

Opis końcowej wersji projektu

Nasz model ma na celu utworzenie N playlist liczącymi M piosenek każda.

W DB posiadamy 22 412 utworów posiadających 14 „technicznych” atrybutów wykorzystanych w naszym modelu. Po znormalizowaniu atrybutów nadawane są im wagi. Następnie wektory atrybutów są grupowane w 100 klastrow algorytmem k_średnich. Przyjęliśmy $k = 100$, ponieważ jest to złoty środek między zbyt grupowaniem niskiej jakości (niskie k), a silnym, gdzie piosenki są niemal identyczne (wysokie k).

```
def k_means(k, df, normalizer, wage): # k = 100
    """
    solve data with KMEANS algorithm
    """
    df = df / normalizer * wage
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(df)
    np.savetxt("./results/songs_in_groups.txt", kmeans.labels_)
    return kmeans.cluster_centers_
```

Po utworzeniu grup szukamy 30 najbardziej zbliżonych „sąsiednich” grup, za pomocą k_neighbours.

```
def k_neighbours(k, cluster_centers): # k = 30
    """
    solve data with KNEIGHBOURS algorithm
    """
    knn = NearestNeighbors(n_neighbors=k)
    knn.fit(cluster_centers)
    _, indicates = knn.kneighbors(cluster_centers)
    np.savetxt("./results/groups_similarity.txt", indicates)
```

Ostatecznie nasz model losuje N grup – tyle ile użytkownik zażyczył sobie playlist. Dla każdej grupy należy wylosować M piosenek. Znając już grupę znamy też hierarchię podobieństwa wszystkich innych grup – wykorzystujemy ją, by zwiększyć zakres różnorodności playlist.

```
def create_playlists(number_of_playlist, playlist_size, dict_groups_songs, groups_similarity): # 30
    """
    create final playlists
    """
    playlists = np.array([])
    for _ in range(number_of_playlist):
        playlist = np.array([])
        random_group = np.random.choice(100)
        for _ in range(playlist_size):
            pick_gauss_neighbour = min(int(abs(np.random.normal(0, 15))), 29)
            #group from which we will pick a song
            song_id = getSongFromAGroup(groups_similarity[random_group][pick_gauss_neighbour], dict_groups_songs)
            playlist = np.append(playlist, np.array([song_id]))
        playlists = np.append(playlists, playlist)
    np.savetxt("./results/playlists_ids.txt", playlists)
```

Za pomocą rozkładu normalnego wybieramy grupę z hierarchii, gdzie 0 to grupa bazowa playlisty, 29 to grupa najbardziej oddalona w hierarchii (mniej podobna technicznie).

Po wybraniu grupy model losuje piosenkę. Piosenki losowane są podobnie do grup, a więc z rozkładu normalnego od 0 do ilości piosenek w grupie. W każdej grupie piosenki posortowane są od 0 do długości grupy względem popularności (0 – najlepsza piosenka), ocenianej na podstawie analizy sesji użytkowników (ilość polubieni, pominięć oraz średniego, procentowego czasu odsłuchu piosenki).

```
def getSongFromAGroup(group_id, dict_groups_songs):  
    """  
    pick song from choosen group by normal distribution  
    """  
    group_size = len(dict_groups_songs[group_id]) - 1  
    gauss_song_id = int(abs(np.random.normal(0, group_size/3)))  
    song_id_in_group = min(gauss_song_id, group_size)  
    return dict_groups_songs[group_id][song_id_in_group]
```

API

Jeśli chodzi o API wystawione jest na localhost:5000 przez flaska. Użytkownik powinien załączyć odpowiednie pliki oraz wybrać docelowe parametry dla modelu. Następnie wcisnąć przycisk prześlij i odczekać do 10 min.

Artists

Wybierz plik `artists.json`

Sessions

Wybierz plik `sessions.json`

Tracks

Wybierz plik `tracks.json`

Wielkość playlisty

15

Ilość playlist

5

Prześlij

Title; Artist

Playlist 1:
Dolla Dolla Bill; Jossylvio
Pourtant; PLK
Vuelve Carinhito; Valentín Elizalde
Diz eil porquês; Benet
Puchda Hi Nahin; Neha Kakkar

Playlist 2:
Only; Jelly Roll
Parklife - 2012 Remaster; Blur
Mailler; Djadja & Dinaz
Unterweg; Apache 207
Ball and Biscuit; The White Stripes

Playlist 3:
Into You; Ariana Grande
Imperfections (Interlude); Pop Smoke
(What A Wonderful World); Sam Cooke
Have You Ever Really Loved A Woman?; Bryan Adams
Go It Alone; Beck

Playlist 4:
Te quiero igual; Andrés Calamaro
Boom Boom Pow; Black Eyed Peas
Since U Been Gone; Kelly Clarkson
Chop Suey!; System Of A Down
With You; Chris Brown

Playlist 5:
Ya Es Muy Tarde; Yuridia
Crazy Love - 2013 Remaster; Van Morrison
God Put a Smile upon Your Face; Coldplay
Sentada Aquí En Mi Alma; Chayanne
Naked - Bonus Track; Ella Mai

Playlist 6:
Dimple; BTS
O Grave Baten; MC Kevinho

Playlist 6:
Dimple; BTS
O Grave Bater; MC Kevinho
Life Goes On; JPac
Du riechst so gut; Rammstein
Balle De Los Pobres; Calle 13

Playlist 7:
Fíjate Que Si; Edwin Luna y La Trakalosa de Monterrey
Drive By; Train
I Could Use a Love Song; Maren Morris
Somebody's Problem; Morgan Wallen
Piano Man; Billy Joel

Playlist 8:
No More Parties; Coi Leray
Reggaeton; J Balvin
MilliPower; Kendrick Lamar
Hell & Back; Kid Ink
Still Waiting; Sum 41

Playlist 9:
Ases y Tercia de Reyes; Juan Gabriel
Nightswimming; R.E.M.
Feeling Good; Ruse
The One That You Love; LP
You Give Good Love; Whitney Houston

Playlist 10:
Alejandro; Lady Gaga
Put Your Records On; Ritt Momney
Una canzone che non so; Gazzelle
The One; Kodakline
Faded; Alan Walker

Playlist 11:
Better; Khalid
Dramaturgy; Eve
Tenemos Que Hablar; Bad Bunny
Freak; Doja Cat
I Don't Wanna Love You Anymore; LANY

Playlist 10:
Alejandro; Lady Gaga
Put Your Records On; Ritt Momney
Una canzone che non so; Gazzelle
The One; Kodakline
Faded; Alan Walker

Playlist 11:
Better; Khalid
Dramaturgy; Eve
Tenemos Que Hablar; Bad Bunny
Freak; Doja Cat
I Don't Wanna Love You Anymore; LANY

Playlist 12:
Te Espero Sentado; Shakin's
Honey, I'm Home; Shania Twain
A Gente Continua - Ao Vivo; Zé Neto & Cristiano
Get Busy; Sean Paul
eu me amo mais; Giulia Be

Playlist 13:
You Talk Way Too Much; The Strokes
Orion - Remastered; Metallica
Somebody To Love (Glee Cast Version) - Cover of Queen Song; Glee Cast
Hate to See Your Heart Break; Paramore
Walk All Over You; AC/DC

Playlist 14:
Half a Heart; One Direction
Te Volvería A Elegir; Calibre 50
Defying Gravity (Glee Cast Version); Glee Cast
裸の心; Ainyon
Be the One; Dua Lipa

Playlist 15:
Stronger (What Doesn't Kill You); Kelly Clarkson
Magic; One Direction
Question; System Of A Down
Reggaetón en lo Oscuro; Wisin & Yandel
Quiero Que Vuelvas; Alejandro Fernández