SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

PREPOZNAVANJE ZVUKA PSA I MAČKE

Seminarski rad

Hrvoje Zec

Osijek, godina. 2023

##### SADRŽAJ

[1. UVOD 1](#_Toc127741679)

[2. KORIŠTENJE TEHNOLOGIJE I ALATI 2](#_Toc127741680)

[2.1. Nadzirano učenje 2](#_Toc127741681)

[2.1.1. Klasifikacija 2](#_Toc127741682)

[2.1.2. Neuronska mreža 3](#_Toc127741683)

[3. IZRADA PROJEKTA ZA PREPOZNAVANJE ZVUKA PSA I MAČKE 4](#_Toc127741684)

[3.1. Podatci za treniranje 4](#_Toc127741685)

[3.2. Model, treniranje i testiranje 5](#_Toc127741686)

[4. TESTIRANJE NA VLASTITOM TESTNOM SKUPU 7](#_Toc127741687)

[5. ZAKLJUČAK 9](#_Toc127741688)

[LITERATURA 10](#_Toc127741689)

[PRILOZI 11](#_Toc127741690)

[P.1. Izvorni kod aplikacije 11](#_Toc127741691)

# UVOD

U ovom seminarskom radu obrađuje se projektni zadatak prepoznavanja zvuka pas i mačke. Za ovaj projekat će se koristiti Python, popularni programski jezik za znanstveno računanje, analizu podataka i strojno učenje, kao i Anaconda, razvojno okruženje koje nudi sve potrebne alate i biblioteke za razvoj strojnog učenja. Korištenje nadziranog učenja podrazumijeva upotrebu klasifikacijskih algoritama i neuronskih mreža kako bi se prepoznali uzorci u podacima, što je osnova za razvoj algoritma za prepoznavanje zvuka pasa i mačaka.

U ovom radu će se detaljno opisati postupak pripreme podataka, treniranja i testiranja algoritma, te će se analizirati dobiveni rezultati. Korištenjem skupa podataka preuzetog iz platforme Kaggle [1], algoritam će se trenirati na velikom broju primjera, te će se zatim testirati na dodatnom vlastitom testnom skupu podataka.

Cilj ovog projekta je razviti precizan i pouzdan sustav prepoznavanja zvuka pasa i mačaka, što može imati primjenu u mnogim područjima, poput nadzora sigurnosti ili kontrole buke.

# KORIŠTENJE TEHNOLOGIJE I ALATI

U ovom projektnom zadatku, korištene su razne tehnologije i alati kako bi se razvio sustav za prepoznavanje zvuka pasa i mačaka. Prije svega, za razvoj projekta korišten je Python, popularni programski jezik za znanstveno računanje, analizu podataka i strojno učenje. Kako bi se olakšao razvoj i rad na projektu, korišteno je razvojno okruženje Anaconda koje nudi sve potrebne alate i biblioteke za razvoj strojnog učenja. Za izradu projekta korišten je interaktivni radni okvir Jupyter Notebook koji omogućava brzo eksperimentiranje, dokumentiranje i vizualizaciju podataka. Korišteno je nadzirano učenje u kojem su podaci označeni, te klasifikacija i neuronske mreže za prepoznavanje uzoraka u podacima. Kako bi se testirala učinkovitost algoritma, korišteni su skupovi podataka preuzeti s platforme Kaggle. Također, dodan je vlastiti testni skup podataka kako bi se dalje evaluirala performansa algoritma.

## Nadzirano učenje

Nadzirano učenje [2] je jedan od najčešćih oblika strojnog učenja koji se koristi za klasifikaciju, regresiju i predviđanje. Ovaj pristup koristi označene podatke za treniranje algoritama strojnog učenja tako da se nauče klasificirati nove primjere. To znači da je nadzirano učenje proces u kojem algoritam uči kako povezati ulazne podatke s izlaznim podacima na temelju skupa primjera za učenje.

U praksi, ovaj pristup uključuje pripremu podataka za učenje, izbor i oblikovanje značajki, odabir odgovarajućeg modela za klasifikaciju i treniranje modela na označenim primjerima. Nakon što je model treniran, testira se na neoznačenim primjerima kako bi se procijenila njegova točnost i učinkovitost. Nadzirano učenje se koristi u mnogim primjenama, uključujući prepoznavanje slika, prepoznavanje govora, klasifikaciju tekstova, predviđanje cijena i još mnogo toga.

### Klasifikacija

Klasifikacija [3] je jedan od najčešćih zadataka u nadziranom učenju, a odnosi se na proces kategoriziranja ulaznih podataka u diskretne klase ili kategorije. U ovom pristupu, model strojnog učenja uči kako klasificirati nove primjere na temelju skupa primjera za učenje, gdje su ulazni podaci povezani s odgovarajućim izlaznim klasama. Na primjer, možemo trenirati model koji će razlikovati zvukove psa od zvuka mačke. U tom slučaju, model bi naučio kako klasificirati nove zvukove kao pas ili mačku na temelju značajki koje su naučene tijekom treniranja.

U procesu klasifikacije, algoritam strojnog učenja prolazi kroz skup značajki koji su izvađeni iz ulaznih podataka kako bi odlučio koju klasu dodijeliti novom primjeru. Algoritam se uči na primjerima za učenje, povezujući ulazne značajke s odgovarajućim klasama. Kada se primijeni na nove primjere, model klasificira ulazne podatke u jednu od naučenih klasa.

Klasifikacija se koristi u mnogim primjenama strojnog učenja, uključujući prepoznavanje slika, klasifikaciju teksta, dijagnostiku medicinskih podataka i još mnogo toga.

### Neuronska mreža

Neuronska mreža [4] je vrsta algoritma strojnog učenja koji je inspiriran načinom na koji radi ljudski mozak. Neuronske mreže sastoje se od više slojeva neurona koji su povezani zajedno i mogu se koristiti za rješavanje složenih problema poput klasifikacije, regresije i predviđanja.

Neuronske mreže se sastoje od tri glavna dijela: ulazni sloj, skriveni slojevi i izlazni sloj. Ulazni sloj prima ulazne podatke koji se prenose kroz skriveni sloj, gdje se značajke podataka izvlače i obrađuju. Izlazni sloj potom generira izlazne vrijednosti ili klasifikaciju.

Svaki neuron u neuronskoj mreži obično ima svoje težine koje se prilagođavaju tijekom treniranja kako bi se poboljšala točnost klasifikacije. Treniranje se obično vrši pomoću algoritama gradijentnog spusta, koji minimiziraju pogrešku klasifikacije u skupu za učenje.

Neuronske mreže se mogu koristiti za razne zadatke strojnog učenja, uključujući prepoznavanje slika, klasifikaciju teksta, dijagnostiku medicinskih podataka, preporučivanje sadržaja, i još mnogo toga. Uz odgovarajuću arhitekturu i treniranje, neuronske mreže mogu postići visoku točnost klasifikacije na složenim zadacima.

# IZRADA PROJEKTA ZA PREPOZNAVANJE ZVUKA PSA I MAČKE

Ovo je seminarski rad o prepoznavanju zvuka pasa i mačaka u Pythonu. Projektni rad koristi bazu podataka s Kagglea i vlastitu bazu podataka. Korištene su biblioteke i modeli strojnog učenja, uključujući logističku regresiju, SVM, FNN i LSTM. Projekt se može razdijelit na tri osnovna koraka, kao što su ekstrakcija značajki zvuka uz pomoć biblioteke *librosa* koja pretvara zvuk u niz značajki, kao što je mfcc, sljedeći korak je priprema podataka za strojno učenje, tu smo koristili *LabelEncoder* za kodiranje oznake psa i mače u numerički oblik, te *train\_test\_split* za podjelu skupa podataka na skupove za treniranje i testiranje. Zadnji korak je trening i testiranje modela strojnog učenja. U ovom koraku se koriste različiti modeli strojnog učenja, uključujući logističku regresiju, SVM, dvije vrste FNN-a i LSTM. Korištena je biblioteka *scikit-learn* za evaluaciju modela, a *confusion matrix* i *classification report* za vizualizaciju točnosti i performansi modela.

## Podatci za treniranje

Skup podatak koji smo koristili za treniranje i testiranje su preuzeti sa kaggle stranice, te su učitane u naš projektni zadatak. Skup podataka se sastoji od dvije kategorije zvučnih datoteka, a to su pas ili mačka. Prolaskom kroz skup podataka uzimamo sve datoteke koje su vrste „wav“. To nam je potrebno kako bi izvukli oznake iz naziva datoteka u skupu podataka kako bi ih se mogle koristiti u procesu strojnog učenja. Za pripremu trening i test podataka koristit ćemo biblioteku *pandas,* kako bi stvorili DataFrame *train\_data* i *test\_data,* koje sadržavaju informacije o putanjama datoteka i njihovim oznakama. Nadalje, učitavamo 4 nasumične zvučne datoteke psa ili mačke kako bi ih vizualno prikazali uz pomoć funkcije *librosa.display.waveshow().* Naposljetku, kako bi obradili zvučne datoteke koristit ćemo metodu mfcc da bi dobili srednju vrijednost sve datoteke. Zatim ih sve spremamo u *mfcc\_features\_df*, gdje svaki redak sadrži vektorsku reprezentaciju značajki i pripadajuću oznaku klase, odnosno x vrijednosti koje nam predstavljaju ulazne podatke u formatu (210,40), te y vrijednosti odnosno izlazni podaci u formatu (210,1)(slika 3.1.).

Text, letter

Description automatically generated

***Slika 3.1****. Prikaz* *mfcc\_features\_df*

Nakon toga se podaci za treniranje i testiranje dijele na skupove *x\_train, x\_test, y\_train* i *y\_test.* Kako koristimo logističku regresiju i SVM klasifikaciju, oznake klase se mijenjaju tako da pretvaramo iz binarnog zapisa u oblik di nam klasa 'mačka' postaje 1, a klasa 'pas' postaje 0 (slika 3.2.).

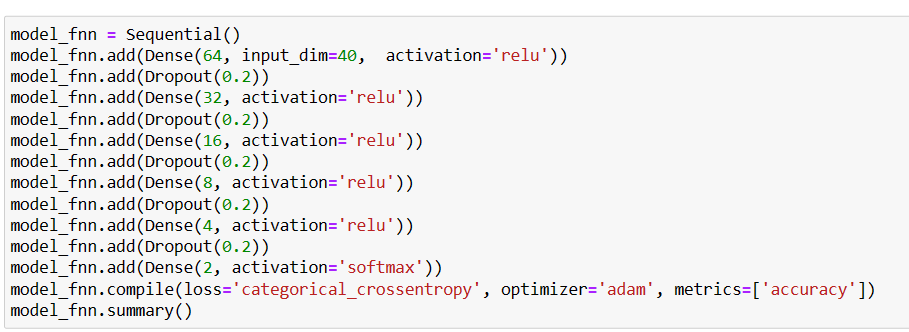
Graphical user interface, text, application

Description automatically generated

***Slika 3.2****. Prikaz podjele podataka*

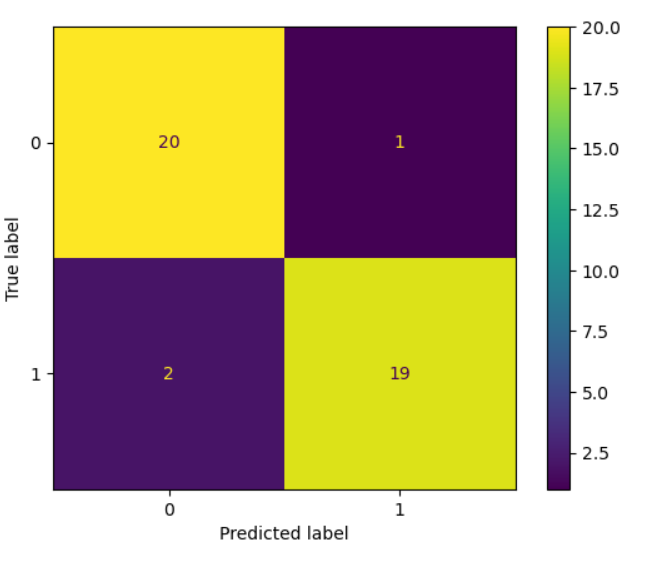
## Model, treniranje i testiranje

Za treniranje i testiranje danih podataka provodimo kroz pet različitih modela SVM, logističku regresiju, FNN, FNN2 i LSTM. U modelu SVM koristim *C\_range* i *gama\_range*, te nam je cilj dobiti hiperravninu koja najbolje razdvaja podatke između dvije klase u prostoru značajki. Logistička regresija je algoritam binarne klasifikacije koji koristi logističku funkciju kako bi modelirao vjerojatnost da će se određeni primjerak pripadati jednoj od dvije klase. FNN (Feedforward Neural Network) je vrsta neuronske mreže koja koristi više slojeva neurona kako bi modelirala složene odnose između ulaznih podataka i ciljne varijable. Ulazni podatci su nam veličine 40, te ih množimo sa težinom i prosljeđujemo ih kroz mrežu neurona. Praksa je pokazala da je najbolje koristiti brojeve na potenciju broja dva kao što su 64, 32, 16, 8, 4, 2.(slika3.3) Potom, moramo doći do broja dva jer imamo dvije izlazne veličine a to su pas i mačka. Kako ne bi došlo do prenaučenja (engl. *overfitting*) modela koristili smo funkciju *EarlyStopping.* Ovaj postupak prekida treniranje modela kada se postigne određeni kriterij zaustavljanja, kao što je prestanak poboljšanja funkcije gubitka (engl. *loss function*) na skupu za validaciju tijekom nekoliko uzastopnih epoha. Na sličnom principu radi i metoda FNN2 samo sa drugim brojem slojeva i neurona. Na kraju, LSTM (Long Short-Term Memory) je vrsta neuronske mreže koja se koristi za obradu sekvencijalnih podataka, kao što su riječi u tekstu ili zvukovi u zvučnom zapisu.



***Slika 3.3*** *prikaz FNN*

Nakon prolaska kroz svih pet metoda, te uz pomoć *confusion matrix, classification report* i prikazivanja *accuracy-a* i *loss* funkcija, dolazimo do zaključka da je najbolja metoda LTSM (slika3.4), a najgora FNN (slika3.5).

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

***Slika 3.4.*** *rezultati metode LTSM*

Chart

Description automatically generated Chart, line chart

Description automatically generated Chart, histogram

Description automatically generated

***Slika 3.5.*** *rezultati metode FNN*

Prema slikama 3.4 i 3.5 promatrajuću *confusion matrix* možemo vidjeti kako LTSM metoda u polju [0][0] ima 20 pogođenih zvukova pasa, te u polju[1][1] 19 pogođenih mačaka. Odnosno *confusion matrix* se sastoji od x labele koja predstavlja *true label* i y labele koja predstavlja *predicted label,* što nam govori da je uz pomoć LTSM metode od 21-jednog ulaza za zvuk psa pogođeno 20, a jedan je prepoznao kao mačku, te kod mačke pogođeno 19, a dva prepoznata kao pas. Nadalje, kod loss funkcije vidimo da je otprilike oko 6 epohe, da nam trening podaci padaju, a test podaci rastu te postaju gori za predvidjeti. Što nam govori da dolazi do prenaučenja (engl. *overfitting*). Kako u našem kodu koristimo funkciju *EarlyStopping* nećemo koristiti svih 500 epoha, nego samo one do koji vrijednost ima svoje značenje, jer se svi kasnije rezultati slabo mijenjaju, te ne utječu toliko na program. Kod FNN metode u *confusion matrix* možemo vidjeti da je pogodio sve ulaze zvuka psa da su psi, a kod zvuka mačke je od 21 pogodio samo 4.

# TESTIRANJE NA VLASTITOM TESTNOM SKUPU

U našem projektu dodajemo novi skup podataka radi boljeg testiranja programa na novim podatcima. Prvo učitavamo nove zvučne datoteke iz mape "barkmeow" u obliku *wav* datoteka. Zatim se izvlače Mel-frequency cepstral coefficients (MFCC) značajke iz tih datoteka. MFCC značajke se zatim koriste za obuku i evaluaciju različitih klasifikacijskih modela.

Prvo se stvaraju popisi imena novih datoteka i oznaka koji se koriste za stvaranje Pandas DataFramea koji sadrži putanju do svake datoteke i pripadajuću oznaku.

Iz DataFramea se izvlače putanje do svake datoteke i oznake koje se zatim koriste za izvlačenje MFCC značajki iz datoteka pomoću funkcije "*extract\_mfcc\_features*". Rezultati izvlačenja značajki se zatim pohranjuju u listu "*mfcc\_features*", koja se koristi za stvaranje DataFramea "*mfcc\_features\_df*" koji sadrži MFCC značajke i pripadajuće oznake klasa.

Zatim se MFCC značajke pretvaraju u oblik pogodan za korištenje u klasifikacijskim modelima, a klasifikacijski modeli se koriste za predviđanje oznaka klasa za nove zvučne datoteke. Izvještaji o klasifikaciji, matrice konfuzije i točnost modela na novim datotekama se zatim ispisuju i prikazuju za svaki model (logistička regresija, SVM, FNN, FNN2 i LSTM)(slika 3.6).

Text

Description automatically generated

***Slika 3.6*** *Prikaz tablice točnosti*

Prema tablici točnosti prikazanoj na slici 3.5. možemo zaključiti, da je na novim podatcima koji rezultati pokazuju da je LSTM model ostvario najbolju točnost u prepoznavanju zvuka pasa i mačaka, dok je FNN bila najlošija. Ova metoda bi mogla biti korisna za razvoj sustava za prepoznavanje zvuka pasa i mačaka u stvarnom vremenu, poput sustava za nadzor kućnih ljubimaca.

# ZAKLJUČAK

Ovaj seminarski rad istražuje primjenu različitih tehnika strojnog učenja za klasifikaciju zvukova mačaka i pasa. U radu su korištene pet vrsta modela: logistička regresija, SVM, dvije vrste FNN-a, te LSTM.

Eksperimentalni rezultati pokazuju da se najbolja performansa postiže s korištenjem neuronskih mreža, posebno LSTM arhitekture, koja postiže najveću točnost klasifikacije na testnom skupu podataka. Također, različite vrste modela imaju različite prednosti i mane, pa se za neke primjene može preferirati jedna metoda u odnosu na drugu.

Zaključno, ovaj rad pruža dobar pregled primjene različitih tehnika strojnog učenja za klasifikaciju zvukova mačaka i pasa te potvrđuje da su neuronske mreže učinkovitije u takvim primjenama.

# LITERATURA

**[1]** Kaggle, Audio dataset cats and dogs, dostupno na: [https://www.kaggle.com/dataset](https://www.kaggle.com/datasets/mmoreaux/audio-cats-and-dogs)[18.2.2023]

**[2]** Supervised learning, What is supervised learning?, dostupno na: [https://www.ibm.com](https://www.ibm.com/topics/supervised-learning) [18.2.2023]

**[3]** Clasification, Clasification in machine learning, dostupno na:

[https://www.simplilearn.com/](https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning)[18.2.2023]

**[4]** Neural network, What is neural network?, dostupno na: [https://deepai.org/](https://deepai.org/machine-learning-glossary-and-terms/neural-network) [18.2.2023]

# PRILOZI

## P.1. Izvorni kod aplikacije

**Dostupno na:** <https://github.com/HrvojeZec/Hrvoje_Zec_RUSU_Projekt.git>