SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij računarstva

**EMAIL SPAM FILTER**

Projektni zadatak

 Antonio Žugec

Osijek, 2023.

# Sadržaj

[1 Sadržaj 2](#_Toc137405490)

[2 UVOD 3](#_Toc137405491)

[3 PRIKUPLJENI PODATCI 4](#_Toc137405492)

[4 KORIŠTENE TEHNOLOGIJE 5](#_Toc137405493)

[1.4 Tensorflow 5](#_Toc137405494)

[2.4 Numpy 5](#_Toc137405495)

[3.4 Sklearn 6](#_Toc137405496)

[4.4 Mathplotlib 6](#_Toc137405497)

[5 STROJNO UČENJE 7](#_Toc137405498)

[1.5 Naive Bayes 7](#_Toc137405499)

[6 DUBOKO UČENJE 9](#_Toc137405500)

[1.6 BERT 10](#_Toc137405501)

[2.6 Konvolucijske mreže 13](#_Toc137405502)

[3.6 LSTM mreža 13](#_Toc137405503)

[7 IZRADA PROJEKTNOG ZADATKA 15](#_Toc137405504)

[1.7 Obrada podataka 15](#_Toc137405505)

[2.7 Obrada podataka za naive bayesov model 16](#_Toc137405506)

[3.7 Obrada podataka za BERT model 16](#_Toc137405507)

[4.7 Obrada podataka za konvolucijsku mrežu 17](#_Toc137405508)

[5.7 Obrada podataka za LSTM mrežu 18](#_Toc137405509)

[8 TRENIRANJE MODELA 19](#_Toc137405510)

[1.8 Treniranje Naive Bayesov model 19](#_Toc137405511)

[2.8 Treniranje BERT modela 19](#_Toc137405512)

[3.8 Treniranje konvolucijskog modela 21](#_Toc137405513)

[4.8 Treniranje LSTM modela 22](#_Toc137405514)

[9 ANALIZA MODELA 24](#_Toc137405515)

[1.9 Analiza Naive Baysevog modela 25](#_Toc137405516)

[2.9 Analiza BERT modela 26](#_Toc137405517)

[3.9 Analiza konvolucijskog modela 27](#_Toc137405518)

[4.9 Analiza LSTM modela 28](#_Toc137405519)

[5.9 Rezultati 29](#_Toc137405520)

[10 ZAKLJUČAK 30](#_Toc137405521)

[11 LITERATURA 31](#_Toc137405522)

# UVOD

Email spam predstavlja dugogodišnji problem s kojim se korisnici diljem svijeta suočavaju svakodnevno. Preplavljenost inbox-a neželjenim porukama ne samo da oduzima vrijeme i pažnju korisnicima, već može predstavljati i potencijalnu prijetnju po sigurnost i privatnost. U cilju rješavanja ovog problema, strojno učenje pruža moćan alat za izgradnju učinkovitih email spam filtara. Projektni zadatak za strojno učenje ima za cilj izgradnju takvog filtra koji će klasificirati email poruke kao "spam" ili "ham" (ne-spam).

Cilj ovog projekta je izgraditi precizan i učinkovit email spam filter, pomoću algoritama za strojna učenje i dubokih neuronskih mreža, koji može filtrirati dolazne poruke u stvarnom vremenu i pouzdano razlikovati između spam i ham sadržaja. Implementacija ovog filtra može donijeti velike koristi korisnicima, štiteći njihovo vrijeme, privatnost i sigurnost prilikom pretraživanja i korištenja njihovih email računa.

# PRIKUPLJENI PODATCI

Skup podataka preuzet je s platforme Kaggle, riječ je o skupu Email-ova, gdje svaki ima svoju odgovarajuću oznaku, "spam" ili "ham". Skup podataka, skinut s Kaggle-a, sam malo modificiraju za svoje potrebe. Izbacio sam NaN podatke, te ostavio 2000 "ham" i 2000 "spam" poruka.

Link s podatcima :  https://drive.google.com/file/d/19jMqNuxHQyno56o1-43nJOuABd7xYlTl/view?usp=drive\_link

# KORIŠTENE TEHNOLOGIJE

Za rješavanje problema projektnog zadatka korištene su biblioteke koje su podržane od strane programskog jezika Python. Za rad modela korištene su biblioteke Tensorflow, Numpy, Sklearn dok je za vizualizaciju korištena biblioteka Mathplotlib.

## Tensorflow

[1]TensorFlow je popularna biblioteka otvorenog koda za strojno učenje koja je razvijena od strane Google Brain tima. Ona pruža fleksibilnost i skalabilnost za izgradnju i treniranje različitih vrsta neuronskih mreža, uključujući duboke modele, za razne zadatke u području umjetne inteligencije. TensorFlow podržava distribuirano izračunavanje, vizualizaciju modela, evaluaciju performansi te omogućava rad s velikim skupovima podataka. Ova moćna biblioteka se koristi u raznim industrijama i akademskim istraživanjima za rješavanje problema kao što su klasifikacija, regresija, obrada prirodnog jezika, računalni vid i mnogi drugi.

U sklopu TensorFlow-a za jedan model koristio se [2]BERT (Bidirectional Encoder Representations from Transformers). To je model dubokog učenja razvijen od strane Google-a koji koristi transformatorsku arhitekturu za obradu teksta. BERT je treniran na velikom skupu podataka i može naučiti bogate reprezentacije riječi i konteksta, omogućavajući napredne zadatke obrade prirodnog jezika poput prevođenja, odgovaranja na pitanja i klasifikacije teksta. TensorFlow pruža implementaciju BERT modela i omogućava njegovo treniranje i primjenu.

## Numpy

[3]NumPy (Numerical Python) je biblioteka za Python koja pruža podršku za rad s višedimenzionalnim nizovima i matricama. Ona je temeljni alat za znanstveno računanje i analizu podataka u Pythonu. NumPy nudi efikasne funkcionalnosti za izvođenje matematičkih operacija, linearnu algebru, statistiku, generiranje pseudo-slučajnih brojeva i druge numeričke manipulacije. Njegova glavna struktura podataka je ndarray, koja omogućava brzu i efikasnu obradu velikih količina podataka. NumPy je također temelj za mnoge druge biblioteke u Python ekosustavu, kao što su Pandas, SciPy i Matplotlib, što ga čini ključnim alatom za znanstvenike podataka, istraživače i programere koji se bave numeričkim računanjem.

## Sklearn

[4]Scikit-learn, poznata i kao sklearn, je popularna biblioteka otvorenog koda za strojno učenje koja je napisana u programskom jeziku Python. Sklearn pruža širok spektar alata i funkcionalnosti za rješavanje raznovrsnih zadataka strojnog učenja, uključujući klasifikaciju, regresiju, grupiranje, smanjenje dimenzionalnosti, evaluaciju modela i još mnogo toga. Ova biblioteka je poznata po svojoj jednostavnosti korištenja, dobro dokumentiranoj sintaksi i dobro optimiziranim implementacijama različitih algoritama strojnog učenja. Sklearn također nudi podršku za rad s podacima, uključujući funkcionalnosti za preprocesiranje, skaliranje i validaciju podataka. Zahvaljujući svojoj fleksibilnosti i bogatom funkcionalnošću, sklearn je postao ključni alat za istraživače, znanstvenike podataka i praktičare strojnog učenja pri razvoju i evaluaciji modela u Python ekosustavu.

## Mathplotlib

[5]Matplotlib je popularna biblioteka za vizualizaciju podataka u programskom jeziku Python. Ona omogućava kreiranje raznih vrsta grafikona, dijagrama, histograma, skica i drugih vizualnih prikaza podataka. Matplotlib je fleksibilan alat koji omogućava detaljno prilagođavanje izgleda i stilova vizualizacija. Može se koristiti za istraživanje podataka, prikazivanje rezultata analiza, prezentacije i mnoge druge svrhe. Uz bogatu funkcionalnost, Matplotlib nudi intuitivan API i jednostavnu sintaksu, što ga čini popularnim izborom za programere, znanstvenike podataka i istraživače koji žele prikazati i interpretirati podatke na vizualno privlačan način.

# STROJNO UČENJE

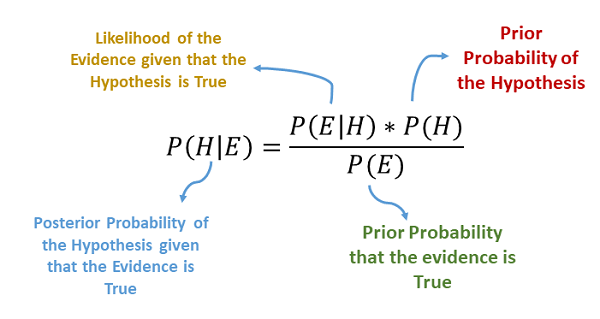
[6]Strojno učenje je grana umjetne inteligencije koja se bavi razvojem algoritama i tehnika koje omogućuju računalima da nauče iz podataka i automatski poboljšaju svoje performanse u obavljanju određenih zadataka. Ključna ideja iza strojnog učenja je da umjesto da programiramo računalo da izvršava specifične zadatke, omogućimo mu da samo nauči kako obavljati te zadatke putem iskustva.

Strojno učenje se temelji na konceptu "učenja" iz podataka. Računalu se daje skup podataka koji sadrži primjere i pripadajuće oznake ili rezultate. Algoritam strojnog učenja zatim analizira te podatke kako bi pronašao obrasce, strukture i zakonitosti. Na temelju tih naučenih informacija, algoritam može donositi zaključke ili predviđati rezultate za nove, neviđene primjere.

Postoje različite vrste pristupa i tehnika u strojnom učenju, kao što su nadzirano učenje i nenadzirano učenje. U nadziranom učenju, algoritam se trenira na označenim primjerima kako bi naučio mapiranje ulaza na odgovarajuće izlaze. U nenadziranom učenju, algoritam se koristi za otkrivanje skrivenih struktura ili grupiranje podataka bez oznaka.

## Naive Bayes

[7]Naive Bayesov model je jednostavan, ali efikasan algoritam za klasifikaciju u strojnom učenju. Temelji se na Bayesovom teoremu i pretpostavci nezavisnosti atributa, odakle dolazi i naziv "naive" (naivan). Model koristi vjerojatnosti kako bi klasificirao nove primjere na temelju njihovih atributa. Iako se pretpostavka nezavisnosti atributa ne ispuni uvijek u stvarnim slučajevima, Naive Bayesov model i dalje nalazi primjene u mnogim područjima kao što su detekcija spam poruka, analiza sentimenta, klasifikacija dokumenata i još mnogo toga.



**Slika 5.1.** Naive Bayesova formula

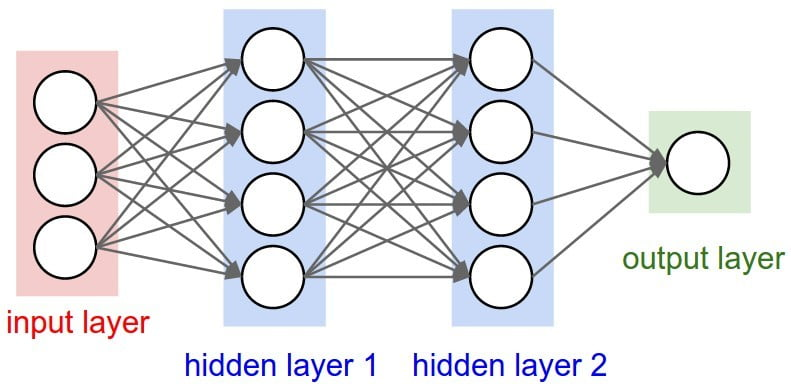
Podatke je potrebno pripremiti u obliku gdje svaki primjer ima skup atributa i pripadajuću oznaku klase. Na temelju skupa podataka, izračunavaju se vjerojatnosti pojavljivanja svake klase i vjerojatnosti svakog atributa za svaku klasu. To se radi pomoću trening faze u kojoj se analizira skup podataka i broji koliko često se pojedini atributi pojavljuju u određenim klasama. Kada se model trenira, može se koristiti za klasifikaciju novih, neviđenih primjera. Za svaki novi primjer, izračunavaju se vjerojatnosti pripadnosti svakoj klasi na temelju vjerojatnosti klasa i atributa. Klasa s najvišom vjerojatnošću smatra se predikcijom modela. Ključna pretpostavka u Naive Bayesovom modelu je nezavisnost atributa, što znači da se vjerojatnosti atributa računaju zasebno bez uzimanja u obzir međuovisnosti između atributa.

# DUBOKO UČENJE

[8]Duboko učenje je tehnika strojnog učenja koja koristi višeslojne neuronske mreže za automatsko učenje i izvlačenje složenih značajki iz podataka. Kroz iterativni proces, modeli se treniraju na velikim skupovima podataka kako bi se naučile reprezentacije koje omogućavaju razumijevanje složenih obrazaca i hijerarhija informacija, što rezultira sposobnošću modela da donose precizne predikcije i donose informirane odluke.

U dubokom učenju se često koriste različite funkcije i metode kako bi se postigla što bolja performansa modela. Neke od ključnih funkcija i metoda koje se koriste u dubokom učenju su aktivacijske funkcije, konvolucijske operacije, pooling, dropout, optimizacijski algoritmi i rekurentne mreže.

Aktivacijske funkcije primjenjuju se na izlazne vrijednosti neurona i daju modelu sposobnost da nauči nelinearne značajke. Neke od uobičajenih aktivacijskih funkcija su sigmoidna funkcija, tangens hiperbolički, ReLU (Rectified Linear Unit) i softmax. Konvolucijske neuronske mreže (CNN) koriste konvolucijske operacije za ekstrakciju značajki iz vizualnih podataka kao što su slike. Konvolucija uključuje primjenu filtera (jezgre) na ulazne podatke kako bi se detektirale različite značajke kao što su rubovi, teksture ili oblici. Pooling se koristi za smanjenje dimenzionalnosti podataka i dobivanje invarijantnosti na pomake i deformacije. Uobičajene tehnike razrjeđivanja uključuju maksimalno razrjeđivanje (max pooling) i srednje razrjeđivanje (average pooling). Dropout je tehnika regularizacije koja slučajno isključuje neke neurone tijekom faze treniranja. To pomaže u sprječavanju prenaučenosti i poboljšava generalizaciju modela. Optimizacijski algoritmi se koriste u dubokom učenju za prilagođavanje težina mreže tijekom treniranja. Neke popularne metode optimizacije su Stochastic Gradient Descent (SGD), Adam, RMSprop i AdaGrad. Rekurentne neuronske mreže (RNN) koriste se za obradu sekvencijalnih podataka, poput teksta ili vremenskih nizova. RNN-ovi koriste povratne veze koje omogućavaju modelu da koristi informacije iz prethodnih stanja za donošenje odluka u trenutnom stanju.



**Slika 6.1.** Neuronska mreža

Duboka neuronska mreža, kao što je prikazano na slici 6.1, može se prikazati kao slojevita struktura, gdje svaki sloj predstavlja skup neurona koji obrađuju ulazne podatke. Prvi sloj u dubokoj neuronskoj mreži je ulazni sloj koji prima ulazne podatke. Ovisno o problemu, ulazni podaci mogu biti slike, tekst, zvuk ili drugi oblik podataka. Nakon ulaznog sloja slijede skriveni slojevi. Svaki skriveni sloj sastoji se od više neurona, a svaki neuron prima podatke od prethodnog sloja. Svaki neuron obavlja linearnu kombinaciju ulaznih podataka s težinama i primjenjuje aktivacijsku funkciju kako bi generirao izlaznu vrijednost. Posljednji sloj u dubokoj neuronskoj mreži je izlazni sloj. Ovaj sloj generira izlazne vrijednosti ili predikcije na temelju obrade podataka u prethodnim slojevima. Broj neurona u izlaznom sloju ovisi o broju klasa koje model treba predvidjeti. Težine između neurona u svakom sloju predstavljaju parametre modela koji se prilagođavaju tijekom faze treniranja. Cilj treniranja je minimizirati grešku između predikcija modela i stvarnih vrijednosti. To se postiže koristeći algoritme optimizacije i metode kao što je Adam.

## BERT

BERT (Bidirectional Encoder Representations from Transformers) je model dubokog učenja koji se koristi za zadatke obrade prirodnog jezika. Radi na osnovi transformer arhitekture, posebno dijela enkodera, kako bi razumio kontekstualne veze između riječi u tekstu. Za razliku od tradicionalnih modela koji obradu tekst sekvencijalno, BERT čita cijeli niz riječi odjednom, čineći ga dvosmjernim ili kontekstualno osviještenim. On hvata značenje riječi na temelju riječi koje je okružuju, kako s lijeve tako i s desne strane. BERT je pred-treniran na velikom korpusu teksta koristeći dva zadatka: maskirano modeliranje jezika (MLM), prikazano na slici 6.2 i predviđanje sljedeće rečenice (NSP), prikazano na slici 6.3.

A diagram of a computer

Description automatically generated with low confidence

**Slika 6.2.** Masked LM (MLM) BERT

  Prije nego što se nizovi riječi unesu u BERT, 15% riječi u svakom nizu zamjenjuje se s *MASK* tokenom. Model zatim pokušava predvidjeti izvornu vrijednost maskiranih riječi, na temelju konteksta koji pružaju druge, nenamaskirane, riječi u nizu. Prvo se dodaje sloj klasifikacije na vrh izlaza enkodera. Zatim, Množe se izlazni vektori s matricom ugradnje, pretvarajući ih u dimenziju rječnika, te se izračunavaju vjerojatnosti svake riječi u rječniku pomoću softmax funkcije.

A picture containing text, screenshot, number, font

Description automatically generated

**Slika 6.3.** Next Sentence Prediction (NSP) BERT

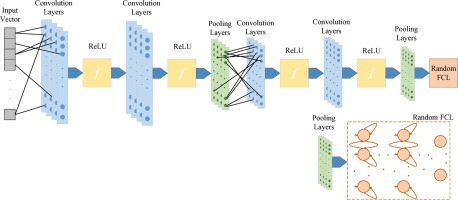
U procesu treniranja BERT modela, model prima parove rečenica kao ulaz i uči predviđati je li druga rečenica u paru sljedeća rečenica u originalnom dokumentu. Kako bi pomogao modelu razlikovati ove dvije rečenice tijekom treniranja, dodaje se na početak prve rečenice token [CLS], a na kraj svake rečenice dodaje se token [SEP]. Kako bi se predvidjelo je li druga rečenica zaista povezana s prvom, cijeli ulazni niz prolazi kroz Transformer model. Zatim, izlaz tokena [CLS] se transformira u vektor oblika 2×1 koristeći jednostavan sloj klasifikacije (matrice naučenih težina i pomaka) te se izračunava vjerojatnost IsNextSequence pomoću softmax funkcije.

Prilikom treniranja BERT modela, Maskirano modeliranje jezika (Masked LM) i Predviđanje sljedeće rečenice (Next Sentence Prediction) treniraju se zajedno, s ciljem minimiziranja kombinirane funkcije gubitka ove dvije strategije.

U procesu fino podešavanja, prethodno obučeni BERT model prilagođava se i fino podešava za određeni zadatak, kao što je klasifikacija teksta, prepoznavanje imenovanih entiteta ili odgovaranje na pitanja. To se postiže primjenom optimizacijskih algoritama na temelju gradijenta, kao što su stohastički gradijentni spust (SGD) ili Adam, kako bi se ažurirale težine i pomaci modela na temelju skupa podataka specifičnog za zadatak. Fino podešavanje omogućava BERT-u da prilagodi i specijalizira svoje reprezentacije za specifične nijanse i zahtjeve ciljnog zadatka, što rezultira poboljšanom izvedbom i boljim predikcijama specifičnim za taj zadatak.

## Konvolucijske mreže

1-D konvolucijske neuronske mreže (1-D CNN) su arhitekture neuronskih mreža koje se koriste za obradu i analizu sekvencijalnih podataka, kao što su vremenski nizovi, jezički podaci ili podaci s jednim prostornim razmjernikom.



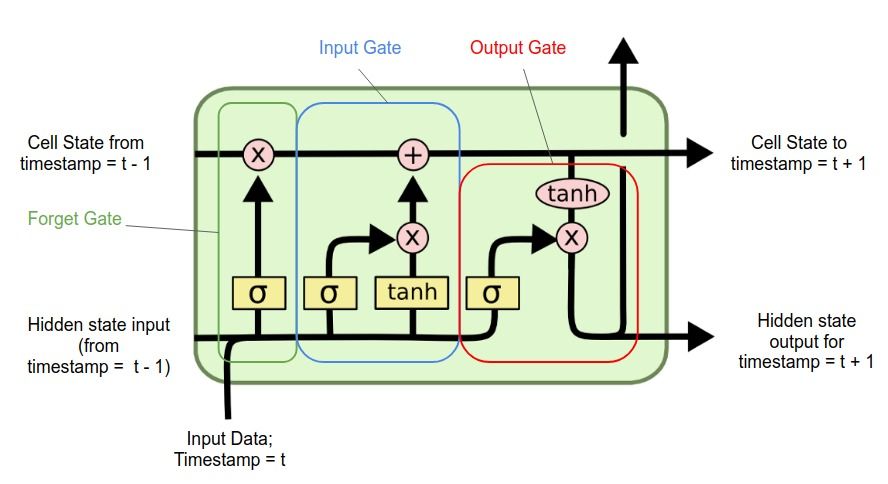
**Slika 6.4.** Konvolucijska mreža

U 1-D CNN, konvolucijski slojevi se primjenjuju na ulazne sekvencijalne podatke kako bi se izvukle lokalne značajke. Svaki konvolucijski sloj ima skup filtara koji se pomiču preko ulazne sekvencije koristeći konvolucijsku operaciju. Ova operacija stvara nove značajke filtriranjem dijelova ulaza. Nakon konvolucije, obično se primjenjuje sloj sažimanja (pooling) koji smanjuje prostornu dimenziju značajki i sažima informacije. Najčešći način sažimanja u 1-D CNN je maksimalno sažimanje (max pooling), gdje se odabire najveći element iz svakog podskupa značajki. Nakon više slojeva konvolucije i sažimanja, značajke se uvode u potpuno povezane slojeve koji služe za klasifikaciju ili regresiju. Ovi slojevi obično imaju funkcije aktivacije kao što su ReLU (rectified linear unit) ili sigmoida kako bi se generirali izlazni rezultati.

## LSTM mreža

[9]LSTM (Long Short-Term Memory) neuronske mreže su posebna vrsta rekurentnih neuronskih mreža (RNN) koje se koriste za obradu sekvencijalnih podataka, kao što su tekstovi, vremenski nizovi ili govorni signali. LSTM mreže su dizajnirane za rješavanje problema nestajanja ili eksplozije gradijenta koji se često javlja kod tradicionalnih RNN-ova.

Ključni elementi LSTM mreže su memorijske ćelije koje čuvaju i ažuriraju informacije tijekom vremena. Svaka memorijska ćelija ima tri osnovna dijela: ulazna vrata (input gate), zaboravna vrata (forget gate) i izlazna vrata (output gate).



**Slika 6.5.** LSTM mreža

Ulazna vrata kontroliraju koliko novih informacija se dodaje u memorijsku ćeliju. One uzimaju u obzir trenutni ulazni podatak i prethodno skriveno stanje mreže kako bi odredile koje informacije treba pohraniti u memorijsku ćeliju. Zaboravna vrata kontroliraju koliko starih informacija će biti zadržano u memorijskoj ćeliji. One ocjenjuju koliko relevantne su informacije iz prošlog stanja u odnosu na trenutno stanje i odlučuju koje informacije treba zaboraviti. Izlazna vrata kontroliraju koliko informacija će biti preneseno iz memorijske ćelije na izlaz. One ocjenjuju trenutno stanje i odlučuju koju informaciju treba prenijeti na izlaz.

# IZRADA PROJEKTNOG ZADATKA

Projekt je izrađen u programskom jeziku Python, koristeći biblioteke Sklearn, Tensorflow, i Numpy. Podaci koji se koriste su prikupljeni na platformi kaggle. Kreirani su 4 različita modela. Naive Bayesov model, konvolucijska mreža, LSTM mreža i BERT model su iskorišteni za rješavanje problema detekcije, koji provjerava je li Email „spam“ ili „ham“.

## Obrada podataka

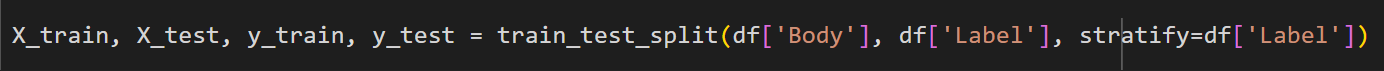
Učitavanje CSV datoteke s Google Drivea i spremanje podataka u varijablu df koristeći pandas biblioteku predstavlja početni korak u obradi podataka, prikazano na slici 7.1. Nakon što su podaci učitani, slijedi izdvajanje relevantnih informacija. U ovom slučaju, bitni podaci su Body, koji sadrži tekstualni sadržaj e-pošte, te Label koji označava da li je e-pošta označena kao "spam" (1) ili "ham" (0). Izdvajanje ovih podataka omogućuje fokusiranje na ključne elemente za analizu ili klasifikaciju e-pošte.

A picture containing text, screenshot, font

Description automatically generated

**Slika 7.1.** Učitavanje CSV datoteke

U drugom koraku, prikazano na slici 7.2, koristi se funkcija train\_test\_split iz scikit-learn biblioteke kako bi se podatci podijelili na skupove za treniranje i testiranje. Na ulazu se nalaze dva skupa podataka: df['Body'], koji sadrži tekstualni sadržaj e-pošte, te df['Label'], koji sadrži oznaku ("spam" ili "ham") za svaki e-mail. Funkcija train\_test\_split vrši podjelu podataka na četiri skupa: X\_train, X\_test, y\_train i y\_test. X\_train predstavlja skup podataka za treniranje koji sadrži tekstualni sadržaj e-pošte za obuku modela. X\_test predstavlja skup podataka za testiranje koji sadrži tekstualni sadržaj e-pošte za evaluaciju modela. y\_train predstavlja oznake za obuku modela, odnosno da li je svaki e-mail "spam" (označeno s 1) ili "ham" (označeno sa 0). y\_test predstavlja oznake za testiranje modela. Parametar stratify=df['Label'] osigurava da se omjeri razreda očuvaju prilikom podjele podataka. To znači da će omjer "spam" i "ham" e-mailova biti sličan u oba skupa, što je važno za uravnoteženu evaluaciju performansi modela.



**Slika 7.2.** Funkcija train\_test\_split

## Obrada podataka za naive bayesov model

CountVectorizer je klasa u biblioteci Scikit-learn koja se koristi za pretvaranje tekstualnih podataka u vektore. Ova tehnika se naziva "Bag of Words" i koristi se za reprezentaciju teksta kao numeričkih značajki koje se mogu koristiti u modelima strojnog učenja. Fit transformacija na trening podacima izvršava dvostruku funkcionalnost. Najprije, metoda fit\_transform() koristi podatke iz X\_train skupa podataka za izgradnju rječnika riječi koji će se koristiti za pretvaranje teksta u vektore brojanja. Nakon toga, izvršava se transformacija trening podataka, pretvarajući tekstualne podatke u vektore koristeći izgrađeni rječnik. Svaki vektor predstavlja jedan dokument (u ovom slučaju, email) i broj pojavljivanja svake riječi u tom dokumentu. Ova matrica se dalje može koristiti kao ulazni podaci za modele strojnog učenja koji zahtijevaju numeričku reprezentaciju teksta.

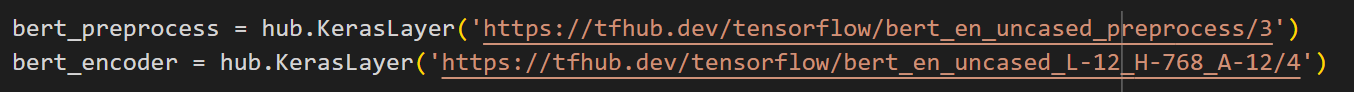
A screen shot of a computer program

Description automatically generated with low confidence

**Slika 7.2.** CountVectorizer

## Obrada podataka za BERT model

Sloj bert\_preprocess, prikazano na slici 7.3, koristi se za predobradu ulaznih podataka prije njihovog ulaska u BERT model. Kada se koristi ovaj sloj, podaci se prilagođavaju BERT modelu na način koji model zahtijeva. To uključuje tokenizaciju teksta, dodavanje posebnih tokena poput [CLS] i [SEP], pretvaranje teksta u brojčane reprezentacije itd. Ovaj sloj omogućava jednostavnu integraciju predobrade podataka s BERT modelom. Sloj bert\_encoder predstavlja BERT model koji će obrađivati ulazne podatke. BERT model sadrži unaprijed obučene težine i arhitekturu potrebnu za obradu teksta. Ovaj konkretni BERT model se naziva "bert\_en\_uncased\_L-12\_H-768\_A-12" i ima 12 slojeva (L), 768 skrivenih jedinica po sloju (H) i 12 glava pažnje (A). Keras sloj bert\_encoder omogućava integraciju BERT modela u TensorFlow model, što omogućava daljnju obradu i treniranje modela.



**Slika 7.3.** BERT preprocess i encoder

Input sloj specificira oblik ulaznih podataka, u ovom slučaju se očekuje samo jedan tekstualni podatak, kao što je prikzano na slici 7.4. shape=() znači da se očekuje 1D tensor (jedan tekstualni podatak), dtype=tf.string označava da se ulazni podaci očekuju kao nizovi znakova, a name="text" je ime sloja koje ga identificira. Ulazni tekst se onda prilagođava BERT modelu tako da se tokenizira. Rezultat predobrade je preprocesirani tekst koji je spreman za ulazak u BERT model. BERT model prima preprocesirani tekst i obrađuje ga kroz slojeve unutar modela. Rezultat izlaza modela su različiti reprezentativni vektori koji kodiraju semantičke značajke teksta.

A screenshot of a computer program

Description automatically generated with low confidence

**Slika 7.4.** BERT slojevi

## Obrada podataka za konvolucijsku mrežu

Tekstovi emailova se moraju pretvoriti u brojčane sekvence koje se mogu koristiti kao ulaz za konvolucijsku mrežu, gdje sve sekvence imaju istu duljinu. Tokenizer, prikazan slici 7.5, je objekt koji se koristi za pretvorbu teksta u sekvence tokena. Parametar num\_words=10000 označava da će se koristiti najčešćih 10000 riječi kao vokabular za pretvorbu. Metoda tokenizer.texts\_to\_sequences pretvara tekstualne podatke u sekvence brojčanih tokena. Svaka riječ se zamjenjuje njenim numeričkim indeksom u vokabularu. Funkcija pad\_sequences([], maxlen=100) osigurava da sve sekvence brojčanih tokena imaju istu duljinu. Ako je sekvencija kraća od zadane maksimalne duljine (100), tada se dodaju nule na kraj sekvence kako bi se postigla željena duljina.

A picture containing text, screenshot, font

Description automatically generated

**Slika 7.5.** Tokenizer za CNN

## Obrada podataka za LSTM mrežu

Kao i u slučaju konvolucijskih neuronskih mreža, potrebno je izvršiti pretvorbu tekstova emailova u brojčane sekvence. Postupak za pretvorbu tekstova u sekvence je identičan kao u prethodnom primjeru, gdje se koristi Tokenizer objekt za izgradnju vokabulara riječi i pretvorbu tekstualnih podataka u numeričke sekvence. Nakon toga, sekvence se dodatno obrađuju kako bi imale istu duljinu, što omogućuje konzistentno ulazno oblikovanje u model strojnog učenja, prikazano na slici 7.6.

A picture containing text, screenshot, font

Description automatically generated

**Slika 7.6.** Tokenizer za LSTM

# TRENIRANJE MODELA

Treniranje modela je ključna faza u postupku razvoja modela strojnog učenja. Tijekom treninga, model prolazi kroz iterativni proces prilagodbe parametara na temelju podataka kako bi minimizirao grešku i postigao što bolje performanse. Postupak treniranja modela uključuje odabir odgovarajuće arhitekture modela i definiranje funkcije gubitka (loss function) koja mjeri razliku između predikcija modela i stvarnih vrijednosti te optimizaciju parametara modela kako bi se minimizirala funkcija gubitka. Kroz više iteracija ili epoha treniranja, model postaje sve bolji u obavljanju željenog zadatka, povećavajući preciznost i sposobnost generalizacije na novim, neviđenim podacima. Važno je pravilno balansirati trening modela kako bi se izbjeglo prenaučenje (overfitting) na trening podacima, što bi rezultiralo slabijim performansama na novim podacima.

## Treniranje Naive Bayesov model

Stvara se instancu objekta MultinomialNB, kao što je prikazano na slici 8.1, što predstavlja Naivni Bayesov model koji se koristi za klasifikaciju podataka. U ovom slučaju, koristi se model s multinomijalnom distribucijom. Metoda model.fit izvršava postupak treniranja modela. Metoda fit() prima dvije ulazne varijable: X\_train\_count, koji predstavlja matricu vektora brojanja pretvorenu iz tekstualnih podataka, i y\_train, koji predstavlja ciljne vrijednosti (oznake) trening skupa. Model se trenira na ovim ulaznim podacima kako bi naučio odnose između vektora brojanja i njihovih pripadajućih oznaka.

A black background with white text

Description automatically generated with low confidence

**Slika 8.1.** Treniranje Naive Bayesov model

## Treniranje BERT modela

Prvi korak u treniranju BERT modela je dodavanje Dropout-a sloja, vodljivo na slici 8.2. Dropout je tehnika regularizacije koja služi za smanjenje prenaučenosti modela. Ovdje se koristi Dropout s vjerojatnošću odbacivanja od 0.15, što znači da će 15% neurona biti isključeno tijekom treninga radi smanjenja prenaučenosti. Idući sloj je gusti sloj, koji ima 1 izlazni neuron i koristi sigmoidnu aktivacijsku funkciju. Ovaj sloj ima zadatak izvršiti konačnu klasifikaciju, generirajući izlaznu vrijednost između 0 i 1 koja predstavlja vjerojatnost pripadnosti pojedinog primjera klasi. Konačni model spaja ulazni tekstualni sloj (text\_input) i izlazni sloj (l). Ovaj model definira arhitekturu mreže koja uzima tekstualni ulaz i generira izlaznu vjerojatnost klase.

A screen shot of a computer code

Description automatically generated with low confidence

**Slika 8.2.** Slojevi BERT modela

Nakon što je model definiran, potrebno ga je konfigurirati za trening postavljanjem određenih parametara, prikazano na slici 8.3. Ovdje se koristi Adam, popularan algoritam za optimizaciju u dubokom učenju. U ovom slučaju, koristi se binarna križna entropija, koja je pogodna za binarnu klasifikaciju (dva izlaza - 0 ili 1).

A picture containing text, screenshot, font

Description automatically generated

**Slika 8.3.** Kompajliranje BERT modela

Tijekom procesa treniranja, prikazano na slici 8.4, model će 10 puta iterativno prolaziti kroz trening skup podataka i prilagođavati svoje težine kako bi minimizirao funkciju gubitka i poboljšao svoju sposobnost predviđanja. Svaka epoha predstavlja jedno prolazak kroz cijeli trening skup podataka.

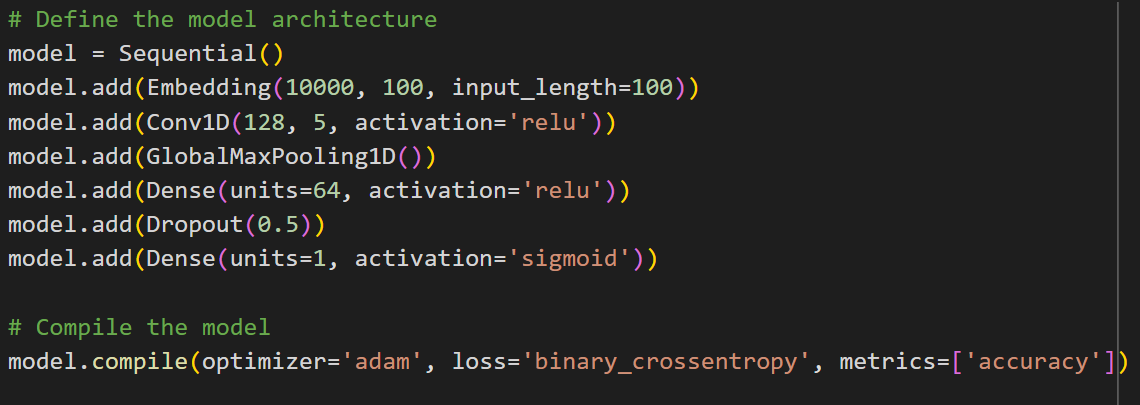
A screenshot of a computer

Description automatically generated with medium confidence

**Slika 8.4.** Treniranje BERT modela

## Treniranje konvolucijskog modela

Prvo se inicijalizira sekvencijalni model, što znači da će slojevi biti dodani jedan za drugim u redoslijedu. Zatim, dodaje se sloj ugnježđivanja (embedding layer). Ugnježđivanje se koristi za pretvaranje riječi u vektore fiksne duljine kako bi se mogli koristiti u neuronskoj mreži. Parametri ovog sloja su, broj riječi u vokabularu (10000), dimenzija ugnježđivanja (100) i duljina ulaznih sekvenci (100). Onda se dodaje konvolucijski sloj sa 128 filtera duljine 5. Konvolucijski slojevi koriste filtriranje preko ulaznih podataka kako bi izdvojili značajke. Aktivacijska funkcija ReLU (Rectified Linear Unit) se koristi za uklanjanje negativnih vrijednosti i dodavanje nelinearnosti. Dodaje se sloj globalnog maksimalnog izvlačenja (Global Max Pooling). Taj sloj izvlači najveću vrijednost iz svake mape značajki kako bi se dobila sažeta reprezentacija. Slijedi potpuno povezani sloj (Dense layer) sa 64 jedinica. Ovaj sloj ima potpunu povezanost između svakog čvora u prethodnom sloju i svakog čvora u ovom sloju. Aktivacijska funkcija ReLU se ponovno koristi. Onda dolazi Dropout sloj sa stopom isključivanja od 0.5. U konačnici, dodaje se završni sloj s jednom jedinicom. Ovaj sloj ima sigmoidnu aktivacijsku funkciju koja daje izlaznu vrijednost između 0 i 1, što je pogodno za binarnu klasifikaciju. Kompajlira se model s optimizatorom Adam.



**Slika 8.5.** slojevi CNN modela

Epochs određuje broj epoha, odnosno koliko puta će se model trenirati na cijelom skupu podataka. Metoda model.fit pokreće postupak treniranja. Tijekom treniranja, model će prilagođavati svoje težine i parametre kako bi minimizirao gubitak (definiran u prethodnom koraku) na temelju usporedbe između stvarnih i predviđenih ciljnih vrijednosti.

A screenshot of a computer program

Description automatically generated with medium confidence

**Slika 8.6.** Treniranje CNN modela

## Treniranje LSTM modela

Kao kod konvulucijske mreže, prvo se inicijalizira sekvencijalni model, koji omogućava dodavanje slojeva mreže jedan za drugim. Također, dodaje se embedding sloj koji mapira riječi u vektore niske dimenzionalnosti. Ovdje se koristi ugrađivanje dimenzionalnosti 100 za vokabular veličine 10000. Ulazna duljina podataka je postavljena na 100. Zatim, dolazi LSTM sloj, rekurentni sloj, koji ima sposobnost zadržavanja informacija na temelju konteksta i učenja dugoročnih zavisnosti. Onda dolazi dense sloj sa 64 jedinica i aktivacijom ReLU. Ovaj sloj služi za izvlačenje značajki iz izlaza LSTM sloja. Idući sloj je dropout sloj sa stopom isključivanja od 0.5. Posljednji sloj je povezani sloj s jednom jedinicom i aktivacijom sigmoid. Ovaj sloj daje izlaznu vrijednost koja se tumači kao vjerojatnost klase. Metoda model.compile konfigurira model za trening. Koristi se optimizator Adam za minimizaciju gubitka binarne križne entropije.

A picture containing text, screenshot, font

Description automatically generated

**Slika 8.7.** slojevi LSTM modela

Kroz pet prolaza (epochs) model se uči prilagoditi svoje parametre kako bi postigao bolju usklađenost s podacima i poboljšao performanse predviđanja.

A screenshot of a computer program

Description automatically generated with medium confidence

**Slika 8.8.** Treniranje LSTM modela

# ANALIZA MODELA

Cilj projekta je pokušaj pronalaska što boljeg modela kako bi klasifikacija Email-ova bila što točnija. Stoga se testiraju 4 različita modela. Na sljedećim slikama su prikazani matrice zabune i izvještaji o klasifikaciji. Matrica zabune (eng. confusion matrix) je tablični prikaz performansi klasifikatora na skupu podataka za testiranje. Matrica zabune se koristi u području strojnog učenja i statistike kako bi se procijenila točnost klasifikatora i otkrile njegove snage i slabosti.

Za testiranje modela koristi se metoda predict kako bi dobili predviđene ciljne vrijednosti za testni skup podataka X\_test. Model primjenjuje naučene parametre na ulazne podatke i generira predviđene izlaze. Metoda flatten se koristi kako bi se predviđene ciljne vrijednosti y\_predicted pretvorile u 1D niz. Ovo je korisno ako su izvorne ciljne vrijednosti ili oblik predviđenih izlaza u obliku višedimenzionalnog niza, a želimo ih predstaviti kao jednodimenzionalni niz.

A black background with white text

Description automatically generated with low confidence

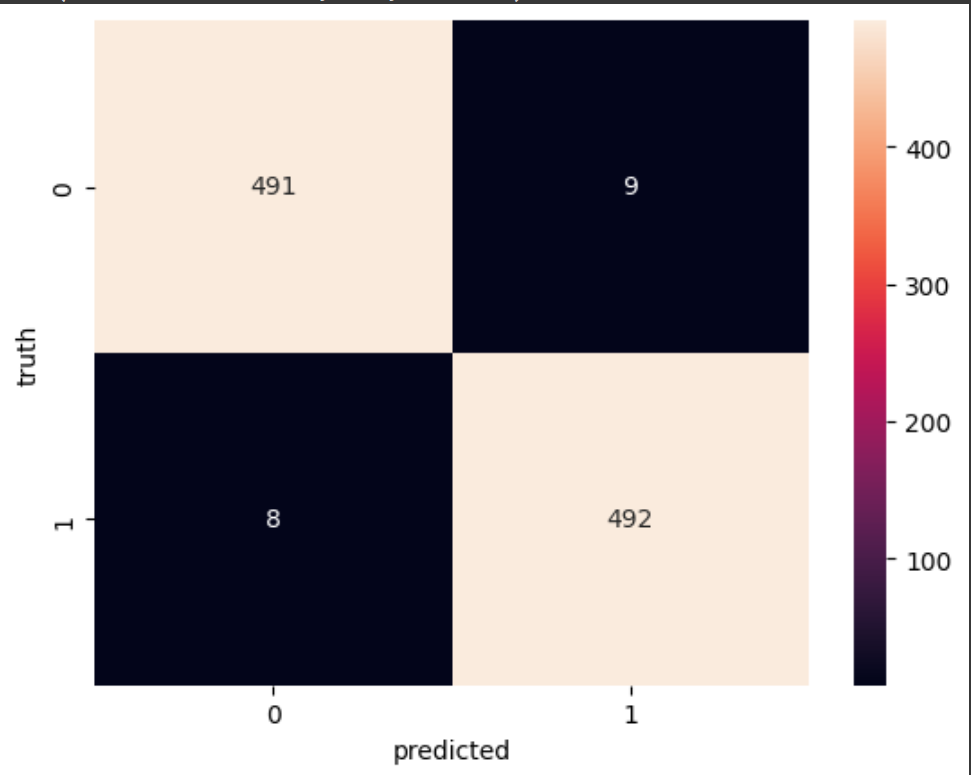
**Slika 9.1.** predviđene vrijednosti

U idućem koraku, koristi se funkcija np.where iz biblioteke NumPy kako bi se prošlo kroz sve vrijednosti u y\_predicted. Ako je vrijednost veća od 0.5, dodjeljuje se vrijednost 1, inače se dodjeljuje vrijednost 0. Drugim riječima, ako je predviđena vrijednost veća od 0.5, smatra se da je pozitivna (1), inače se smatra negativnom (0).



**Slika 9.2.** Podešavanje rezultata predviđene vrijednosti

## Analiza Naive Bayesovog modela



**Slika 9.3.** Matrica zabune za Naive Bayesov model

A screenshot of a computer

Description automatically generated with low confidence

**Slika 9.4.** Izvješće o klasifikaciji za Naive Bayesov model

## Analiza BERT modela

A picture containing text, screenshot, diagram, design

Description automatically generated

**Slika 9.5.** Matrica zabune za BERT model

A screenshot of a computer

Description automatically generated with low confidence

**Slika 9.6.** Izvješće o klasifikaciji za BERT model

## Analiza konvolucijskog modela

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

**Slika 9.7.** Matrica zabune za konvolucijski model

A screenshot of a computer

Description automatically generated with low confidence

**Slika 9.8.** Izvješće o klasifikaciji za konvolucijski model

## Analiza LSTM modela

A picture containing screenshot, text, diagram, square

Description automatically generated

**Slika 9.9.** Matrica zabune LSTM model

A screenshot of a computer

Description automatically generated with low confidence

**Slika 9.10.** Izvješće o klasifikaciji za LSTM model

## Rezultati

Najiznimniji rezultat je ostvaren upotrebom Naive Bayesovog modela koji je postigao izuzetno visoku preciznost od 98%. Konvolucijski model i LSTM model nisu daleko iza, ostvarujući izvanrednu preciznost od 97%. S druge strane, BERT model se ističe s najnižom preciznošću od 88%, no važno je napomenuti da je BERT sofisticirana arhitektura koja je možda pretjerana za ovako jednostavan problem klasifikacije.

# ZAKLJUČAK

U zaključku ovog rada prikazano je istraživanje razvoja filtra za emailove s neželjenom poštom primjenom tehnika strojnog učenja. Korištene su tehnologije poput Numpy-a, TensorFlow-a, BERT-a i scikit-learn-a kako bi se izradili i evaluirali različiti modeli, uključujući konvolucijsku neuronsku mrežu (CNN), LSTM mrežu, BERT model i Naive Bayes model.

Rezultati su pokazali da je Naive Bayes model ostvario najbolje rezultate, postigavši impresivnu razinu točnosti. Međutim, važno je istaknuti da su CNN model i LSTM model slijedili vrlo blizu, pokazujući svoju učinkovitost u rješavanju zadatka klasifikacije emailova s neželjenom poštom. S druge strane, BERT model je pokazao najslabiju izvedbu, što se može pripisati njegovoj složenosti i potencijalno pretjeranim mogućnostima za ovaj specifičan problem.

Općenito, ovo istraživanje ističe važnost korištenja tehnika strojnog učenja u razvoju pouzdanih filtara za emailove s neželjenom poštom. Rezultati naglašavaju učinkovitost tradicionalnih modela poput Naive Bayes-a te dubokih modela poput CNN-a i LSTM-a. Iako je izvedba BERT modela možda bila manje zadovoljavajuća za ovaj konkretan problem, on služi kao podsjetnik da je potrebno prilagoditi složenost modela prema zahtjevima zadatka. Buduća istraživanja mogu istražiti daljnje usavršavanje i optimizacije kako bi se poboljšala izvedba ovih modela, te razmotriti alternativne pristupe u rješavanju izazova klasifikacije emailova s neželjenom poštom.

# LITERATURA

[1] TensorFlow - https://www.integrate.io/glossary/what-is-tensor-flow/

[2] BERT - https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[3] NumPy - https://www.w3schools.com/python/numpy/numpy\_intro.asp

[4] Sklearn - https://en.wikipedia.org/wiki/Scikit-learn

[5] MathPlotLib - https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/#:~:text=Matplotlib%20is%20a%20cross%2Dplatform,open%20source%20alternative%20to%20MATLAB.

[6] Strojno učenje - https://www.ibm.com/topics/machine-learning#:~:text=the%20next%20step-,What%20is%20machine%20learning%3F,learn%2C%20gradually%20improving%20its%20accuracy.

[7] Naive Bayes - https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/#:~:text=Frequently%20Asked%20Questions-,What%20Is%20the%20Naive%20Bayes%20Algorithm%3F,presence%20of%20any%20other%20feature.

[8] Duboko učenje - https://aws.amazon.com/what-is/deep-learning/#:~:text=Deep%20learning%20is%20a%20method,produce%20accurate%20insights%20and%20predictions.

[9]LSTM model- https://www.tutorialspoint.com/time\_series/time\_series\_lstm\_model.htm#:~:text=An%20LSTM%20module%20has%20a,only%20a%20few%20linear%20interactions.