Recommendation System for creating Custom Playlist

Siddharth Sundararajan
Rutgers University
ss3177@scarletmail.rutgers.edu

Dev Shah
Rutgers University
ds1560@scarletmail.rutgers.edu

Tejasva Tomar
Rutgers University
tt425@scarletmail.rutgers.edu

***Abstract*—** In this report, we consider the problem of building a recommendation system for a set of playlist's containing songs. We investigate several algorithms to help recommend a playlist of songs for a set of users. A user-interface is built on the recommendation system for ease of use and better user experience. A dataset containing approximately 100k songs is used to evaluate the performance of the system.

I. Project Description

Recommendation systems arise in a variety of areas such as songs, movies, games, news, books, social tags, etc. A recommendation system is defined as an application that predicts the preference or importance a user would give to an item, or identifies a set of N items that would interest the user.

Our aim is to build a user-interface on a recommendation engine, which suggests a playlist of songs to the user given a set of input playlists. The project is divided into three main phases - Data preprocessing, Feature extraction and Model building, and building an user interface.

In phase 1, a dataset containing approximately 100k songs is chosen and various methods are implemented to clean the data and select relevant columns. In the next phase, feature selection and extraction is performed. Also, based on space and time complexity, a set of algorithms are tested before finding the optimum algorithm to build a recommendation model. In phase 3, a simple user interface is build and tested.

This recommendation engine is novel as an optimum algorithm will be implemented. It is useful as it has many real world applications and a lot of future scope. This project is feasible and can be completed in 45 days (accounting all delays possible). There are three major milestones as explained in the three phases of the project. A major roadblock that could arise is in phase 2 - feature extraction. Each phase of the project will be worked on by each project member. Phase 1 by Dev Shah, phase 2 by Siddharth Sundararajan and phase 3 by Tejasva Tomar. This project type falls under the category of Massive Algorithmics (As mentioned on sakai).

The project has four stages: Gathering, Design, Infrastructure Implementation, and User Interface.

A. *Stage1 - The Requirement Gathering Stage.*

Deliverables for Stage1 as follows:

- The general system description: The aim of this project is to build a recommendation system for a user. The user inputs three playlists, each containing ten songs. The output is a recommended playlist of top ten songs that are similar to the songs present in the three playlists. The input and output are seen on a user interface that is build on top of the recommendation system.

- The three types of users (grouped by their data access/update rights): The three types of users are -

  1. Database Manager - This user type has access only to the input data. Any update that occurs to the data is handled by this user. This user type creates a clean database by preprocessing.

  2. Administrator - This user type has access to every part of the system. This user type can change the user interface, update/modify/change the algorithm and access the input database.

  3. User - This user type can only view and use the user interface. The user can only input playlist's of songs and the output is a playlist of top ten similar songs.

- Real world scenarios
  Scenario 1 -
  - Scenario1 description: A family of three who would like to listen to songs that each of them would enjoy.
  - System Data Input for Scenario1: The user would input three playlists of ten songs each.
  - Input Data Types for Scenario1: The input data type would be three arrays of size ten each.
  - System Data Output for Scenario1: The output would be a playlist of top ten similar songs.
  - Output Data Types for Scenario1: The output data type is an array of ten songs.

  Scenario 2 -
  - Scenario2 description: When more data is added to the database.

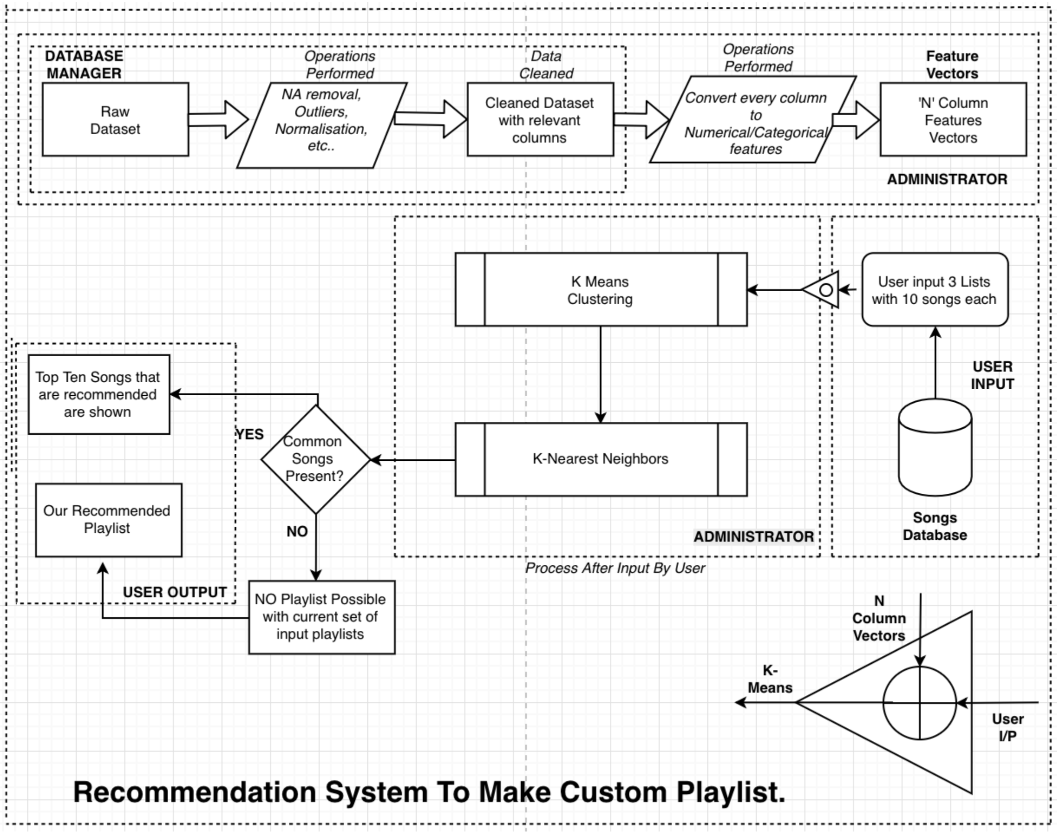

DATABASE MANAGER
Raw Dataset
Operations Performed
NA removal, Outliers, Normalisation, etc..
Data Cleaned
Cleaned Dataset with relevant columns
Operations Performed
Convert every column to Numerical/Categorical features
Feature Vectors
'N' Column Features Vectors
ADMINISTRATOR
K Means Clustering
User input 3 Lists with 10 songs each
USER INPUT
Songs Database
K-Nearest Neighbors
Common Songs Present?
YES
NO
Top Ten Songs that are recommended are shown
Our Recommended Playlist
USER OUTPUT
NO Playlist Possible with current set of input playlists
ADMINISTRATOR
Process After Input By User
N Column Vectors
K-Means
User I/P
Recommendation System To Make Custom Playlist.

Fig. 1. Flow Diagram

- System Data Input for Scenario2: The data input is a file. The database manager takes the input data and preprocesses it.
- Input Data Types for Scenario2: The input data type is in one of the formats such as .csv, .tsv, .json, etc
- System Data Output for Scenario2: The database manager creates a new database that is fed into the model after feature extraction.
- Output Data Types for Scenario2: The output data type is a .csv file format.

• Project Time line and Divison of Labor. The timeline of the project is 45 days. The time taken for the three major phases of the project is as follows -
1. Phase 1 - 14 days by Dev Shah
2. Phase 2 - 17 days by Siddharth Sundararajan
3. Phase 3 - 7 days by Tejasva Tomar
The remaining 7 days will be used for testing, documentation, evaluation, project report and power point presentation. All these will be equally distributed between the three members of this project.

B. *Stage2 - The Design Stage.*

Deliverables for Stage2 as follows:

• Short Textual Project Description : Figure 1 represents the flow diagram of the recommendation system for creating a custom playlist. The proposed solution is to use k-means clustering algorithm (or variants of this method) along with k-Nearest Neighbor(k-NN) search algorithm to create a recommended playlist of ten songs.

First, data pre-processing and cleaning is performed by the database manager, by performing operations such as NA removal, outlier treatment, normalization, etc. All relevant columns are chosen and converted to create a d dimensional feature vector. These operations take place in the backend and are kept ready.

The UI shows the user a database of songs from which he/she can pick and create three playlists, each containing ten songs. k-means clustering is run on the backend, on the d dimensional feature vector along with the input data and the output is fed into the k-NN algorithm. k-NN search algorithm finds songs based on the clusters formed. If there is no similarity in the songs present in each playlist given by the user, a common recommended songs list is provided as an output. In case of similarity, top ten recommended songs is shown to the user in the UI.

$$TimeComplexity : O(ndki)$$

$$SpaceComplexity : O((n+k)d)$$

where,
n is the number of data points in the dataset
d is the number of features
k is the number of clusters in k-means
i is the number of iterations the algorithm runs

Note : The time and space complexity calculated is mainly based on the two major algorithms implemented - k-means clustering algorithm and k-NN search algorithm.

- High Level Pseudo Code
  Algorithm 1 presents the pseudo code for this project. Algorithms 2 and 3 describe steps for the two major algorithms implemented.

Algorithm 1: Recommendation system for creating custom playlist

```
User input: 3 vectors of size 10 each containing 10
  songs
Input: Music dataset, User input
Output: Vector of size 10 containing 10 songs
Procedures:
cleanData = preProcessData(Music dataset)
featureVectors = extractFeatures(cleanData)
kClusters = kMeans(featureVectors, User input)
userOutput = kNN(kClusters, User input)
```

Algorithm 2: k-means Clustering

```
Input: Feature vector for all data points
Output: k Clusters
Determine the optimum value for $k$
Randomly generate initial values for $k$ centroids
while Not converged do
    Compute the distance of each data point to the
      centroids and assign them based on its proximity;
    Recompute centroids
end
```

Algorithm 3: k-Nearest Neighbor Search

```
Input: User input, feature vector for all data points
Output: k nearest neighbors
1. Initialize the value of $k$ based on similarity in songs
  (given by the user)
2. Compute the distances between the feature vectors
  of user input and training data
3. Sort and select the top 10 songs
```

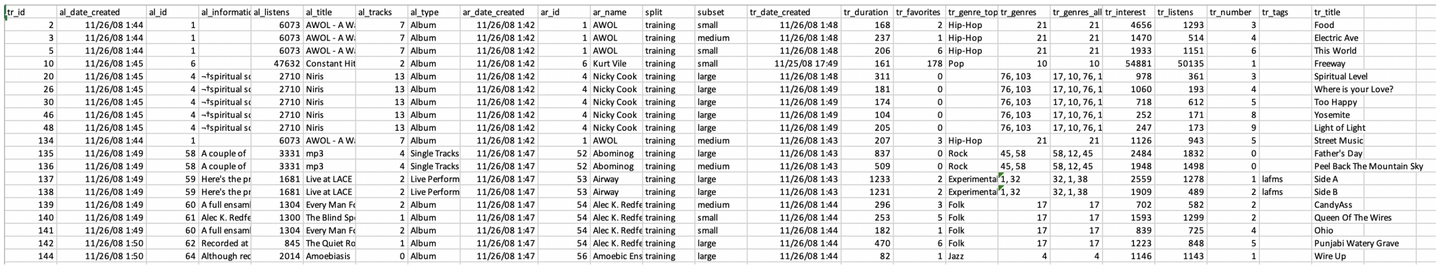
- Algorithms and Data Structures : The description for two major algorithms implemented are discussed below -
  - *k-means Clustering Algorithm* : k-means clustering is a method to partition data points into k clusters. The main aim of this algorithm is to find groups in the data based on the feature vector similarity. The distance metric used in this algorithm is Euclidean distance.
    Data structure - Feature vectors in the form of a $n \times d$ matrix
  - *k-Nearest Neighbor Search Algorithm* : kNN is a method to find the k nearest neighbors for a data point based on its feature vector in a d dimensional space. Euclidean distance is used in this algorithm to compute the distance between the input data point and the training data points.
    Data structure - Feature vectors in the form of a $n \times d$ matrix

  Note : Other distance metrics like Manhattan distance, correlation distance etc., will also be used for testing in this project.
- Flow Diagram Major Constraints:
  - Integrity Constraint: Any new input given by the user should contain the necessary features. In case all the features used to build k-clusters are not present the recommendation might not be accurate.

C. Stage3 - The Implementation Stage.

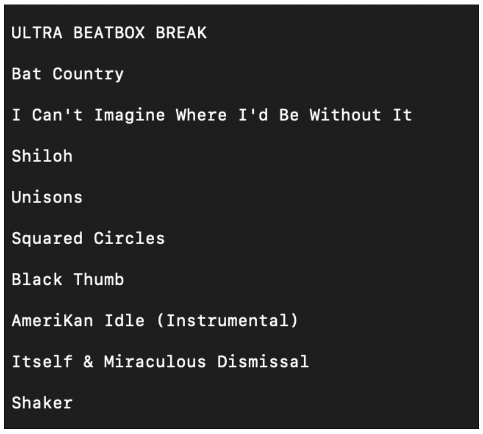
We build our recommendation system using python as the programming language and is build under the windows environment.

- Sample small data snippet: Figure 2 is the screen shot of the reduced dataset post processing. We have retrieved the columns which are important for our system, few examples being track name, artist name, track favorites, track genre, track listens, etc.
- Sample small output: Our system recommends a list of ten songs which is the final output of our recommendation system and figure 3 shows one such output.
- The complete code is in the code bundle
- Demo and sample findings:
  - Our present data-set size is 101 MB and the current execution time of the application is 3.09 minutes in total, which includes performing of elbow point test to find optimal value of k for k-means clustering, running the k-means clustering, running and training of the kNN model. Obtaining the results does not occupy a significant amount of time in the execution. The system takes 3.09 minutes every time because the model is trained each time it is run. We can minimize the time to obtain results by passing the input values on an already trained model, then our running time will be in milliseconds.
  - From the data-set, we have used only few columns for k means clustering, we plan to use more features to see if the accuracy of the system increases by a significant amount or not. From the elbow point test, we see that optimized value of k is 10 right now with the present data-set which might change when more feature columns are taken into consideration.



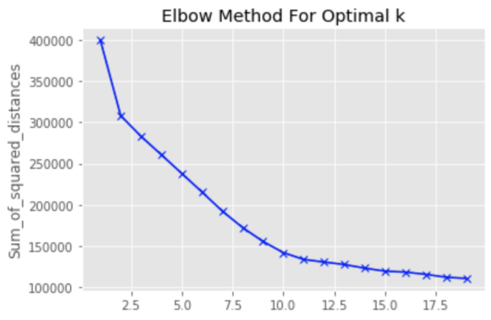
tr_id	al_date_created	al_id	al_informatic	al_listens	al_title	al_tracks	al_type	ar_date_created	ar_id	ar_name	split	subset	tr_date_created	tr_duration	tr_favorites	tr_genre_top	tr_genres	tr_genres_all	tr_interest	tr_listens	tr_number	tr_tags	tr_title
2	11/26/08 1:44	1		6073	AWOL - A Wa	7	Album	11/26/08 1:42	1	AWOL	training	small	11/26/08 1:48	168	2	Hip-Hop	21	21	4656	1293	3		Food
3	11/26/08 1:44	1		6073	AWOL - A Wa	7	Album	11/26/08 1:42	1	AWOL	training	medium	11/26/08 1:48	237	1	Hip-Hop	21	21	1470	514	4		Electric Ave
5	11/26/08 1:44	1		6073	AWOL - A Wa	7	Album	11/26/08 1:42	1	AWOL	training	small	11/26/08 1:48	206	6	Hip-Hop	21	21	1933	1151	6		This World
10	11/26/08 1:45	6		47632	Constant Hit	2	Album	11/26/08 1:42	6	Kurt Vile	training	small	11/25/08 17:49	161	178	Pop	10	10	54881	50135	1		Freeway
20	11/26/08 1:45	4	¬†spiritual sc	2710	Niris	13	Album	11/26/08 1:42	4	Nicky Cook	training	large	11/26/08 1:48	311	0		76, 103	17, 10, 76, 1	978	361	3		Spiritual Level
26	11/26/08 1:45	4	¬†spiritual sc	2710	Niris	13	Album	11/26/08 1:42	4	Nicky Cook	training	large	11/26/08 1:49	181	0		76, 103	17, 10, 76, 1	1060	193	4		Where is your Love?
30	11/26/08 1:45	4	¬†spiritual sc	2710	Niris	13	Album	11/26/08 1:42	4	Nicky Cook	training	large	11/26/08 1:49	174	0		76, 103	17, 10, 76, 1	718	612	5		Too Happy
46	11/26/08 1:45	4	¬†spiritual sc	2710	Niris	13	Album	11/26/08 1:42	4	Nicky Cook	training	large	11/26/08 1:49	104	0		76, 103	17, 10, 76, 1	252	171	8		Yosemite
48	11/26/08 1:45	4	¬†spiritual sc	2710	Niris	13	Album	11/26/08 1:42	4	Nicky Cook	training	large	11/26/08 1:49	205	0		76, 103	17, 10, 76, 1	247	173	9		Light of Light
134	11/26/08 1:44	1		6073	AWOL - A Wa	7	Album	11/26/08 1:42	1	AWOL	training	medium	11/26/08 1:43	207	3	Hip-Hop	21	21	1126	943	5		Street Music
135	11/26/08 1:49	58	A couple of	3331	mp3	4	Single Tracks	11/26/08 1:47	52	Abominog	training	large	11/26/08 1:43	837	0	Rock	45, 58	58, 12, 45	2484	1832	0		Father's Day
136	11/26/08 1:49	58	A couple of	3331	mp3	4	Single Tracks	11/26/08 1:47	52	Abominog	training	medium	11/26/08 1:43	509	0	Rock	45, 58	58, 12, 45	1948	1498	0		Peel Back The Mountain Sky
137	11/26/08 1:49	59	Here's the pr	1681	Live at LACE	2	Live Perform	11/26/08 1:47	53	Airway	training	large	11/26/08 1:43	1233	2	Experimental	1, 32	32, 1, 38	2559	1278	1	lafms	Side A
138	11/26/08 1:49	59	Here's the pr	1681	Live at LACE	2	Live Perform	11/26/08 1:47	53	Airway	training	large	11/26/08 1:43	1231	2	Experimental	1, 32	32, 1, 38	1909	489	2	lafms	Side B
139	11/26/08 1:49	60	A full ensaml	1304	Every Man Fc	2	Album	11/26/08 1:47	54	Alec K. Redfe	training	medium	11/26/08 1:44	296	3	Folk	17	17	702	582	2		CandyAss
140	11/26/08 1:49	61	Alec K. Redfe	1300	The Blind Spc	1	Album	11/26/08 1:47	54	Alec K. Redfe	training	small	11/26/08 1:44	253	5	Folk	17	17	1593	1299	2		Queen Of The Wires
141	11/26/08 1:49	60	A full ensaml	1304	Every Man Fc	2	Album	11/26/08 1:47	54	Alec K. Redfe	training	small	11/26/08 1:44	182	1	Folk	17	17	839	725	4		Ohio
142	11/26/08 1:50	62	Recorded at	845	The Quiet Ro	1	Album	11/26/08 1:47	54	Alec K. Redfe	training	large	11/26/08 1:44	470	6	Folk	17	17	1223	848	5		Punjabi Watery Grave
144	11/26/08 1:50	64	Although rec	2014	Amoebiasis	0	Album	11/26/08 1:47	56	Amoebic Ens	training	large	11/26/08 1:44	82	1	Jazz	4	4	1146	1143	1		Wire Up

Fig. 2. Snippet of the DataSet



```
ULTRA BEATBOX BREAK

Bat Country

I Can't Imagine Where I'd Be Without It

Shiloh

Unisons

Squared Circles

Black Thumb

AmeriKan Idle (Instrumental)

Itself & Miraculous Dismissal

Shaker
```

Fig. 3. Sample Output



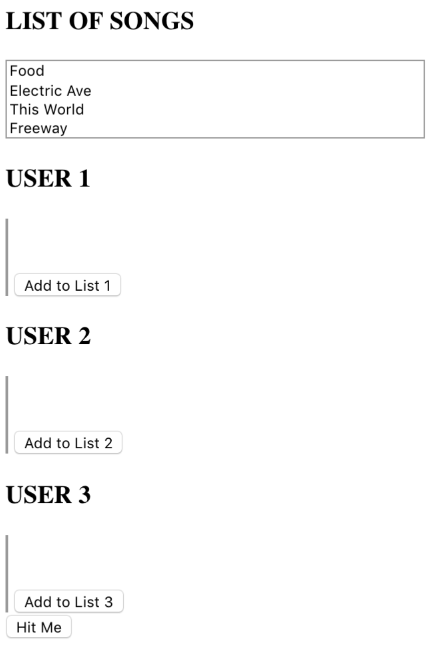
test.png

Fig. 4. Elbow Method for Optimal K

- This application will be useful for a group of people who listen to songs together, when in a group we cannot accommodate every one's choice of music. Our application is made to that specific thing i.e. it will recommend them a list of songs which will be of common interest of everyone.
- The novelty in such application is that, right now every other application in the market recommends you songs on the basis of the songs one user listens to. Our application widens the scope, it recommends a playlist based on number of users and their respective playlist.

D. Stage4 - User Interface.

The user interface serves the sole purpose to take input of 10 songs per user and generate a custom playlist based on the input from three users. The main page of the application is the Custom Playlist generator where users are directed to another page based on the events triggered. After the system receives the user event, it will direct the user to a new page with the output. The user interface is simple and user friendly, and the output results are clear and concise.



LIST OF SONGS

Food
Electric Ave
This World
Freeway

USER 1

USER 2

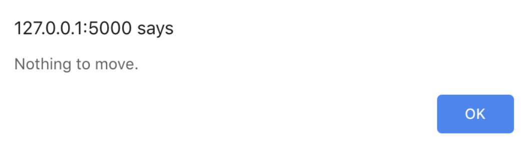
USER 3

Fig. 5. Snippet of the MAIN PAGE

- The application activates when we go to the server page. The system after initializing, shows the user a list of all songs, three empty lists to fill their song choices and few buttons. Users can add the songs from the list of the songs to their respective list using the respective buttons. Once the three play-lists are filled the users can use the button to trigger the function and generate a custom play-list.
- The initial page can be seen in Figure 5, the buttons "ADD TO LIST" can be used to add the songs from the list to the user list 1,2,3. After the lists are filled with the

songs, they can click the "HIT ME" button which will trigger a new page with the output of the list of songs.

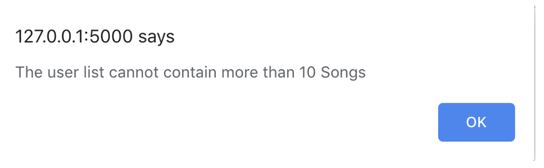
- The error messages will come up when unable to meet the requirements of the system (along with explanation and examples):
  - The error message type 1:
  - The user clicks on the ADD TO LIST button without selecting any song from the list of the songs and the error is popped up as shown.


127.0.0.1:5000 says

Nothing to move.

Fig. 6. Snippet of the ERROR 1

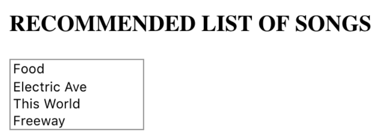
  - The error message pops up because the user is trying to add a song to the user list without selecting any song from the list of songs. The system will ask to select a song that is to be moved. This is seen in Figure 6.
  - The error message type 2:
  - If the user tries to select more than 10 songs then this error shows up, system will warn the user and suggest them to not add more songs.


127.0.0.1:5000 says

The user list cannot contain more than 10 Songs

Fig. 7. Snippet of the ERROR 2

  - The error message pops up because the maximum number of songs per list is kept to 10 as the system takes total of 30 songs in total i.e. 10 songs in one list. This is seen in Figure 7.
- The results that pop-up in the response to user interface events is just a single list which is displayed on a different page. This is our recommended list of songs. This is seen in Figure 8.


RECOMMENDED LIST OF SONGS

Food
Electric Ave
This World
Freeway

Fig. 8. Snippet of RESULT PAGE

- The interface mechanism that activate different views - The "hit me" button is the only event trigger which calls the function and the recommended playlist is displayed on the other view.