta je bio razviti model za klasifikaciju novinskih članaka. Korištene su neuronske mreže i TF-IDF vektorizacija sa bigramima, čime je postignuta ukupna tačnost od 80%. Ovaj rezultat je solidan i zadovoljavajući s obzirom na već spomenute rizike.

Problem je dosta kompleksniji od binarne klasifikacije ili klasifikacije sa par kategorija. S obzirom da imamo 14 kategorija, model bi trebao da nauči najmanje razlike među kategorijama. Nakon izvršene evaluacije, vidjeli smo da imbalans klasa ima velik uticaj. Klasa sa najviše instanci je postigla F1-score od 0.88, dok su klase sa najmanjim brojem instanci postigle znatno niži koji se kreće do 0.64. Na osnovu ovoga vidimo da je model imao probleme sa klasifikacijom manje zastupljenih kategorija.

Matrica konfuzije je pokazala postojanje semantičkog preklapanja između kategorija. Ovo je i logično pošto imamo dosta kategorija sa sličnim vokabularom. Ova saznanja su dalje potvrđena prilikom testiranja modela na novim podacima, gdje je pogrešno klasifikovan BUSINESS u MEDIA.

5.2 USPOREDBA SA REFERENTNIM RADOVIMA

USPOREDBA SA RADOM HEIDI MANAI

U ovom radu su korišteni tradicionalni algoritmi kao što su Random Forest, Logistic Regression i postignuta ukupna tačnost od 98% na BBC News datasetu. Ovaj rezultat je superioran u odnosu na naš, ali je bitno uzeti u obzir da ovaj dataset ima samo 5 kategorija koje su jasnije razdvojene: tech, business, politics, sport i entertainment. Naš dataset je predstavljao veći izazov i baš zbog toga su odabrane neuronske mreže, kako bi model mogao naučiti kompleksnije nelinearne obrasce.

USPOREDBA SA RADOM “VIETNAMESE NEWS CLASSIFICATION”

Ovaj rad je najsličnjem našem s obzirom da koristi TF-IDF i neuronsku mrežu. Postigli su F1-score od 89% na datasetu sa 30 kategorija. Ovakav rezultat se može pripisati većem datasetu i kompleksnijoj arhitekturi mreže sa 3 skrivena sloja. Ovo ukazuje na to da bi i naš model potencijalno mogao postići slične ili bolje rezultate sa više članaka i dubljom arhitekturom.

USPOREDBA SA RADOM “WORD2VEC CONVOLUTIONAL NEURAL NETWORKS”

U ovom radu su korištene kompleksnije tehnologije poput Word2vec embeddinga i konvolucionih neuronskih mreža. Postignut je F1-score od 93%. Ovakvi modeli su dosta napredniji jer Word2vec čuva semantički kontekst riječi, a CNN prepoznaje ključne fraze. Model zasnovan na TF-IDF nije u stanju da uhvati složene jezičke odnose i samim tim ovo ukazuje na jedan od pravaca za buduća poboljšanja.

Rezultat od 80% tačnosti je konkurentan i opravdan s obzirom na složenost problema i korištenu metodologiju. Naš model čini odličnu i efikasnu osnovu.

5.3 DISKUSIJA O POTENCIJALNIM POBOLJŠANJIMA

Identifikovano je par pravaca za unapređenje finalnog modela:

1. **Rješavanje imbalansa klasa:** Moglo bi se primijeniti tehnike kao što su ponderisanje klasa ili tehnike resamplinga. Za ponderisanje bi se dodjeljivale veće težine manje zastupljenim klasama tokom treniranja sa `class_weight` parametrom. Prilikom resamplinga bi se mogla koristiti tehnika SMOTE, prilikom koje bi se generisali sintetički podaci za manjinske klase. Bilo koja od ove dvije tehnike bi pomogle pri balansiranju dataseta.
2. **Naprednija vektorizacija:** Za napredniju vektorizaciju bi se mogao koristiti word embedding. Ova tehnika bi predstavljala riječi kao vektore u višedimenzionalnom prostoru, pri čemu bi riječi sa sličnim semantičkim značenjem imale slične vektore. Ovako bi fokus bio ne samo na frekvenciju riječi, već i na njihov kontekst.
3. **Bolja arhitektura modela:** Arhitektura modela bi se mogla poboljšati sa dubljim slojevima ili sa korištenjem drugih arhitektura kao što su CNN ili rekurentne mreže. CNN bi potencijalno efikasnije prepoznavao ključne fraze, a rekurentne mreže bi razumijevale sekvensijalne zavisnost kod dužih rečenica i pasusa.
4. **Optimizacija hiperparametara:** Korišteni Keras Tuner bi se mogao poboljšati tako što bi se testirao sa većim brojem skrivenih slojeva, aktivacijskih funkcija i raspona za stopu učenja i dropout.

6. IZRADA WEB APLIKACIJE UZ ANVIL FRAMEWORK

Za što jednostavniju upotrebu ovog modela razvijena je aplikacija uz Anvil Framework [10]. Izabran je zbog svoje jednostavnosti pri dizajnu korisničkog interfejsa, laganoj integraciji sa Google Collab Notebook-om, te podrške za kompleksne Python biblioteke.

ARHITEKTURA APLIKACIJE

Arhitektura ove aplikacije je prvobitno bila napravljena tako da je čitava aplikacija, uključujući i backend logiku sa Tensorflow modelom izvršavala na Anvil serverima. Na početku je ovo radilo bez problema i omogućavalo dostupnost aplikacije 24/7.

Međutim, tokom dorada se pojavio tehnički izazov. Zbog promjena u Anvil okruženju, veličina paketa koji su korišteni je premašio memorijski limit od 500 MB. Ovaj limit je dostupan na besplatnom planu i zbog ovog je došlo do greške `Image size limit exceeded 500 MB`. Nismo više mogli da pokrenemo server.

Package	Version
joblib	1.5.1 ×
nltk	3.9b1 ×
scikit-learn	1.7.0rc1 ×
gdown	5.2.0 ×
tensorflow	2.15.0 ×

[Add package...](#)
Version (leave blank for latest)

[Edit requirements.txt directly](#)

Status: Error – Image size limit of 500 MB exceeded 

Slika 31: Greška u Anvil Frameworku sa paketima

Za rješenje ovog problema, koristio se isti model ali je arhitektura aplikacije promijenjena tako da se oslanjala na Anvil Uplink.

Aplikacije je podijeljena na sljedeći način:

1. Frontend - i dalje se izvršava na Anvil serverima
2. Backend - dio koji se sada izvršava u Google Colab Notebooku
3. Anvil Uplink - most između frontenda na Anvilu i backenda u Colabu

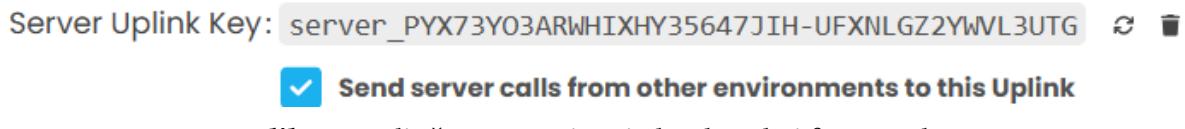
6.1 FRONTEND

Frontend se sastoji od područja za unos teksta članka (textarea), dugmeta za pokretanje procesa klasifikacije (button) i oznake u kojoj se prikaže finalni rezultat (label).

Svi ovi elementi su dodani pomoću Drag and Drop. Logika vezana za njih je napisana u Client Code dijelu, a CSS stilovi su napisani u theme.css. Postavljen je listener na dugme, pri čemu se aktivira metoda klasifikuj_button_click. Ona uzima tekst koji je unesen, poziva funkciju klasifikuj_clanak sa backenda u Colabu pomoću Uplinka i čeka da se završi obrada. Nakon toga se rezultat prikazuje u rezultat_label.

6.2 BACKEND

Backend je čitav smješten u Collab-u. U svesci se sve čelije izvrše kako bi se do bile sve potrebne funkcije, objekti kao što su tfidf_vectorizer i label_encoder te finalni model za predikciju. Na kraju je dodan kod koji uspostavlja vezu sa Anvil aplikacijom pomoću Server Uplink ključa i funkcija klasifikuj_clanak dostupna za frontend.



Slika 32: Ključ za povezivanje backenda i frontenda

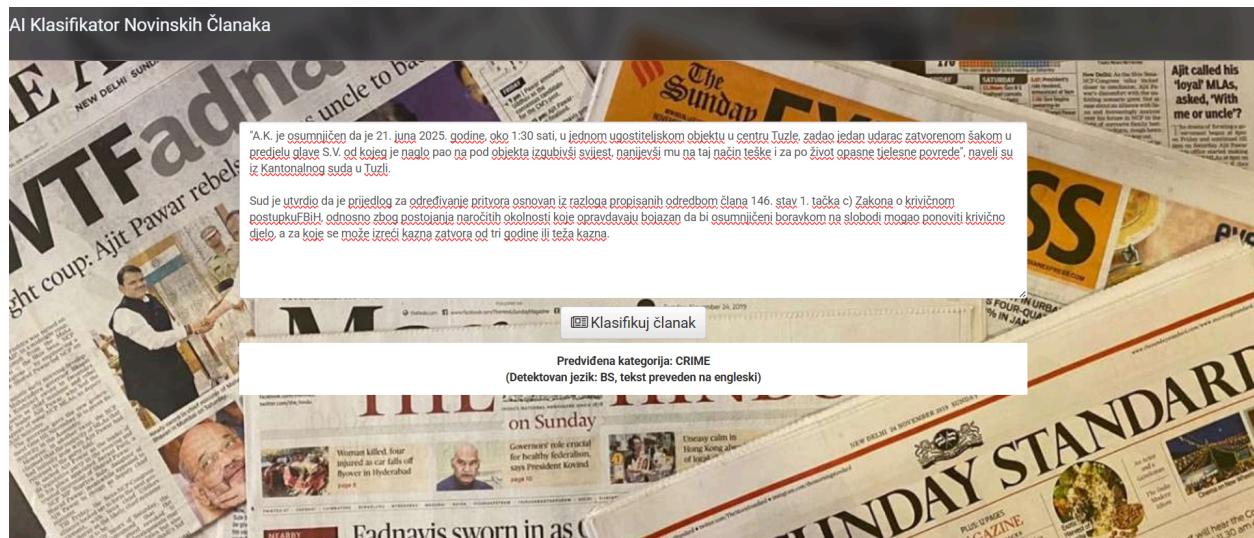
Ova funkcija ima sljedeći niz koraka. Prima tekst preko frontenda i u slučaju da nije bio engleski jezik, vrši se prevod. Nakon toga se proslijeđuje na pretprocesiranje. Taj tekst se transformiše u TF-IDF vektor uz pomoć vectorizer-a. Ovaj vektor se proslijeđuje u finalni model, pri čemu se koristi numpy.argmax() kako bi se pronašla kategorija sa najvećom vjerovatnoćom. Na kraju učitani label encoder pretvara indeks u naziv kategorije, te vraća ovaj rezultat na frontend.

U skripti je također dodan anvil.server.wait_forever() koji drži ovu sesiju aktivnom i spremnom za nove zahtjeve sa frontend-a.



Slika 33: Izgled web aplikacije za klasifikaciju novina

Čitavi tok aplikacije je sljedeći. Korisnik pokrene sve ćelije u Google Colabu, zatim otvoriti aplikaciju "**AI Klasifikator Novinskih Članaka**" [11] putem odgovarajućeg linka. Korisnik unosi naslov i tekst članka u predviđeno polje (može biti članak na bilo kojem jeziku), pritisne dugme "Klasifikuj članak". Čitava frontend i backend logika se izvrši, te na izlazu dobije predviđenu kategoriju i detektovani jezik ukoliko nije engleski.



Slika 34: Primjer rezultata za vijest na bosanskom jeziku

7. REFERENCE

- [1] "Dutch News Articles Dataset" Dostupno na:
<https://www.kaggle.com/datasets/maxscheijen/dutch-news-articles>
- [2] "News Articles Dataset" Dostupno na:
<https://www.kaggle.com/datasets/asad1m9a9h6mood/news-articles>
- [3] "Turkish News Articles Dataset" Dostupno na:
<https://www.kaggle.com/datasets/dilemre/turkish-news-article>
- [4] "Machine Learning : Text Classification of News Articles" Dostupno na:
<https://hedimanai.medium.com/machine-learning-text-classification-of-news-articles-bd5d70473037>
- [5] "BBC News Classification Dataset" Dostupno na:
<https://www.kaggle.com/c/learn-ai-bbc/data>
- [6] "Vietnamese News Articles Classification Using Neural Networks" Dostupno na:
<https://www.jait.us/uploadfile/2021/1029/20211029035100618.pdf>
- [7] "Word2vec convolutional neural networks for classification of news articles and tweets" Dostupno na: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0220976>
- [8] "News Article Category Dataset" Dostupno na:
<https://www.kaggle.com/datasets/timilsinabimal/newsarticlecategories>
- [9] "Machine Learning Design" Dostupno na:
<https://medium.com/data-science/how-to-efficiently-design-machine-learning-system-caef9e05d5fb>
- [10] "Anvil Framework" Dostupno na:
<https://anvil.works/learn/tutorials/google-colab-to-web-app>
- [11] "AI Klasifikator Novinskih Članaka aplikacija" Dostupno na:
<https://elementary-spirited-sandgrouse.anvil.app/>