Univerzitet u Banjoj Luci

Prirodno-matematički fakultet

PROJEKTNI ZADATAK

Predmet: Uvod u vještačku inteligenciju

Tema: Spam detection using a multi-layer perceptron

Profesor:
Doc. Dr Marko Đukanović

Student:
Jelena Komljenović

SADRŽAJ

1 Uvod

U savremenom digitalnom okruženju, elektronska pošta (e-mail) je postala nezaobilazno sredstvo komunikacije, međutim, sa porastom važnosti e-mail pošte, dolazili su i izazovi u obliku neželjenih informacija, poznatih kao "spam" e-mail poruke. Ovaj projekat ima za cilj da istraži razvoj i evaluaciju neuronske mreže kao efikasnog alata za detekciju spam e-mail poruka, kako bi se unaprijedio kvalitet elektronske komunikacije. Projekat je fokusiran na analizu teksta unutar e-mail poruka i koristi dva ključna atributa za proučavanje: sadržaj poruke i oznaku, koja ukazuje da li se radi o spam-u ili ne. Kroz temeljno istraživanje i eksperimentalni pristup biće razvijen model, koji će precizno klasifikovati e-mail poruke u dvije kategorije.

2 Opis problema

Na raspolaganju imamo dvije baze e-mail poruka koje se sastoje od dvije kolone i koje se razlikuju po balansiranosti instanci između klasa.
Prva kolona, pod nazivom "email" sadrži tekstualni sadržaj samih poruka koje bi trebalo obraditi, dok druga kolona, pod nazivom "spam" sadrži binarne vrijednosti. Ukoliko je poruka klasifikovana kao spam, binarna vrijednost će biti 1, a ukoliko nije onda 0. Osnovni cilj jeste razviti sofisticiran model neuronske mreže koji će kroz slojevitu arhitekturu, efikasno i precizno ocijeniti da li je određena elektronska poruka spam ili nije.

3 Način implementacije

Kako na raspolaganju imamo skup podataka koji sadrži e-mail poruke, prvi korak jeste njihova priprema za dalju obradu. To uključuje čišćenje i pretprocesiranje podataka kako bi bili u pogodnom formatu za samu analizu.
Nakon toga, pristupamo obradi samog sadržaja poruka, i za to koristimo TF-IDF (*engl. Term Frequency-Inverse Document Frequency*) matrice, koje će nam pomoći da izdvojimo sve riječi koje se pojavljuju u e-mail porukama. Ove riječi će

formirati kolone TF-IDF matrice, omogućavajući nam da predstavimo e-mail poruke kao vektore u prostoru riječi.
Nakon što smo uspješno pripremili podatke, sljedeći korak je podjela skupa podataka na dvije grupe: trening (*engl. train*) i test (*engl. test*) podatke. Ovo je važno kako bismo mogli da treniramo model na jednom skupu podataka, a zatim pravilno testiramo na drugom. Zatim dolazimo do kreiranja višeslojne neuronske mreže, koja će analizirati pripremljene podatke i naučiti kako da klasifikuje e-mail poruke na one koju su spam i na one koje nisu. Arhitektura mreže, broj slojeva i parametri će biti pažljivo odabrani kako bismo postigli najbolje rezultate.
Na samom kraju, kako bismo procijenili kvalitet modela, koristićemo različite metrike kao što su: matrica konfuzije, tačnost (preciznost) i druge relevantne evaluacione metrike. Ovo će nam pomoći da ocijenimo kako se model nosi sa zadatkom detekcije spam poruka i koliko je efikasan u razvrstavanju e-mailova.

3.1 Pretprocesiranje i transformacija ulaznih podataka

Nakon što smo učitali ulazne podatke, vršimo nekoliko ključnih koraka u fazi pretprocesiranja. Prvo, sprovodimo eliminaciju duplikata, koja nam omogućava da radimo sa jedinstvenim setom podataka, što smanjuje potencijalno nepotrebno ponavljanje.
Zatim provjeravamo da li postoje null vrijednosti u podacima. Ukoliko postoje bilo kakvi nepotpuni podaci, oni se brišu kako bismo osigurali kvalitet i tačnost daljih analiza.
Podatke zatim dijelimo u dvije grupe: "X" grupa - sadrži podatke iz kolone "email", odnosno tekstualni sadržaj e-mail poruka i "Y" grupa - sadrži podatke iz kolone "spam", tj. informacije o tome da li je e-mail poruka spam (označeno sa 1) ili nije (označeno sa 0).
Ova podjela je značajna, jer nam omogućava da dalje radimo sa podacima za treniranje modela. Grupa "X" će predstavljati ulazne podatke, dok će grupa "Y" biti ciljna promjenljiva na osnovu koje će model učiti i vršiti predviđanja.

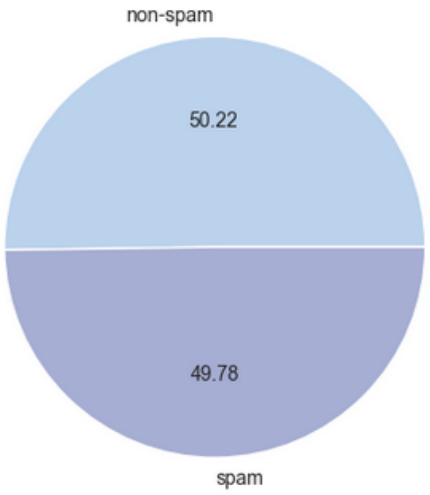
	email	spam
0	Subject: naturally irresistible your corporate...	1
1	Subject: the stock trading gunslinger fanny i...	1
2	Subject: unbelievable new homes made easy im ...	1
3	Subject: 4 color printing special request add...	1
4	Subject: do not have money , get software cds ...	1

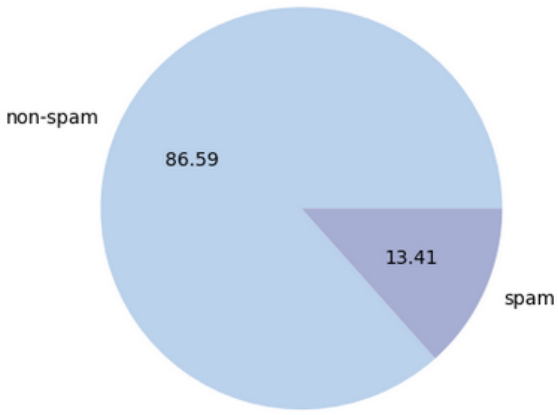
Slika 1. Prikaz prvog seta podataka

	spam	email
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Slika 2. Prikaz drugog seta podataka

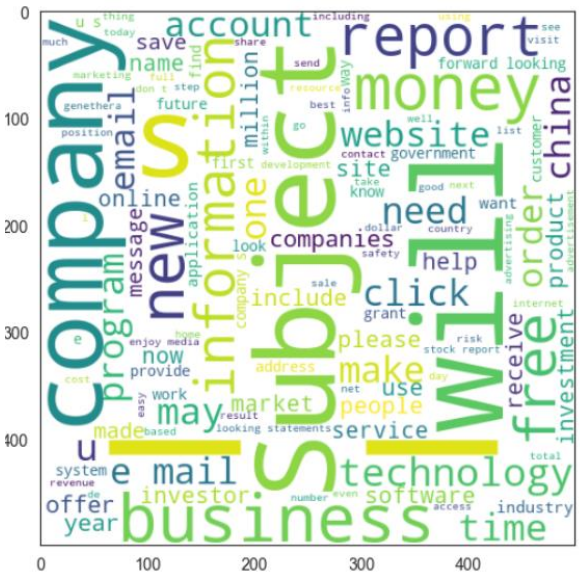
Ono što nam još može biti korisno da bismo bolje razumjeli skup podataka i identifikovali uzorke jeste vizualizacija podataka. Pogledajmo prvo odnos između poruka koje su spam i onih koje nisu, prvo za balansirani skup podataka, a onda i za nebalansirani:



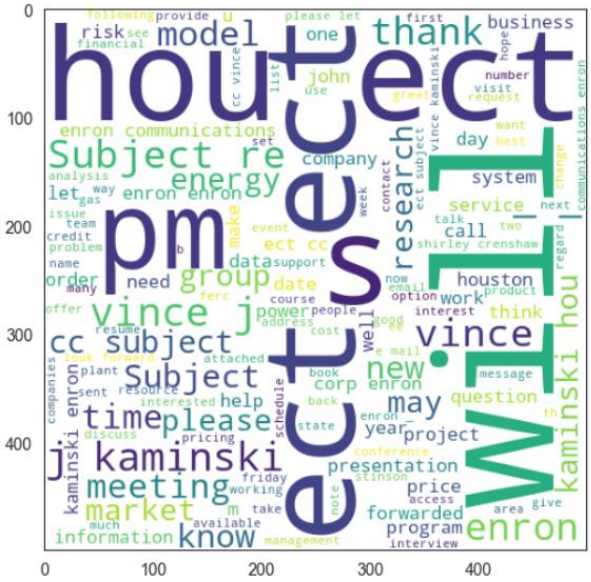


Zatim prikažimo frekvenciju riječi u skupu podataka pomoću word cloud vizualizacije. To je vizualizacija u kojoj se riječi koje se pojavljuju češće prikazuju većim fontom, dok se riječi koje se pojavljuju rjeđe prikazuju manjim fontom.

Vizualizacija balansiranog skupa podataka:

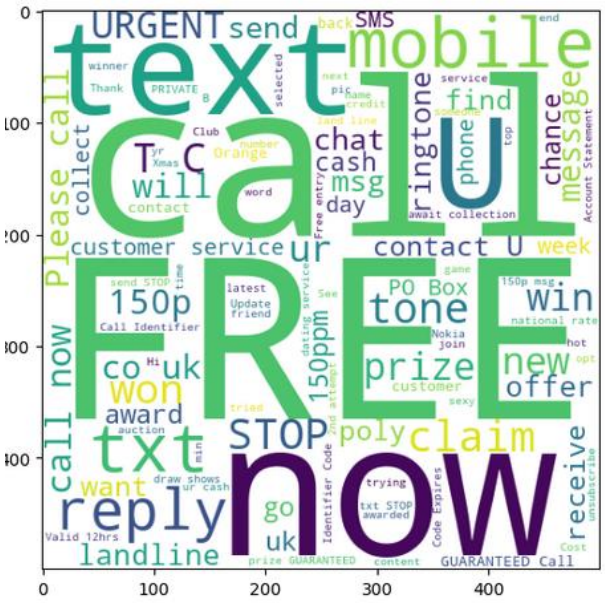


Slika 3. Vizualizacija spam poruka

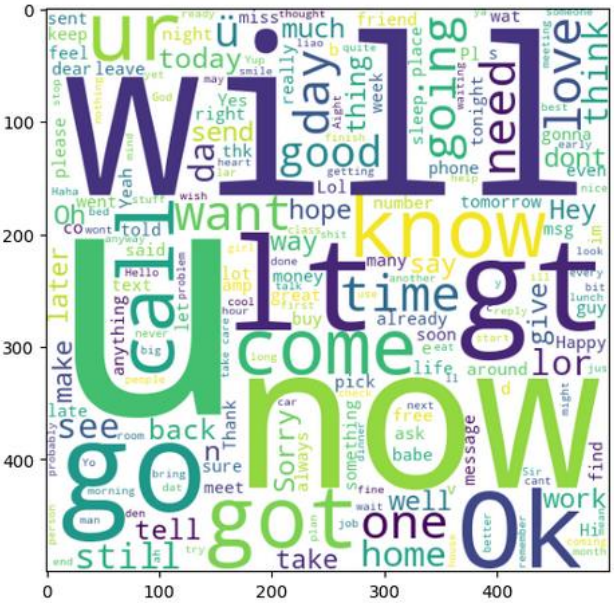


Slika 4. Vizualizacija poruka koje nisu spam

Vizualizacija nebalansiranog skupa podataka:



Slika 5. Vizualizacija spam poruka



Slika 6. Vizualizacija poruka koje nisu spam

3.2 Formiranje TF-IDF matrice

TF-IDF (*engl. Term Frequency-Inverse Document Frequency*) matrica je moćan alat u analizi teksta koji se koristi za obradu i reprezentaciju tekstualnih podataka. TF (engl. term frequency) predstavlja broj pojavljivanja pojedinačnih riječi u odnosu na ukupan broj riječi u datom dokumentu.
Formula za izračunavanje TF vrijednosti za riječ *t* u dokumentu *d* je:

$$TF(t,\ d) = \frac{Broj\ pojavljivanja\ riječi\ t\ u\ dokumentu\ d}{Ukupan\ broj\ riječi\ u\ dokumentu\ d}$$

IDF (*engl. Inversed Document Frequency*) je termin koji se koristi u analizi teksta kako bi se utvrdila važnost određenih riječi u dokumentima unutar šireg skupa dokumenata, kao što su u ovom slučaju e-mail poruke.
IDF vrijednost ima ulogu da smanji značaj riječi koje se često pojavljuju, jer se smatraju manje informativnim, dok sa druge strane povećava značaj riječi koje su rijetke i specifične, jer se smatraju relevantnijim.
Računanje vrijednosti IDF za riječ *t* u dokumentu *d*, gdje dokument pripada skupu dokumenata *D* je:

$$IDF(t,\ d,\ D) = \log\left(\frac{Ukupan\ broj\ dokumenata\ u\ skupu\ D}{Broj\ dokumenata\ u\ skupu\ D\ koji\ sadrže\ riječ\ t}\right)$$

Da bismo dobili konačnu TF-IDF vrijednosti za određenu riječ u dokumentu, potrebno je pomnožimo prethodno dobijene vrijednosti:

$$TF - IDF\ (t,\ d,\ D)\ = TF(t,\ d)\ \cdot\ IDF\ (t,\ d,\ D)$$

Za kreiranje TF-IDF matrice koristićemo alat TfidfVectorizer iz biblioteke *sklearn*. Pored ove matrice moguće je koristiti i CountVectorize matricu, gdje bismo e-mail poruke pretvarali u matricu tokena, pri čemu bi se svaki token označavao broj ponavljanja date riječi u svakom dokumentu (email-u).
Ključna razlika je ta što TF-IDF matrica uzima u obzir i značaj riječi u cjelokupnom korpusu dokumenata. Ova metrika "kažnjava" riječi koje se često pojavljuju u svim dokumentima i na taj način, dobijamo bogatiju i informativniju matricu koja bolje reflektuje suštinske karakteristike teksta i pomaže u izdvajanju ključnih informacija iz dokumenta.

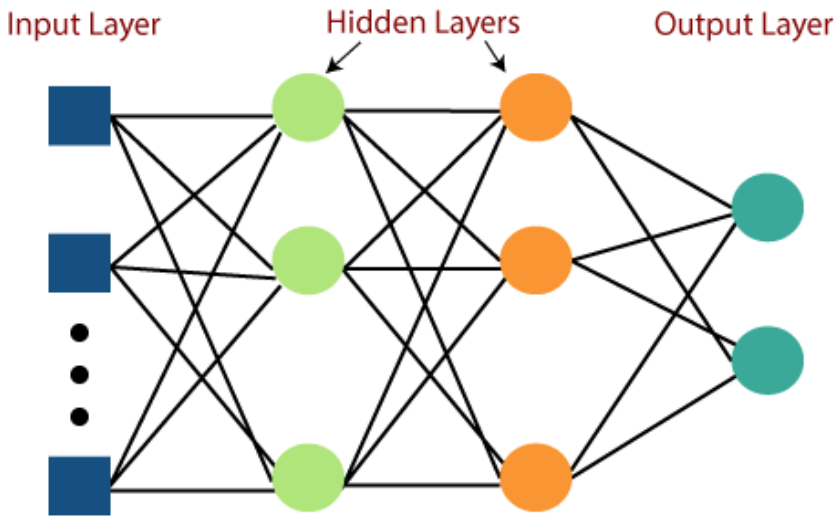
3.3 Podjela na trening i test podatke

Kako bismo iskoristili podatke za obuku modela na najbolji mogući način, vršimo podjelu podataka na dva skupa: trening skup (za učenje) i test skup (za evaluaciju). Model će prvo biti treniran na na trening skupu, a zatim će to znanje iskoristiti u evaluaciji testnih podataka kako bismo dobili objektivnu procjenu njegovih performansi.

3.4 Višeslojna neuronska mreža

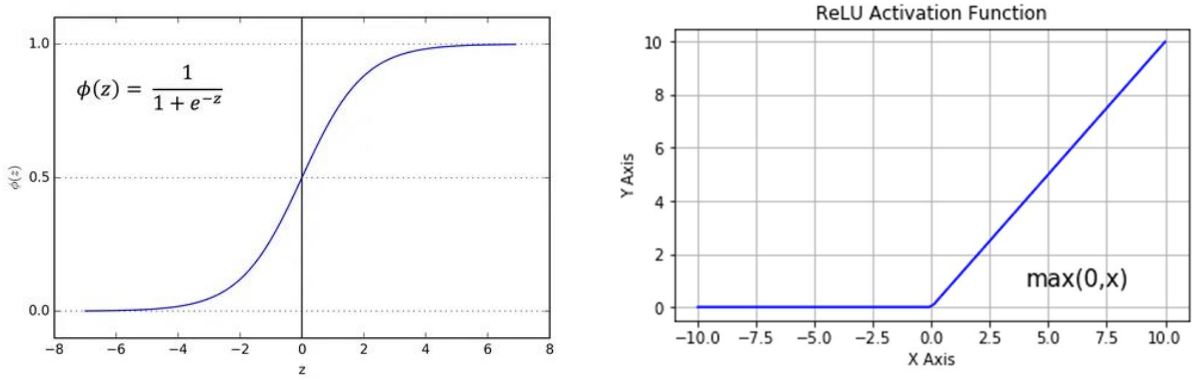
Višeslojna neuronska mreža je tip neuronske mreže koji sadrži više od jednog sloja neurona između ulaznih i izlaznih slojeva. Ovaj tip mreže često se koristi za rješavanje kompleksnih problema kod mašinskog učenja, poput prepoznavanja slika, obrade prirodnog jezika, analize teksta, i slično.
Osnovni elementi višeslojne neuronske mreže uključuju:

- **Ulazni sloj** (*engl. Input Layer*): Ovdje se ulazni podaci unose u mrežu. Svaki neuron u ovom sloju odgovara jednoj varijabli iz ulaznih podataka.
- **Skriveni slojevi** (*engl. Hidden Layers)*: To su slojevi koji se nalaze između ulaznog i izlaznog sloja. Svaki neuron u skrivenim slojevima prima izlaz iz prethodnog sloja i generiše ulaz za sljedeći sloj. Dubina mreže je određena brojem skrivenih slojeva.
- **Izlazni sloj** (*engl. Output Layer*): Ovaj sloj generiše konačne izlazne predikcije ili rezultate mreže. Broj neurona u izlaznom sloju zavisi od vrste zadatka. Na primjer, za binarnu klasifikaciju možemo imati jedan neuron sa sigmoidnom aktivacijom, dok za višeklasnu klasifikaciju obično imamo više neurona.



Slika 7. Višeslojna neuronska mreža, izvor [7]

Da bismo implementirali višeslojnu neuronsku mrežu, koristićemo popularnu biblioteku TensorFlow u okviru Keras API-ja.
Modelovaćemo našu neuronsku mrežu koristeći trening podatke, gdje ćemo u ulaznom sloju za dimenzije ulaza imati X_train.shape[1]. Sa obzirom na to da se bavimo problemom binarne klasifikacije, na izlaznom sloju ćemo imati samo jedan čvor sa vrijednostima između 0 i 1. Ako ta vrijednost prelazi prag od 0.5, označićemo e-mail kao spam, dok u suprotnom neće biti spam.
Za funkciju aktivacije ulaznog sloja ćemo koristiti relu, a za funkciju aktivacije izlaznog sloja ćemo koristiti sigmoidnu funkciju, koja mapira ulazne vrijednosti u interval [0, 1] što je pogodno za binarnu klasifikaciju:



Slika 8. Sigmoidna i ReLu funkcija aktivacije, izvor [2]

Nakon što smo definisali strukturu mreže, slijedi korak kompilacije modela. Za optimizaciju koristimo adam algoritam, dok za loss funkciju, sa obzirom na binarni karakter problema, koristimo *binary crossentropy*. Kao metriku za praćenje, koristićemo accuracy.
Nakon postavljanja modela, treniramo ga na trening podacima (X_train i y_train). Veličinu uzorka (*engl. batch size*), broj epoha (*engl. epochs*), kao i validation split podešavamo u zavisnosti od rezultata kako bismo postigli najbolje performanse modela.

3.5 Ocjena modela

Kako bismo ocijenili performanse našeg modela, koristićemo različite metode iz sklearn.metrics biblioteke. Ove metode nam pomažu procijeniti kako se naš model ponaša u klasifikacijskim zadacima pružajući različite aspekte njegove uspješnosti. Neke od ključnih metoda koje ćemo koristiti su:

- Izvještaj o klasifikaciji (*engl. Classification Report)* - informativan izvještaj koji nam pruža detaljne informacije o performansama našeg modela u klasifikacijskim zadacima. Uključuje metrike kao što su:
  - Preciznost (*engl. Precision*) - omjer pravilno pozitivno klasifikovanih instanci u odnosu na ukupan broj pozitivnih instanci. Mjeri koliko je model tačno prepoznao pozitivne primjere.
  - Odziv (*engl. Recall)* - mjeri sposobnost modela da prepozna sve pozitivne primjere.
  - F1-score - sredina između preciznosti i odziva. Koristi se kako bi se dobila balansirana mjera performansi modela.
  - Tačnost (*engl. Accuracy*) -mjeri ukupnu tačnost modela, odnosno omjer ispravno klasifikovanih instanci u odnosu na ukupan broj instanci.
- Matrica konfuzije (*engl. Confusion Matrix*) - matrica koja prikazuje broj stvarno pozitivnih, stvarno negativnih, lažno pozitivnih i lažno negativnih klasifikacija koje je model napravio. Ova matrica pruža detaljan uvid u modelove greške i sposobnost razlikovanja između različitih klasa. Model se smatra dobrim ukoliko ima visoku vrijednost za TP i TN, a nisku vrijednost za FP i FN.

		Predicted	
		Negative **(N)** **-**	Positive **(P)** **+**
Actual	Negative **-**	True Negative **(TN)**	**False Positive (FP)** **Type I Error**
	Positive **+**	**False Negative (FN)** **Type II Error**	True Positive **(TP)**

Slika 9. Matrica konfuzije, izvor [1]

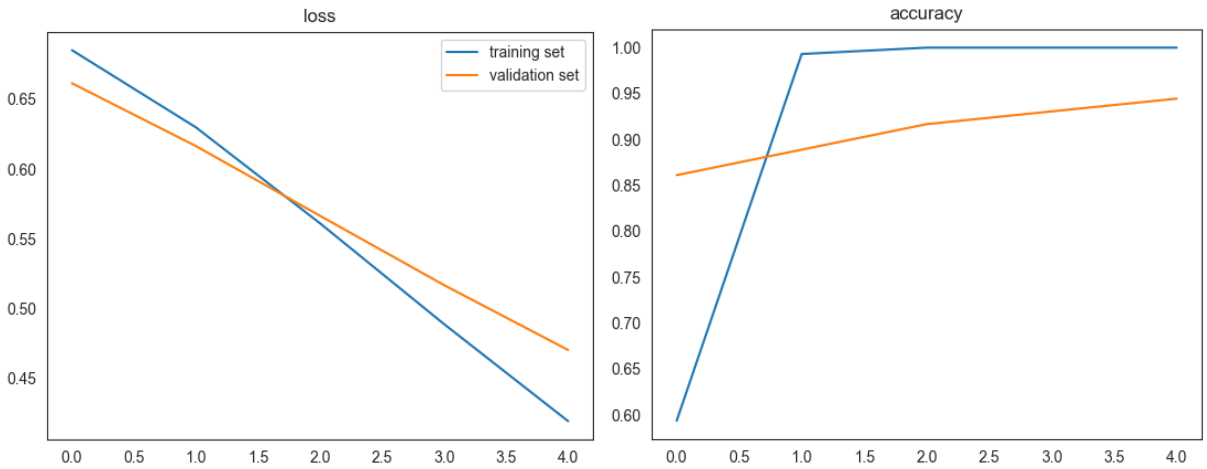
- Preciznost (*engl. Precision Score)* - numerička vrijednost koja predstavlja preciznost modela u prepoznavanju pozitivnih primjera. Veća vrijednost preciznosti ukazuje na bolju sposobnost modela da minimizuje lažno pozitivne klasifikacije. Računa se pomoću sljedeće formule:

$$\frac{TP}{(TP+FP)}$$

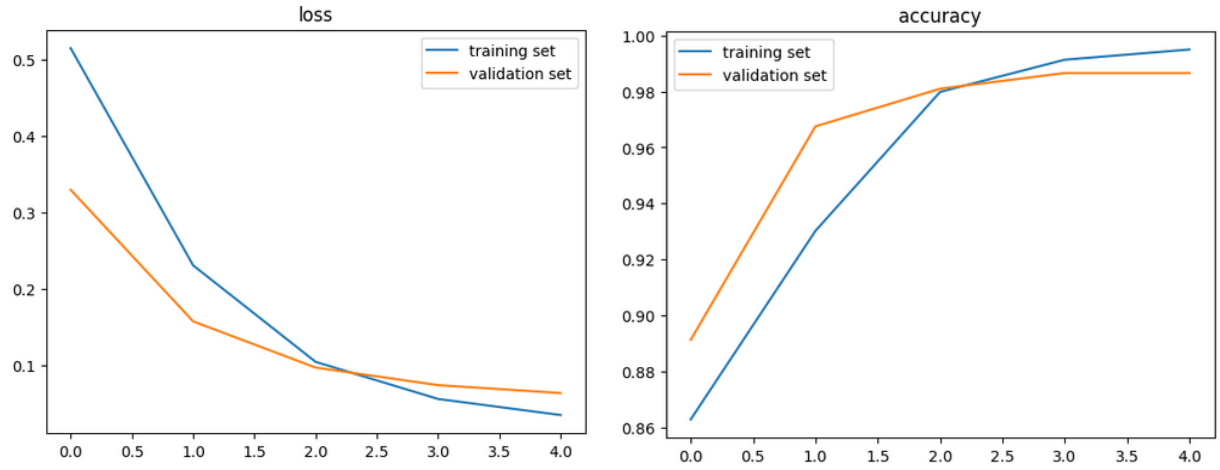
Ove metode nam omogućavaju da prilagodimo model i poboljšamo njegove performanse, ukoliko je to potrebno.

4 Rezultati modela

Uzmimo prvo da početni model sadrži 2 sloja, ulazni i izlazni i neka je broj epoha 5.



Slika 10. Rezultati balansiranog skupa podataka

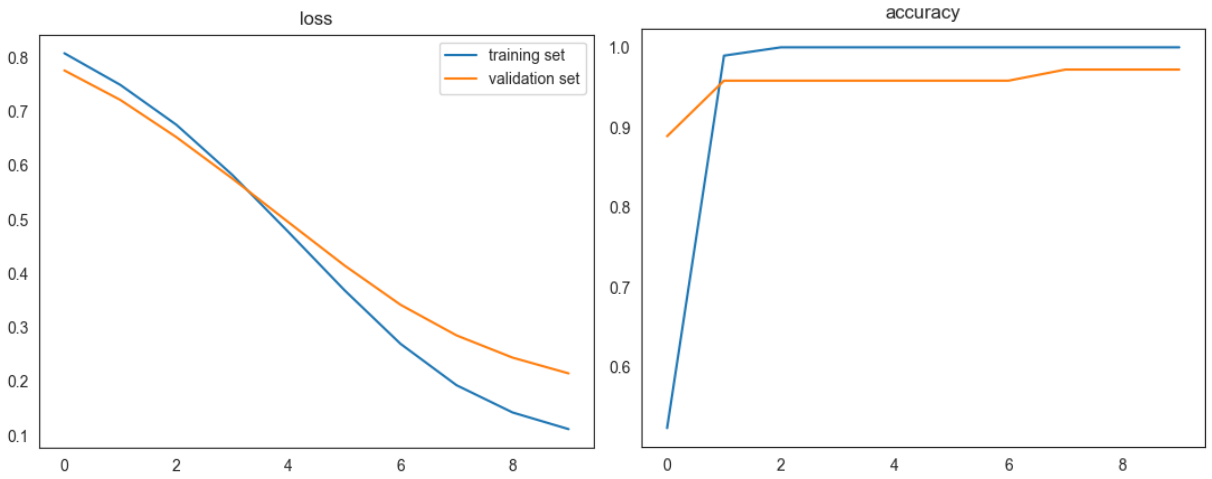


Slika 11. Rezultati nebalansiranog skupa podataka

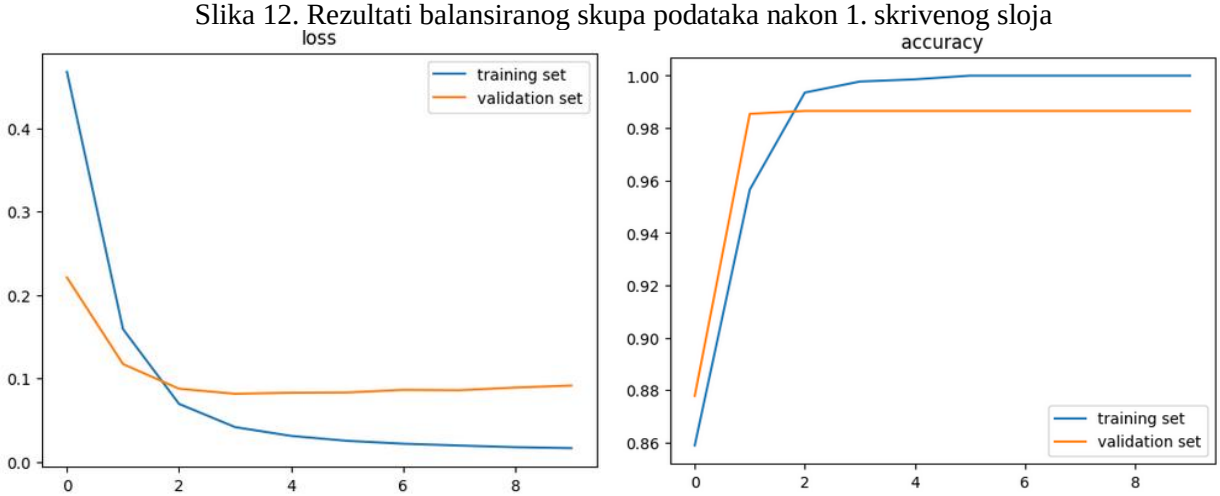
Kada je riječ o balansiranom skupu podataka, iz priloženog možemo uočiti da je loss na validacionom skupu (0.4705) veći nego na trening skupu (0.4196), što nam govori da je došlo do preprilagođavanja podataka. Isto tako vidimo da validacioni skup dolazi do tačnosti 0.9444.
U slučaju kada se radi o nebalansiranom skupu podataka loss na validacionom skupu (0.0637) je takođe veći nego na trening skupu (0.0350). Validacioni skup u ovom slučaju dolazi do tačnosti od 0.9865.

Pokušaćemo unaprijediti naš model korišćenjem tehnika regularizacije, koje su dostupne u TensorFlow biblioteci.
Regularizacija obuhvata različite tehnike, čiji je cilj smanjiti kompleksnost modela tokom treninga, čime se sprečava prenaučenost. U ovom slučaju koristićemo L1 regularizaciju.
Nakon primjene L1 regularizacije, kod balansiranih podataka primjećujemo povećanje tačnosti na validacijskom skupu na 0.9722, dok je gubitak (loss) ostao sličan kao prije primjene regularizacije, što ukazuje na poboljšanje sposobnosti modela. Kod nebalansiranih podataka došlo je do blagog povećanja loss vrijednosti kada je u pitanju validacioni skup (0.0984), dok je tačnost ostala približno ista (0.9854).

Pokušajmo sada poboljšati model dodavanjem jednog skrivenog sloja koji sadrži 64 čvora i koristi relu kao funkciju aktivacije. Broj epoha će biti povećan na 10.



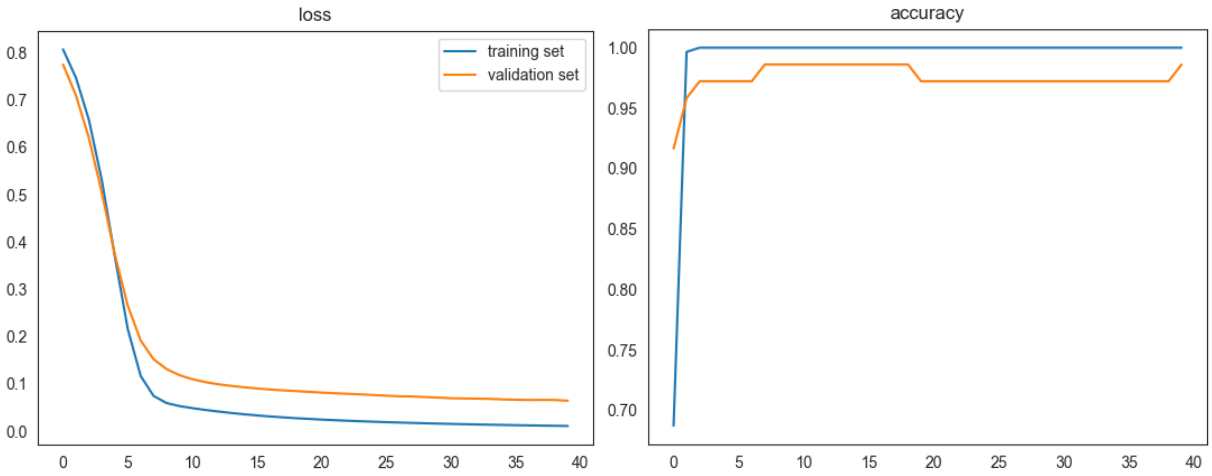
Slika 12. Rezultati balansiranog skupa podataka nakon 1. skrivenog sloja



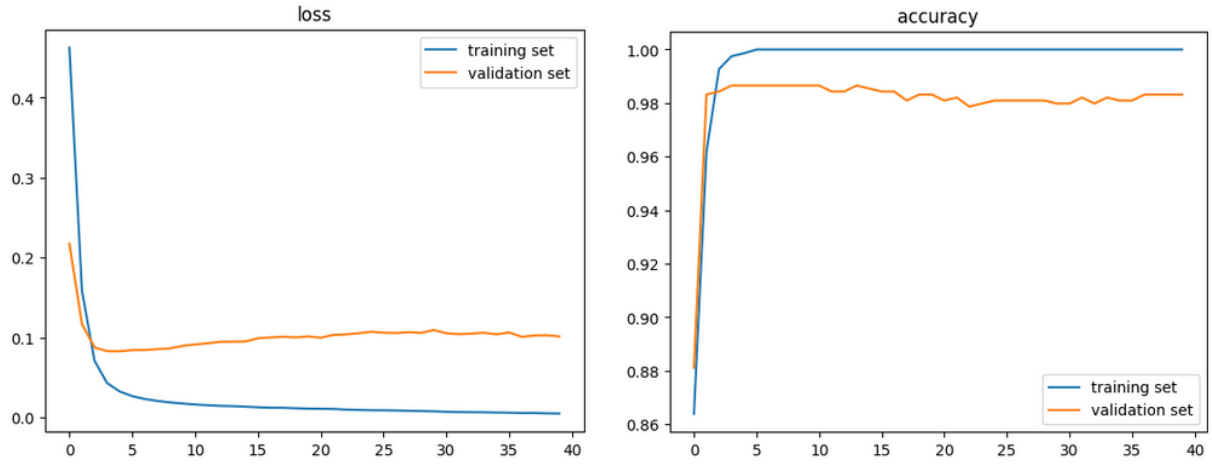
Slika 13. Rezultati nebalansiranog skupa podataka nakon 1. skrivenog sloja

Nakon dodavanja jednog skrivenog sloja kod balansiranog skupa podataka, primjećujemo poboljšanje modela, gdje je sada tačnost na validacionom skupu 0.9722, sa loss-om od 0.2152.
Kada je riječ o nebalansiranom skupu podataka, nije došlo do značajnijeg poboljšanja.

Pokušaćemo poboljšati model dodavanjem novih skrivenih slojeva. Neka sada novi skriveni sloj sadrži 128 čvorova i neka također koristi relu kao funkciju aktivacije. Povećajmo i broj epoha na 40 kako bi trening podaci prošli veći broj puta kroz mrežu.



Slika 14. Rezultati balansiranog skupa podataka nakon 2. skrivenog sloja



Slika 15. Rezultati nebalansiranog skupa podataka nakon 2. skrivenog sloja

Nakon dodavanja i drugog skrivenog sloja tačnost kod balansiranog skupa podataka na validacionom skupu je sada 0.9861, a loss je 0.0648, što predstavlja poboljšanje u odnosu na jedan skriveni sloj. Na test skupu dobijamo tačnost od 1.0000 sa loss-om od 0.0113.
Kod nebalansiranog skupa podataka nije došlo do značajnog poboljšanja.

Dodavanjem još skrivenih slojeva, primjećujemo da model ne napreduje, nego čak i dolazi do smanjivanja njegove efikasnosti, tako da za 2 sloja dobijemo optimalno rješenje.
Isto tako izbor brojeva neurona u skrivenim slojevima je rezultat eksperimentisanja sa različitim vrijednostima, sa ciljem pronalaženja konfiguracije koja donosi najbolje rezultate.
Ostale ocjene su:

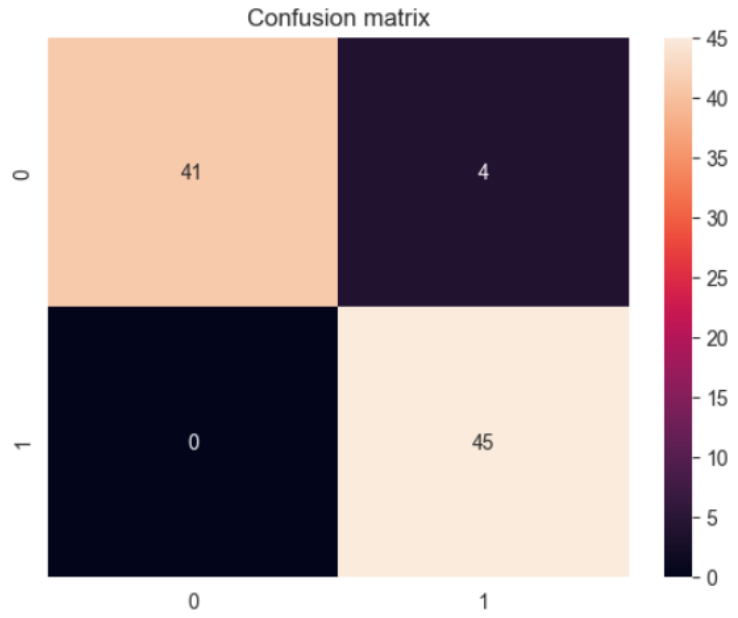
	precision	recall	f1-score	support
0	1.00	0.91	0.95	45
1	0.92	1.00	0.96	45
accuracy			0.96	90
macro avg	0.96	0.96	0.96	90
weighted avg	0.96	0.96	0.96	90

Slika 16. Ocjene balansiranog skupa podataka

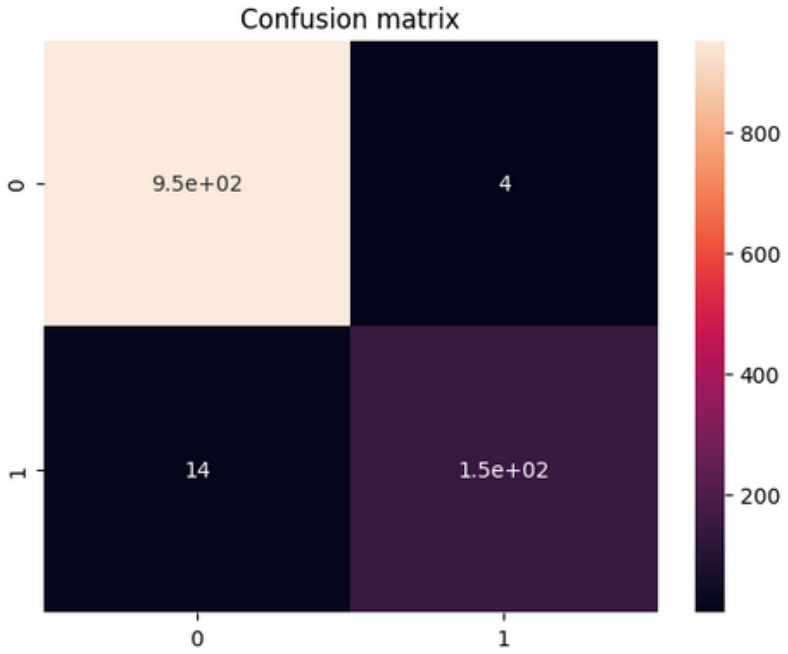
	precision	recall	f1-score	support
0	0.99	1.00	0.99	955
1	0.97	0.91	0.94	160
accuracy			0.98	1115
macro avg	0.98	0.95	0.97	1115
weighted avg	0.98	0.98	0.98	1115

Slika 17. Ocjene nebalansiranog skupa podataka

Na osnovu matrice konfuzije vidimo da je model dovoljno dobar, jer ima visoku vrijednost za TP i TN, a nisku vrijednost za FP i FN u oba slučaja.



Slika 18. Matrica konfuzije balansiranog skupa podataka



Slika 19. Matrica konfuzije nebalansiranog skupa podataka

5 Poređenje sa drugim algoritmima klasifikacije

	Logistic Regression	MLP	MultinomialNB	Random Forest
Accuracy	94.444444	95.555556	97.777778	92.222222
F1_score	94.427245	95.546759	97.776680	92.174885
Recall	94.444444	95.555556	97.777778	92.222222
Precision	94.444444	95.555556	97.777778	92.222222

Slika 20. Poređenje kod balansiranog skupa podataka

Na osnovu datih metrika vidimo da MultinomialNB model ima najviše vrijednosti, pa samim tim on postiže najbolje rezultate u ovom slučaju. MLP takođe pokazuje dobre rezultate, dok su logistička regresija i Random Forest nešto manje tačni.

	Logistic Regression	MLP	MultinomialNB	Random Forest
Accuracy	97.219731	98.385650	96.233184	98.385650
F1_score	93.909691	96.628024	91.370223	96.553063
Recall	97.219731	98.385650	96.233184	98.385650
Precision	97.219731	98.385650	96.233184	98.385650

Slika 21. Poređenje kod nebalansiranog skupa podataka

Kada je riječ o nebalansiranom skupu podataka, vidimo da MLP i Random Forest daju najbolje rezultate, sa tim da MLP ima malo bolji F1 score.
Na osnovu ovih rezultata vidimo da MLP model daje veoma zadovoljavajuće rezultate za problem klasifikacije između spam i ne-spam poruka. Model je veoma tačan i ima dobar balans između sposobnosti identifikacije pozitivnih instanci (spam) i minimizacije lažnih pozitiva.

6 Zaključak

Na temelju analize rezultata iz prethodnih plotova i matrica, možemo zaključiti da model pruža izrazito dobre rezultate i da ima visoku sposobnost prepoznavanja spam e-mail pošte. MLP model daje konzistentno visoke performanse, što ga čini dobrim izborom bez ozbira na raspodjelu klasa u skupu podataka.
Isto tako možemo primijetiti da dodavanje velikog broja skrivenih slojeva u neuronsku mrežu neće dati bolji rezultat, nego može dovesti do smanjenja efikasnosti modela i povećanja njegove složenosti.
Zato je važno pravilno odabrati arhitekturu mreže kao i broj skrivenih slojeva. Potrebno je pažljivo odabrati procenat skupa za treniranje i za testiranje, kao i za validaciju. Također, kvalitet samog skupa podataka ima značajan uticaj na to koliko dobro će model raditi, pa su iz tog razloga, obrada i priprema podataka podjednako bitni kao i sam proces kreiranja neuronske mreže.

7 Literatura

1. https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

2. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

3. https://ceur-ws.org/Vol-3340/paper32.pdf

4. https://mariamsarfraz.github.io/attachments/ml_report.pdf

5. https://comodemia.comodo.com/repository/495964.pdf

6. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9222163

7. https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow

8. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

9. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

10. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html