Klasifikacija glazbenih djela po žanrovima

Bošnjak, Dominik

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstvo

KLASIFIKACIJA GLAZBENIH DJELA PO ŽANROVIMA

Diplomski rad

Dominik Bošnjak

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 14.09.2021.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Dominik Bošnjak
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1039R, 06.10.2019.
OIB studenta:	82614748782
Mentor:	Izv. prof. dr. sc. Emmanuel Karlo Nyarko
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv.prof.dr.sc. Ratko Grbić
Član Povjerenstva 1:	Izv. prof. dr. sc. Emmanuel-Karlo Nyarko
Član Povjerenstva 2:	Dr. sc. Petra Pejić
Naslov diplomskog rada:	Klasifikacija glazbenih djela po žanrovima
Znanstvena grana rada:	**Umjetna inteligencija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Treba istražiti različite algoritme prikladne za klasifikaciju glazbenih žanrova. Implementirati i usporediti nekoliko takvih algoritama.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.09.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 30.09.2021.

Ime i prezime studenta:	Dominik Bošnjak
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1039R, 06.10.2019.
Turnitin podudaranje [%]:	2

Ovom izjavom izjavljujem da je rad pod nazivom**: Klasifikacija glazbenih djela po žanrovima**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Emmanuel Karlo Nyarko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD

Glazba jest umjetnost zvuka, izražava se pjevanim ili sviranim tonovima, aranžiranim u vremenu i prostoru. Sveprisutna je u ljudskom životu te je neraskidivi dio ljudskog okruženja. Postoji otkada i čovjek te je usko vezana uz razvoj čovjeka, što dokazuju prapovijesni ostatci raznih instrumenata. Razvojem civilizacije razvija se i glazba, počinje se upotrebljavati u religijske svrhe i obrede (dozivanje kiše, uspješnost u lovu…), pri svakodnevnom radu kako bi olakšala posao, pri podizanju borbenog duha i pripremanja ljudi za bitku. Glazba nas prati u svim životnim okolnostima, od vožnje liftom do svečanosti i slavlja koja obilježavamo. Čini nas sretnima, pomaže nam u teškim trenutcima, ima brojne svrhe i brojne oblike te ju je teško podijeliti i kategorizirati. Jedan od načina kategorizacije glazbe su žanrovi. Žanr je oznaka stvorena i dogovorena među ljudima koja označava set karakteristika koje opisuju specifični ugođaj glazbe. Međutim, žanrovi nemaju strogo definirane granice, a glazbena djela zbog svoje kompleksnosti često kombiniraju karakteristike različitih žanrova. S druge strane, klasifikacija glazbenih djela po žanrovima sveprisutna je u glazbenoj industriji. Od raznih *streaming* platformi kao što su *YouTube, Spotify, Tidal,* koji ju primjenjuju pri kreiranju personaliziranih popisa reprodukcije (engl. *playlist*), predlaganjem djela koja bi se mogla svidjeti slušatelju na osnovi dosad preslušanih. Nadalje, pomaže pronaći popularne žanrove i trendove, što pridonosi stvaranju novih glazbenih djela. Također, tehnike koje se upotrebljavaju za postizanje klasifikacije prisutne su i u drugim područjima glazbenog pretraživanja (engl. *Music Information Retrieval - MIR*) koje je relativno nova znanstvena grana, a one su: prepoznavanje instrumenata, automatsko generiranje glazbenih transkripcija pa čak i automatsko stvaranje glazbenih djela. U ovome radu pričat će se isključivo o klasifikaciji glazbenih djela po žanrovima na sljedeći način. Pregled postignuća i problema unutar MIR-a i nekoliko algoritama strojnog učenja koji se primjenjuju u svrhu klasifikacije bit će dan u Poglavlju 2. Opis odabranog skupa podataka (engl. *Dataset*) i predobrada (engl. *preprocessing*) istih objašnjena je u Poglavlju 3. Poglavlje 4 opisuje postupak treniranja nekoliko različitih algoritama klasifikacije na predobrađenom podatkovnom skupu, dok Poglavlje 5 sumira i uspoređuje rezultate primijenjenih algoritama.

2. PREGLED PODRUČJA TEME

Klasifikacija glazbenih djela po žanrovima jedan je od osnovnih problema glazbenog pretraživanja (engl. *Music Information Retrieval*) dalje u tekstu MIR, koje je pak interdisciplinarna znanost koja se bavi procesiranjem, pretraživanjem i organizacijom informacija vezanih za glazbu [1]. Ono kombinira tehnike obrade signala i strojnog učenja kako bi se izvukle osnovne značajke glazbe (engl. *audio features*). G. Tzanetakis i P. Cook definiraju tri glavna oblika glazbenih značajki [2]:

1. tekstura boje tona (engl. *timbral texture*) – opisuje razliku između istog tona proizvedenog na različitim instrumentima. Isto tako različiti ukrasi glazbenika i način na koji izvodi pjesmu također utječu na boju tona.
2. sadržaj visine tona (engl. *pitch conten*) - opisuje razlike između tonova, odnosno što razlike ton A od tona C#
3. ritmičke (engl. *rhytmic*) – uključuje tempo, mjere, duljinu tona i naglaske.

Nakon njih J. S. Downie navodi i druge moguće aspekte glazbe [1] kao što su:

4. Tekstualni (engl. *textual*) – odnosni se na lirički sadržaj. Problem ovog aspekta jest neovisnost teksta o melodiji. Postoje različite pjesme koje imaju istu melodiju no drugačiji tekst. Isto tako, postoji mnoštvo glazbe koja uopće ne sadrži tekst.
5. Bibliografski (engl. *bibliographic*) – odnosni se na nazive, skladatelje, aranžere, izvođače, pisce, odnosno sve meta-podatke o stvorenom djelu.

Različite su metode primijenjene pri svladavanju navedenog problema, no uglavnom se svode na procesiranje podatkovnog skupa audio zapisa, izvlačenje značajki te njihova upotreba u treniranju klasifikatora strojnog učenja. Pa tako, prema [3] primijenjen je hibridni model između specifične neuronske mreže (engl. *neural network*) i stroj s potpornim vektorima (engl. *support vector machines – SVM*). Nadalje, H. Bahuleyan unutar svoga rada [4], upotrebljava logističku regresiju (engl. *logistic regression*) koja je obično binarni klasifikator, pri čemu je prilagođava problemu više-klasne klasifikacije, uspoređujući s ranije navedenim metodama te s metodom slučajnih šuma (engl. *random forest*) koja stvara nekoliko stabla odlučivanja koja većinskim glasanjem određuju konačni žanr glazbenog djela. Potaknuti radom J. S. Downiea [1], u određeni radovima izabiru se značajke tekstova pjesama, recenzija istih te čak i omote albuma [5, 6]. Opravdanje ovakvog pristupa je to što stvaratelji svakog glazbenog djela nekog žanra imaju različiti pristup svim aspektima gotovog proizvoda te su se

ovakve klasifikacije također pokazale uspješnima. Na primjer, omot albuma jednog žanra često ima logo benda uz sliku njihove maskote koja je obično čudovišnog oblika, dok omot albuma drugog žanra se obično sastoji od izvođača u šeširu sa svojim instrumentom, te je očito o kojemu se žanru radi i bez opisa (Slike 1.1. – 1.2.). Nadalje, postoje metode koje su inspiraciju preuzele iz digitalne obrade slike: ideja je pretvoriti audio zapis u vizualnu reprezentaciju, točnije spektogram te primijeniti konvolucijske mreže koje su se pokazale uspješnima pri klasifikaciji slika i tako audio klasifikaciju prebacuju u područje klasifikacije slika [7].

Slika 1.1. Omot *metal* albuma

Slika 1.2. Omot *country* albuma

Problemi na koje se pak nailazi pri razvoju MIR-a uključuju ranije naveden problem više-aspektne prirode glazbe, no on nije jedini [1]. Znanstvenici unutar ovog područja moraju uzeti u obzir sve različite načine na koje glazba može biti prezentirana, različitim simbolima (note, tekst, tablature) pa onda različitim audio formatima bilo digitalnim ili analognim (*.mp3, .wav,* ploča, CD, kazeta). Također, kako je navedeno u uvodu, glazba nadilazi vrijeme i kulturološke granice, no svako povijesno razdoblje i kultura stvorila je vlastiti način izražavanja kroz glazbu, stoga je teško definirati stroge granice između žanrova. S obzirom na to, umjesto pridruživanja jednog žanra, postoje metode koje pridružuju nekoliko žanrova određenoj pjesmi (engl. *multi-label genre classification*) [8], implementirajući različite ansamble tehnike kako bi prilagodili algoritam klasifikacije problemu gdje svaki uzorak može pripadati nekoliko klasa odjednom. U sklopu ovoga rada, odabrane su značajke teksture boje tona, te su uspoređena tri klasifikatora:

1. K najbližih susjeda – iz razloga što se smatra najjednostavnijim algoritmom, želja je ispitati koliko je primjenjiv za dani problem.
2. Stroj s potpornim vektorima – zapravo pripada skupini binarnih klasifikatora, no to ne znači da ga nije moguće prilagoditi više-klasnom problemu.
3. Neuronske mreže – sveprisutna, prilagodljiva metoda, koja se pokazala uspješnom u raznim područjima i problemima.

3. PODATKOVNI SKUP I PREDOBRADA PODATAKA

Strojno učenje uglavnom zahtjeva podatke numeričkog tipa, često u obliku vektora značajki (engl. *feature vector*). Unutar poglavlja naveden je opis odabranog podatkovnog skupa te alternativa koje postoje, uz objašnjenje izvučenih glazbenih značajki koje pripadaju skupinama navedenim u prošlom poglavlju.

2.1. Podatkovni skupovi

2.1.1. **The Million Song Dataset (MSD)**

Kao što samo ime predlaže, ovaj podatkovni skup sastoji se od milijun pjesama, koje su klasificirane različitim žanrovima i pod-žanrovima (engl. *subgenres*) [9]. MSD se razlikuje od drugih glazbenih podatkovnih skupova značajno, s obzirom da ne daje direktan pristup audio zapisima jer sadrži trenutno popularne pjesme na koja nemaju prava s gledišta *copyright-a*. Nadalje, podatkovni skup koji bi sadržavao milijun pjesama zauzimao bi puno memorije što otežava njegovu upotrebu. Iz navedenih razloga, MSD se sastoji od već izvučenih značajki i metapodataka. Iz razloga što je direktan pristup audio zapisima potreban kako bi se izvukle specifične značajke koje nisu sadržane u podatkovnom skupu, ovaj skup se neće koristiti.

2.1.2. **The Free Music Archive (FMA)**

FMA sastoji se od 100,000 klasificiranih pjesama te također postoji nekoliko verzija podatkovnog skupa od: 'male' verzije (8,000 uzoraka trajanja 30 sekundi), 'velike' (106,574 uzorka duljine 30 sekundi) do 'cijele' verzije (106,574 pjesama u svojoj cijelosti) [10]. Problem ovog podatkovnog skupa jest što nije balansiran, odnosno svaki žanr nije jednako zastupljen.

2.1.3. **GTZAN**

GTZAN podatkovni skup sastoji se od 1000 audio zapisa, svaki duljine 30 sekunda [11]. Sadrži 10 žanrova te svaki čini 10 % ukupnog podatkovnog skupa, odnosno svaki žanr ima 100 primjera. Audio zapisi su uzorkovani sa 22050 Hz te su zapisani u *.wav* formatu. Podržani žanrovi su:

- *blues*
- *classical*
- *country*
- *disco*

- *hiphop*
- *jazz*
- *metal*
- *pop*
- *reggae*
- *rock*

Osim audio zapisa, podatkovni skup sadrži i vizualnu reprezentaciju svakog podatkovnog primjera u obliku Mel Spektograma (engl. *Mel Spectogram*). Također sadrži i dvije *.csv* datoteke, jednu s značajkama izvučenim iz svakog podatkovnog primjera duljine 30 sekundi, dok druga ima istu strukturu no prije izvlačenja podatkovni primjeri su podijeljeni u audio zapise duljine 3 sekunde. Ove datoteke nisu upotrijebljene pri treniranju nego pri provjeri sa samostalno izvučenim značajkama dobivenim na način opisan u nastavku.

2.2. Izvlačenje značajki

Izvlačenje značajki proces je izračunavanja numeričke reprezentacije podatkovnih uzoraka, najčešće u obliku vektora. Za potrebe klasifikacije glazbenih djela po žanrovima odabrane su sljedeće značajke [2]:

1. Spektralni centroid (engl .*spectral centroid*)
2. Frekvencija većinske spektralne snage (engl. *spectral rolloff*)
3. Spektralni tok (engl. *spectral flux*)
4. Broj prolazaka kroz ništicu (engl. *zero crossing rate*)
5. Mel-frekvencijski kepstralni koeficjenti (engl. *mel-frequency cepstral coefficients-MFCC*)
6. Postotak segmenata niske energije (engl. *low-energy)*

Navedene značajke predstavljaju skupinu značajki nazvanu tekstura boje tona (engl. *timbral texture*) [2] te pripadaju uobičajenim značajkama koje se također primjenjuju i za prepoznavanje govora. Izračun značajki baziran je na kratko-vremenskoj Fourierovoj transformaciji (engl. *short time Fourier transform*) ili skraćeno *STFT*.

2.2.1. **STFT**

Koncept *STFT-a* jest opisati frekvencijske karakteristike signala koje se mijenjaju tijekom vremena. S obzirom da se primjenom diskretne Fourierove transformacije (DFT) dobiva znanje o prisutnim frekvencijama u signalu, nastaje problem jer se zvučni signal mijenja

tijekom vremena, mijenjaju se i frekvencije pa je potrebno znati kada se promjene frekvencije pojavljuju u signalu. Kao rješenje, ideja je cijeli signal podijeliti na manje dijelove ili segmente, odnosno vremenske okvire te na njima raditi DTF [12], formula (3.1). Dvije su različite, no ekvivalentne interpretacije STFT-a. Prva fiksira signal, te na njega primjenjuje nisko propusni filter, odnosno prozorsku (engl. *window*) funkciju, te izračunava DFT. Druga, pak fiksira prozor te pomjera signal, također uzimajući DFT za svaki segment. Rezultat ovoga pripada vremensko-frekvencijskoj domeni, koja lijepo opisuje audio signal, jer prikazuje frekvencijske informacije te njihovu promjenu kroz vrijeme:

$$X_n\left(e^{j\omega_k}\right) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega_k m} \tag{3.1.}$$

pri čemu je: x(n) signal definiran za svaki n, $X_n\left(e^{j\omega_k}\right)$ STFT za x(n) u trenutku n i frekvenciji ω_k, w(n) prozorska funkcija.

Ono što je bitno kod implementacije STFT-a jest odrediti duljinu segmenta na kojemu se uzima DTF. Kraći segmenti daju bolju vremensku rezoluciju, odnosno lakše se diskriminiraju impulsi koji između sebe imaju mali vremenski razmak, no pogoršavaju frekvencijsku rezoluciju, odnosno sposobnost razlikovanja čistih tonova koji imaju mali frekvencijski razmak. Dodatno, prilikom segmentiranja signala javlja se problem gdje krajnje točke segmenta postaju diskontinuirane, a time se u frekvencijskoj domeni transformiraju u obliku komponenti visokih frekvencija koje nisu prisutne u originalnom signalu (engl. *Spectral leakage*). Ovo se rješava pomoću preklapanja segmenata (engl. *overlapping*), drugim riječima uvodi se dodatna varijabla koja definira za koliko se pomjera prozor u sljedećem koraku.

Takvom primjenom STFT-a iz izvornog audio zapisa izračunavaju se značajke za svaki segment signala te su konačne značajke koje se upotrebljavaju, srednja vrijednost i varijanca značajki preko svakog segmenta signala, a objašnjene su u nastavku.

2.2.2. **Spektralni centroid**

Spektralni centroid je težinska sredina frekvencija prisutnih u signalu, pri čemu su magnitude težine. Drugim riječima on je „centar mase“ magnituda STFT-a:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{3.2.}$$

gdje $M_t[n]$ predstavlja magnitudu spektra dobivenu Fourierovom transformacijom za vremenski okvir *t* i frekvenciju *n.*

Spektralni centroid opisuje „svjetlinu tona“, to jest, veće frekvencije predstavljaju „svjetliji ton“. Za primjer ako dva instrumenta sviraju isti ton jednakom glasnoćom, trubu će se doživjeti „svjetlije“ nego rog, obou od flaute, čembalo od klavira. Također isto vrijedi i za udaraljke, udarac o triangl zvuči svjetlije nego udarac o drvenu kocku [13].

2.2.3. **Frekvencija većinske spektralne snage**

Frekvencija većinske spektralne snage definirana je kao frekvencija R_t ispod koje se nalazi 85% magnituda STFT-a. Služi za razlikovanje bezvučnog od zvučnog govora i glazbe. Unutar prepoznavanja govora pokazano je kako snaga bezvučnog govora podjednako raspoređena na nižim i višim frekvencijama, dok se većina energije glazbe i zvučnog govora nalazi na niskim frekvencijama [2]:

$$\sum_{n=1}^{R_t} M_t[n] \;=\; 0.85 * \sum_{n=1}^{N} M_t[n] \tag{3.3.}$$

2.2.4. **Spektralni tok**

Spektralni tok je mjera koja pokazuje koliko brzo se mijenja spektar snage. Izračunava se kao suma kvadratnih razlika normaliziranih magnituda susjednih vremenskih segmenata [2]:

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{3.4.}$$

gdje su $N_t[n]$ i $N_{t-1}[n]$ normalizirane magnitude Fourierove transformacije trenutnog vremenskog segmenta t i susjednog vremenskog segmenta t-1

2.2.5. **Broj prolazaka kroz ništicu**

Broj prolazaka kroz ništicu definira stopu kojom signal prelazi iz pozitivne vrijednosti u negativnu ili obrnuto. Drugim riječima prikazuje koliko puta amplituda signala prolazi kroz ništicu u danom vremenskom intervalu [2]:

$$Z_t = \frac{1}{2}\sum_{n=1}^{N} |sgn[x_i(n)] - sng[x_i(n-1)]| \; gdje\ je\ sng[x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0 \\ -1, & x_i(n) < 0 \end{cases} \tag{3.5.}$$

Broj prolazaka kroz ništicu moguće je interpretirati kao mjeru prisutnosti šuma (engl. *noise*) u signalu, jer uobičajeno poprima veće vrijednosti kada signal sadrži mnogo šuma. Također, standardna devijacija ove značajke pokazuje se veća za signale govora u odnosu na glazbene signale.

2.2.6. **Mel-Frekvencijski kepstralni koeficjenti**

MFCC jedan od najčešće primjenjivanih značajki pri klasifikaciji audio zapisa, primijenjena je u problemima prepoznavanja govora, zvuka okoline te klasifikaciji glazbenih djela. Također se temelji na STFT, a izvlači se na sljedeći način [14]:

1. Primijeniti STFT
2. Dobiveni spektar snaga provući kroz niz filtera, *mel filterbank* te zbrojiti dobivene energije za svaki filter:
    - Ljudsko uho puno bolje diskriminira razlike u tonovima pri niskim frekvencijama, primjenjuje se *mel-scale* pri kreiranju filtera, na način da su filteri manje razmaknuti i uži pri niskim frekvencijama.
3. Izračunati logaritam dobivenih filtriranih energija:
    - Također, ljudsko uho ne doživljava glasnoću na linearnoj skali, da bi zvuk bio doživljen dva put glasnijim, potrebno je uložiti 8 puta više energije.
4. Izračunati diskretnu kosinusnu transformaciju (engl. *Discrete Cosine Transformation* - DCT):
    - Zbog preklapanje filtera, dobivene filtrirane energije su korelirane; DCT ih dekorelira
5. Izabrati prvih n koeficijenata koji će se sačuvati:
    - Tipično se uzima prvih 13 koeficijenata pri reprezentaciji govornog zvuka, no za potrebe klasifikacije glazbenih djela prvih 5 se pokazalo dovoljno [2].

MFCC dobro opisuju „boju“ zvuka (engl. *timbre*), odnosno ono što razlikuje dva tona iste glasnoće i visine.

2.2.7. **Postotak segmenata niske energije**

Za razliku od prijašnjih značajki, postotak segmenata niske energije bazira se na duljim segmentima, odnosno definira se kao postotak kraćih segmenata koji imaju manji korijen srednje kvadratne energije (engl. *root mean square (RMS) energy*) nego srednja vrijednost RMS energija preko duljeg segmenta [2]. Za primjer, govor ili glazba s tišinama imat će veći postotak segmenata niske energije nego kontinuirani zvuk.

4. PRIMIJENJENI KLASIFIKACIJSKI MODELI

Unutar poglavlja dan je detaljni pregled primijenjenih modela strojnog učenja pri rješavanju problema klasifikacije te načina navedene primjene pri klasifikaciji glazbenih djela po žanrovima.

Strojno učenje može se podijeliti u tri kategorije:

1. Nadzirano (engl. *Supervised*) – cilj je odrediti nepoznatu funkcionalnu ovisnost između ulaza i izlaza na temelju podatkovnog primjera.
2. Nenadzirano (engl. *Unsupervised*) – na raspolaganju su samo ulazni podaci te je potrebno naći pravilnosti u njima kako bi se dobio izlaz.
3. Podržano (engl. *Reinforced*) – postoji agent koji poduzima akcije te metodom pokušaja i pogreški uz dobivenu povratnu informaciju o uspješnosti poduzete akcije dolazi do rješenja.

Iz razloga što se rad svodi isključivo na klasifikaciju, koja pripada kategoriji nadziranog učenja, ostale kategorije neće biti dalje objašnjavane. Dakle, kod nadziranog učenja podatkovni skup koji se sastoji od ulaznog i izlaznog vektora; za svaki ulazni vektor $\boldsymbol{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}]$, postoji odgovarajuća izlazna veličina te ovisno je li ona kontinuirana ili diskretna, govori se o regresiji, odnosno klasifikaciji. Klasifikacija se jednostavno može objasniti kao pridruživanje ispravnih oznaka (engl. *label*) ulaznim veličinama. Prilikom određivanja nepoznate funkcionalne ovisnost između ulaznih i izlaznih veličina, nastoji se minimizirati empirijsku pogrešku (engl. *empirical error*), koja predstavlja broj ili postotak odstupanja previđenih vrijednost od stvarnih (engl. *ground truth*) [15], a dana je formulom:

$$E(h|D_{train}) = \sum_{i=1}^{n} f\left(h\left(\boldsymbol{x}^{(i)}\right) \neq \boldsymbol{y}^{(i)}\right) \; gdje\; je\; f(a \neq b) = \begin{cases} 1, & a \neq b \\ 0, & a = b \end{cases} \tag{4.1.}$$

pri čemu *h* predstavlja jedan od odabranih modela koji predviđa izlaznu veličinu na osnovi ulaznog vektora $\boldsymbol{x}$:

$$h\left(\boldsymbol{x}^{(i)}\right) = \begin{cases} 1, & ako\ h\ klasificira\ \boldsymbol{x}^{(i)}\ kao\ pozitivan\ primjer \\ 0, & ako\ h\ klasificira\ \boldsymbol{x}^{(i)}\ kao\ negativan\ primjer \end{cases} \tag{4.2.}$$

dok je skup za učenje:

$$D_{train} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{n_{train}} \tag{4.3.}$$

Dane jednadžbe (4.1. - 4.3.) odnose se na problem binarne klasifikacije. Za probleme više-klasne klasifikacije kao što je klasifikacija glazbenih djela po žanrovima, klasa K pripada skupu $C_k (k = 1, \dots, K)$. Pri tome izlazna veličina je vektor od K elemenata:

$$\boldsymbol{y}_k^{(i)} = \begin{cases} 1, & ako\ je\ \boldsymbol{x}^{(i)} \in C_k \\ 0, & ako\ je\ \boldsymbol{x}^{(i)} \in C_j, j \neq k \end{cases}, \tag{4.4.}$$

pa se problem može svesti na K problema binarne klasifikacije pri čemu treba naučiti K modela:

$$h_k(\boldsymbol{x}^{(i)}) = \begin{cases} 1, & ako\ je\ \boldsymbol{x}^{(i)} \in C_k \\ 0, & ako\ je\ \boldsymbol{x}^{(i)} \in C_j, j \neq k \end{cases}, \tag{4.5.}$$

Empirijska pogreška tada predstavlja sumu preko svih predikcija za sve klase:

$$E(\{h_k\}_{k=1}^{K} | D_{train}) = \sum_{i=1}^{n_{train}} \sum_{k=1}^{K} f(h_k(\boldsymbol{x}^{(i)}) \neq \boldsymbol{y}_k^{(i)}) \tag{4.6.}$$

3.1. *k* najbližih susjeda

Klasifikator *k* najbližih susjeda (engl. *K nearest neighbors* – *k*-NN), neparametarska je metoda, što znači da radi na principu „slični ulazi daju slične izlaze“. Za svaki novi mjerni uzorak metoda izgrađuje novi lokalni model te je potrebno imati na raspolaganju čitav skup za učenje tijekom eksploatacije modela. Izgradnja modela se odgađa sve dok nije potrebno provesti predikciju. Algoritam *k*-NN-a radi na sljedeći način:

1. Odrediti konstantu *k* koja označava broj susjeda koji će utjecati na rezultat.
2. Izračunati udaljenost između uzorka nepoznate klase i svih ostalih uzoraka podatkovnog skupa.
3. Poredati udaljenost uzlazno.
4. Zbrojiti broj klasa prvih *k* elemenata poredanog niza.
5. Većinskim glasanjem odrediti klasu nepoznatog uzorka.

Prednost *k*-NN metode jest jednostavna implementacija, no bez obzira na nju omogućava aproksimaciju složenih nelinearnih odnosa jer koristi lokalnu informaciju pri predikciji. S druge strane zbog istih razloga ima nedostatke u pogledu velike memorijske zahtjevnosti iz razloga što je potreban cijeli podatkovni skup i pri predikciji. Nadalje, osjetljiv je na sve ulazne veličine bez obzira na dimenziju prostora.

3.1.1. **Razrada *k*-NN modela**

Kako je ranije navedeno, *k*-NN je jedan od jednostavnijih algoritama klasifikacije pa je tako i njegova izrada poprilično intuitivna. Potrebno je odrediti broj susjeda *k*; ako se uzme premala vrijednost, predikcije postaju manje stabilne, jer uzorci neke klase, koji se značajno razlikuju od ostalih uzoraka iste klase (engl. *outlieri*) imaju veći utjecaj. S druge strane velike vrijednosti izglađuju predikciju, no čine granice između klasa teže uočljivima [15].

Također prilikom primjene *k*-NN algoritma, potrebno je definirati način na koji se određuje udaljenost između uzorka nepoznate klase i svih ostalih uzoraka podatkovnog skupa. Najčešće se primjenjuje euklidska udaljenost:

$$d(p,q) = \sqrt{(p-q)^2} \tag{4.7.}$$

Nadalje, moguće je odrediti težinu svakog glasa koji pridonosi klasifikaciji, ovakva implementacija poboljšava slučaj predikcije kada distribucija uzoraka unutar podatkovnog skupa nije proporcionalna, odnosno kada sve klase nisu jednako zastupljene. Pri tome, najčešće se kao težina uzima udaljenost, drugim riječima glasovi bližih susjeda vrijede više.

Za implementaciju algoritma primijenjena je *KNeighborsClassifier* funkciju iz *scikit learn* biblioteke [16]. Pri određivanju modela podatkovni skup je standardiziran skaliranjem ulaznih veličina tako da imaju srednju vrijednost 0 te varijancu 1. Ovo se naziva standardizacija (engl. *z-score normalization*):

$$\tilde{x}_j = \frac{x_j - \bar{x}_j}{\sigma_j} \tag{4.8.}$$

pri čemu je srednja vrijednost:

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_j^{(i)} \tag{4.9.}$$

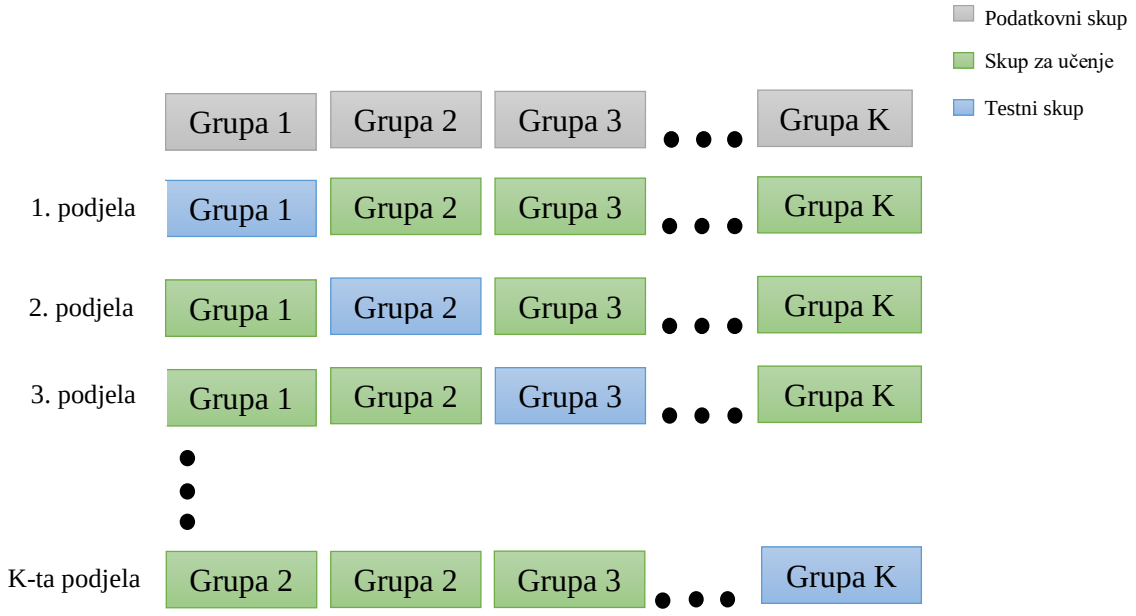
a varijanca :

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} - \bar{x}_j)^2} \tag{4.10.}$$

gdje $\boldsymbol{x^{(i)}}$ predstavlja jedan mjerni uzorak.

Nadalje, originalni skup se dijeli na dva podskupa, jedan za učenje te drugi za testiranje. Nakon toga, prilikom određivanja *k*-NN modela, primijenjena je funkcija *GridSearchCV* iz iste biblioteke koja prima niz parametara koji se žele isprobati, koja značajno olakšava testiranje jer nije potrebno ručno mijenjati vrijednosti parametara i ponovno pokretati kod, već je dovoljno joj predati rječnik koji sadrži parametre i sve njihove vrijednosti koje se žele isprobati. Dodatno, funkcija uključuje i metodu unakrsnog vrednovanja (engl. *cross – validation - CV*) koja poboljšava točnost algoritma iz razloga što nije potrebno izvorni podatkovni skup podijeliti još i na validacijski, nego se skup za učenje može upotrijebiti za validaciju. Upotrebom *k-fold CV* koja dijeli ulazni skup na *k* grupa otprilike iste veličine; Za validaciju se uzima jedan skup, dok se ostalih *k-1* skupova primjenjuje za učenje. Navedeni postupak se ponavlja *k* puta. Na slici 4.1. je prikazan način na koji se podatkovni skup dijeli pri svakoj iteraciji algoritma:

$$KCV = \frac{1}{k}\sum_{i=1}^{k} MSE_i \;\; gdje\ je\ MSE_i = (y^{(i)} - y'^{(i)})^2 \tag{4.11.}$$



Slika 4.1. Ilustracija *k-fold* metode

Konačno, dobivanjem najboljih parametara pokreće se testiranje na testnom skupu koji je kreiran na početku programa, te su rezultati prikazani u poglavlju 5.

3.2. Stroj s potpornim vektorima

SVM je također metoda nadziranog učenja, te se može primjenjivati i za klasifikaciju i za regresiju. Isto tako vodi se pretpostavkom da se slične vrijednosti grupiraju zajedno u prostoru. Metoda tako pokušava definirati granice između klasa na način da je udaljenost između klasa što veća. Drugim riječima, definira *n-1* dimenzionalnu hiperravninu gdje *n* predstavlja broj značajki. Ovisno na koju stranu hiperravnine novi podatak padne, pridružuje mu se ta klasa.

S obzirom da uspješnost metode ovisi o raspodjeli podataka, odnosno koliko se dobro grupiraju i na koji način, metodu je moguće inicijalizirati sa različitim funkcijskim jezgrama (engl. *kernel functions*). Ako su podatci linearno odvojivi dovoljno je upotrijebiti linearnu jezgru, dok kod podataka koji zahtijevaju polinomske granice, jer se podatci preklapaju u prostoru moguće je upotrijebiti različite funkcijske jezgre od kojih su najčešće polinomska (engl. *polynomial*), jezgra sa radijalnom osnovnom funkcijom (engl. *radial basis function*),…[16]

3.2.1. **Određivanje parametara SVM modela**

Implementacija modela odrađena je primjenom *SVC* (engl. *Support-vectro classifier*) funkciju iz *sklearn.svm* biblioteke. Za pretraživanje najboljih parametara upotrijebljena je ranije objašnjena funkcija *GridSearchCV*, no zbog vremenskih zahtjeva nisu ispitane sve moguće kombinacije parametara. Također, standardizacija i podjela izvornog podatkovnog skupa, koji su objašnjeni u prijašnjem postupku su upotrijebljeni i ovdje.

Radi jednostavnosti proces treniranja će se objasniti za slučaj binarne klasifikacije [15]. Za podatkovni skup S = {(x_1,y_1), … , (x_n, y_n)}, pri čemu je (y = 1 ili -1). potrebno je pronaći *w i* w_0 takve da:

$$w^T \boldsymbol{x}^t + w_0 \geq 1 \quad za\ \boldsymbol{y}^t = 1 \tag{4.12.}$$

$$w^T \boldsymbol{x}^t + w_0 \leq -1 \quad za\ \boldsymbol{y}^t = -1 \tag{4.13.}$$

odnosno

$$\boldsymbol{y}^t(w^T\boldsymbol{x}^t + w_0) \geq 1 \tag{4.14.}$$

tada je udaljenost uzorka $\boldsymbol{x}^t$ od funkcije koja razdvaja klase:

$$\frac{\boldsymbol{y}^t(w^T\boldsymbol{x}^t + w_0)}{\|w\|} \tag{4.15.}$$

a ona treba biti barem neka vrijednost ρ:

$$\frac{\boldsymbol{y}^t(w^T\boldsymbol{x}^t + w_0)}{\|w\|} \geq \rho\,, \forall t \tag{4.16.}$$

Kako je potrebno maksimizirati ρ, ali za to postoji beskonačan broj rješenja, postavlja se ρ||w|| = 1. Kako bi se maksimizirala margina, potrebno je minimizirati ||w|| te se dobiva optimizacijski problem:

$$min\frac{1}{2}\|w\|^2 \; za \; \boldsymbol{y}^t(w^T\boldsymbol{x}^t + w_0) \geq 1 \tag{4.17.}$$

SVM ima nekoliko parametara koji se moraju definirati ukoliko se želi dobiti što bolji model. Kako bi se spriječila prenaučenost (engl. *overfitting*) modela, na skup za učenje primjenjuje se regularizacija uvođenjem parametra *C* koji se naziva i faktorom pogreške[17]. *Overfitting* je najlakše objasniti na primjeru učitelja i učenika: Učitelj sprema učenika za test tako što mu daje pitanja koja će biti u testu. Učenik tada nauči sva pitanja napamet te riješi test velikim uspjehom. Nakon toga učenik dolazi na završno testiranje te ne zna niti jedno pitanje koje će se pojaviti u završnom ispitu. S obzirom da nije naučio zaključivati sam, već se oslanjao samo na učenje točnih odgovora, na završnom testu pogrešno odgovara na još neviđena pitanja te mu je uspjeh znatno niži nego na normalnom testu. Regularizacija sprječava *overfitting* tako što penalizira uspjeh modela za svaki pogrešno klasificiran uzorak. Postoji nekoliko načina regularizacije, no SVC implementira *L2* regularizaciju koja se naziva još i *Ridge regularization,* a za penalizaciju primjenjuje kvadrat sume težina. Tada (4.17.) postaje:

$$min\frac{1}{2}\|w\|^2 + C\sum_t \xi^t \; za \; y^t(w^T x^t + w_0) \geq 1 - \; \xi^t \tag{4.18.}$$

Drugi parametar *gamma* određuje koliko utjecaja pojedini uzorak ima [17]. Što je *gamma* veća, određuju se strože granice, što povećava opasnost od *overfittinga.*

Ključ dobre razredbe SVM-a leži u odabiru jezgrene funkcije te odgovarajuće vrijednosti pripadajućih parametara. Pomoću jezgrene funkcije (engl. *kernel*) određuje se način iscrtavanja margine. Linearna jezgra je osnovna i većinom se primjenjuje kada su podaci linearno odvojivi:

$$K(x^t, x) = (x^T x^t) \tag{4.19.}$$

Polinomska jezgra stupnja *d*:

$$K(x^t, x) = (x^T x^t + 1)^d \tag{4.20.}$$

Radial-basis jezgra najčešće je primjenjivana, jer se pokazala efektivnom kod linearno neodvojivih podataka. Definira sfernu jezgru gdje *s* predstavlja radijus*:*

$$K(x^t, x) = exp\left[-\frac{\|x^T - x^t\|^2}{2s^2}\right] \tag{4.21.}$$

Nakon određivanja najboljih parametara, model je testiran na testnom skupu, a rezultati su prikazani u poglavlju 5.

3.3. Neuronske mreže

Neuronske mreže (engl. *Neural network*) inspirirane su promatranjem ljudskog mozga. Mozak se sastoji od brojnih neurona, koji prenose signale jedan na drugoga. Ideja je, dakle, načiniti mrežu jednostavnih čvorova koji će odraditi izračun te rezultat poslati na sljedeći čvor. Neuronske mreže tako počinju od *perceptrona* koji je jednostavan algoritam za nadziranu binarnu klasifikaciju. Ulazne se podatke združuje u obliku težinske sume te se na izlazu daje logička jedinica „1“ ili logička nula „0“, ovisno je li suma veće od neke granične vrijednosti ili nije.

Neuronske mreže se tako sastoje od slojeva, koje pak čine čvorovi, a svaki čvor predstavlja jedan neuron. Postoje različiti oblici neuronski mreža: potpuno povezane, djelomično povezane, unaprijedne, cikličke… Također varijabilni su brojevi slojeva, čvorova u svakom sloju i implementiraju se različite aktivacijske funkcije.

3.3.1. **Treniranje neuronske mreže**

U radu je primijenjena unaprijedna mreža (engl. *feedforward neural network*) s potpuno povezanim (engl. *fully connected*) slojevima. Definiranje mreže podrazumijeva strukturiranje

mreže, to jest odabir broja slojeva, broja neurona u svakom sloju te aktivacijske funkcije [15]. Neuronska mreža implementirana je pomoću *Sequential* funkcije *Keras* biblioteke [18]. Navedena funkcija omogućava jednostavnu izgradnju neuronske mreže, sloj po sloj, pri čemu podatci putuju sekvencijalno iz jednog sloja u drugi. Svakom sloju moguće je definirati broj čvorova te aktivacijsku funkciju, a za ulazni sloj potrebno je odabrati ispravan broj ulaza; najčešće se za to uzima broj značajki skupa za učenje. Isprobano je nekoliko struktura mreže, s 1, 2 i 3 skrivena sloja te je izabrana struktura mreže koja se sastoji od ulaznog sloja, nakon čega slijede 3 skrivena sloja sa 128, 64 odnosno 32 čvora, te konačno izlazni sloj. Skriveni slojevi za aktivacijsku funkciju implementira *ReLu* (engl. *rectified linear activation function*), koja na izlazu daje ulazu vrijednost ako je ona pozitivna, a u suprotnom vraća nulu:

$$f(x) = \max(0, \mathrm{x}) \tag{4.22.}$$

Izlazni sloj za aktivacijsku funkciju primjenjuje *softmax* koja je generalizacija logističke regresije na višedimenzionalni prostor, dok broj njegovih čvorova određuje broj klasa, konkretno: 10 čvorova s obzirom na 10 žanrova od koji se sastoji upotrijebljeni podatkovni skup. Odnosno kako logistička regresija daje vjerojatnost (vrijednost između 0 i 1), isto radi *softmax* funkcija, no za višeklasne probleme:

$$f\left(y^{(i)}\right) = \frac{e^{y^{(i)}}}{\sum_{j=0}^{k} e^{y_j^{(i)}}} \tag{4.23.}$$

Primijenjeno je slučajno izostavljanje neurona tijekom učenja (engl. *dropout*). Kako bi se postiglo navedeno, između svakog sloja pozvana je *dropout* funkcija koja slučajnim odabirom ulaz postavlja na 0, određenom frekvencijom pri svakom koraku treniranja kako bi se spriječila prenaučenost [16].

Nakon što je definirana struktura potrebno je odrediti funkciju gubitka (engl. *loss-function*) koja definira način određivanja pogreške pri klasifikaciji za trenutno stanje modela. upotrijebljena je kategorijska empirijska pogreška (engl. *categorical cross entropy*). Zatim, za način optimizacije, odabran je algoritam *Adam* koji je oblik stohastičke metode gradijentnog spusta. Za razliku od klasične metode gradijentnog spusta, gdje se kreće od neke početne vrijednosti te se težine ažuriraju u smjeru negativnog gradijenta kriterijske funkcije na temelju svakog mjernog uzorka konstantom stopom učenja koja se odabire proizvoljno, *Adam* adaptira stopu učenja tijekom izračuna.

Konačno treniranje se provodi s različitim vrijednostima hiperparametara: broj iteracija (engl. *epochs*) i broj uzoraka (engl. *batch size*). Broj uzoraka odnosi se na broj uzoraka za treniranje koje treba ispitati prije nego se težine ažuriraju. Jedan iteracija označava jedan prolaz kroz sve uzorke podataka za učenje, odnosno znači da je svaki uzorak imao priliku ažurirati težine.

5. REZULTATI

U sklopu poglavlja, diskutira se o programskim alatima i bibliotekama upotrijebljenim pri izračunavanju dobivenih rezultata. Nakon toga objašnjen je primijenjeni postupak evaluacije rezultata te naravno objašnjenje dobivenih rezultata. Valja napomenuti, kako je upotrijebljen GTZAN podatkovni skup (potpoglavlje 3.1.3) koji se sastoji od 1000 audio zapisa te je podijeljen na dva podskupa, jedan za učenje te drugi za testiranje u omjeru 80 : 20.

5.1. Python

Za izradu skripta potrebnih za implementaciju klasifikacije glazbenih djela po žanrovima upotrijebljen je skriptni jezik *Python.* Odabran je zbog svoje pogodnosti pri rješavanju problema gdje su podatci predstavljeni na kompleksne načine, odnosno zbog jednostavnog načina na koji omogućuje manipuliranje matricama i nizovima jer su dani podatkovni skupovi najčešće u tome obliku. Dodatno podržava brojne biblioteke od numeričkih, vizualizacijskih, sve do biblioteka za strojno učenje. U konačnici, jezik je otvorenog koda (engl. *open source*), što znači da je besplatan te je korisnička zajednica velika, što značajno olakšava razvoj koda.

Od alternativnih programskih alata često se primjenjuje *Matlab* koji je također skriptni jezik te pruža brz i jednostavan rad s numeričkim proračunima te su svi podatci spremani u obliku matrica, što otežava rad ako se problem ne može predstaviti numeričkom matricom. Nadalje, potrebna je licenca za korištenje, što znači i manja korisnička zajednica.

C programski jezik za razliku od *Python-a* daje bolje performanse, odnosno brži je, no nedostatak je što programer mora sam brinuti o upravljanju memorijom te nema toliko mnogo razvijenih biblioteka za strojno učenje kao što ima *Python*.

5.1.1. **Biblioteke**

- *Matplotlib* – biblioteka upotrijebljena za različite grafičke prikaze.
- *Pandas* – biblioteka upotrijebljena za učitavanje i analizu podatkovnih skupova.
- *ScikitLearn i Keras* [16, 18] – biblioteke primijenjene pri implementaciji algoritama strojnog učenja te evaluaciji rezultata
- *Librosa* [19] – biblioteka primijenjena za obradu zvučnih signala i pri izvlačenju značajki

5.2. Načini evaluacije

Za evaluaciju modela pri problemu klasifikacije najčešće se primjenjuju matrice zbunjenosti (engl. *confusion matrix*). Zbog jednostavnosti, slijedi objašnjenje na problemu binarne klasifikacije $y = \begin{cases} 1 \\ -1 \end{cases}$. Matrica zbunjenosti jednostavno prikazuje kombinacije predviđenih i stvarnih klasa za dane uzorke (Slika 5.1.), moguće su 4 kombinacije:

Stvarno pozitivni (engl. *True positive – TP*) – predviđena je klasa 1 te je predviđanje točno, odnosno ($y=1 \; i \; y'=1$)

Stvarno negativni (engl. *True negative – TN*) – predviđena je klasa -1 te je predviđanje točno, odnosno ($y=-1 \; i \; y'=-1$)

Lažno pozitivni (engl. *False positive – FP*) – predviđena je klasa 1 te je predviđanje netočno, odnosno ($y=-1 \; i \; y'=1$)

Lažno negativni (engl. *False negative – FN*) – predviđena je klasa -1 te je predviđanje netočno, odnosno ($y=1 \; i \; y'=-1$)

		Predviđena vrijednost	
		Pozitivno (1)	Negativno (0)
Stvarna vrijednost	Pozitivno (1)	TP	FN
	Negativno (0)	FP	TN

Slika 5.1. matrica zabunjenosti

Nadalje iz dobivene matrice jednostavno je izračunati metrike koje prikazuju uspješnost klasifikacije:

1. Preciznost (engl. *precision*) pokazuje za svaku predviđenu klasa 1, koliko je puta bilo točno predviđanje:

$$P = \frac{TP}{TP + FP} \tag{5.1.}$$

2. Odziv (engl. *recall*) pokazuje koliko je puta predviđena klasa 1, kada je stvarna klasa 1:

$$R = \frac{TP}{TP + FN} \tag{5.2.}$$

3. Točnost (engl. *accuracy*) pokazuje koliko predviđanja odgovara stvarnoj klasi:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.3.}$$

4. F-mjera (engl. *F-measure*) je harmonijska sredina preciznosti i odziva. Uvodi se jer je teško usporediti dva modela s niskim preciznostima i visokim odzivima ili obrnuto.

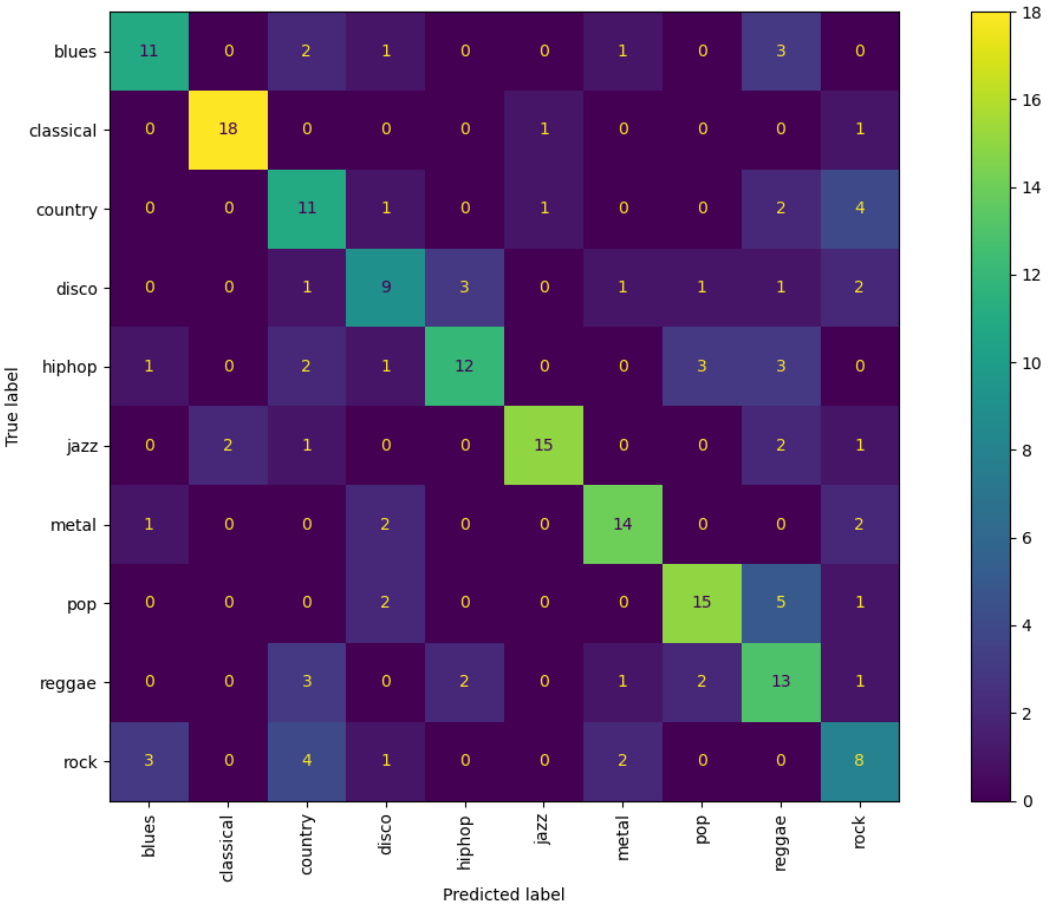
$$F = \frac{2PR}{P + R} \tag{5.4.}$$

Ovisno o tipu problema koji se rješava treba odrediti je li važnije minimizirati *FP* ili *FN*. Odnosno, hoće li se tražiti veća preciznost (manje FP), ili će se tražiti veći odziv (manje FN).

5.3. *k* najbližih susjeda

Metoda k najbližih susjeda, iako najjednostavnija, svejedno daje obećavajuće rezultate prikazane u tablici 5.1. te slici 5.2. Treniranje je ponovljeno nekoliko puta no parametar *k* nije točno određen, odnosno varirao je pri svakom ponovnom treniranju algoritma, ali je utvrđen raspon unutar kojega točnost ne varira značajno, tako su se najbolje vrijednosti pokazale u rasponu 3-11 pri čemu točnosti variraju ± 4 % pri testiranju ako svi žanrovi nisu jednoliko zastupljeni, što se može pridodati neparametarskoj prirodi implementiranog algoritma. Za metodu K najbližih susjeda najbolji način izračuna udaljenosti pokazao se euklidski izračun te je također utvrđeno da pri glasovanju treba primijeniti težinu koja je u ovom slučaju izračunata udaljenost.

Tablica 5.1. i Slika 5.2. prikazuju evaluaciju modela. Cjelokupna točnost algoritma (63 %) čini se niskom, no uzevši u obzir kako se klasificira 10 žanrova, ona nije stvarni pokazatelj uspješnosti već valja promotriti postotke individualnih žanrova.



Slika 5.2 Matrica zbunjenosti metode *k*-NN

Tablica 5.1. Preciznost, Odziv, F-mjera po žanrovima i Točnost *k*-NN

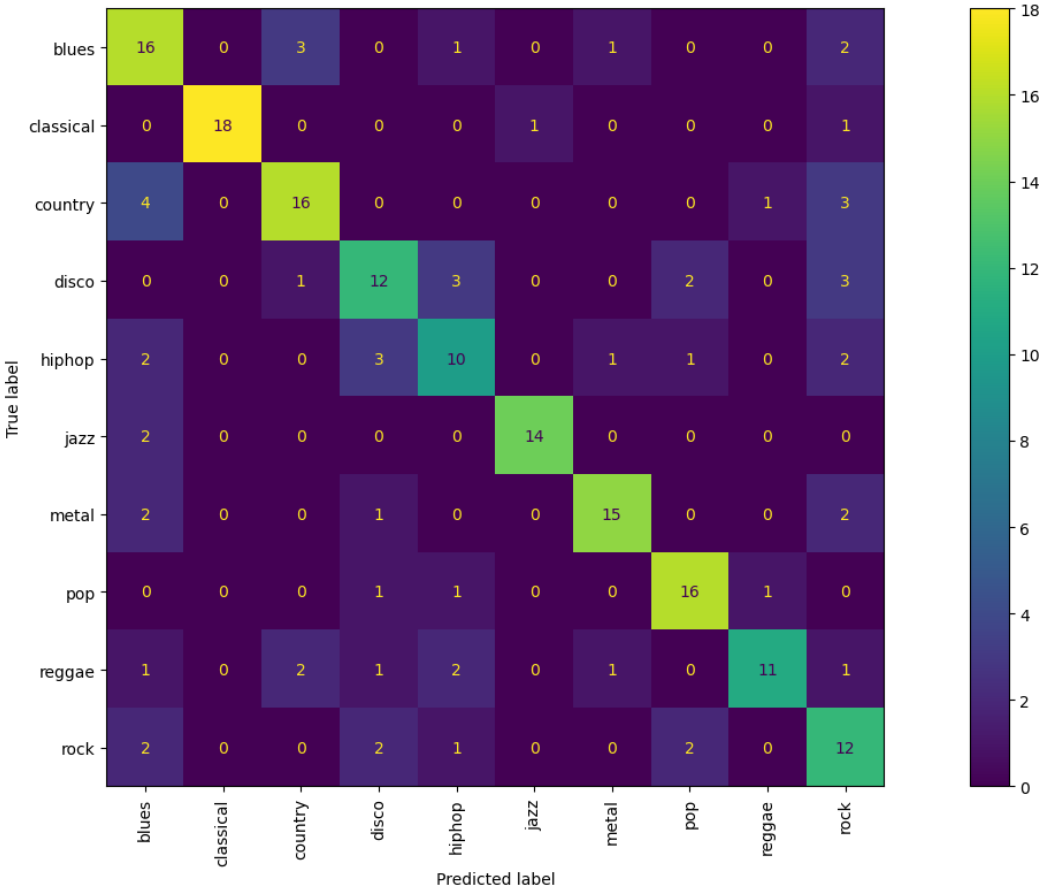
Klasa	**Preciznost (%)**	**Odziv (%)**	**F-mjera (%)**
Blues	69	61	65
Classical	90	90	90
Country	46	58	51
Disco	53	50	51
HipHop	71	55	62
Jazz	88	71	79
Metal	74	74	74
Pop	71	65	68
Reggae	45	59	51
Rock	40	44	42
Točnost (%)		63	

Bez obzira na odabrani *k* iz utvrđenog raspona pokazalo se kako je žanr klasične glazbe redovno najbolje sortiran (90 %), dok *rock* pokazuje najgore rezultate (42 %). Dobiveni rezultati sugeriraju sličnosti među podacima različitih žanrova, to jest razumno je zaključiti kako se uspješnost klasifikacije klasične glazbe može pripisati njenoj specifičnosti u odnosnu na ostale žanrove, dok niska uspješnost klasifikacije *rock* glazbe i njena najčešće pogrešna klasifikacije u *blues*, *country* ili *metal* iskazuje značajne sličnosti tih žanrova. Preciznost *hiphopa* je relativno visoka što je razumljivo s obzirom na njegovu specifičnost vokalnog aspekta, dok se nizak odziv može objasniti činjenicom da koristi mnogo *samplova*, odnosno isječaka iz drugih glazbenih žanrova na koje se onda dodaje navedeni vokalni aspekt. Najčešća zamjena *disco* sa *hiphopom* može se pak pripisati „elektroničkoj" naravi tih dvaju žanrova, odnosno često koriste sintetički proizvedene melodije. No zamjena za *metal, rock* i *reggae* sugeriraju postojanje pogrešaka u podatkovnom skupu.

5.4. Stroj s potpornim vektorima

SVM klasifikator nešto je kompliciraniji od jednostavne metode *k* najbližih susjedna, no isto tako daje i bolje rezultate, koje prikazuje tablica 5.2. te slika 5.3. Ispitivanjem različitih vrijednosti parametara SVM metode za klasifikaciju najbolji parametri su se pokazali: C=100, *gamma = 0.01* te funkcijska jezgra *Radial-basis.*

Nadalje, ponovno se pokazuje kako je najbolje klasificiran žanr klasična glazba što potvrđuje tezu uspostavljenu prijašnjim algoritmom o njenoj specifičnosti. Uspješnost klasifikacije *rock* glazbe porasla je čak 10 %, no i dalje je najviše zamijenjena sa *blues, country* i *metal* žanrom. *Hiphop* nastavlja biti zamijenjen za nekoliko različitih žanrova što ponovno pridonosi korištenje *samplova* iz tih žanrova. Klasifikacije svih žanrova značajno su poboljšane u usporedbi s metodom *k* najbližih susjeda. Ovo sugerira kako *SVM* ipak ima jaču mogućnost pri klasifikaciji podataka koji su poprilično slični, odnosno linearno neodvojivi.



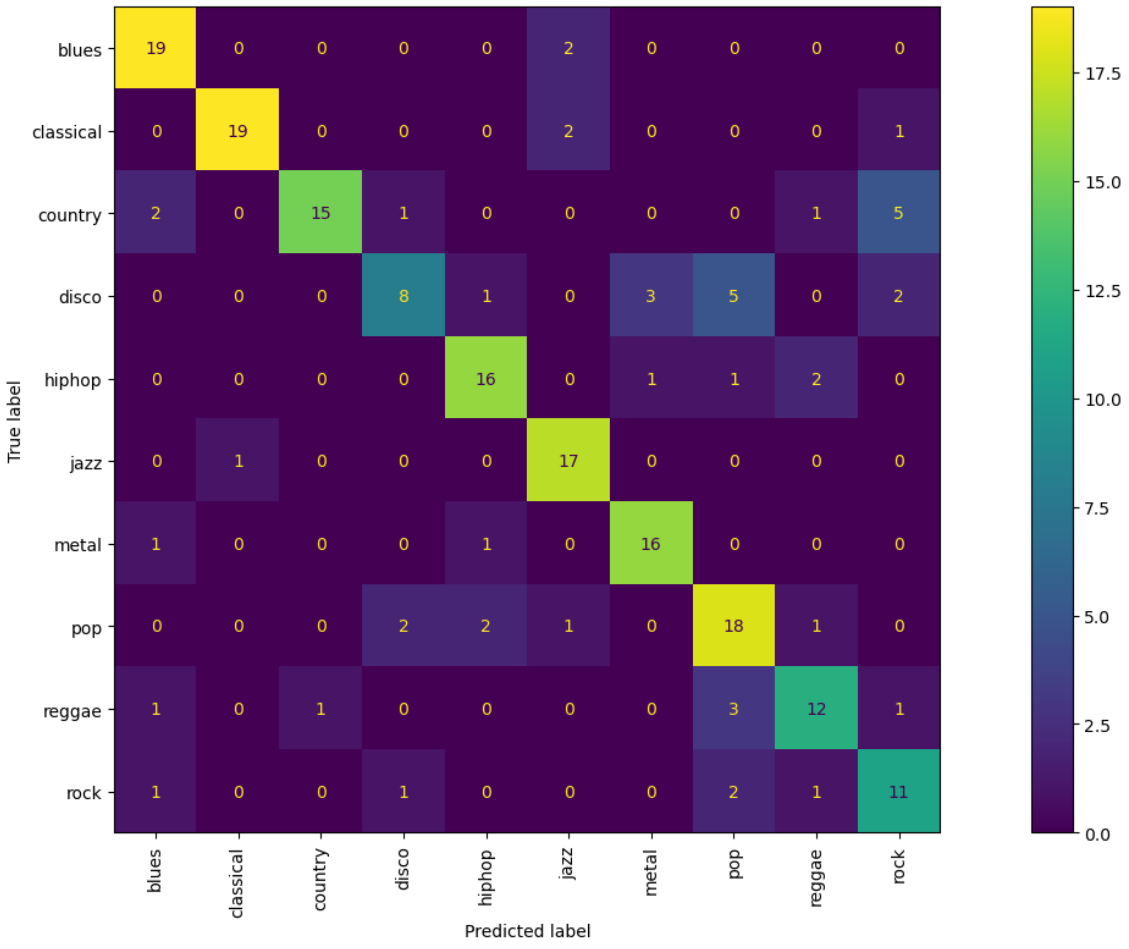
Slika 5.3. matrica zbunjenosti za metodu SVC

Tablica 5.2 . Preciznost, Odziv, F-mjera po žanrovima i Točnost SVC

Klasa	Preciznost (%)	Odziv (%)	F-mjera (%)
Blues	55	70	62
Classical	100	90	95
Country	73	67	70
Disco	60	57	59
HipHop	56	53	64
Jazz	93	88	90
Metal	83	75	79
Pop	76	84	80
Reggae	85	58	69
Rock	46	63	53
Točnost (%)		70	

5.5. Neuronska mreža

Prateći temu složenosti algoritama neuronske mreže najsofisticiranija je metoda strojnog učenja i kao takva očekivano daje najbolje rezultate (Tablica 5.3, Slika 5.4.). Za neuronsku mrežu također je primijenjen *GridSearchCv* radi pronalaska optimalnih hiperparametara, a najbolje vrijednosti su se pokazale: *epochs = 100, batch size = 20.*



Slika 5.4. matrica zbunjenosti za metodu Neuronske mreže

Tablica 5.3. Preciznost, Odziv, F-mjera po žanrovima i Točnost Neuronske mreže

Klasa	Preciznost (%)	Odziv (%)	F-mjera (%)
Blues	79	90	84
Classical	95	86	90
Country	94	62	75
Disco	67	42	52
HipHop	80	80	80
Jazz	77	94	85
Metal	80	89	84
Pop	62	75	68
Reggae	71	67	69
Rock	55	69	61
Točnost (%)		76	

Ponovno je klasična glazba najbolje klasificirana, što je očekivano, iako je F-mjera nešto niža nego prije, no ostale klase pokazuju poboljšanje rezultata u odnosu na prijašnje metode. Ovo je prihvatljivo i ne oduzima od vrijednosti algoritma pošto je razumljivo da se za cjelokupno poboljšanje mora žrtvovati na određenim dijelovima. Zamjena *disco* za *metal* je poprilično neobjašnjiva što može značiti pogreške u podatkovnom skupu, dok je zamjena za *pop* objašnjiva s obzirom da većina popularne glazbe zvuči jako slično *disco* glazbi.

5.6. Usporedba rezultata

Uspoređujući rezultate prema točnosti svake metode (tablice 5.1. – 5.3. i slike 5.2. – 5.4.), neuronske mreže su se pokazale najboljima (76%), nakon njih stroj s potpornim vektorima (70 %), a na zadnjem mjestu se nalazi metoda *k* najbližih susjeda (63%). Ovakvi rezultati su očekivani, neuronske mreže su daleko najkompleksnije te ih je moguće vrlo precizno kalibrirati kako bi riješile čak i najsloženije probleme. Nadalje, metoda *k* najbližih susjeda je jedna od najjednostavnijih metoda strojnog učenja, te značajno ovisi o linearnoj odvojivosti podataka, što u ovom problemu nije bio slučaj, no bez obzira na to i dalje daje prihvatljive rezultate s obzirom na svoju jednostavnost.

Nadalje, valjalo bi usporediti rezultate s obzirom na pojedine žanrove (Tablica 5.4.). Kao što je ranije navedeno, Klasična glazba je daleko najbolje klasificirana bez obzira na metodu koja se upotrijebi. Isto tako sve metode su najgore klasificirale *rock* glazbu. Ovo se može intuitivno objasniti: Klasična glazba definitivno je najspecifičnija od ostalih žanrova, odnosno pokazuje najviše razlika, a razlog za to je najvjerojatnije što koristi drugačije instrumente od svih ostalih žanrova, isto tako često je instrumentalna, što znači da u njoj nema pjevanja. S druge strane *rock* glazba je najgore klasificirana iz razloga što je najsličnija ostalim žanrovima, odnosno često je zamjenjivana sa *blues* i *country* žanrom iz samog razloga što je proizašla iz *blues* glazbe, te koristi iste ili slične instrumente i strukturu pjesama.

Tablica 5.4. F-mjere svih metoda za pojedine žanrove i točnost za metode

Klasa	**NN**	***k*-NN**	**SVM**
Blues	84 %	65 %	62 %
Classical	90 %	90 %	95 %
Country	75 %	51 %	70 %
Disco	52 %	51 %	59 %
HipHop	80 %	62 %	64 %
Jazz	85 %	79 %	90 %
Metal	84 %	74 %	79 %
Pop	68 %	68 %	80 %
Reggae	69 %	51 %	69 %
Rock	61 %	42 %	53 %
Točnost	76 %	63 %	70 %

Konačno, moguće je argumentirati kako točnosti ovih metoda nisu optimalne te imaju mnogo prostora za poboljšanje. S druge strane, ako se pak pogleda uspješnost ljudi pri obavljanju istog zadatka, može se argumentirati kako su podatci relativno dobri što se tiče automatskog klasificiranja žanrova glazbenih djela pomoću računala. Naime, u istraživanju provedenom nad studentima prve godine psihologije *Northwestern* sveučilišta [20], pokazano je kako je uspješnost klasifikacije na segmentima glazbe trajanja 250 ms, bila samo 53 %, a pri klasifikaciji na segmentima duljine 3 s do 70 %. Nadalje ovi rezultati dodatno potvrđuju kompleksnost problema, odnosno teškoću s kojom se mogu definirati razlike između žanrova. Dodatno, osim pronalaženja boljih značajki pri klasifikaciji trebalo bi sagledati i podatkovne

skupove koji se upotrebljavaju. Konkretno, GTZAN je kreiran 2001. godine, te se glazbeni pejzaž značajno promijenio od tada na prijem današnja *pop* glazba više ne zvuči toliko slično *disco* glazbi kao što je u to vrijeme, pa bi se danas ista glazbena djela možda klasificirala u neki drugi žanr.

6. ZAKLJUČAK

Cilj rada bio je riješiti problem klasifikacije glazbenih djela po žanrovima, odnosno istražiti različite algoritme prikladne za klasifikaciju glazbenih žanrova te implementirati i usporediti nekoliko takvih algoritama. Pri tome su odabrane metode K najbližih susjeda koja je jedna od jednostavnijih metoda klasifikacije te je dala relativno dobre rezultate s obzirom na svoju jednostavnost te kompleksnost problema. Drugo je ispitana metoda stroja s potpornim vektorima koja je pokazala poboljšanje u pogledu točnosti klasifikacije te se pokazala boljom pri klasifikaciji pojedinačnih žanrova zbog mogućnosti stvaranja kompleksniji granica odluke. Konačno najbolje rezultate dala je metoda neuronskih mreža, koja je najsofisticiranija metoda od ispitanih. U konačnici sva tri algoritma su se pokazala dostatna izazovu te je utvrđeno kako za poboljšanje rezultata nisu potrebni novi algoritmi i tehnologije, već je potrebno prebaciti fokus na stvaranje cjelovitijih podatkovnih skupova te bolje razumijevanje područja glazbe kako bi se definirale stavke koje bi vjerodostojnije razlikovale glazbena djela po žanrovima, jer u usporedbi s ljudskim performansama pri rješavanju identičnog problema, rezultati ne odstupaju značajno. Područje glazbenog pretraživanja kojemu pripada ovaj problem, relativno je nova znanstvena grana, te probleme koje nastoji riješiti su šaroliki, ali svaki od njih beneficira poboljšanjem uspješnosti konkretne problematike klasifikacije glazbenih djela po žanrovima. Primjene ovih rezultata već danas se koriste u raznim *online streaming* uslugama, kako pri preporučivanju personaliziranih glazbenih popisa za reprodukciju, pa sve do identifikacije o kojoj se pjesmi radi uz pruženi mali isječak pjesme (*Shazam* aplikacija [21]) ili pak uz pjevušenje ili fićukanje melodije pjesme od strane čovjeka. Zaključno, već danas postoje brojne komercijalne primjene za navedenu problematiku, te poznavanje vještina i tehnologija koje ju rješavaju otvara brojna vrata i mogućnosti.

LITERATURA

[1] J. S. Downie, „Music information retrieval“, Annual Review of Information Science and Technology 37: 295-340, 2003.

[2] G. Tzanetakis, P. Cook, „Musical genre classification of audio signal“, IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, srpanj 2002., dostupno na: https://ieeexplore.ieee.org/document/1021072 [09.07.2021.]

[3] P. Fulzele, R. Singh, N. Kaushik, K. Pandey, „A Hybrid Model for Music Genre Classification Using LSTM and SVM“ , *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1-3, kolovoz 2018., dostupno na: https://ieeexplore.ieee.org/document/8530557 [09.07.2021.]

[4] H. Bahuleyan, „ Music Genre Classification using Machine Learning Techniques“, *arXiv:1804.01149,* travanj 2018., dostupno na: https://arxiv.org/abs/1804.01149 [09.07.2021.]

[5] J. Li, D. Sun, T. Cai, Genre Classification via Album Cover, CS230: Deep Learning, ljeto 2019., Stanford University, CA

[6] A. Tsaptsinos, Lyrics-based music genre classification using a hierarchical attention network. *arXiv preprint arXiv:1707.04678*., srapnj 2017., dostupno na: https://arxiv.org/abs/1707.04678 [09.07.2021.]

[7] N. M R and S. Mohan B S, "Music Genre Classification using Spectrograms," *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, 2020, pp. 1-5, doi: 10.1109/PICC51425.2020.9362364

[8] C. Sanden, & J. Z. Zhang, Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 705-714), 2011.

[9] Million Song Dataset, official website by Thierry Bertin-Mahieux, dostpno na: http://millionsongdataset.com/

[10] M. Defferrard, FMA: A Dataset For Music Analysis, Cornell University, 2016., dostupno na: https://arxiv.org/abs/1612.01840v3

[11] Marsyas, GTZAN Genre Collection, Jakob Leben - Powered by Nikola, 2015., dostupno na: http://marsyas.info/downloads/datasets.html

[12] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," in Proceedings of the IEEE, vol. 65, no. 11, pp. 1558-1564, Nov. 1977, doi: 10.1109/PROC.1977.1077

[13] J. Noble, Brightness / Darkness: Timbre Lingo #10, ACTOR Project, 2020, dostupno na: https://www.actorproject.org/tor/timbre-lingo/2020/4/22/brightness-darkness

[14] Lyons, J. (2012). Mel Frequency Cepstral Coefficients, dostupno na: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[15] E. Alpaydin, Introduction to Machine Learning, The MIT Press, Cambridge, Massachusetts, 2014.

[16] Scikit-learn:Machine Learning in Python, Pedregosa et al., JMLR 12, pp.2825-2830, 2011. Dostupno na: https://scikit-learn.org/stable/modules/svm.html#svm, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html , https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

[17] Mace. S. T., Wagoner, C. L., Teachout, D. J., & Hodges, D. A. (2012). Genre identification of very brief musical excerpts. Psychology of Music, 40(1), 112-128.

[18] F. Chollet, keras, F. Chollet and others, 2015. Dostupno na: https://keras.io/api/layers/regularization_layers/dropout/

[19] B. McFee, et al., librosa/librosa: 0.8.1rc2, librosa development team, 2013. – 2021., dostupno na: https://librosa.org/doc/latest/index.html [09.08.2021.]

[20] Gjerdingen, Robert O. and Perrott, David(2008) 'Scanning the Dial: The Rapid Recognition of MusicGenres', Journal of New Music Research, 37: 2, 93 — 100

[21] S. Jarvis, Shazam, Apple Inc., London, UK, 2002. Dostupno na: https://www.shazam.com/

SAŽETAK

Kroz rad diskutiran je problem klasifikacije glazbenih djela po žanrovima. Postavljena je problematika te su sagledani mogući pristupi rješavanju problema. Dani su dostupni podatkovni skupovi za problematiku te je obrazložen proces izabiranja značajki koje bi mogle dostatno prikazati različite aspekte glazbe ovisno o žanru. Nadalje, uspoređeno je nekoliko mogućih metoda za klasifikaciju, konkretno metoda K najbližih susjeda, stroj sa potpornim vektorima te neuronske mreže. Za svaku od metoda pružen je kratki opis te proces treniranja. Nakon treniranja prikazani su dobiveni rezultati te je odrađena usporedba koja je pokazala uspješnost modela. Neuronske mreže pokazale su se najuspješnijima, za njima slijedi stroj sa potpornim vektorima te je „najlošije“ rezultate dala metoda K najbližih susjeda. Također, usporedbom je utvrđeno kako postoje određeni trendovi vezani za klasifikaciju žanrova pa je tako bez obzira na metodu, Klasična glazba uvijek bila najtočnije klasificirana, dok su najgori rezultati dobiveni za *Rock* žanr. U konačnici, pokazano je kako odabrane metode daju podjednake rezultate iz čega slijedi zaključak da za poboljšanje rezultata nisu potrebi novi, sofisticiraniji algoritmi i tehnologije, već da bi se fokus trebao prebaciti na kreiranje cjelovitijih podatkovnih skupova te na proces izvlačenja značajki koje bi vjernije prikazivale glazbeni žanr u pogledu numeričkih vrijednosti.

Ključne riječi : klasifikacija, glazbeni žanrovi, strojno učenje, glazbeno pretraživanje, neuronske mreže,

ABSTRACT – Music Genre Classification

The problem of music genre classification is discussed in this paper. The scope of the problem is given along with possible solutions. Next, available datasets are listed and feature extraction process for sufficient music genre representation is explained. Furthermore, several classification methods are compared, specifically K Nearest Neighbors, Support Vector Machines and Neural Networks. For each method a short description along with training process is provided. Afterwards, obtained results are compared between methods. Neural Networks have proven to be most successful, followed by Support Vector Machines and finally K Nearest Neighbors with the lowest success rate. Likewise, comparison has shown that there are some trends within the dataset samples hence no matter which method was used, Classical music always had the highest, while Rock music showed the lowest accuracy amongst the genres. Ultimately it has been shown how the evaluated methods give similar results from which it can be concluded that for obtaining higher accuracies there is no need for more sophisticated algorithms or new technologies, but more effort needs to be invested in creating better datasets. Likewise, features for better musical genre representation could be considered.

Key words: classification, music genre, machine learning, music information retrieval, neural networks

ŽIVOTOPIS

Dominik Bošnjak, rođen je 1997. godine u Požegi gdje pohađa i završava, s odličnim uspjehom Osnovnu Školu Julia Kempfa. Zbog strasti prema Tehnici i Informatici, koju su mu prenijeli nastavnici u navedenoj osnovnoj školi, upisuje i završava sa istim uspjehom Gimnaziju Požega, prirodoslovno-matematički smjer. Daljnje obrazovanje nastavlja na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku gdje 2020 godine završava preddiplomski studij te stječe status sveučilišni prvostupnik inženjer računarstva. Trenutno na navedenom fakultetu završava diplomski studij računarstva smjer podatkovne i informacijske znanosti.