Music Playlist Generation And Shuffling

Umang Jain(17074016)
Anurag Singh(17074003)

Music Playlist Generation And Shuffling

- **Music Playlist Generation : -**

  Music Playlist Generation simply means the task of generating a sequence of songs suitable for a particular user based on various inputs given by the user.

- **Music Playlist Shuffling : -**

  Music Playlist Shuffling means rearranging the playlist songs in such a order which is appealing to the user.

Proposed Methodology

1. Music Playlist Generation
2. Music Playlist shuffling

Music Recommendation System

Recommendation System :

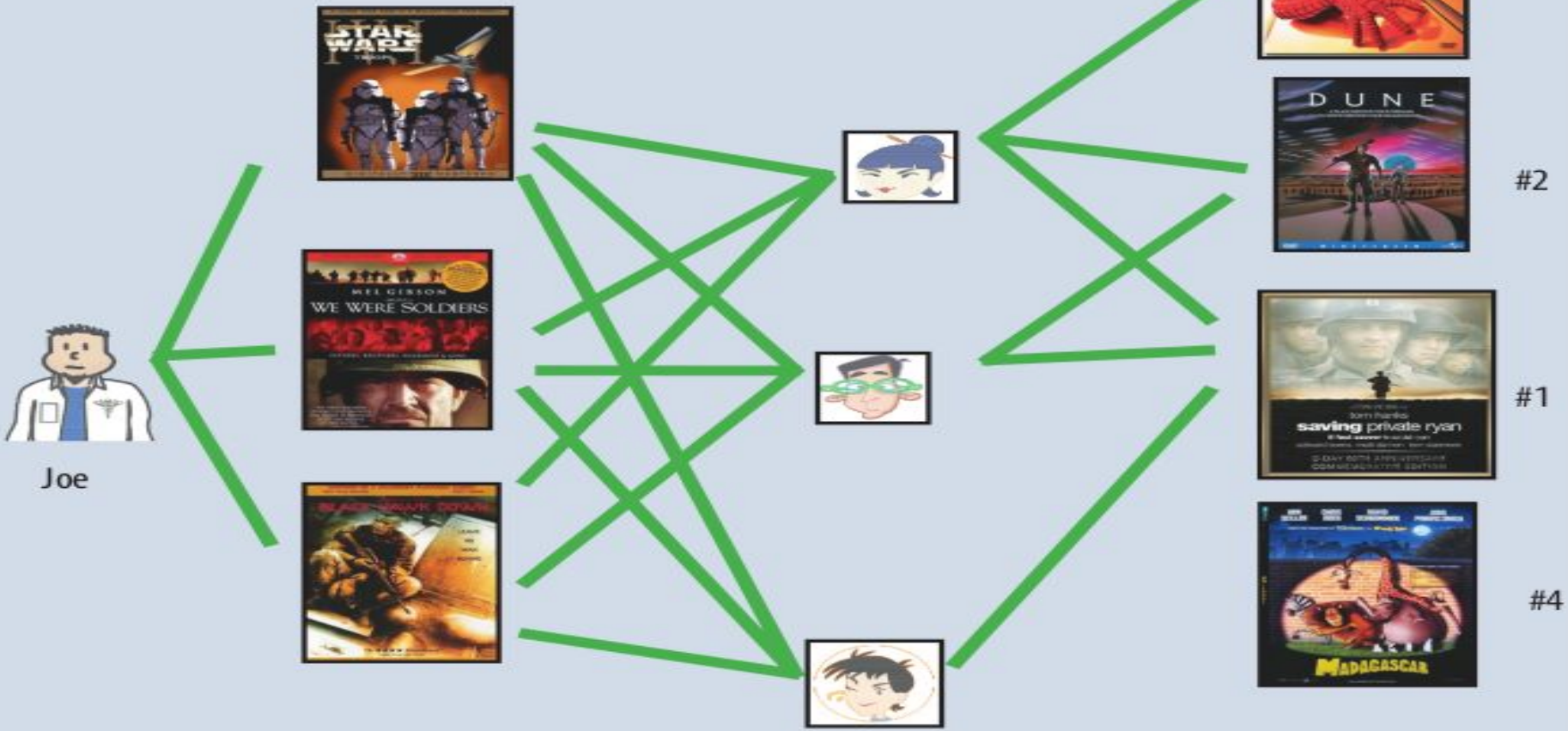
Recommender systems are techniques used for providing suggestion for a item related to various decision making process such as what items to buy or what music to listen.

These music recommendation systems are part of a broader class of recommender systems, which filter information to predict a user's preferences when it comes to a certain item.

Types Of Recommendation System

- **Content based filtering** : Content-based filtering closely examines the actual item to determine which features are most important in making recommendations and how those features interact.
  - Music Genome Project
- **Collaborative filtering** : In this approach we find the user to user closeness means based on various features we try to find other users which are most similar to user in question. **We have used collaborative approach.**
  - More accurate and domain free

Collaborative Filtering



Music Playlist Generation

For Music playlist, we found the top 10 Music recommendation for the user taking artist name whom he/she want to listen as input. Then these 10 recommendation is treated as a playlist.

K-Nearest Neighbour

we have used KNN model on last.fm data set to form recommendation system.

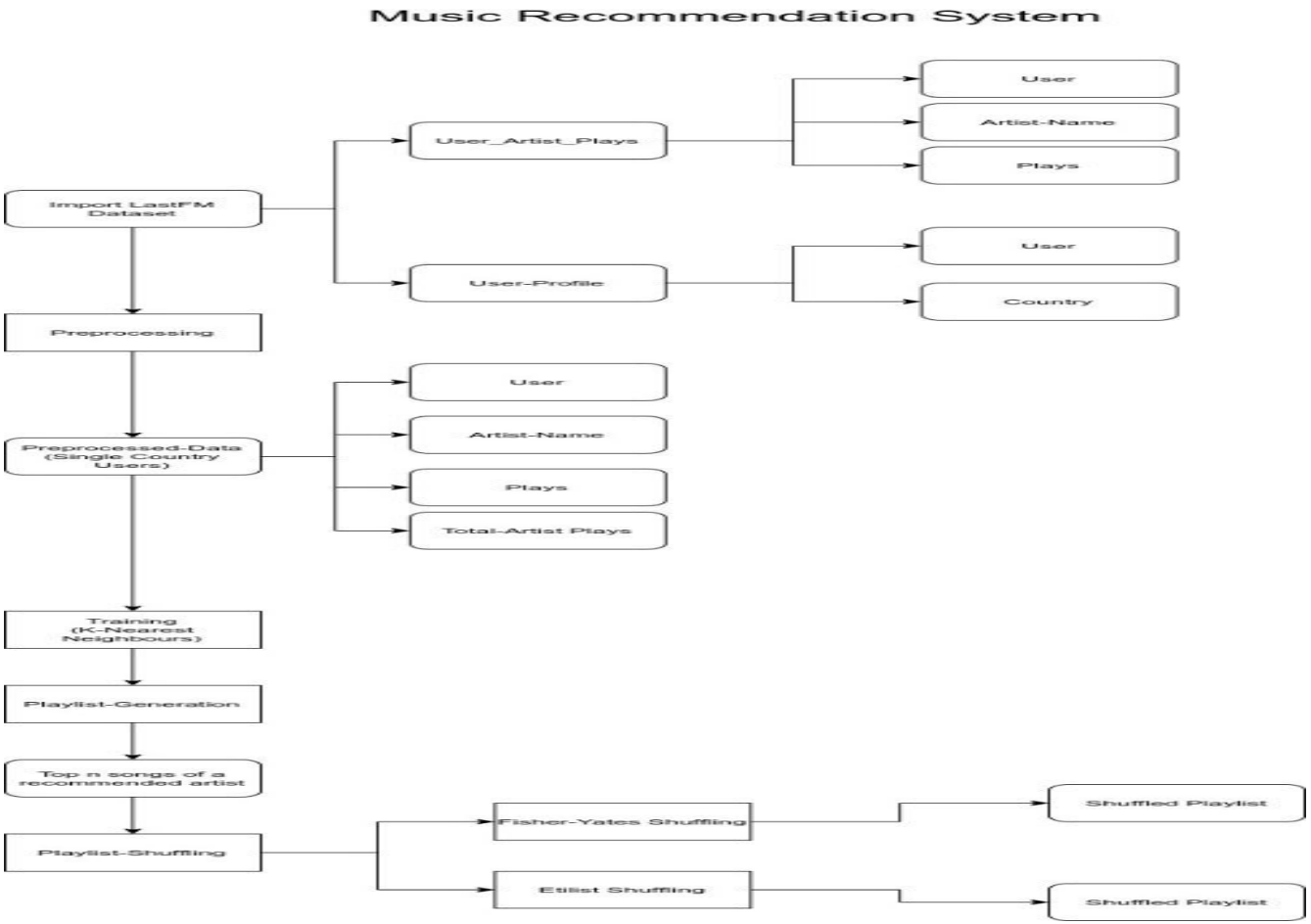
Basically, KNN algorithm assumes that similar thing exist in close proximity. We have to define a distance function to find distance between two points (could be as simple as Manhattan distance or Cartesian distance). Then using this function we find K nearest neighbour to the query point

Music Playlist Shuffling

Music Playlist Shuffling means rearranging the playlist songs in such a order which is appealing to the user. It also mean changing the order dynamically based on user inputs(like whether user skipped current track or listened to the whole song or skipped in between).

Shuffling Used

- **Fisher-Yates Shuffle :** The idea is to start from the last element, swap it with a randomly selected element from the whole array (including last). Now consider the array from 0 to n-2 (size reduced by 1), and repeat the process till we hit the first element.
- **Etilist Shuffle:** In this method of shuffling, every item has given some weight then we calculate weighted probability. Items will be chosen for shuffling based on its weighted probability.


Music Recommendation System
Import LastFM Dataset
User_Artist_Plays
User
Artist-Name
Plays
User-Profile
User
Country
Preprocessing
Preprocessed-Data (Single Country Users)
User
Artist-Name
Plays
Total-Artist Plays
Training (K-Nearest Neighbours)
Playlist-Generation
Top n songs of a recommended artist
Playlist-Shuffling
Fisher-Yates Shuffling
Shuffled Playlist
Etilist Shuffling
Shuffled Playlist

Dataset Used

Last.fm-dataset-360K Version 1.2

Dataset has contains two files :-

1. First file(usershal-profile.tsv) contains general information about users

    columns = [user , gender , age, country, signup]

2. Second file(usersha1-artmbid-artname-plays.tsv) contains information about user and artist song plays

    columns = [user, musicbrainz-artist-id, artist-name, plays]

Preprocessing

user-profile pre-processing

columns = [user, country]

user-artist-plays file pre-processing

columns = [user, artist-name, plays]

Unpopular artist are removed and after some modification final data set

columns = [user, artist-name, plays, total artist plays, country]

Final Table Formed

	users	artist-name	plays	total_artist_plays	country
134	00007a47085b9aab8af55f52ec8846ac479ac4fe	devendra banhart	456	2366807	United States
135	00007a47085b9aab8af55f52ec8846ac479ac4fe	boards of canada	407	6115545	United States
136	00007a47085b9aab8af55f52ec8846ac479ac4fe	cocorosie	386	2194862	United States
137	00007a47085b9aab8af55f52ec8846ac479ac4fe	aphex twin	213	4248296	United States
138	00007a47085b9aab8af55f52ec8846ac479ac4fe	animal collective	203	3495537	United States

INPUT FILE

Usersha1-artmbid-artname-plays.tsv:

000063d3fe1cf2ba248b9e3c3f0334845a27a6bf af8e4cc5-ef54-458d-a194-7 cannibal corpse 48

usersha1-profile.tsv

000063d3fe1cf2ba248b9e3c3f0334845a27a6bf m 19 Mexico Apr 28, 2008

Data Statistics:

- Total Lines : 17,559,530
- Unique Users : 359,347

HyperParameter Used

- popularity_threshold := 1,00,000
- User_Nationality := American
- K := 10

Results For Music Playlist Generation

MusicReccommendationSystemFinal

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

Print Top Music recommendations

```
[ ]  1 def getMusicByArtistName(recommended_artists):
     2     songList=[]
     3     for idx,artist_name in enumerate(recommended_artists):
     4         artist = lc.Artist(artist_name.title())
     5         albumList = artist.get_album_list()
     6         for song in albumList:
     7             songList.append(str(song))
     8     return songList
```

```
[ ]  1 songList = getMusicByArtistName(recommended_artists)
     2 for song in songList:
     3     print(song)
```

```
Album(title=Music From Big Pink, artist=Artist(name=The Band))
Album(title=The Band, artist=Artist(name=The Band))
Album(title=Stage Fright, artist=Artist(name=The Band))
Album(title=Cahoots, artist=Artist(name=The Band))
Album(title=Rock Of Ages: The Band In Concert, artist=Artist(name=The Band))
Album(title=Moondog Matinee, artist=Artist(name=The Band))
Album(title=Northern Lights - Southern Cross, artist=Artist(name=The Band))
Album(title=The Best Of The Band, artist=Artist(name=The Band))
Album(title=Islands, artist=Artist(name=The Band))
Album(title=Anthology, Volume I, artist=Artist(name=The Band))
Album(title=The Last Waltz, artist=Artist(name=The Band))
Album(title=Jericho, artist=Artist(name=The Band))
Album(title=High On The Hog, artist=Artist(name=The Band))
Album(title=Jubilation, artist=Artist(name=The Band))
Album(title=The Last Waltz, artist=Artist(name=The Band))
Album(title=The Grateful Dead, artist=Artist(name=Grateful Dead))
Album(title=Anthem Of The Sun, artist=Artist(name=Grateful Dead))
Album(title=Aoxomoxoa, artist=Artist(name=Grateful Dead))
Album(title=Live/Dead, artist=Artist(name=Grateful Dead))
```

Results After Shuffling

MusicReccommendationSystemFinal

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

Fisher Yates

```
[ ]  1 def Fisher_yates(songList):
     2     import random
     3     for i in range(len(songList)-1, 0, -1):
     4         j = random.randint(0, i + 1)
     5         if(j>=len(songList) or i>=len(songList)):
     6             print("NOT GOOD")
     7         songList[i], songList[j] = songList[j], songList[i]
     8     return songList
```

```
[ ]  1 songList_acc_to_fisher_yates = Fisher_yates(songList)[:10]
     2 for idx,row in enumerate(songList_acc_to_fisher_yates):
     3     print(str(idx)+" : "+row+"\n")
     4 # print(songList_acc_to_fisher_yates)
```

```
0 : Album(title=Dreams, artist=Artist(name=The Allman Brothers Band))

1 : Album(title=Muswell Hillbillies, artist=Artist(name=The Kinks))

2 : Album(title=Wipe The Windows, Check The Oil, Dollar Gas, artist=Artist(name=The Allman Brothers Band))

3 : Album(title=The Ultimate Experience, artist=Artist(name=Jimi Hendrix))

4 : Album(title=History Of The Grateful Dead, Vol. 1, artist=Artist(name=Grateful Dead))

5 : Album(title=Forty Licks, artist=Artist(name=The Rolling Stones))

6 : Album(title=Seven Turns, artist=Artist(name=The Allman Brothers Band))

7 : Album(title=The Allman Brothers Band, artist=Artist(name=The Allman Brothers Band))

8 : Album(title=Schoolboys In Disgrace, artist=Artist(name=The Kinks))
```

Results After Shuffling

MusicReccommendationSystemFinal

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Etilist Shuffling

```
[ ]  1 def Etilist_Shuffle(items,inequality):
     2     weights = np.power(np.linspace(1, 0, num=len(items), endpoint=False),inequality)
     3     weights = weights / np.linalg.norm(weights, ord=1)
     4     return np.random.choice(items, size=len(items), replace=False, p=weights)
```

```
[ ]  1 inequality=5
     2 songList = Etilist_Shuffle(songList,inequality)[:10]
     3 for idx,row in enumerate(songList):
     4     print(str(idx)+" : "+row+"\n")
```

```
0 : Album(title=Low Budget, artist=Artist(name=The Kinks))

1 : Album(title=Cahoots, artist=Artist(name=The Band))

2 : Album(title=The Rolling Stones, artist=Artist(name=The Rolling Stones))

3 : Album(title=Time Fades Away, artist=Artist(name=Neil Young))

4 : Album(title=Hitchhiker, artist=Artist(name=Neil Young))

5 : Album(title=December's Children, artist=Artist(name=The Rolling Stones))

6 : Album(title=Sleepwalker, artist=Artist(name=The Kinks))

7 : Album(title=Some Girls: Live In Texas '78, artist=Artist(name=The Rolling Stones))

8 : Album(title=Led Zeppelin, artist=Artist(name=Led Zeppelin))

9 : Album(title=Wipe The Windows, Check The Oil, Dollar Gas, artist=Artist(name=The Allman Brothers Band))
```

Stay Safe

Future Possible Work

- For the Playlist Generation, More advanced deep learning based algorithm can be used.
- For the Playlist Shuffling, Weights can be given to songs based on features like likeness to artist,genre,etc and then some weight based shuffling algorithm can be developed.
- For the Playlist Generation, More explicit features like current user mood, genre, etc can be used. we could develop algorithms that will include these features.**But for this data set is not easily available**.
- Also, Implicit features like music history, facial expressions, current time can also be taken but again data set becomes a problem.

Github Repository Containing Code

https://github.com/umangja/Music-Playlist-Generation-And-Shuffling

THANK YOU