SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Izrada sustava preporuke glazbenog sadržaja.

Seminar 1

Tin Popović

Mentor: doc. dr. sc. Damir Pintar

Zagreb, 2023.

Sadržaj

1. Uvod

Sustavi preporuke predstavljaju sustave za filtriranje informacija koji daju prijedloge za stavke koje su najrelevantnije za određenog korisnika u određenom trenutku. Oglasi proizvoda , prijedlog filmova, glazbe samo su neki od primjera rezultata rada sustava preporuke. Dobra implementacija jednog takvog sustava uvelike poboljšava kvalitetu platforme nad kojom radi. Uzmimo društvenu mrežu TikTok na primjer. U svega par kratkih godina TikTok je zaprimio ogromnu popularnost za koju je u većini odgovoran njihov algoritam preporuke.

Postoje dva temeljna pristupa pri izradi sustava preporuke. To su:

1. **Zajedničko filtriranje** - je pristup dizajnu sustava preporuke koji se temelji na pretpostavci da korisnicima sličnih karakteristika su potrebni isti proizvodi.
2. **Filtriranje na temelju sadržaja** - je oblik sustava preporuke gdje se preporuke temelje na sadržaju artikla i korisnikovih preferencija.

U ovom seminaru predstavit ćemo implementaciju jednog sustava preporuke za glazbu i nećemo se detaljno fokusirati na osnovne i temeljne karakteristike sustava preporuke već ćemo njih prikazati kroz implementaciju i na kraju kroz sami rad implementiranog sustava. Sustav kojeg ćemo implementirati je izrađen po principu filtriranja na temelju sadržaja, i napravljen je po uzoru na sustave preporuke kojeg koriste Spotify i Apple Music.

Spotify i Apple Music imaju u svojoj implementaciji razvijene sustave preporuke za glazbu. Tako Spotify tjedno generira niz personaliziranih playlista za svakog korisnika, kao što su „Discover Weakly“, „Made For You“ i slično. Playliste mogu biti generirane na temelju raspoloženja korisnika ili na temelju prijašnjih slušanja.

Spotify je popularan po svojim sustavima preporuke i skoro uvijek mnogo hvaljen zbog točnosti tih sustava. Ali kako zapravo ti sustavi rade? U nastavku ćemo odgovoriti na to pitanje tako da predstavimo nužne korake za implementaciju jednog sustava preporuke. Napominjemo da je u ovom seminaru implementirana samo jedna metoda preporuke, te da Spotify koristi niz drugih i kompleksniji implementacija.

2. Implementacija sustava preporuke.

Finalni produkt ovog seminara je sustav koji će na temelju određene kolekcije pjesama preporučiti niz drugih pjesama. Jedan ciklus rada takvog sustava preporuke je prikazan na Slici 1 Rad sustava preporuke.

Slika.2.1 Rad sustava preporuke

Implementacija sustava ove prirode zahtjeva par koraka. Početni korak naše implementacije je obrada podataka. Koristimo javni podatkovni skup koji sadrži veliki broj pjesama i kolekcija pjesama te niz značajki koji detaljno opisuju svaku pjesmu. Te podatke je potrebno obraditi adekvatnim metodama kako bismo mogli generirati preporuke putem našeg sustava. Metode koje su potrebne za nužnu obradu podataka će biti predstavljeni u nastavku ovog seminara. Naš podatkovni skup sadrži 67 499 zapisa te 30 značajki. Pogledajmo djelomični sadržaj naših podataka.

	pos	artist_name	track_uri	artist_uri	track_name	album_uri	duration_ms_x	album_
0	0	Missy Elliott	0UaMYEvWZi0ZqiDOoHU3YI	spotify:artist:2wIVse2owClT7go1WT98tk	Lose Control (feat. Ciara & Fat Man Scoop)	spotify:album:6vV5UrXcfyQD1wu4Qo2I9K	226863	Coo
1	73	Missy Elliott	0UaMYEvWZi0ZqiDOoHU3YI	spotify:artist:2wIVse2owClT7go1WT98tk	Lose Control (feat. Ciara & Fat Man Scoop)	spotify:album:6vV5UrXcfyQD1wu4Qo2I9K	226863	Coo
2	14	Missy Elliott	0UaMYEvWZi0ZqiDOoHU3YI	spotify:artist:2wIVse2owClT7go1WT98tk	Lose Control (feat. Ciara & Fat Man Scoop)	spotify:album:6vV5UrXcfyQD1wu4Qo2I9K	226863	Coo
3	42	Missy Elliott	0UaMYEvWZi0ZqiDOoHU3YI	spotify:artist:2wIVse2owClT7go1WT98tk	Lose Control (feat. Ciara & Fat Man Scoop)	spotify:album:6vV5UrXcfyQD1wu4Qo2I9K	226863	Coo
4	1	Missy Elliott	0UaMYEvWZi0ZqiDOoHU3YI	spotify:artist:2wIVse2owClT7go1WT98tk	Lose Control (feat. Ciara & Fat Man Scoop)	spotify:album:6vV5UrXcfyQD1wu4Qo2I9K	226863	Coo

Slika 2.2 Sadržaj podatkovnog skupa

Na gornjoj slici ne možemo vidjeti čitav sadržaj jednog zapisa jer jedan podatak je opisan s 30 značajki. Iz gornje slike vidimo da se pjesma „Lose Control“ od Missy Elliot ponavlja. Nije riječ o

duplikatu već o činjenici da se ta pjesma nalazi u nekoliko različitih playlista. Daljinom analizom utvrdili smo da u našem podatkovnom skupu je prisutno 35 000 pjesama. Za izradu sustava preporuke nisu potrebni svih 30 značajki stoga fokusirat ćemo se na one koje najbolje opisuju pjesme. Izabrali smo 16 značajki koje smo podijelili na tri skupine zbog bolje preglednosti. To su sljedeće skupine:

1. **Metadata** – značajke koje opisuju temeljne karakteristike pjesme. To su u našem skupu: artist_name, id, artist_pop (popularnost glazbenika), track_pop (popularnost pjesme).
2. **Audio značajke** – Danceability, Energy, Key, Loudness, Mode, Speechines, Acousticness, Instrumentalness, Liveness, Valence, Tempo.
3. **Tekstualne značajke** – track_name. Ovu značajku smo izdvojili jer će zahtijevati poseban oblik obrade u nastavku.

METADATA	AUDIO ZANČAJKE	TEKSTUALNE ZNAČAJKE
Artist_name, id, artist_pop, Track_pop	Danceability, Energy, Key, Loudness, Mode, Speechines, Acousticness, Instrumentalness, Liveness, Valence, Tempo.	Track_name

Naš podatkovni skup nije imao nedostajućih vrijednosti i sadrži veliki broj zapisa te stoga nije bilo potrebno uvoditi tehnike generiranja novih podataka. Na temelju gornje podjele naših varijabli vidimo da naše značajke su različite prirode. Zbog toga one zahtijevaju različit pristup u daljnjoj obradi podataka. U svrhu implementiranja dobrog glazbenog sustava preporuke potrebno je obraditi značajke na adekvatan način. U ovom seminaru je provedena sentimentalna analiza, „One-Hot Encoding“ , „TF-IDF“ metoda te normalizacija značajki. Započnimo sa sentimentalnom analizom naslova pjesme

2.1 Sentimentalna analiza

Dobar i kreativan naslov pjesme može uvelike pomoći u popularnosti pjesme a time i u preporuci, stoga želimo zadobiti određenu količinu informacije iz njega. Informacija iz teksta se može dobiti putem besplatne Python TextBlob biblioteke. Putem te biblioteke mi ćemo proširiti naš podatkovni skup s varijablama „Subjectivity“ i „Polarity“.

Varijabla „Subjectivity“ predstavlja količinu osobnog mišljenja i činjeničnih informacija sadržanih u tekstu. Izražena je u obliku kategoričke varijable s vrijednostima : nisko, srednje, visoko.

Varijabla „Polarity“ predstavlja stupanj snažnog ili jasno definiranog negativnog osjećaja prema tekstu. Također je izražena kao kategorijska varijabla s vrijednostima: negativan, neutralan,pozitivan.

Pogledajmo sada vrijednosti ovih varijabli nad prvi pet pjesama.

Out[10]:

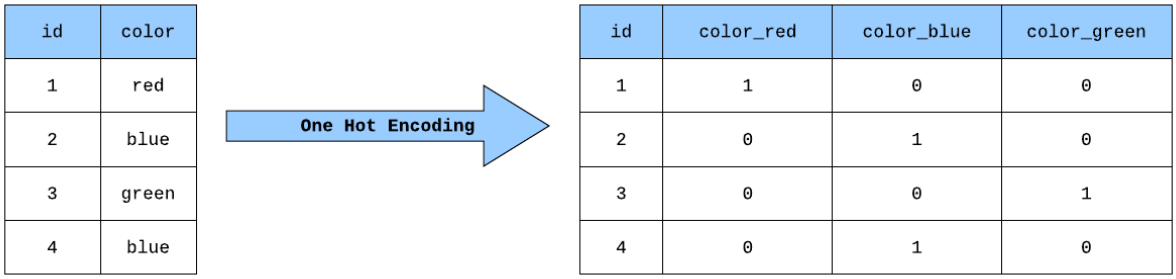
	track_name	subjectivity	polarity
0	Lose Control (feat. Ciara & Fat Man Scoop)	low	Neutral
6	Toxic	low	Neutral
19	Crazy In Love	high	Negative
46	Rock Your Body	low	Neutral
55	It Wasn't Me	low	Neutral

Slika 2.1.1 Vrijednosti varijabla subjektivnosti i polariteta

Vidimo npr. da pjesma „Crazy In Love" ima visok stupanj subjektivnosti, a nisku vrijednost polariteta što znači da ovakav naziv ne bi trebao uzrokovati negativne osjećaje kod korisnika.

U svrhu implementiranja dobrog sustava preporuke potrebno je određene stupce obraditi putem „One-Hot encoding“ operacije. To je operacija koja pretvara kategoričku varijablu u

skupinu bitova s jednom jedinicom i ostalim nulama. Jednostavni primjer koji opisuje rad i rezultat ove operacije prikazan je na slici ispod.



Slika 2.1.2 One Hot Encoding

Varijable subjektivnosti i polariteta koje smo stvorili iznad su kategoričke varijable koje ćemo obraditi putem “OHE“ operacije. Zbog preglednosti, rezultat te operacije nećemo ovdje prikazati, ali svi međurezultati su prisutni u Jupyter bilježnici. Ovakav oblik obrade podataka je potreban jer želimo da naše varijable budu numeričkog tipa. To želimo jer ćemo računati sličnost između pjesama na temelju udaljenosti u nekom n-dimenzijskom prostoru. Zbog toga je operacija „OHE“ od velike važnosti. Mana ove operacije je ogromno proširenje našeg podatkovnog skupa. Primjerice generirali smo jedan novi stupac „Subjectivity“ koji može poprimiti jednu od tri vrijednosti. Nakon „OHE“ metode taj jedan stupac će se proširiti na tri stupca , po jedan za svaku vrijednost te varijable. Tako ćemo imati subjectivity_low, subjectivity_mid, subjectivity_high stupac. Nakon sentimentalne analize moramo se detaljnije posvetiti žanrovima glazbe jer oni predstavljaju jednu važnu karakteristiku pjesama.

2.2 TF-IDF

Žanr glazbe zasigurno je najvažnija varijabla prilikom generiranja preporuka sličnog glazbenog sadržaja. Prilikom istraživanja nove glazbe većina ljudi se orijentira najviše na žanr glazbenika ili pjesme. Stoga je potrebno osjetljivo pristupiti analizi i obradi žanrova prilikom izrade sustava preporuke za glazbu jer ne želimo osobi koja je veliki hip-hop obožavatelj preporučiti pjesme „country“ žanra. Podatkovni skup s kojim mi raspolažemo jednoj pjesmi ne pridružuje samo jedan žanr, već jedna pjesma može biti klasificirana u više različitih žanrova. To možemo vidjeti i na slici ispod.

	track_name	genres_list
0	Lose Control (feat. Ciara & Fat Man Scoop)	[dance_pop, hip_hop, hip_pop, pop, pop_rap, r&...
6	Toxic	[dance_pop, pop, post-teen_pop]
19	Crazy In Love	[dance_pop, pop, r&b]
46	Rock Your Body	[dance_pop, pop]
55	It Wasn't Me	[pop_rap, reggae_fusion]

Slika 2.2.0.1 Žanr pjesme

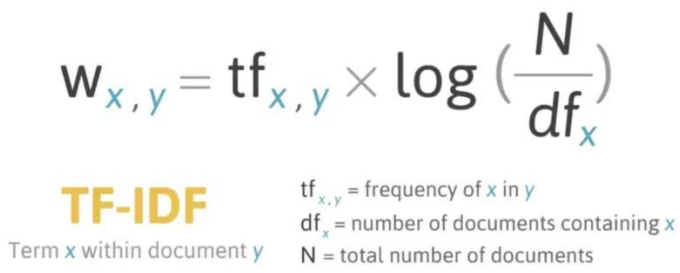
Ovakav način klasifikacije pjesama prisutan je i kod Spotify-a. Vidimo da ne postoji samo pop glazba, već teen-pop, dance-pop, pop rap, itd… Ovakav detaljan pristup klasifikaciji je zapravo dobar za sustave preporuke jer ne bi bilo dobro kategorizirati sve pop pjesme u jednu kategoriju. Na ovaj način postoji jasna razlika između njih.

Sustavi preporuke su zapravo modeli strojnog nenadziranog učenja gdje podaci nemaju oznaku. U metodama nenadziranog učenja zadatak modela je grupiranje neoznačenih podataka. Žanr glazbe predstavlja jedan kriteriji grupiranja podataka.

Iz prijašnje slike skupa podataka mogli smo vidjeti da neke vrijednosti žanra su općenitije od drugih, te možemo i razumno pretpostaviti da nisu svi žanrovi jednako prisutni niti važni u našem podatkovnom skupu. Stoga je potrebno dodijeliti težinu žanrovima. To ćemo učiniti putem **TF_IDF** (engl. Term Frequency-Inverse Document Frequency) metode. Ta metoda uzima u obzira dva faktora , a to su:

1. **Term frequency (TF)**: broj ponavljanja riječi (tj. termina) unutar nekog dokumenta, podijeljen s ukupnim brojem riječi u tom dokumentu.
2. **Inverse Document Frequency (IDF)**: logaritamska vrijednost ukupnog broj dokumenata gdje je određena riječ (tj. termin) prisutan.

Vrijednost TF predstavlja važnost riječi u jednom dokumentu, dok IDF predstavlja važnost riječi u svim dokumentima. U našem slučaju dokumenti su zapravo pjesme, a termini ili riječi su pojedinačni žanrovi. Želimo znati koliko je dominantan žanr u jednoj pjesmi te koliko je on dominantan u čitavom podatkovnom skupu. Na ovaj način svakom tipu žanra ćemo dodijeliti njemu odgovarajuću težinu. Pregledniji prikaz metode TF-IDF prikazan je ispod.



Slika 2.0.2.2 TF-IDF metoda

Slično kao i kod „OHE“ metode i ovdje ćemo svaki žanr pretvoriti u zasebni stupac. To je potrebno kako bi uspjeli ostvariti usporedbu između različitih kolekcija pjesama. Kao i kod „OHE“ metode i ovdje ćemo se susresti s velikim povećanjem dimenzionalnosti skupa. Nakon provođenja TF-IDF metode utvrdili smo da postoje sveukupno 2147 različitih klasifikacija žanrova u našem skupu. Zbog preglednosti nećemo ih ovdje prikazati, ali još jednom nam to potvrđuje koliko zapravo Spotify detaljno razlikuje pojedine žanrove.

Sažmimo sve potrebne korake koje smo predstavili. Prvo smo proširili podatkovni skup s varijablama subjektivnosti i polariteta na temelju naziva pjesme. Iste smo obradili putem OHE metode. Zatim smo koristeći TF-IDF metodu dodijelili težinu žanrovima pjesme. Preostaje nam još jedan korak obrade podataka , a to je normalizacija značajki.

2.3 Normalizacija značajki.

Naglasili smo da se sustavi preporuke mogu smatrati kao modeli strojnog učenja. Klasičan korak prilikom izrade modela strojnog učenja je normalizacija značajki. Ona je nužna jer ne želimo da značajke veće magnitude imaju veću važnost prilikom generiranja novih preporuka. Postoje različiti oblici skaliranja značajki kao što su standardno skaliranje, MinMax skaliranje i slično. U ovom seminaru provedeno je MinMax skaliranje nad audio značajkama.

Na slici 2.2.3 Metoda obrade podataka predstavljen je kod koji sažima sve potrebne korake obrade podataka.

```python
def create_feature_set(df, float_cols):

    # Tfidf genre lists
    tfidf = TfidfVectorizer()
    tfidf_matrix =  tfidf.fit_transform(df['genres_list'].apply(lambda x: " ".join(x)))
    genre_df = pd.DataFrame(tfidf_matrix.toarray())
    genre_df.columns = ['genre' + "|" + i for i in tfidf.get_feature_names()]
    genre_df.drop(columns='genre|unknown') # drop unknown genre
    genre_df.reset_index(drop = True, inplace=True)

    # Sentiment analysis => subjectivity and polarity
    df = sentiment_analysis(df, "track_name")

    # One-hot Encoding => for categorical values
    subject_ohe = ohe_prep(df, 'subjectivity','subject') * 0.3
    polar_ohe = ohe_prep(df, 'polarity','polar') * 0.5
    key_ohe = ohe_prep(df, 'key','key') * 0.5
    mode_ohe = ohe_prep(df, 'mode','mode') * 0.5

    # Normalization
    # Scale popularity columns
    pop = df[["artist_pop","track_pop"]].reset_index(drop = True)
    scaler = MinMaxScaler()
    pop_scaled = pd.DataFrame(scaler.fit_transform(pop), columns = pop.columns) * 0.2

    # Scale audio columns
    floats = df[float_cols].reset_index(drop = True)
    scaler = MinMaxScaler()
    floats_scaled = pd.DataFrame(scaler.fit_transform(floats), columns = floats.columns) * 0.2

    # Concanenate all features
    final = pd.concat([genre_df, floats_scaled, pop_scaled, subject_ohe, polar_ohe, key_ohe, mode_ohe], axis = 1)

    # Add song id
    final['id']=df['id'].values

    return final
```

Slika 2.3.2.3.1 Metoda obrade podataka

Nakon obrade podatkovnog skupa, možemo krenuti s preporukom pjesama. U nastavku seminara predstavit ćemo kako radimo preporuku, kako mjerimo sličnost između pjesama te kako ocjenjujemo rad sustava preporuke. Važno je naglasiti kako prilikom izrade sustava preporuke korak obrade podataka je veoma važan i zahtjeva osjetljivo rukovanje jer on je u najvećoj mjeri odgovoran za uspješan ili neuspješan rad sustava.

3. Generiranje preporuka.

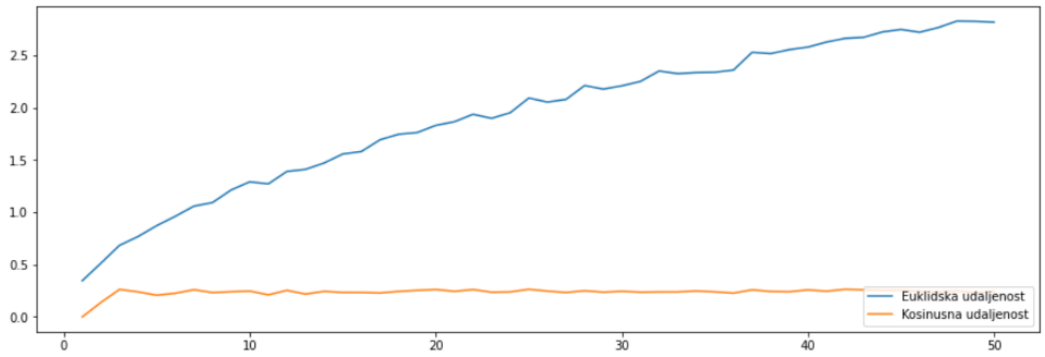
Obradili smo podatke te sada konačno možemo započeti s preporukom pjesama. Na početku seminara smo naglasili kako naš sustav na ulaz prima testnu playlistu , te na temelju njenih karakteristika generira preporuke. Za testni primjeri iskoristit ćemo "Mom's playlist", listu pjesama čiji djelomični sadržaj možemo vidjeti na slici 3.1 Sadržaj playliste.

artist_name	track_name
The Killers	Mr. Brightside
Rihanna	We Found Love
American Authors	Best Day Of My Life
Clean Bandit	Rather Be (feat. Jess Glynne)
Sia	Chandelier
Hozier	Jackie And Wilson
Aloe Blacc	I Need a Dollar
Aloe Blacc	Wake Me Up - Acoustic
John Legend	All of Me
Pharrell Williams	Happy - From "Despicable Me 2"

Slika 3.1 Sadržaj playliste

Ovu playlistu je potrebno obraditi na isti način kao što je opisano ranije. Na taj način ćemo i ovom skupu proširiti broj varijabli na 2179. Nakon obrade i proširenja testne playliste izdvojit ćemo one pjesme koje se pojavljuju u playlisti iz našeg skupa jer ne želimo korisniku predložiti pjesmu koja se već nalazi u njegovoj kolekciji pjesama.

Usporedbu između pjesama playliste i ostatke pjesme ćemo ostvariti putem kosinusne udaljenosti. Razlog zašto koristimo upravo ovu mjeru sličnosti je činjenica da je kosinusna udaljenost otporna na porast dimenzionalnosti , a mi imamo podatkovni skup s preko dvije tisuće varijabli po zapisu. Usporedba euklidske i kosinusne udaljenosti pri porastu dimenzije prostora prikazana je na slici 3.2 Euklidska i kosinusna udaljenost.



Slika 3.2 Euklidska i kosinusna udaljenost

Cilj nam je pronaći one pjesme koje imaju što manju udaljenost od naše playliste. Pjesme koje imaju slične karakteristike audio značajki, subjektivnosti, polarnosti će imati malenu udaljenost , i stoga sustav će preporučiti upravo te pjesme. Prilikom implementacije „TF-IDF“ metode naglasili smo da je prisutan velik broj žanrova u našem skupu te da smo sve žanrove morali odvojiti u zasebne stupce. Zbog toga većina značajki u našem obrađenom skupu se odnosi isključivo na žanr pjesme. Stoga sličnost između žanrova dviju pjesama uvelike smanjuje kosinusnu udaljenost između pjesama. Na slici ispod prikazan je kod koji je odgovoran za generiranje preporuka novih pjesama.

```
In [29]: def generate_playlist_recos(df, features, nonplaylist_features):
             '''
             Generated recommendation based on songs in aspecific playlist.
             ---
             Input:
             df (pandas dataframe): spotify dataframe
             features (pandas series): summarized playlist feature
             nonplaylist_features (pandas dataframe): feature set of songs that are not in the selected playlist

             Output:
             non_playlist_df_top_40: Top 40 recommendations for that playlist
             '''

             non_playlist_df = df[df['id'].isin(nonplaylist_features['id'].values)]
             # Find cosine similarity between the playlist and the complete song set
             non_playlist_df['sim'] = cosine_similarity(nonplaylist_features.drop('id', axis = 1).values, features.values.reshape(1, -1))
             non_playlist_df_top_40 = non_playlist_df.sort_values('sim',ascending = False).head(40) #return top 40 recomendations

             return non_playlist_df_top_40
```

Slika 3.3 Kod za generiranje preporuke

Nakon što smo obradili našu testnu playlistu, te odvojili ostatak pjesama pokušali smo pronaći najsličnije pjesme na temelju kosinusne udaljenosti. Rezultat našeg sustava je prikazan ispod.

artist_name	track_name
American Authors	Believer
American Authors	Go Big Or Go Home
The 1975	She's American
Neon Trees	Sleeping With A Friend
American Authors	Luck
WALK THE MOON	Aquaman
Neon Trees	Animal
The 1975	Menswear
The 1975	Heart Out
The 1975	Somebody Else
Andy Grammer	Keep Your Head Up
American Authors	Hit It
The 1975	Girls
The 1975	Sex
The 1975	This Must Be My Dream
American Authors	Think About It
The 1975	Chocolate
The 1975	Head.Cars.Bending
COIN	Hannah
WALK THE MOON	One Foot

Slika 3.4 Rezultat sustava preporuke

Uspjeli smo implementirati sustav koji na temelju određenog seta pjesama generira preporuku novih pjesama sličnih karakteristika. Jedna mana sustava preporuke je ta što ne postoji mjera točnosti koja može evaluirati rad modela. Ovdje jedino možemo sami procijeniti ispravan rad sustava. Vidimo da većina preporučenih pjesama je od benda „The 1975“ čije pjesme dijele slične karakteristike s onima iz „Mom's playlist“. Stoga možemo tvrditi da sustav radi relativno dobro, ali naravno to je moja subjektivna procjena. Ovo je samo jedan način izrade sustava ove prirode.

4. Zaključak

U ovom seminaru predstavili smo samo jednu metodu izrade sustava preporuke za glazbu. Generirali smo preporuke na temelju žanra pjesme, naslova pjesme , popularnosti glazbenika i audio značajkama pjesama. Ovakav tip preporuke se koristi u „Discover Weakly“ playlistama koje Spotify generira tjedno za svakog korisnika. Spotify koristi niz drugih metoda generiranja preporuka među kojima je i zajedničko filtriranje koje nismo obradili u ovom seminaru. Cilj ovog seminara je upoznati se s radom i glavnim osobinama sustava preporuke kroz praktičnu implementaciju. Važno je za napomenuti da postoji velika sloboda prilikom implementacije sustava preporuke. Na primjer mi smo najveću važnost dodijelili žanrovima pjesme ali to nije bilo nužno. Mogli smo generirati preporuke na temelju popularnosti pjesme ili glazbenika i slično. Implementacija ovog sustava uvelike ovisi o inženjeru te što on smatra da su najvažnije značajke proizvoda kojeg želimo preporučiti.