

**Klasifikacija glazbenih žanrova**

*-seminarski rad-*

**Kolegij**: Osnove digitalne obrade govora i slika

**Mentor**: doc. dr. sc. Miran Pobar

**Studenti**: Maja Trstenjak

*Rijeka, rujan 2020.*

Sadržaj

[UVOD 3](#_Toc50551255)

[DATASET 4](#_Toc50551256)

[Spektrogram 4](#_Toc50551257)

[Librosa 5](#_Toc50551258)

[Proučavane značajke 6](#_Toc50551259)

[Mel-frequency cepstral coefficients (MFCCs) 6](#_Toc50551260)

[Chroma Frequencies 7](#_Toc50551261)

[Root Mean Square Error (rmse) 7](#_Toc50551262)

[Spectral Centroid 8](#_Toc50551263)

[Spectral Bandwidth 9](#_Toc50551264)

[Spectral Roll-off 9](#_Toc50551265)

[Zero Crossing Rate 10](#_Toc50551266)

[Zapis značajki u csv datoteku 11](#_Toc50551267)

[Obrada podataka 12](#_Toc50551268)

[NEURONSKA MREŽA 13](#_Toc50551269)

[Validacija 15](#_Toc50551270)

[Predviđanje 16](#_Toc50551271)

[REZULTATI 18](#_Toc50551272)

[ZAKLJUČAK 22](#_Toc50551273)

[POPIS SLIKA 23](#_Toc50551274)

[POPIS TABLICA 23](#_Toc50551275)

[LITERATURA 24](#_Toc50551276)

# UVOD

Tema ovog rada je klasificiranje glazbe po žanrovima.

Klasifikacija glazbenih žanrova bazirana na vizualnoj reprezentaciji je zastupljena u istraživanjima posljednjih godina. Klasifikatori trenirani sa teksturalnim deskriptorima ekstrahirani iz spektrograma su postigli zadivljujuće rezultate na nekoliko glazbenih datasetova.

U ovom radu modeliran je *klasifikator* za klasificiranje glazbe u 10 različitih glazbenih žanrova.



Korišten je GTZAN dataset koji sadrži 1000 zapisa duljine 30 sekundi, podijeljenih u 10 žanrova po 100 pjesama. Sadržani žanrovi su blues, klasična glazba, country, disco, hiphop, jazz, reggae, rock, metal i pop.

Za rad bilo je potrebno izvući značajke iz audio zapisa da bi se algoritmom za klasifikaciju zapisi klasificirali po žanrovima.

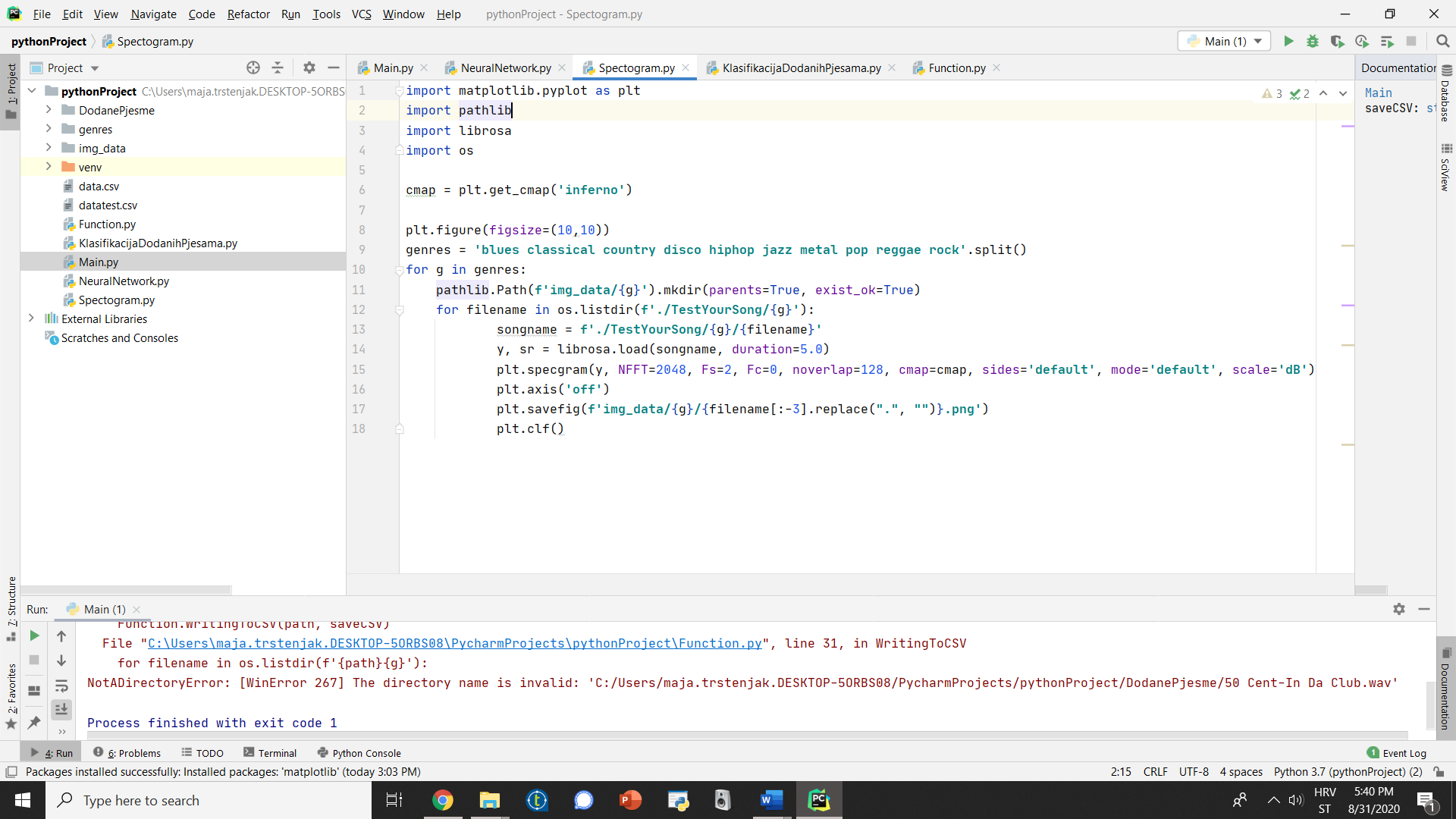
# DATASET

U svrhu ovog rada korišten je GTZAN dataset od 1000 zapisa duljine 30 sekundi, podijeljenih u 10 žanrova po 100 pjesama. Sadržani žanrovi su blues, klasična glazba, country, disco, hiphop, jazz, reggae, rock, metal i pop. Navedeni žanrovi pohranjeni su u varijablu *genres*.

Zatim su svi audio zapisi konvertirani u *spektrograme* za lakše ekstrahiranje značajki.

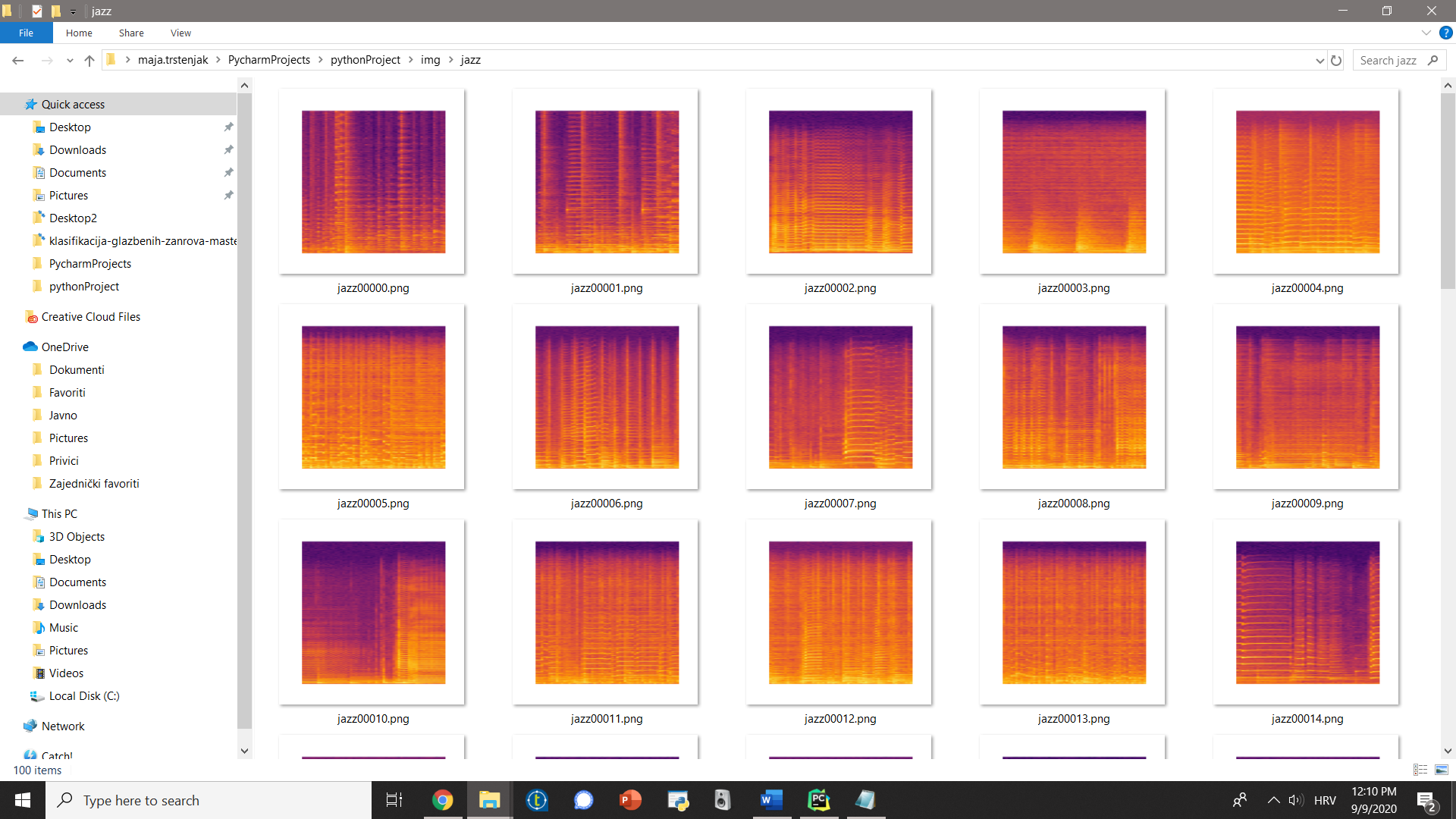
## Spektrogram

Spektrogram je vizualna reprezentacija frekvencija zvuka ili drugih signala koji variraju kroz vrijeme. U dvodimenzionalnom prostoru prva os predstavlja frekvenciju, a druga vrijeme. Spektrogram se može prikazati naredbom *librosa.display.specshow*.



*Slika 1. - Python file Spectrogram*

Slika 1. prikazuje kod konverzije audio zapisa u spektrograme, te se i na temelju njih iz zapisa mogu izvlačiti značajke koje se žele proučavati.



*Slika 2. – Prikaz nekoliko generiranih spektrograma za žanr jazz*

## Librosa

Librosa je python paket za analizu glazbe i zvuka. Pruža građevne blokove potrebne za stvaranje sustava za pronalaženje glazbenih podataka.



*Slika 3. – Logo Librosa*

U nastavku je u 2 linije koda za primjer prikazano izvlačenje MFCC karakteristika *librosa* bibliotekom i podržanim metodama korišteno u praktičnom dijelu ovog rada.

Y, sr = librosa.load(songname, mono=True, duration=30)

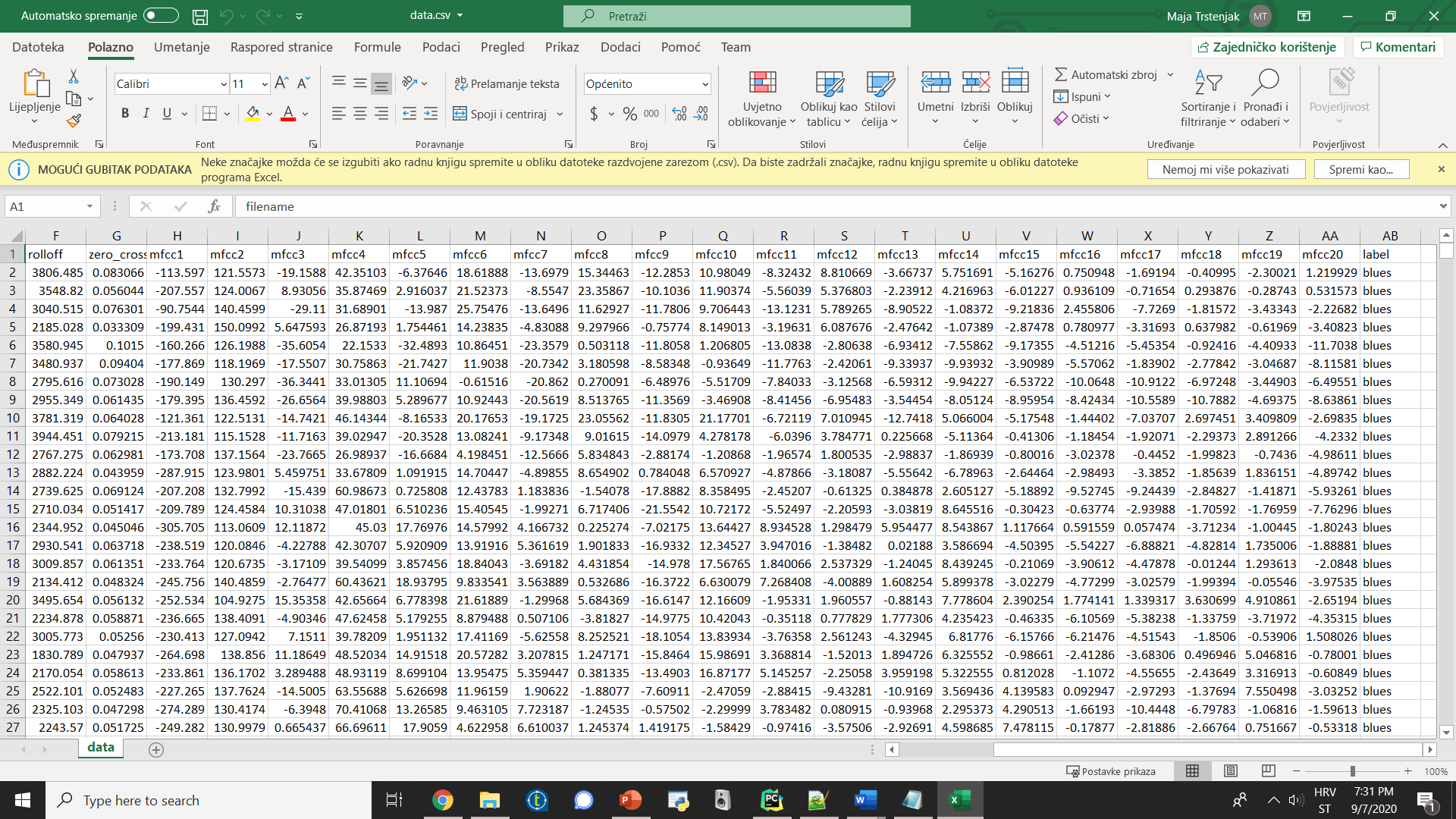
mfcc = librosa.feature.mfcc(y=y, sr=sr)

## Proučavane značajke

Vađenje značajki je važan dio analize, podaci koje zvuk daje modeli ne mogu izravno razumjeti, stoga ih je potrebno pretvoriti u razumljiv format.

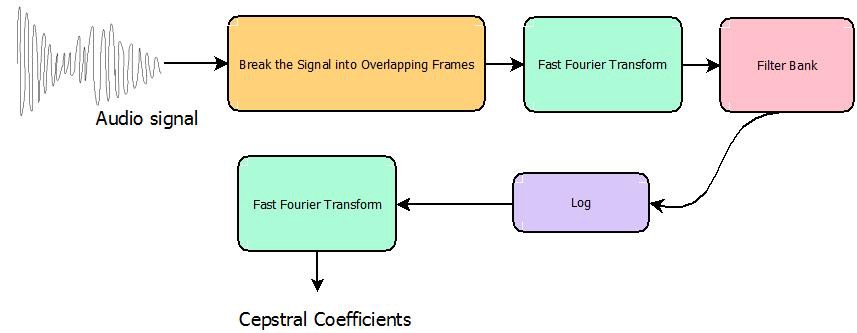
### Mel-frequency cepstral coefficients (MFCCs)

MFCC jemali skup značajki (10-20, za ovaj rad 20) koje opisuju ukopni oblik spektralne ovojnice. Dakle, to su koeficijenti koji zajedno čine MFC, te su izvedeni iz centralne reprezentacije audio zapisa. MFC je prikaz kratkotrajnog spektra snage zvuka temeljen na linearnom kosinusu od transformacije logaritamskog spektra na nelinearnoj mel-skali frekvencije. Kod MFC-a su frekvencijski pojasevi podjednako raspoređeni na mel ljestvici, što aproksimira reakciji ljudskog slušnog sustava sličnije nego linearno raspoređeni frekvencijski pojasevi korišteni u običnom cepstrumu.



*Slika 4. – Vrijednosti 20 MFCC koeficijenata za jedan od zapisa*

Bilo koji zvuk generiran od ljudi je određen oblikom njihovog glasovnog trakta. Ako se ovaj oblik može pravilno odrediti, svaki proizvedeni zvuk može se točno predstaviti. Omot vremenskog spektra snage govornog signala reprezentativan je za vokalni trakt, a tada MFCC točno predstavlja ovojnicu. U nastavku je prikazan blok dijagram koji je sažetak kako se dolazi do MFCC-a.



*Slika 5. – MFCC*

Postupak prikazan koracima na prethodnoj slici je Postupak vađenja značajki MFCC algoritmom koji započinje uzimanjem Fourierove transformacije signala, nakon toga se mapiraju tonovi spektra na mel – skalu koristeći preklapajuće okvire. Idući korak je uzimanje zapisa tonova za svaku mel – frekvenciju, te slijedi uzimanje diskretne kosinusove transformacije svih tonova kao da su signali. Konačno, MFC koeficijenti su amplitude dobivenog spektra.

### Chroma Frequencies

Pojam kroma značajka ili chromagram se odnosi na 12 različitih klasa tona glazbene oktave koje prikazuju koliko je energije od svake klase tona prikazano u signalu. Svaki ton se razlikuje po oktavi, dok svaka oktava ima 12 polutonova. Navedeni skup tonova sastoji se od 12 – C, C#, D, D#, E, F, F#, G, G#, A, A# i B. Glavno svojstvo kroma značajki je ta da hvataju harmonične i melodijske karakteristike glazbe. Ideja značajki je da sažmu u određenom vremenskom okviru informacije koje se odnose na dani audio zapis u jedinstveni koeficijent. Pomicanje vremenskog prozora rezultira promjenama u nizu karakteristika krome, pri čemu se ona svaki put raspodjeljuje na 12 dijelova. Za potrebe ovog rada koristi se normalizirana energija za svaku kroma značajku na svakom okviru.

### Root Mean Square Error (rmse)

RMSE je standardna devijacija residuala (pogreške predviđanja), tj. odstupanje srednje vrijednosti korijena. Residual je mjera koliko su točke podataka regresijske linije udaljene. RMSE je mjera koliko su residuali rašireni, te govori kako su podaci koncentrirani oko linije najbolje optimizacije.

,

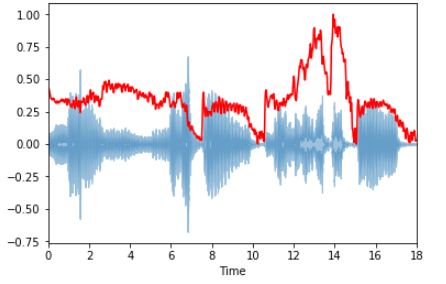
gdje je f nepoznati rezultat i o poznati rezultat.

### Spectral Centroid

Spektralni centroid je mjera koja se koristi u digitalnoj obradi signala za karakterizaciju spektra, a pokazuje gdje se nalazi središte mase spektra; svaki okvir spektrograma je normaliziran, a centroid dobivamo aritmetičkom sredinom za svaki okvir po formuli:

*centroid[t] = sum\_k S[k, t] \* freq[k] / (sum\_j S[j, t]),*

gdje je x(n) magnituda od pretinca n i f(n) je centar frekvencije tog pretinca.

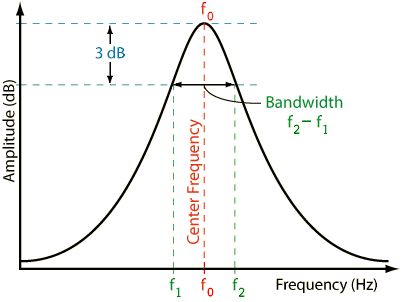


*Slika 6. - Primjer za prikaz porasta spektralnog centroida prema kraju*

Za bolje objašnjenje spektralnog centroida uspoređene su metal i blues pjesme: Metal pjesma ima više frekvencija prema kraju u usporedbi s blues pjesmom, koja je ista tijekom cijele duljine, pa spektralni centroid za blues pjesmu leži negdje blizu sredine spektra, dok bi za metal pjesmu bio pri kraju.

### Spectral Bandwidth

Spektralna širina pojasa je razlika između gornje i donje frekvencije u kontinuiranom frekvencijskom pojasu.



*Slika 7. – Spektralna širina pojasa*

Spektralna širina pojasa po okviru se računa po formuli:

*BW = f2 − f1= f0/Q,*

gdje je f0 centar frekvencije i Q je Q faktor

*Q=f0/(f2-f1)*

### Spectral Roll-off

Spectral Roll-off je mjera oblika signala. Predstavlja frekvenciju ispod specificiranog postotka ukupne energije spektra.

### Zero Crossing Rate

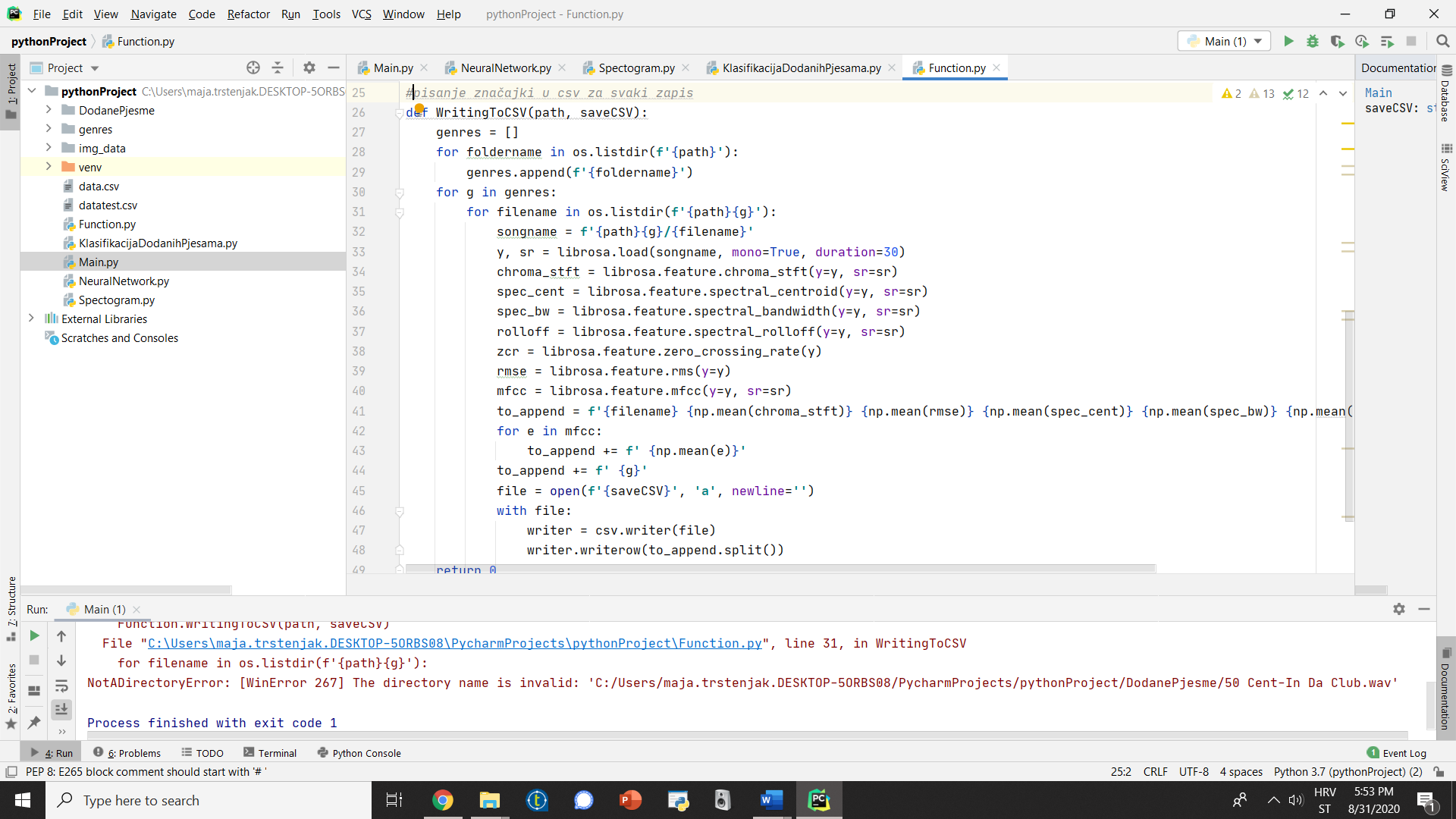
Zero crossing rate ili nulta stopa prijelaza je brzina promjene znaka duž signala, tj. broj gdje se signal promjeni iz negativnog u pozitivan ili obratno. U nastavku je navedene formula dobivanja zcr-a.

,

gdje je s signal od dužine T i 1R<0 je indikator funkcije. Obično ima veće vrijednosti za vrlo udarne zvukove poput onih u metalu i rocku.

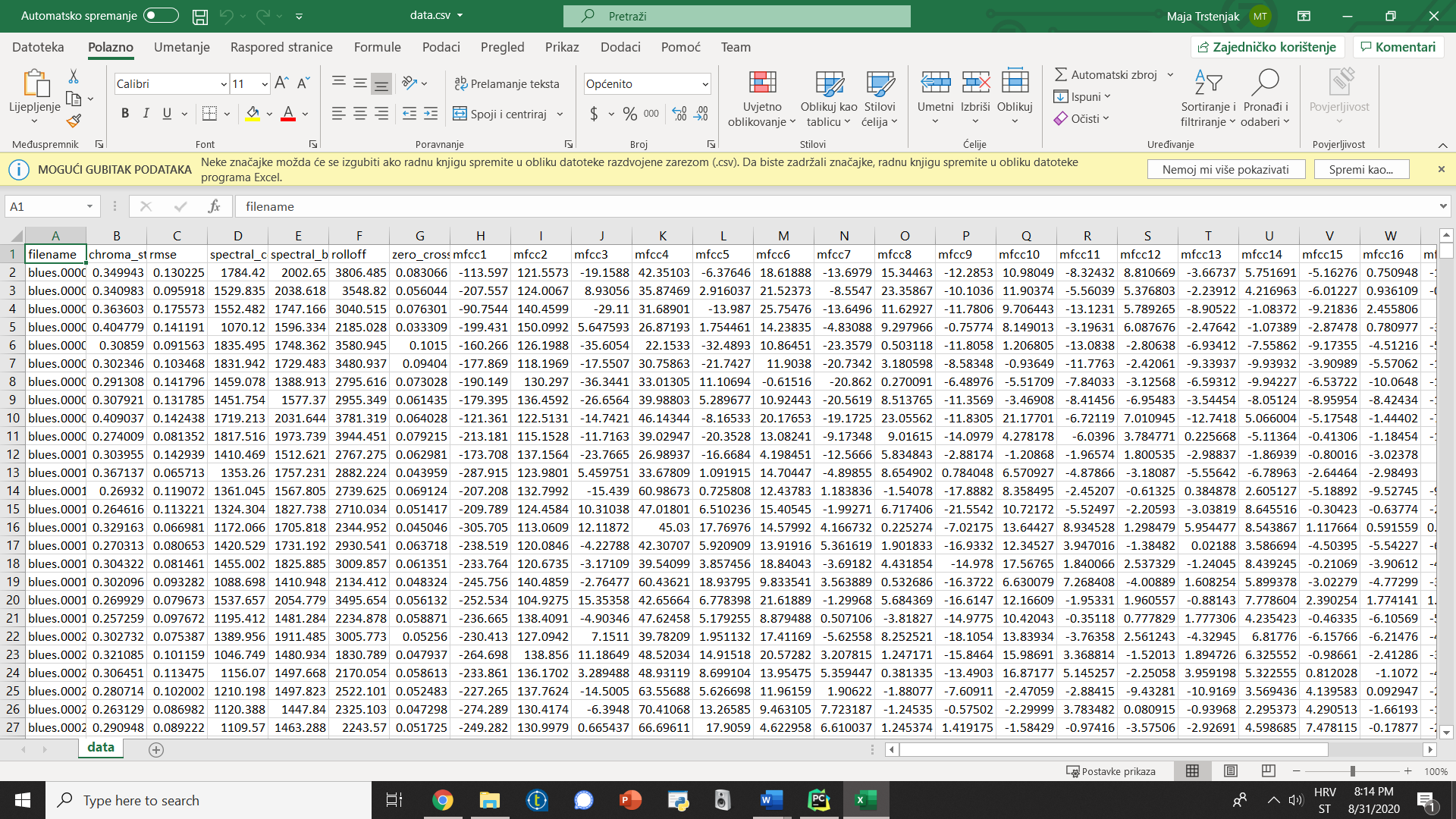
## Zapis značajki u csv datoteku

Značajke svakog audio zapisa pohranjene su u csv datoteku WritingToCSV funkcijom.



*Slika 8. - Funkcija WritingToCSV*

Slika u nastavku prikazuje datoteku sa parametrima spremljenim za pojedine audio zapise.



*Slika 9. - Dobivena csv datoteka*

## Obrada podataka

Nakon pripreme podataka, korištenjem *pandas* biblioteke učitava se *data.csv* datoteka, te se odstranjuje prvi stupac koji sadržava nazive datoteka. Nadalje se encodiraju oznake, te se skaliraju elementi u varijancu standardiziranjem značajki.

Standardna ocjena uzorka x izračunava se formulom: *Z = (x-u)/ s, g*dje je srednja vrijednost uzorka za treniranje nula ako je *s(mean)* False, u suprotnom je *s* standardna devijacija uzorka za treniranje. Centriranje i skaliranje odvijaju se neovisno o svakoj značajki izračunavanjem relevantnih statistika o uzorcima. Srednja i standardna devijacija se pohranjuju kako bi se kasnije koristile u metodi za transformiranje.

Metoda *scaler.fit\_transform* skalira i prevodi svaku značajku pojedinačno tako da je u danom rasponu (između 0 i 1 u ovom primjeru) na skupu za treniranje. Transformacije daju:

*X\_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))*

*X\_scaled = X\_std \* (max - min) + min,*

*g*dje su min i max minimalni i maksimalni raspon značajki.

Zatim se transformacija računa formulama:

*X\_scaled = scale \* X + min - X.min(axis=0) \* scale*

*where scale = (max - min) / (X.max(axis=0) - X.min(axis=0))*

Idući korak je podjela podataka na skup za treniranje i skup za testiranje.



*Slika 10. - Analiza podataka i podjela podataka u skup za treniranje i skup za testiranje*

# NEURONSKA MREŽA

Pomoću Keras biblioteke gradimo neuronsku mrežu. Navedena biblioteka je napisana u Pythonu, otvorenog je koda, a dizajnirana je kako bi omogućila brzo eksperimentiranje s dubokim neuronskim mrežama. Kroz BuildingNetwork funkciju formirani su slojevi mreže i setovi podataka.

Prije učenja mreže potrebno je definirati model, podatke treba podijeliti u dva poduzorka (za treniranje i za testiranje). Ukupan uzorak podijeliti na tri poduzorka (za treniranje, testiranje i konačnu validaciju) ukoliko se za vrijeme učenja planiraju koristiti optimizacijske tehnike za optimiranje duljine učenja i strukture mreže. Pravila za podjelu nema, ali se preporuča najveći dio podataka ostaviti za treniranje mreže, a manji dio podataka za testiranje i validaciju.

Mreža se gradi od 4 dubinska sloja, počinje se sa 256 neurona, a završava sa 10 neurona na izlazu. Model se trenira sa 30 epoha. Jedno predstavljanje cjelokupnog skupa za učenje je jedna epoha.

Korišten je sekvencijalni model, koji je prikladan za običan niz slojeva gdje svaki sloj ima točno jedan ulazni tenzor i jedan izlazni tenzor. *Layers.****dense*** implementira operaciju:

output = activation(dot(input, kernel) + bias)

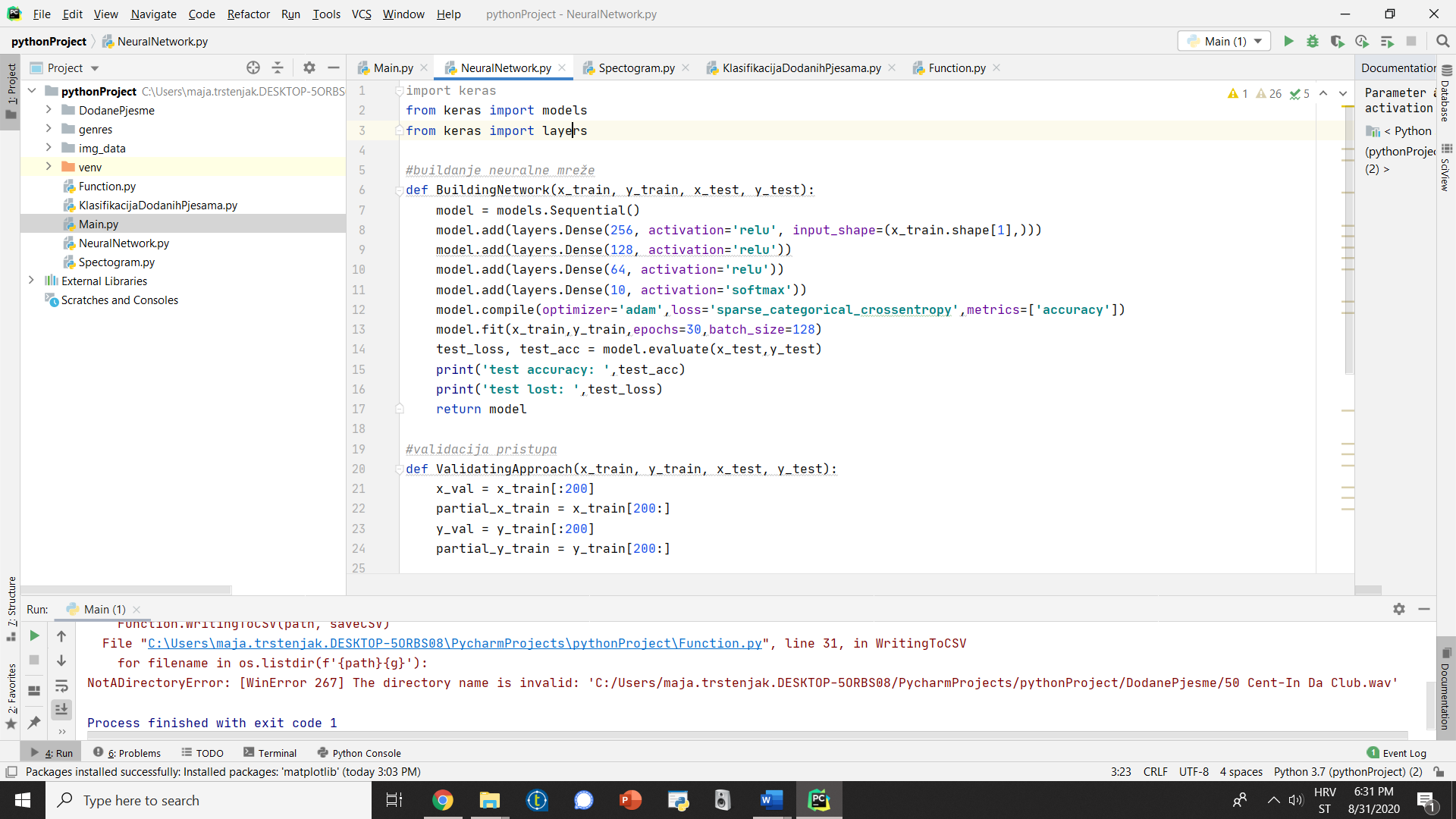
gdje je aktivacija elementna funkcija aktiviranja koja se prenosi kao aktivacijski argument, kernel je matrica pondera koju je stvorio sloj, a pristranost je vektor pristranosti stvoren slojem.

U umjetnim neuronskim mrežama *funkcija aktiviranja* čvora definira izlaz tog čvora s obzirom na ulaz ili skup ulaza. Standardni integrirani krug može se promatrati kao digitalna mreža funkcija aktiviranja koja može biti "ON" ili "OFF", ovisno o ulazu. To je slično ponašanju linearnog perceptrona u neuronskim mrežama.

*ReLU* je jedinica koja koristi ispravljač, a naziva se i ispravljena linearna jedinica, te ima izlaz 0 ako je ulaz manji od 0, u suprotnom ima sirovi izlaz. Odnosno, ako je ulaz veći od 0, izlaz je jednak ulazu. Djelovanje ReLU-a blisko je načinu rada naših bioloških neurona.

*Softmax* je aktivacijska funkcija koja preslikava izlaz u raspon [0,1] i mapira svaki izlaz na način da je ukupan zbroj 1. Izlaz Softmaxa je raspodjela vjerojatnosti, te se softmax funkcija često koristi u završnom sloju klasifikatora utemeljenog na neuronskoj mreži.

*Adam* je adaptivni algoritam za optimizaciju stope učenja koji je dizajniran posebno za trening dubokih neuronskih mreža. Algoritam koristi snagu metoda prilagodljivih stopa učenja za pronalaženje individualnih stopa učenja za svaki parametar.

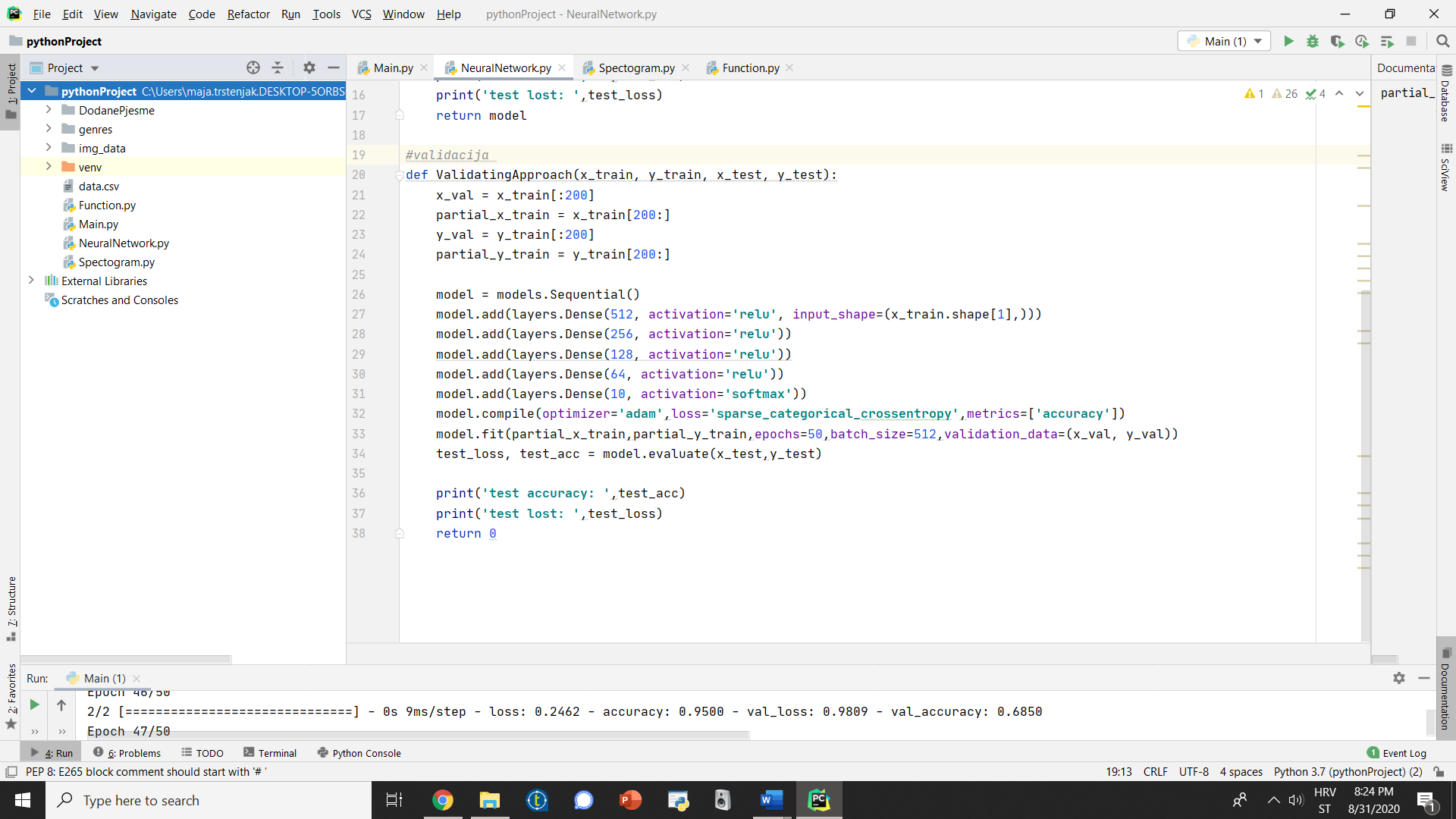


*Slika 11. - BuildingNetwork funkcija*

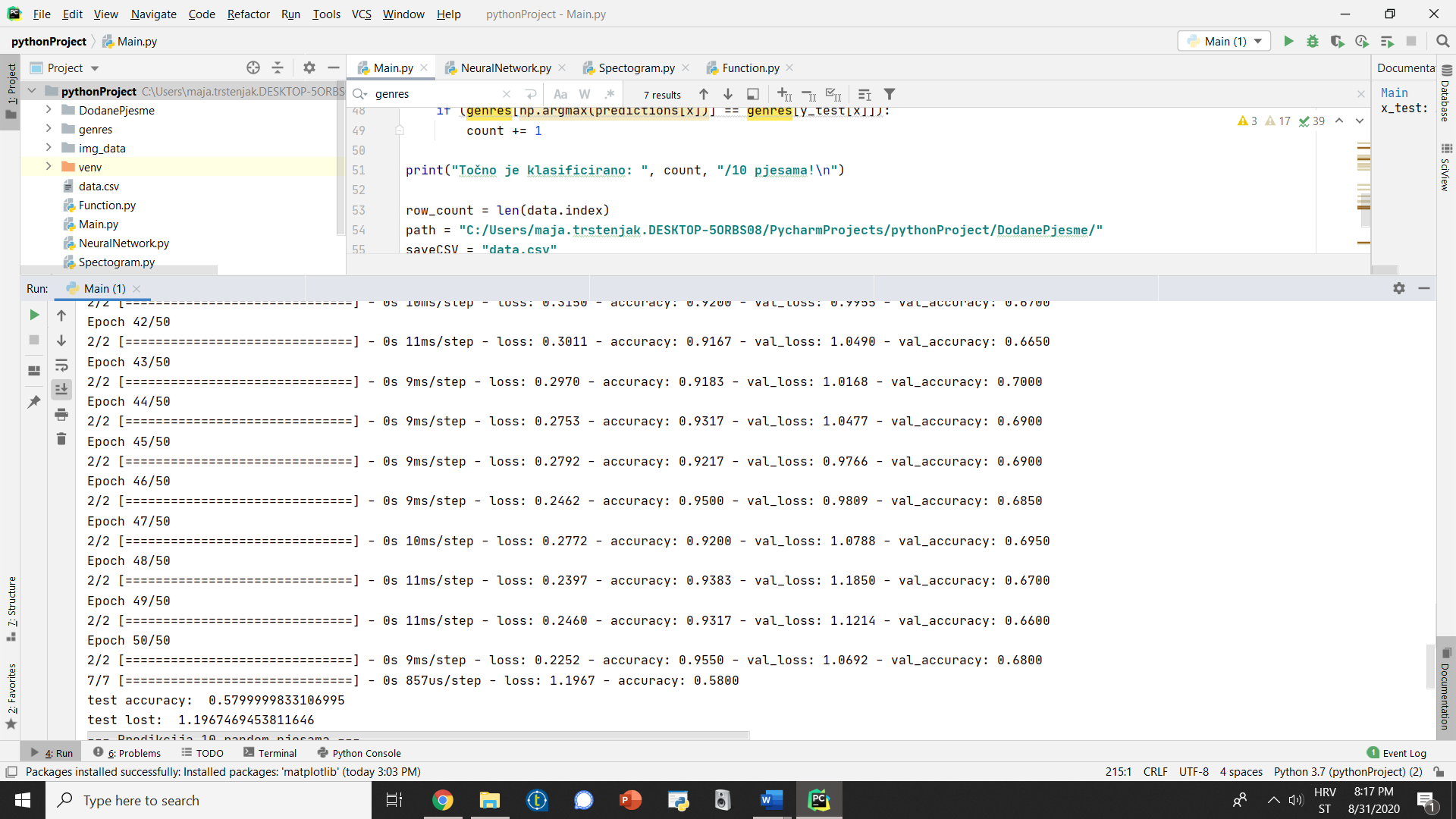
Testiranje mreže je druga faza rada neuronske mreže, te je ona odlučujuća za ocjenjivanje mreže. Razlika između faze treniranja i faze testiranja je u tome što u ovoj fazi mreža više ne uči, dakle težine su fiksne na vrijednostima koje su dobivene kao rezultat prethodne faze treniranja.

## Validacija

Nakon treniranja modela na 600 zapisa, model je validiran na uzorku od 200 zapisa funkcijom *model.evaluate()*.



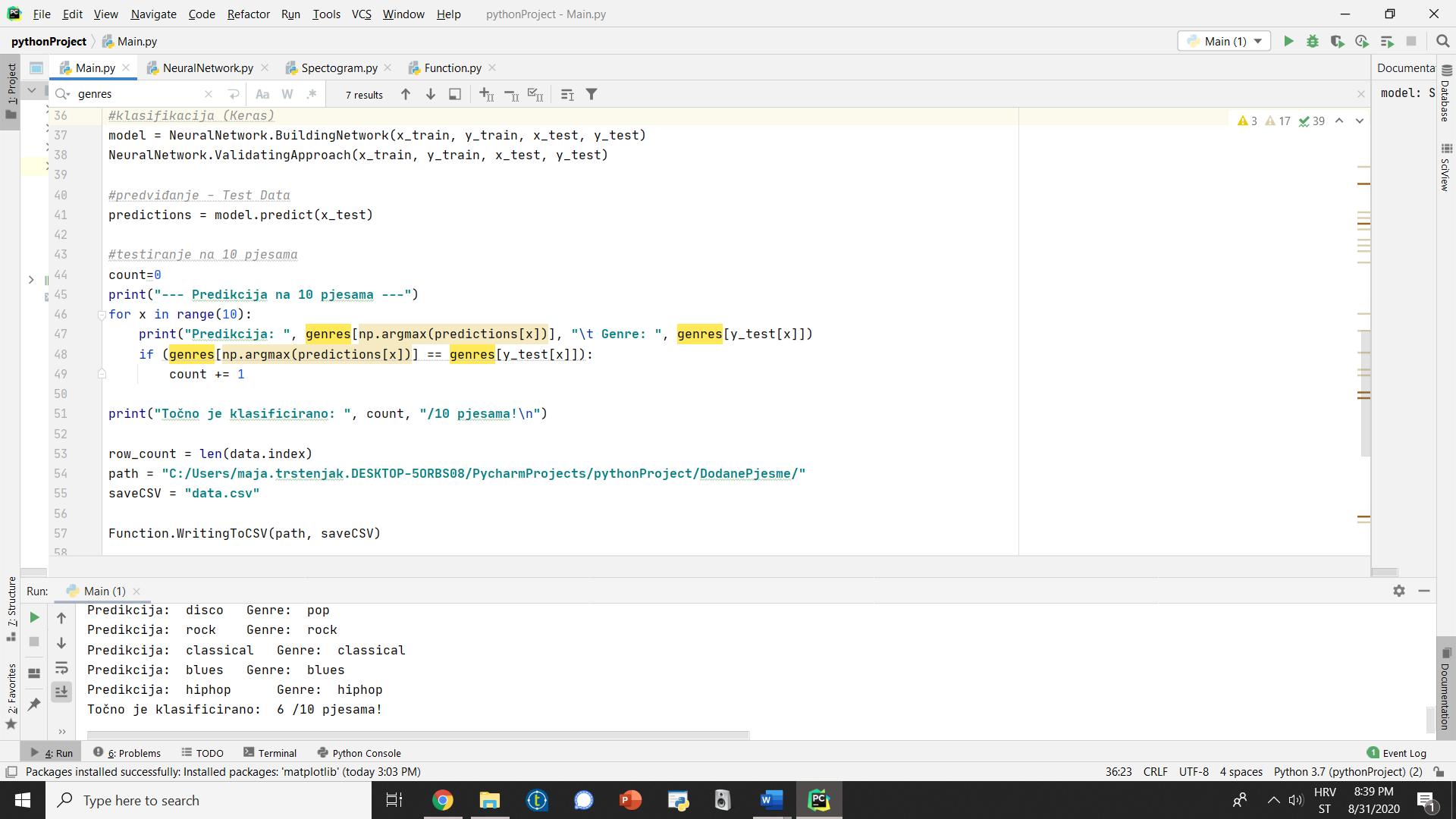
*Slika 12. - ValidatingApproach funkcija*



*Slika 13. - Preciznost i gubitak u modelu*

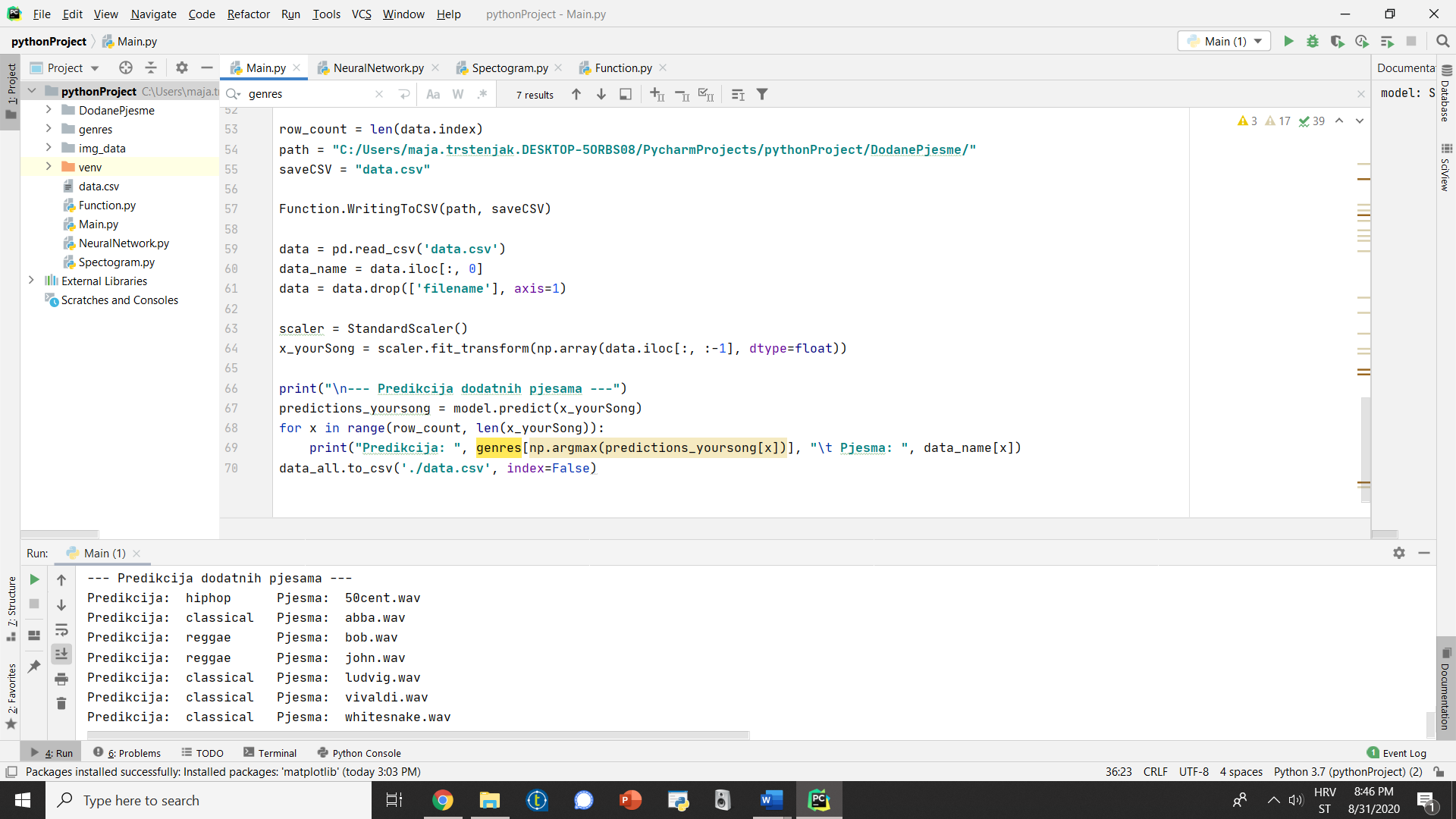
## Predviđanje

Funkcijom *model.predict()* postiže se predviđanje na zadanom skupu, što je prikazano na 10 pjesama na slici u nastavku.



*Slika 14. - Prikaz jednog predviđanja i točnosti*

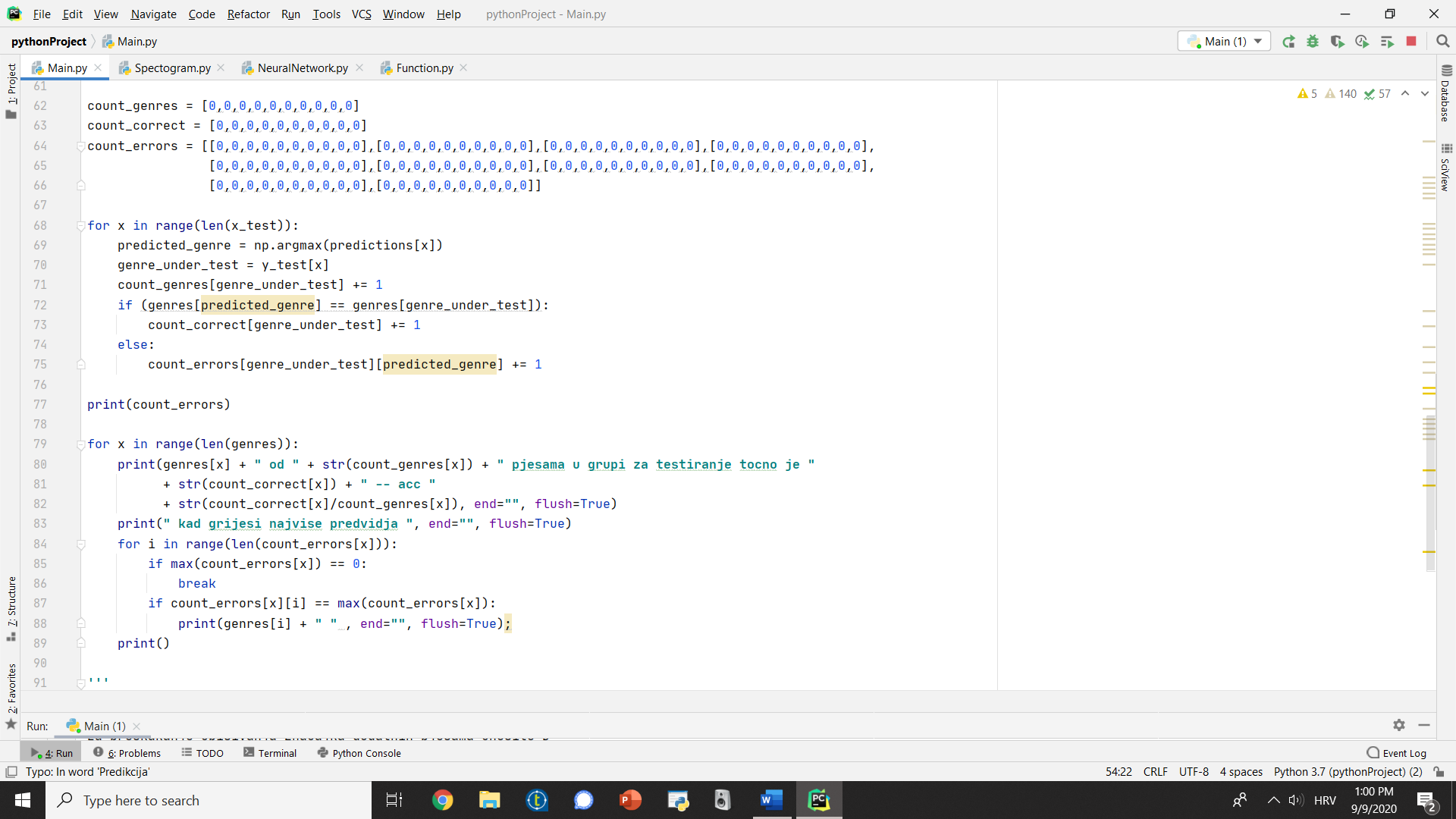
U nastavku rada omogućeno je klasificiranje dodatnih pjesama pohranjenih u zasebnu mapu. Pomoću *librosa* biblioteke čitaju se značajke pjesama, koje se pohranjuju u csv file, nadalje se *scaler* funkcijom skaliraju parametri te se može predvidjeti žanr određenog audio zapisa funkcijom *predict*. Točnost prikazana na slici ispod je 4/7, što je zadovoljavajuće obzirom na kompleksnost pjesama, te možebitno pripadanje jedne pjesme više žanrova.



*Slika 15. - Predikcija žanrova za 7 naknadno dodanih pjesama*

# REZULTATI

Za analizu rezultata klasifikacije po žanrovima korišteno je 200 audio zapisa iz cijelog skupa za testiranje, te su proučavani izlazi za nekoliko pokretanja programa.

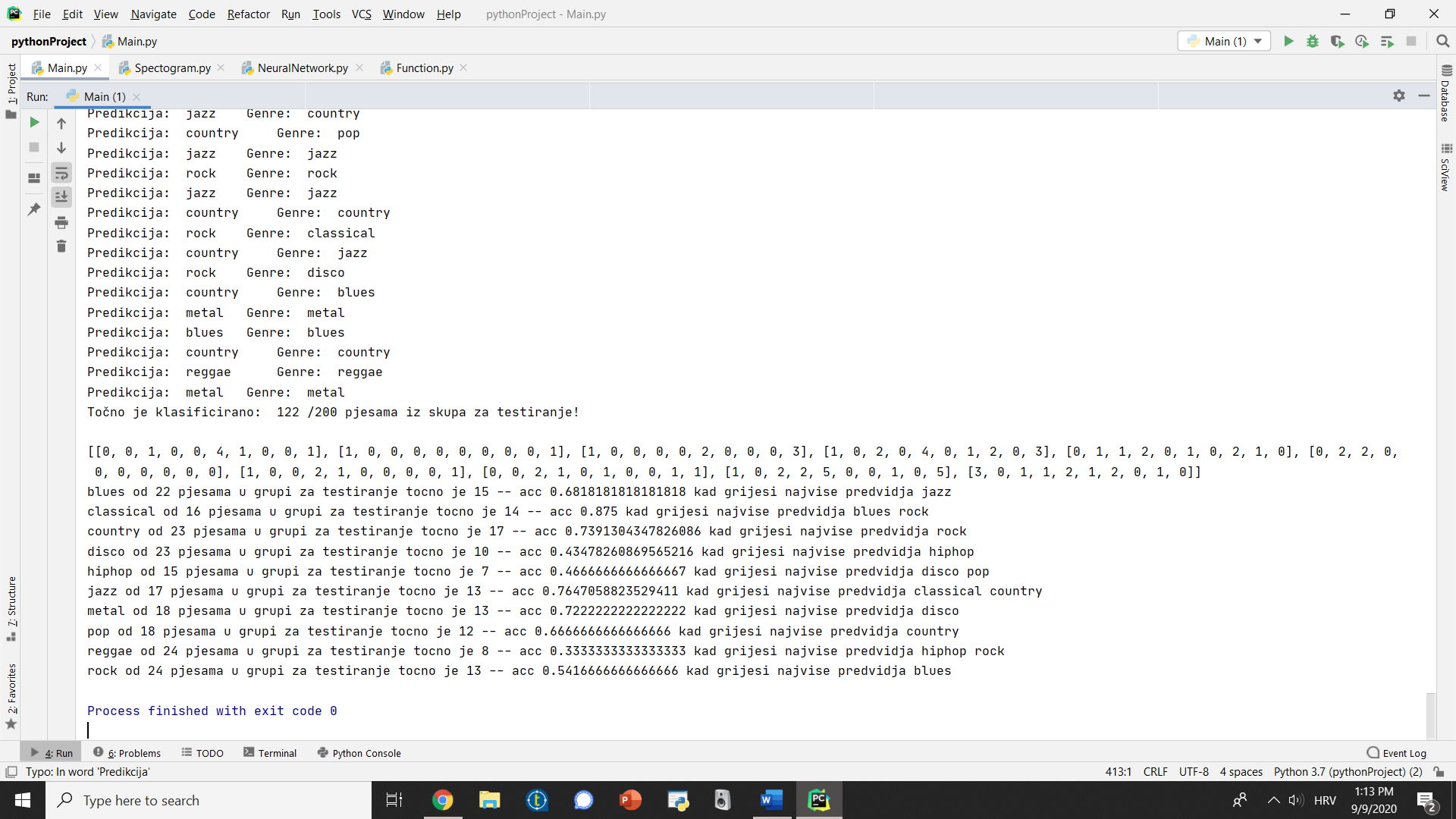


*Slika 16. Kod za usporedbu krivih klasifikacija*

Na prethodnoj slici prikazan je kod kojim je dobivena točnost klasifikacije po pojedinom žanru, te ispis žanrova kojima je pojedini žanr zamijenjen u jednom pokretanju programa.

**Prva izvedba programa**

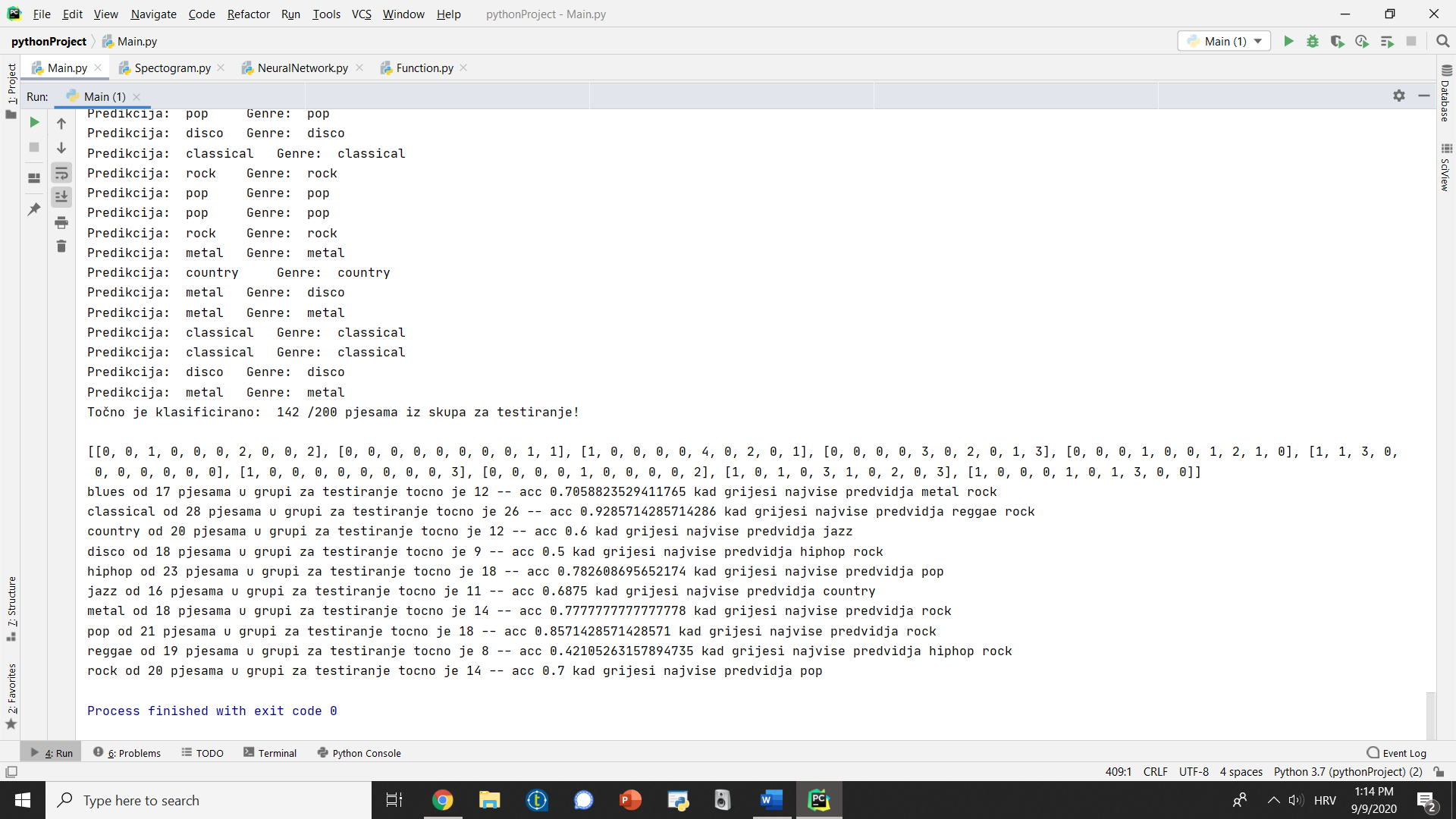
U prvoj izvedbi programa ukupna točnost klasifikacije iznosila je 65% (test accuracy: 0.6499999761581421), najtočnije je klasificiran žanr klasične glazbe sa 88% točnosti gdje je od 16 pjesama, njih 14 ispravno prepoznato, a 2 su svrstane u žanrove blues i rock. Najlošiju točnost imao je žanr raggae sa 33% točnosti, gdje je od 24 promatrane pjesme njih 8 točno klasificirano, a ostale su najčešće svrstane u žanrove hiphop i rock.



*Slika 17. Prva izvedba programa*

**Druga izvedba programa**

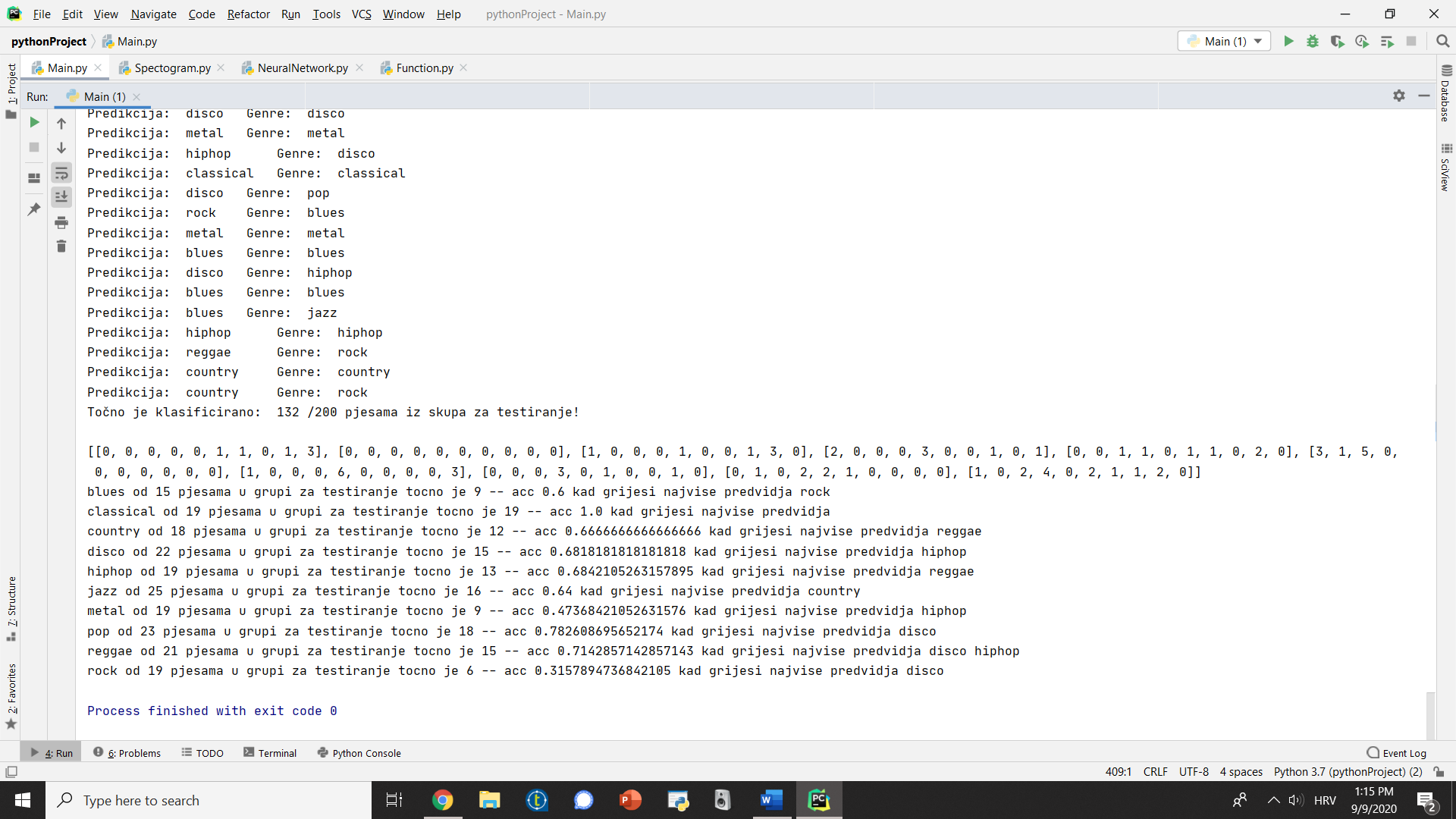
U drugoj izvedbi programa ukupna točnost klasifikacije iznosila je 71% (test accuracy: 0.7149999737739563), najtočnije je klasificiran žanr klasične glazbe sa 93% točnosti gdje je od 28 pjesama, njih 26 ispravno prepoznato, a 2 su svrstane u žanrove raggae i rock. Najlošiju točnost imao je žanr raggae sa 42% točnosti, gdje je od 19 promatranih pjesama njih 8 točno klasificirano, a ostale su najčešće svrstane u žanrove hiphop i rock.



*Slika 18. Druga izvedba programa*

**Treća izvedba programa**

U trećoj izvedbi programa ukupna točnost klasifikacije iznosila je 62% (test accuracy: 0.6200000047683716), najtočnije je klasificiran žanr klasične glazbe sa 100% točnosti gdje je od 19 pjesama svih 19 ispravno prepoznato. Najlošiju točnost imao je žanr rock sa 32% točnosti, gdje je od 19 promatranih pjesama njih 6 točno klasificirano, a ostale su najčešće svrstane u žanr disco.



*Slika 19. Treća izvedba programa*

U nastavku slijedi prikaz točnosti klasifikacije za svih 10 žanrova u 3 izvedbe programa.

|  |  |  |  |
| --- | --- | --- | --- |
| **Žanr** | **Točnost klasifikacije** | | |
| 1. *Pokretanje* | 1. *Pokretanje* | 1. *pokretanje* |
| Blues | 68% | 71% | 60% |
| Classical | **88%** | **93%** | **100%** |
| Country | 74% | 60% | 67% |
| Disco | 43% | 50% | 68% |
| Hiphop | 46% | 78% | 68% |
| Jazz | 76% | 69% | 64% |
| Metal | 72% | 78% | 47% |
| Pop | 66% | 86% | 78% |
| Raggae | **33%** | **42%** | 71% |
| Rock | 54% | 70% | **32%** |

*Tablica 1. Prikaz točnosti klasifikacije u tri izvedbe programa sa istaknutim najboljim i najgorim rezultatom*

Dobiveni rezultati nisu uvijek dosljedni. U nekim pokretanjima pogreška klasifikacije je u npr. žanrovima klasične i rock glazbe, no ponekad algoritam zamjenjuje rock sa glazbom pop žanra.

Blues glazba najčešće je kod pogreške klasificirana kao rock ili jazz, dok je rock kod pogrešne klasifikacije uz blues klasificiran i kao disco i pop. Jazz je kod pogrešne klasifikacije najčešće svrstan u country žanr, disco u hiphop, a hiphop u pop žanr glazbe.

Točnost algoritma bila bi veća na većem skupu podataka, te bi i osvrt mogao biti precizniji.

# ZAKLJUČAK

Rad sa zvučnim zapisima je zanimljiv, te omogućava kreativnost prilikom odlučivanja pristupa obrade podataka.

Cilj ovoj projekta je bio uspješno pohraniti te obraditi podatke postojećim metodama, te izgraditi neuronsku mrežu koja bi sa dobro obrađenim podacima i raznovrsnim značajkama uspješno naučila razlikovati bazu audio zapisa.

Baza za učenje mreže sadržavala je jasne i jednostavne zapise, pa je točnost kod klasifikacije kompleksnih pjesama bila očekivano slabija.

# POPIS SLIKA

Slika 1. - Python file Spectrogram

Slika 2. – Prikaz nekoliko generiranih spektrograma za žanr jazz

Slika 3. – Logo Librosa

Slika 4. – Vrijednosti 20 MFCC koeficijenata za jedan od zapisa

Slika 5. – MFCC

Slika 6. - Primjer za prikaz porasta spektralnog centroida prema kraju

Slika 7. – Spektralna širina pojasa

Slika 8. - Funkcija WritingToCSV

Slika 9. - Dobivena csv datoteka

Slika 10. - Analiza podataka i podjela podataka u skup za treniranje i skup za testiranje

Slika 11. - BuildingNetwork funkcija

Slika 12. - ValidatingApproach funkcija

Slika 13. - Preciznost i gubitak u modelu

Slika 14. - Prikaz jednog predviđanja i točnosti

Slika 15. - Predikcija žanrova za 7 naknadno dodanih pjesama

Slika 16. Kod za usporedbu krivih klasifikacija

Slika 17. Prva izvedba programa

Slika 18. Druga izvedba programa

Slika 19. Treća izvedba programa

# POPIS TABLICA

Tablica 1. Prikaz točnosti klasifikacije u tri izvedbe programa sa istaknutim najboljim i najgorim rezultatom

# LITERATURA

* <https://www.kaggle.com/carlthome/gtzan-genre-collection>
* <https://www.sciencedirect.com/science/article/pii/S1110866512000151>
* <https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8>
* <https://medium.com/@matanlachmish/music-genre-classification-470aaac9833d>
* <http://www.cs.cmu.edu/~yh/files/GCfA.pdf>
* <https://keras.io/>
* <https://librosa.org/doc/latest/index.html>

**LINK NA RAD:** https://easyupload.io/j864u8