# Music Recommendation System

Shail Patel
AU1940142
shail.p2@ahduni.edu.in
B.Tech CSE

Sahil Miskeen
AU1940267
sahil.m@ahduni.edu.in
B.Tech CSE

Varun Deliwala
AU1940034
varun.d@ahduni.edu.in
B.Tech CSE

***Abstract*—Using past data of the listening preferences of the user in order to predict his future recommendations of songs. Use the different features related to songs (liveness, acoustic, dynamic etc) to predict what taste the user has in music is what drives all the music streaming companies. To get these predictions right is a tough task as the human mind keeps changing preferences continuously. We aim to use machine learning techniques in order to provide the best possible recommendations of song personalised according to the user.**

***Key Words*—K-nearest neighbour, Machine Learning, Regression, Dependency, Recommendation, Spotify, API**

## I. Introduction

The number of hours an average person spends on listening to music, be it while working or studying or during one's leisure time, is substantially increasing day by day. With the increase of user base on applications like spotify, it often becomes difficult to discover new music that matches one's interests. So, the main aim of the project here is to identify the best solutions for music recommendation.

## II. Literature Survey

The approaches taken till now for music recommendation systems are based on deep neural networks. These deep neural networks work on the basis of content based filtering taking into account different variables like the past listening history of the user, listening time of various genres etc. (1)
Logistic regression has been applied to build a recommendation system. This takes into account the listening history of the user and predicts whether or not the user will like another song or not. However this is a very strict recommendation system and the results are quite poor. (2)
KNN is another ML algorithm which allows the recommendation system to make a comparison between K different songs and predict the next closest neighbor which corresponds to the song and gives us that song as a recommended song. (3)

### A. Deep Learning Based Recommendation System

The approaches taken till now for music recommendation systems are based on deep neural networks. These deep neural networks work on the basis of content based filtering taking into account different variables like the past listening history of the user, listening time of various genres etc (1).

### B. Logistic Regression

Logistic regression is commonly used in recommendation system where the model predicts a user will watch a movie or in our case, like a song suggested based on the movies he/she has watched previously. So for a given input pair (U, S) of user U and song S, we want to predict whether the user will (0) or will not (1) like the song. As a logistic regression model can also predict the probability of the interaction in addition to a binary label, it is popularly used to the predict probilities to sort the object in terms of users and recommend some fixed number of top-ranked Objects, the objects ranging from movies, songs to food dishes too.

### C. KNN Algorithm

k-Nearest neighbors is a machine learning algorithm which is commonly used in recommendation systems so as to find clusters of similar objects, based on the common likings or experiences. The model makes predictions using the average distance (in our case the previous listens) of the top-k nearest neighbors. There can be a content based approach where the discrete characteristics of an item are utilized in order to recommend addition items with similar properties and there can be Collaborative filtering approach where the model becomes more accurate with increase in information about the user's past behavior. Majorly, models currently used in recommendation systems are based on a hybrid approach so as to improve accuracy.

## III. Implementation

The implementation process for the KNN bases music recommendation system is quite extensive and broad. The process starts from collection of the required data. For this purpose we have used the data provided to us by spotify. The dataset contains 170653 songs belonging to different genres. Another dataset provided to us is the genre dataset. The genre dataset contains 2953 different genres of music according to which each and every one of the song would be classified into. Following this the dataset contained 19 different columns with each column providing us a feature related to the song.The feature included ranged from danceability to acouscticness of the song. In order to understand the features we needed to have a good domain knowledge. We tried to develop a new feature by performing feature engineering on the variables. Moving on to the next step was the visualization of the available data. We used different visualization techniques in order to do so. We performed a correlation matrix, plotted the distribution of the variables to see the range, type of distribution and the spread of the variables which were available to us.

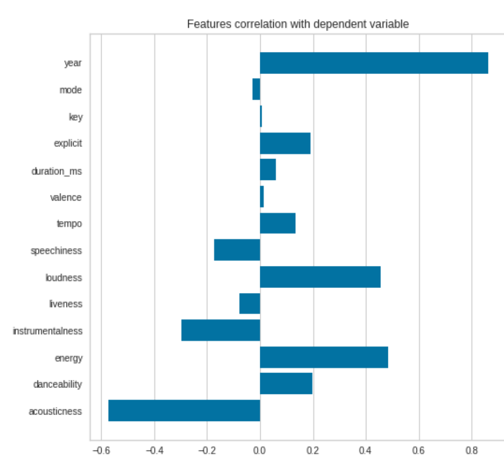


Fig-1: Correlation between variables and "popularity" variable

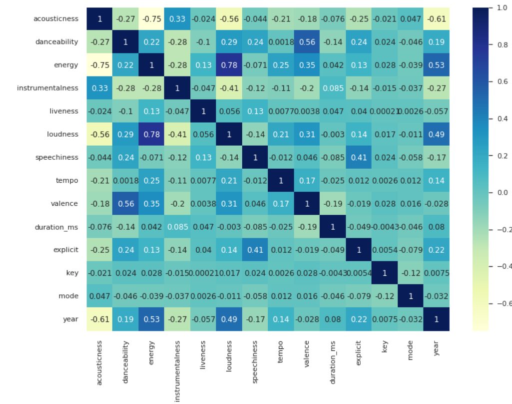


Fig-2: Correlation Graph of variables

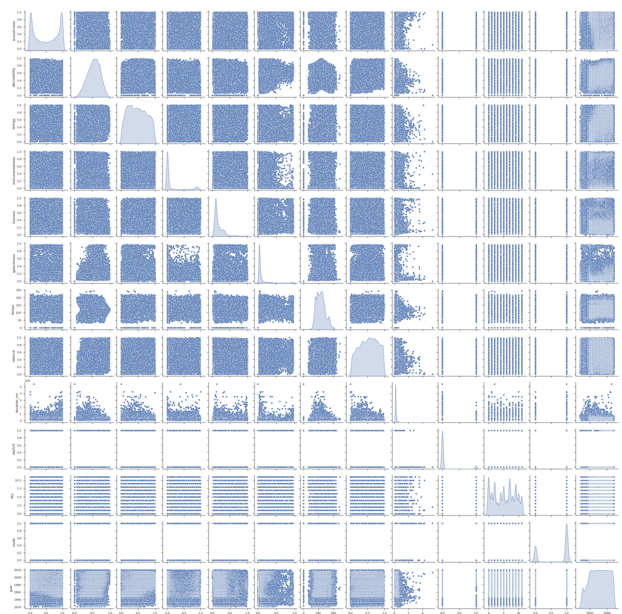

Fig-3: Correlation Graph of variables

After performing the exploratory data analysis we got an idea which variables needed to be scaled in order to bring every variable to same the scale for better clustering afterwards. We performed standard scaler on the the required variables.

| | valence | acousticness | danceability | duration_ms | energy |
|---|---|---|---|---|---|
| count | 170653.000000 | 170653.000000 | 170653.000000 | 1.706530e+05 | 170653.000000 |
| mean | 0.528587 | 0.502115 | 0.537396 | 2.309483e+05 | 0.482389 |
| std | 0.263171 | 0.376032 | 0.176138 | 1.261184e+05 | 0.267646 |
| min | 0.000000 | 0.000000 | 0.000000 | 5.108000e+03 | 0.000000 |
| 25% | 0.317000 | 0.102000 | 0.415000 | 1.698270e+05 | 0.255000 |
| 50% | 0.540000 | 0.516000 | 0.548000 | 2.074670e+05 | 0.471000 |
| 75% | 0.747000 | 0.893000 | 0.668000 | 2.624000e+05 | 0.703000 |
| max | 1.000000 | 0.996000 | 0.988000 | 5.403500e+06 | 1.000000 |

Fig-4: Spread of some variables

As clearly visible the spread of some variables is more than compared to others and for that reason we need to perform standard scaler on the variables. Further after this step we needed to reduce the dimensions of the genre and the song dataset. Since KNN works on the number of clusters it is necessary that the dimensions to work with are less so that clusters can be accurately formed. In order to reduce the dimensions we worked out with 2 different techniques. 1. PCA 2. TSNE. We tried both the techniques for both the dataset and selected the best results we got by doing the reduction techniques on the dataset.

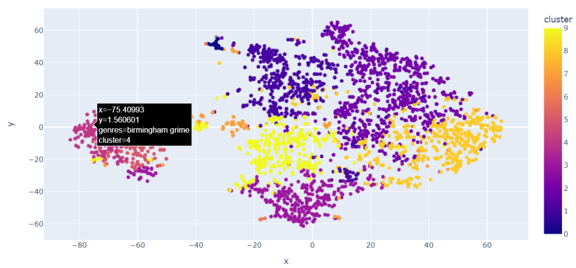

Fig-5: Cluster of genre using TSNE

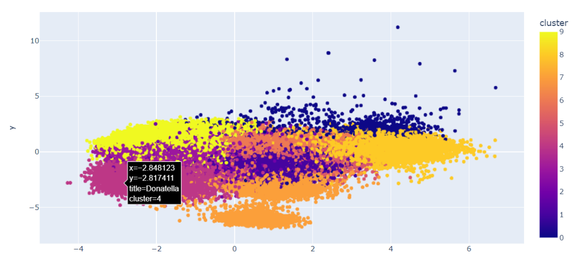

Fig-6: Cluster of songs using PCA

Based on the clusters formed by PCA and TSNE we move on to building KNN model for recommending songs. Number of nearest neighbors taken into consideration are 10.
Following the creation of model the next task was to create a api service that would lead to connecting the songs for which we want the recommendations to the recommendations received from the KNN model. We used a spotify library provided by python in order to connect it with the spotify application.
The final model was developed in such a way that the user had an option to enter the songs for which they want the recommendation with or without the year. However the results received were quite different for both of the inputs provided. This was an anomaly that we detected while implementing the KNN model for music recommendation system. We have designed a user friendly system that would allow the user to enter the songs and receive an output for the same. The user has an option to get the data from his spotify account or to enter the songs on his own. The recommended songs will then be displayed to him in a manner easily recognized.

## IV. Results

The results achieved are a part of the UI perspective as seen by the user. The model runs in the backend of the server which is not visible to the user. The user has an option of entering the songs or getting data from spotify. The user can enter the songs along with the year of the song or just the song. The output for both the processes differ due to a change in the constraint and the decision making process which relies on the year.

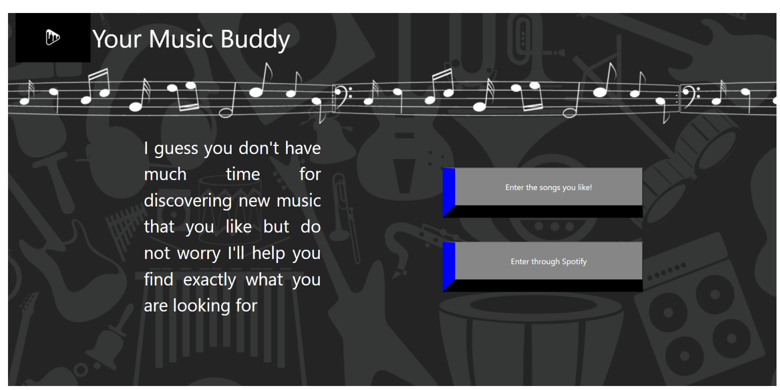


Figure-7 : Home Page

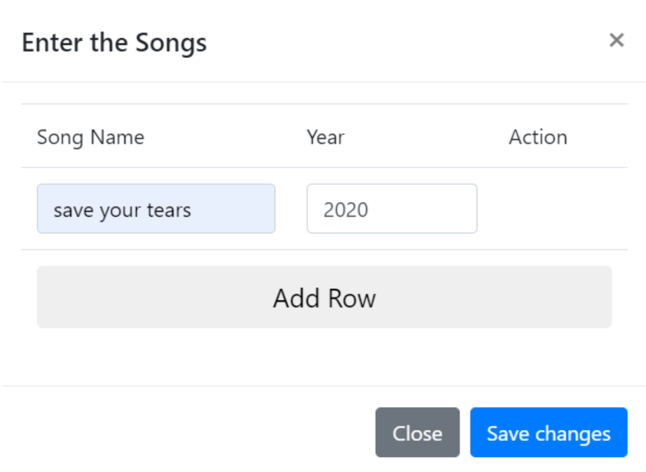


Figure-8 : User enters the song he likes along with the year of the song


| Song Name | Artist | Year |
|---|---|---|
| Open Wide (feat. Big Sean) | Calvin Harris, Big Sean | 2014 |
| Roly Poly | Chuck Berry | NA |
| Save Your Tears | The Weeknd | 2020 |
| See See Rider | The Animals | NA |
| Space Bound | Eminem | 2010 |
| Tell Me U Luv Me (with Trippie Redd) | Juice WRLD, Trippie Redd | 2020 |
| The Return of Jackie and Judy - 2002 Remaster | Ramones | NA |
| Time of Our Lives | Pitbull, Ne-Yo | 2014 |
| Venom - Music From The Motion Picture | Eminem | 2018 |
| You're Gonna Miss Me | 13th Floor Elevators | NA |

Figure-9 : User receives the recommendation along with the years of the song.

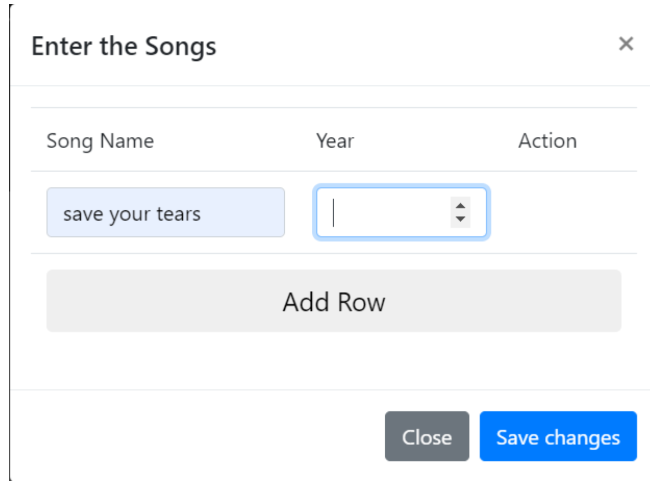


Figure-10 : User enters the song he likes along without the year of the song


| Song Name | Artist | Year |
|---|---|---|
| Brand New Numb | Motionless In White | NA |
| Burbujas de Cristal | Cartel De Santa, Millonario | NA |
| Numb / Encore | JAY-Z, Linkin Park | NA |
| Numb / Encore | JAY-Z, Linkin Park | NA |
| Open Wide (feat. Big Sean) | Calvin Harris, Big Sean | NA |
| Save Your Tears | The Weeknd | NA |
| Space Bound | Eminem | NA |
| Tell Me U Luv Me (with Trippie Redd) | Juice WRLD, Trippie Redd | NA |
| Time of Our Lives | Pitbull, Ne-Yo | NA |
| Venom - Music From The Motion Picture | Eminem | NA |



Figure-11 : User receives the recommendation without the years of the song.

## V. Conclusion

While solving the problem of song recommendation the major issue that we faced was recommendation vs prediction. While predictive algorithms worked on the basis that once a user listens to a song from a particular artist it becomes a high possibility that the user might listen to another song from the same artist. However appropriate it may seem it be be delusional sometimes. Artists often incorporate various elements in their different songs and this leads to different genres of music. This would be undesirable for a recommendation system as it should suggest songs based on the genre and not on basis of the artist.
Year played a major role (than expected) in providing recommendations. This was observed when the songs we wanted recommendations for were antiquated; the recommended songs were also of the same time. Thus we could infer that even though the genre of the song may be the same; the features attributed to the genre changes over time. As when the songs recommended were analyzed there seemed to be a slight change in the genre of songs that were recommended with year / without year.


Fig-7: Trend of variables with year

## References

[1] Liao, K., 2018. Prototyping a Recommender System Step by Step Part 1: KNN Item-Based Collaborative Filtering. [online] Medium. Available at: ¡https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea¿ [Accessed 20 March 2022].

[2] M. Schedl, "Deep Learning in Music Recommendation Systems," Frontiers, 01-Jan-1AD. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fams.2019.00044/full. [Accessed: 28-Mar-2022].

[3] "A music recommendation system based on logistic regression ..." [Online]. Available: https://www.researchgate.net/publication/336162555 A Music Recommendation System Based on logistic regression and eXtreme Gradient Boosting. [Accessed: 28-Mar-2022].

[4] https://silo.tips/download/project-report-cs-240a-applied-parallel-computing-k-nearest-neighborhood-based-m

[5] https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify

[6] https://deepnote.com/@zi-xiao-li/Spotify-fd3c414d-44ef-42d2-b9f4-51d7faf159c7

[7] https://www.kaggle.com/code/richardcsuwandi/spotify-time-series-analysis/notebook