



Univerzitet u Novom Sadu  
Fakultet tehničkih nauka



## Predlog projekta

Aplikacija za preporuku muzike

Studenti:	Dimitrije Gašić	SV31-2021
	Maša Ivanov	SV54-2021
Predmet:	Sistemi bazirani na znanju	

## Motivacija

Postoji veliki broj servisa za slušanje muzike i kreiranje playlisti koji imaju ugrađen sistem za preporučivanje nove muzike. Ova rešenja obično ne pružaju korisniku dovoljno kontrole nad preporukama i pretragom slične muzike. Takođe često imaju naklonost ka popularnoj muzici, što nije uvek ono što korisnici žele.

Ideja aplikacije je da omogući korisnicima da kreiraju nove kolekcije muzike na osnovu svoje istorije slušanja ali i da generišu potpuno nove *Discover* playliste koje bi sadržale muziku koju korisnik nije slušao do sada.

## Pregled problema

Spotify ima funkcionalnost kreiranja sličnih playlisti, *For you* kolekcije i *Smart shuffle* koji mogu da budu suviše naklonjeni velikim hitovima ili postojećoj muzici koju korisnik sluša. Apple Music nema funkcionalnost kreiranja playlisti za specifične kategorije muzike, iako ima predefinisane playliste koje su zasnovane na muzičkom ukusu korisnika.

Takođe, neki korisnici žele preporuke muzike koja je slična specifičnoj pesmi koju su slušali, bez obzira na njihov muzički ukus. Spotify *Go to song radio* pravi plejlistu na osnovu jedne pesme, ali umesto da istakne novu i sličnu muziku, često ponavlja numere koje korisnik već redovno sluša. Postoje web aplikacije koje imaju ovu funkcionalnost, ali često nisu zadovoljavajuće ili nemaju opciju da se u preporuke uključi muzički ukus korisnika.

## Metodologija rada

### Ulazi u sistem

Prilikom registracije korisnik unosi godište i podešavanje da li dozvoljava eksplicitne sadržaje. Sistem formira početni **UserTaste** (afinitete po žanru) na osnovu korisničke kolekcije muzike. Generisanje preporuka može početi na dva načina:

- izborom profila iz predefinisane liste stilova (učenje, opuštanje, trening, raspoloženje...)
- unosom konkretne pesme ili detektovanjem trenutne (**CurrentActivity**)

U oba slučaja poštuju se korisnička podešavanja (eksplicitnost) i iz preporuka se izbacuju nedavno slušane numere.

### Izlazi iz sistema

Izlaz *forward chaining* procesa je lista preporučenih pesama. Sistem najpre formira selekciju pesama koje ispunjavaju korisnička pravila (eksplicitnost, isključenje sopstvene kolekcije i nedavno slušane).

Zatim svakoj pesmi dodeljuje bodove prema definisanim pravilima: poklapanje žanrova sa korisničkim ukusom i/ili sličnost sa odabranim profilom ili zadatom pesmom (žanrovi i audio karakteristike). Pesme sa najviše bodova čine izlaznu listu.

## Baza znanja

Sistem ima bazu podataka od 50.683 pesme sa autorima, karakteristikama i žanrovima. Pored toga, u bazi se čuvaju korisnici koji se registruju, njihova podešavanja i **UserTaste** (afiniteti po žanru). Postoji hijerarhija žanrova koju koristimo pri proceni sličnosti. Za svakog korisnika se čuva i lista nedavno slušanih pesama.

## Interakcije na osnovu podataka korisnika

- Ako korisnik označi da ne želi eksplicitne pesme, biće uzeto u obzir prilikom svake sledeće preporuke
- Ako korisnik označi da ne želi pesme koje je slušao skoro, neće biti uzete u obzir prilikom preporuke (dobijamo informacije o slušanju pomoću CEP-a)
- Ako korisnik označi da ne želi pesme koje ima u kolekciji, neće biti uzete u obzir prilikom preporuke (u bazi postoji informacija o korisničkim kolekcijama)

## Ulazni zahtevi

Postoje dva ulazna moda:

- **RequestProfile**(user, profile, topN) - korisnik bira profil
- **RequestSeed**(user, seedTrackId, topN) - korisnik zadaje konkretnu pesmu (ili koristimo trenutno puštanu - **CurrentActivity**)

Za svaki zahtev, pravila Z1 i Z2 konstruišu dve specifikacije (nastaju zajedno):

- **FilterSpec** koji ima parametre: allowExplicit, excludeOwned, excludeRecent, forbiddenTrackIds, genreDistanceMax?, yearDeltaMax?, contextGenres? i contextYear?
  - Ako je ulazni zahtev **RequestSeed**, context polja su popunjena
- **ScoreSpec** koji ima težine za bodovanje: wTaste, wProfileGenre, wProfileAudio, wSeedGenre i wSeedAudio
  - U slučaju **RequestProfile**, težine koje počinju sa Seed imaju vrednost 0 i obrnuto

## Pravila filtriranja

Pre nego što pesme uđu u bodovanje, primenjuju se pravila koja filtriraju (imaju veći *salience*):

- F1 - ako je korisnik isključio eksplicitne sadržaje, a pesma je označena kao eksplicitna, ne prolazi filter
- F2 - ako je korisnik isključio preporuke iz svoje kolekcije, a pesma mu već pripada, ne prolazi filter
- F3 - ako se pesma nalazi u listi nedavno slušane muzike, ne prolazi filter
- F4 - ako **FilterSpec** ima vrednost contextGenres (potiče iz **RequestSeed**) i pesma ima preveliku razliku žanrova, ne prolazi filter

## Bodovanje kandidata (sve komponente nezavisno doprinose)

Pesme koje prođu filtere dobijaju poene iz više izvora. Svako pravilo upisuje svoj doprinos koji se kasnije sumira:

- B1 - poeni rastu proporcionalno poklapanju žanrova kandidata i korisničkog ukusa

Ako su prisutni parametri wProfileGenre i wProfileAudio, aktiviraju se sledeća pravila:

- B2 - profil ima definisane karakteristike koje grupišu žanrove; kandidat dobija bonus/penal u skladu sa tim da li se poklapa sa navedenim žanrovima
- B3 - porede se normalizovane osobine (valenca, akustičnost, glasnoća...) sa ciljevima profila

Ako su prisutni parametri wSeedGenre i wSeedAudio, aktiviraju se sledeća pravila:

- B4 - sličnost žanrova kandidata i seed pesme donosi dodatne poene
- B5 - sličnost audio karakteristika kandidata i seed pesme

## Templates

Može da postoji veći broj specifičnih pravila za bodovanje pojedinačnih audio karakteristika. Sva mogu da se formalizuju kao formula koja proverava sličnost i na osnovu toga dodeljuje određene bodove (proporcionalno uticajnosti te karakteristike). Zbog smanjenja redundantnosti, ova pravila su implementirana putem template.

## Agregacija

- A1 - Svi doprinosi se sabiraju u ukupne bodove kandidata za tog korisnika i pesmu

Zatim sledi rangiranje i izbor najboljih N preporuka za korisnika.

## Backward chaining

Kada proveravamo da li se korisniku sviđa neki žanr, često možemo dobiti false negative rezultat iako se korisniku sviđa neki blizak žanr. Hijerarhija ima velike grupe kao: rok pop i metal a rok može da ima kategoriju alternativni rok sa specifičnim žanrovima kao grunge, britpop, indie rok i post-rok. Osnovni slučaj je da se korisniku sviđa uneti žanr. U suprotnom, proveravamo da li korisniku sviđa bilo koji podžanr ili bazni žanr unetog kao i njegov "sibling". Provera baznog žanra i podžanra se vrši rekurzivnim query. Ovaj backward chain se koristi prilikom ažuriranja korisničkih preferenci o žanrovima

Drugi *backward chaining* je implementiran kod provere da li se određeni žanr uklapa u neki profil. Profili imaju karakteristike koje mogu da grupišu više žanrova (nezavisno od njihove hijerarhije). Na primer, profil Teretana može da ima karakteristike Energija i Motivacija. Ako trenutno bodujemo pesmu sa žanrom trep, query će proveriti da li može da "dođe" do profila Teretana. Chain se koristi prilikom generisanja preporuka gde se specifikuju žanrovi koji su dobri za izabrani profil.

## Complex event processing

Korisnik prilikom slušanja muzike može da izvrši i dodatne akcije: sviđanje i preskakanje pesme. Zbog boljeg korisničkog iskustva, ovi događaji se detektuju i obrađuju. Time imamo sledeće ulazne događaje:

- **ListenEvent** - kada se slušanje neke pesme završi na bilo koji način, sadrži informaciju o procentu poslušane pesme
- **LikeEvent** - kada korisnik doda pesmu u svoju kolekciju
- **SkipEvent** - kada korisnik preskoči pesmu

Događaji koji su rezultat obrađivanja navedenih mogu da budu: SongLiked, SongDisliked, GenreAffinity, GenreLiked i GenreDisliked. Prilikom kreiranja **SongLiked** i **SongDisliked**, kreira se događaj **GenreAffinity** za svaki pojedinačni žanr te pesme. Prvi sloj pravila predstavlja agregiranje ulaznih događaja:

- S1 - 3 ili više **ListenEvent**-ova pesme koja je poslušana više od 75% u protekla 24 sata - kreira **SongLiked** i **GenreAffinity** sa intenzitetom +2
- S2 - **LikeEvent** - kreira **SongLiked** i **GenreAffinity** sa intenzitetom +2
  - Ako se desio u roku od 2 minuta **ListenEvent**-a, **GenreAffinity** je povećan na +3
- S3 - **SkipEvent** ako je pesma poslušana manje od 25% - kreira **SongDisliked** i **GenreAffinity** sa intenzitetom -2
  - Ako se desio u roku od 10 prvih sekundi slušanja intenzitet **GenreAffinity** je -3

Događaji koji označavaju preference prema žanru su ređi i zato se kreiraju agregiranjem **GenreAffinity** događaja prema sledećim pravilima:

- G1 - ako je suma intenziteta događaja istog žanra u proteklom satu pozitivna, preko 5 i nije se skoro desio **GenreLiked** event - kreiraj **GenreLiked** event
- G2 - ako je suma intenziteta događaja istog žanra u proteklom satu negativna, ispod -3 i nije se skoro desio **GenreDisliked** event - kreiraj **GenreDisliked** event

Događaji **GenreAffinity** su povezani sa bazom i koriste se za ažuriranje preferenci žanrova tog korisnika.

## Primer rezonovanja

1. Korisnik je definisao da ne želi eksplicitne sadržaje, želi da njegov muzički ukus utiče na preporuke muzike i ne želi preporuke muzike koju ima u svojoj kolekciji
2. Korisnik je pustio pesmu *Dancing in the moonlight* od King Harvest
3. Kreira se zahtev **RequestSeed** sa id pesme koja se trenutno pušta
4. Pravilo Z2 kreira:
  - a. FilterSpec sa contextGenres Pop i R&B i contextYear postavljenim na 1972
  - b. ScoreSpec pri čemu su vrednosti parametara wSeedGenre i wSeedAudio > 0
5. U sesiju se dodaju **TrackCandidate** za sve pesme iz baze podataka
6. Pravila F1 do F5 pozivaju *delete* nad kandidatima koji ne zadovoljavaju bilo kojih od njih
7. Pravilo B1 dodeljuje bodove na osnovu sličnosti žanrova kandidata i korisničkog ukusa
8. Aktiviraju se i pravila B4 i B5 zato što postoje parametri wSeedGenre i wSeedAudio
  - a. Sličnosti žanra sa *Dancing in the moonlight*
  - b. Sličnosti audio karakteristika sa *Dancing in the moonlight*
9. Pravilo A1 agregira bodove koje su generisali B1, B4, B5 i ažurira kandidate
10. U poslednjem koraku se uzima N najboljih preporuka za izlaz