Audio content-based Music Recommendation System

Akanksha Pandey
MT20048
IIIT Delhi
akanksha20048
@iiitd.ac.in

Atul
2017032
IIIT Delhi
atul17032
@iiitd.ac.in

Prashant Jain
2018253
IIIT Delhi
prashant18253
@iiitd.ac.in

Shivani Mishra
MT20062
IIIT Delhi
shivani20062
@iiitd.ac.in

ABSTRACT

As online music streaming becomes the dominant medium for people to listen to their favorite songs, music streaming services are now able to collect large amounts of data from their users. These streaming services, like Spotify, Apple Music or Pandora, are using this data to provide recommendations to their listeners.
This project tries to recommend songs depending on the features extracted from the music data.

1. PROJECT INTRODUCTION

"Music is a moral law. It gives soul to the universe, wings to the mind, flight to the imagination, and charm and gaiety to life and to everything" - Plato. The words of Plato rightly describe the importance of music in the world. As the field of music is evolving, large number of songs are being published every day.

A recommender (or recommendation) system (or engine) is a filtering system which aims to predict a rating or preference a user would give to an item, i.e. a song. Recommendations done using content-based recommenders can be seen as a user-specific classification problem. This classifier learns the user's likes and dislikes from the features of the song.

2. LITERATURE SURVEY

Recommendation System of Music can be attempted in a variety of ways. A typical approach involves processing a dataset of audio files, extracting features from them, and then using a dataset of these extracted features to train a machine learning classifier.

In [1] J. Fang et al. propose a system that incorporates user profiling to provide a strong set of initial recommendations to the user. Reinforcement learning is then used as each recommendation is accepted or rejected in order to ensure that subsequent recommendations are also likely to be approved. Test subjects who used the proposed system rated the playlists it provided more highly than those provided by a prior state-of-the-art reinforcement learning-based music recommendation system and also did not need to reject as many songs before being satisfied with their recommendations, both when receiving recommendations based on individual profiles, and when receiving recommendations based on aggregate profiles formed by grouping the users.

In [2] B. McFee et al. propose a method for optimizing content-based similarity by learning from a sample of collaborative filter data. The optimized content-based similarity metric can then be applied to answer queries on novel and unpopular items, while still maintaining high recommendation accuracy. The proposed system yields accurate and efficient representations of audio content, and experimental results show significant improvements in accuracy over competing content-based recommendation techniques.

In [3] D.Kim et al. suggests a method for personalized services. They extract the properties of music from music's sound wave. They use STFT (Shortest Time Fourier Form) to analyze music's property. And they infer user's preferences from user's music list. To analyze users' preferences they propose a dynamic K-means clustering algorithm. The dynamic K-means clustering algorithm clusters the pieces in the music list dynamically adapting the number of clusters. They recommend pieces of music based on the clusters.
By using our K-means clustering algorithm, they can recommend pieces of music which are close to user's preference even though he likes several genres. They perform experiments with one hundred pieces of music. In this paper, they present and evaluate algorithms to recommend music.

M. Soleymani et al. [4] proposed a music recommender-system based on psychological study [5] done by P.J Rentflow et al. to describe music preferences. The study provided five set of attributes namely Melow, Unpretentious, Sophisticated, Intense and Contemporary (MUSIC) to narrate the preference of music.
For dataset, they used 249 audio files collected from 5 substudies.The dataset contained nine points user rating system, some metadata like artist, genre and title, and scores on five factors of the music preference model (MUSIC).
Timbral features which shows the perceived quality of sound or musical tone and auditory temporal features which helps in revealing the small and sudden stimuli in

audio were extracted for genre recognition.
while training the model, PCA was used for dimensionality reduction and MLR, SVR, and RSS were used. For performance measure they used RMSE and the coefficient determination of r^2.
The authors concluded that the best performance for attribute detection was found when we combine the audio modulation feature with sparse representation. They also showed that the model did not have any cold start problem.

In [6], S.D.Teh Chao Ying et al. presented an approach for lyrics based genre classification which useed mood information. The authors selected 10 genres(pop,blue,country, etc) and 10 mood categories(happy, sad, angry etc). For dataset, they collected 1000 English songs.The study assumed that genre and mood are complementary with each other.
For pre-processing the data, they manually cleaned the lyrics removing phrases like "back to intro" and replacing them with words itself to get complete text. Authors used variants of tf-idf including wf-idf. For training purpose they used kNN, Naive Bayes and SVM with ten-fold cross-validation which was further averaged over five repeated runs.
They noted that with Lwf-idf, Pop hadAuthors noted highest accuracy of 76.94 for pop using Lwf-idf and with Nlwf-idf, 75.71. They concluded that Lwf-idf and NLwf-idf weighting equations indicated thar additional weights in this domain is a promising approch and it can greatly help in classifying Genre using a lyrics based system.

[7] proposes a Shazam system audio identification system which computes a spectrogram from an audio using STFT (short term fourier transform). Peak picking strategy is used to extra all the local maxima in the magnitude spectrogram (these time frequency points represent closest search to the audio). The graph is further reduced to a constellation map, a low-dimensional sparse representation of the original signal by means of a small set of time-frequency points.The peaks are highly characteristic, reproducible, and robust against many, even significant distortions of the signal which facilitates its high identification rate, while scaling to large databases.

In [8] Tags form a basis for many music recommendation systems, which uses information about moods, musical key, etc to recommend audio content. However, such tags tend to be less accurate, subjective, and rather noisy. Crowd (or social) tagging, one popular strategy in this context, employs voting and filtering strategies based on large social networks of users for "cleaning"the tags. However, this approach is that it relies on a large crowd of users for creating reliable annotations. While mainstream pop/rock music is typically covered by such annotations, less popular genres are often scarcely tagged. This phenomenon is also known as the "long-tail" problem. The accuracy of these system is very less 30-40% and is entirely dependent on the crowd's choice. To overcome these problems, content-based retrieval strategies are preferred as they do not rely on any manually created metadata but are exclusively based on the audio content and cover the entire audio material in an objective and reproducible way.

3. METHODOLOGY

3.1 Dataset and Data Analysis

We are using FMA (free music archive) dataset for music recommendation which is freely available. This dataset contains songs of different different genres.
For this project, we have used 5 different genres Pop, Hip-Hop, RnB, folk and Rock. Each song is of at least 1 minute long. For each of these, we then split each song into 20 second chunks using pydub. The motivation behind this is to gather as much information possible from each song. We want to analyse the effect of segmenting the song into smaller chunks, and thereby enchancing the local regions of the song, on the recommendation performance. Also, the use of smaller song excerpts makes the dataset much easier to acquire than if more number of full tracks had to be downloaded.For analyzing the data we plotted waveforms of a songs and spectrogram to visualize it.

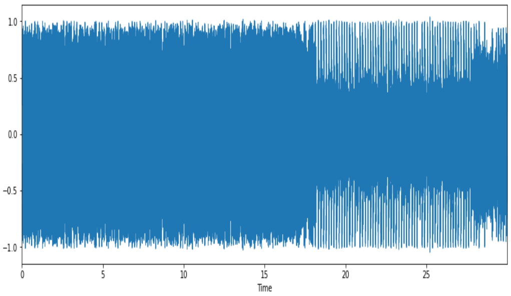


Figure 1: Visualized Waveform

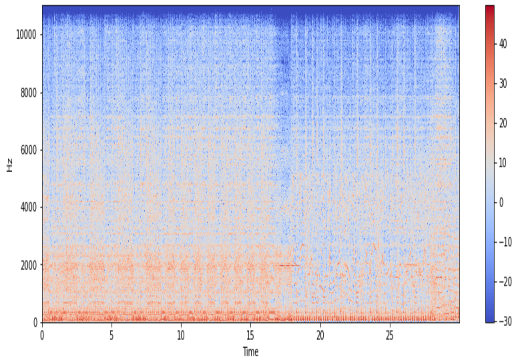


Figure 2: Visualized Spectogram

3.2 Data Preprocessing

Several stages of preprocessing were required in order to ready the dataset for recommendation. First we have downloaded the data on the basis of genre like rock, hiphop, pop, folks and RnB. Then we have divided these data into chunk of 20 second of each music segments. After that we have total of approx 5000 songs in the final dataset which we are using. We handled missing values by removing them from our dataset. Then we normalized the dataset so that they will use a common scale and standardized features by removing the mean and scaling to unit variance which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.. Later we split the data into 80% training and 20% testing for genre prediction task.

3.3 Feature Extraction

In order to represent the tracks numerically, twenty eight audio features were extracted from each track after careful analysis of importance of audio features and the selecting the best features. Features can be broadly classified as time domain and frequency domain features. The feature extraction was done using libROSA a Python library.

i) Time Domain Features
These are features which were extracted from the raw audio signal.

1) Zero Crossing Rate (ZCR) : A zero crossing point refers to one where the signal changes sign from positive to negative. The average of the ZCR across all frames are chosen as representative features.

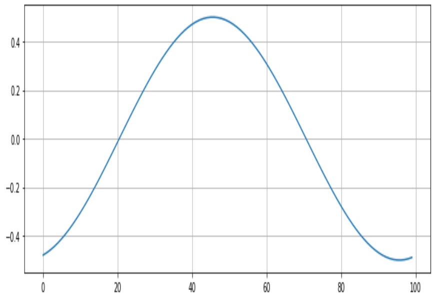


Figure 3: Visualization of ZCR feature

2) Root Mean Square Energy (RMSE) : The root mean square value can be computed as:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{n} x(n)^2 = 1}$$

RMSE is calculated frame by frame and then we take the average across all frames.

3) Tempo : In general terms, tempo refers to the how fast or slow a piece of music is; it is expressed in terms of Beats Per Minute (BPM).

ii) Frequency Domain Features
The audio signal can be transformed into the frequency domain by using the Fourier Transform. We then extract the following features.

1) Mel-Frequency Cepstral Coefficients (MFCC) : It is representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

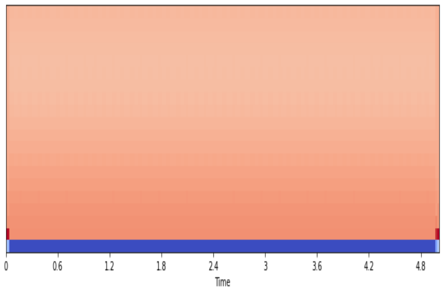


Figure 4: Visualization of MFCC features

2) Chroma Features : This is a vector which corresponds to the total energy of the signal in each of the classes.

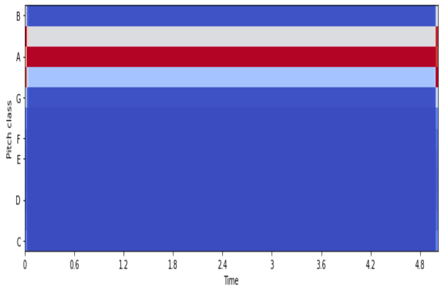


Figure 5: Visualization of Chroma feature

3) Spectral Centroid : For each frame, this corresponds to the frequency around which most of the energy is centered.

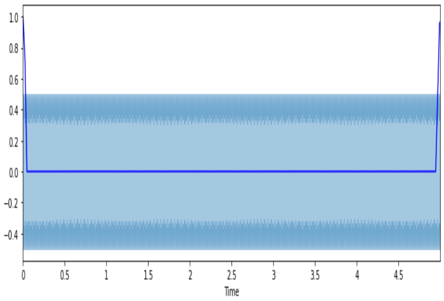


Figure 6: Visualization of Spectral Centroid feature

4) Spectral Roll-off : This feature corresponds to the value of frequency below which 85% (this threshold can be defined by the user) of the total energy in the spectrum lies
There are many more features available. For each of the spectral features described above, the mean of the values taken across frames is considered as the representative final feature that is fed to the model.

We are also finding Correlation matrix of whole dataset as shown below.

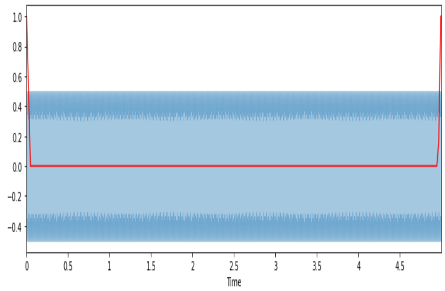


Figure 7: Visualization of Spectral Roll-off feature

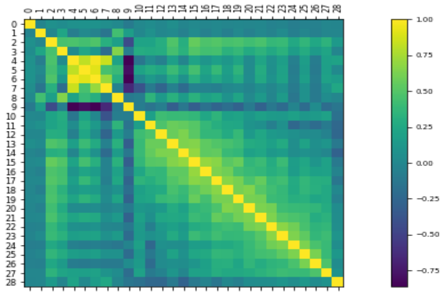


Figure 8: Correlation Matrix

Then we analyse the TSNE plot in 2D and 3D to visualize the data perfectly.
Following is the TSNE plot for 20 second length data.

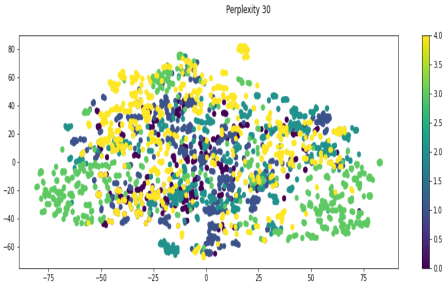


Figure 9: TSNE plot in 2D

3.4 Genre Classification

This section provides a brief overview of the machine learning classifiers adopted in this project for the purpose of predicting genre. We have used various techniques to classify the given song in different genres.

1) Logistic Regression (LR) : This linear classifier is generally used for binary classification tasks.

2) Support Vector Machines (SVM) : SVMs transform the original input data into a high dimensional space using a kernel trick. The transformed data can be linearly separated using a hyperplane. The optimal hyperplane maximizes the margin. For this multi-class classification task, the SVM is implemented as a one-vs-one method.

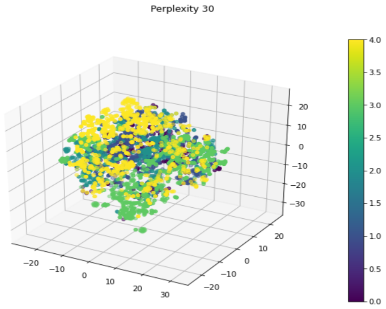


Figure 10: TSNE plot in 3D

3) K-Nearest Neighbour (k-NN) : It is very famous for its simplicity of execution. The k-NN is by design non-linear and it can detect direct or indirect spread information. It also slants with a huge amount of data. The essential computation in our k-NN is to measure the distance between two tunes. We implement 7-KNN model after careful analysis of the effect of the neighbors to choose the optimal k.

4) Decision Tree : The decision trees are the robust, non linear classifiers that use entropy and information gain to classify the features.

5) Neural Network : Neural Networks have the ability to learn by themselves and then produce the output that is not limited to the input provided to them.
We have used the following structure for the Neural Network.

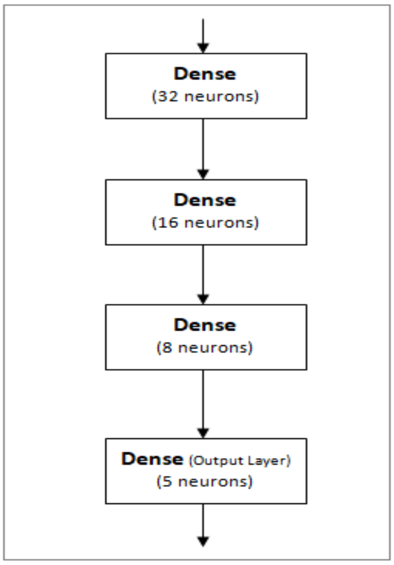


Fig: Structure of the Neural Network

We are using approx 5000 samples of the songs as mentioned above. We are analysing by taking one song at a time and also visualzing it in high dimensional space. Then we are doing feature extraction in which some number of audio features will be extracted from each track and then making final dataset based on features.
We are the using different models to final dataset as men-

tioned above for predicting genre of the particular song. Then we have calculated the similarity of one lyric to another based on features. We have used the cosine similarity. Once we get the similarities, we have then used that similarity scores to access the most similar items and output a recommendation based the number of recommendation user want.
We are also using euclidean distance for finding the distance of each song to the test song data and recommended on the basis of that distance.
Also, we are using knn for the same as it also predict the genre of songs very accurately compared to other algorithms.
Finally we are making an end-to-end pipeline using TKinter in python in which our system will recommend songs based on some input.

3.5 Music Recommendation

For recommending music we are asking the user to input a song from their device and then we are extracting the required features from the input song also. Then have compared four different types of techniques to find the similarity between the songs
1) Cosine Similarity : It measures the similarity between two given vectors by measuring their cosine angles.
2) Euclidean Distance with dtw : It is the distance between two segments. Formula for calculating Euclidean Distance is :

$$d(p,q) = \sqrt{\sum_{n=1}^{n}(pi - qi)^2}$$

3) K Nearest Neighbor : For giving recommendations, K Nearest Neighbor checks how many Neighbors have selected a particular song. The most selected songs are then given as the output depending upon the user requirements.
4) K Nearest Neighbour with Manhattan distance : It is the distance between two vectors when measured at right angle.

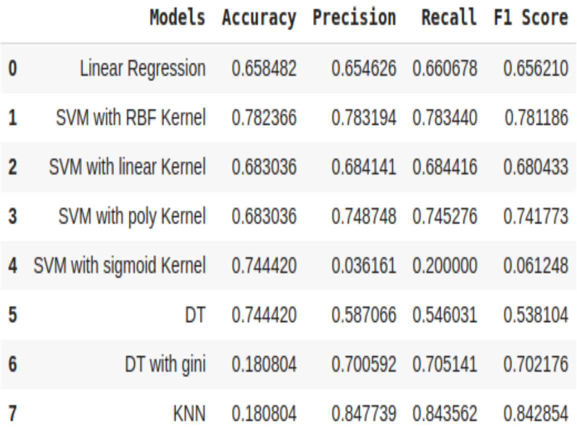
$$d = |x2 - x1| + |y2 - y1| + ...$$

Then we compared the variance between these models and found out that K Nearest Neighbour was performing best at k=25.

4. RESULT

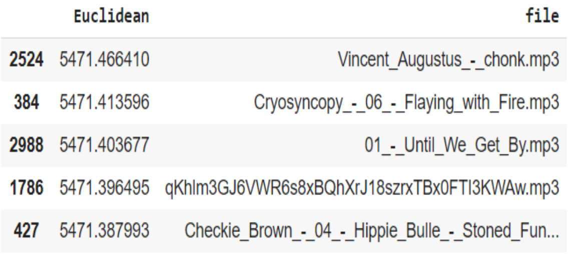
We are using different metrics like accuracy, precision recall and f1-score to evaluate the models and recorded the accuracy of prediction of genre by various models. The table given below lists out the accuracy for the different models used. We use the separately formed test data for measurement of its performance.

Then we are using Cosine Similarities, euclidean distance and knn for the recommendation of the particular song given by user. The best variance was given by K Nearest Neighbour at k = 25. Output of songs given by these models are following :



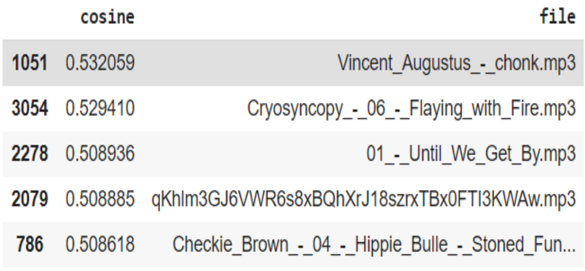
	Models	Accuracy	Precision	Recall	F1 Score
0	Linear Regression	0.658482	0.654626	0.660678	0.656210
1	SVM with RBF Kernel	0.782366	0.783194	0.783440	0.781186
2	SVM with linear Kernel	0.683036	0.684141	0.684416	0.680433
3	SVM with poly Kernel	0.683036	0.748748	0.745276	0.741773
4	SVM with sigmoid Kernel	0.744420	0.036161	0.200000	0.061248
5	DT	0.744420	0.587066	0.546031	0.538104
6	DT with gini	0.180804	0.700592	0.705141	0.702176
7	KNN	0.180804	0.847739	0.843562	0.842854

Figure 11: Comparison of models



	Euclidean	file
2524	5471.466410	Vincent_Augustus_-_chonk.mp3
384	5471.413596	Cryosyncopy_-_06_-_Flaying_with_Fire.mp3
2988	5471.403677	01_-_Until_We_Get_By.mp3
1786	5471.396495	qKhlm3GJ6VWR6s8xBQhXrJ18szrxTBx0FTI3KWAw.mp3
427	5471.387993	Checkie_Brown_-_04_-_Hippie_Bulle_-_Stoned_Fun...

Figure 12: Recommendation Based on euclidean distance using dtw



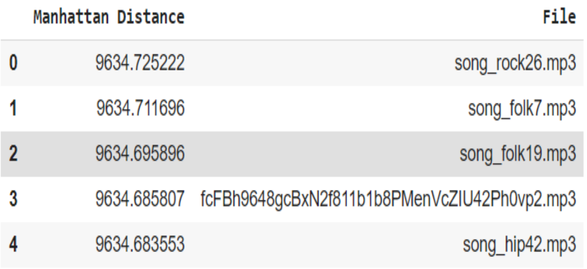
	cosine	file
1051	0.532059	Vincent_Augustus_-_chonk.mp3
3054	0.529410	Cryosyncopy_-_06_-_Flaying_with_Fire.mp3
2278	0.508936	01_-_Until_We_Get_By.mp3
2079	0.508885	qKhlm3GJ6VWR6s8xBQhXrJ18szrxTBx0FTI3KWAw.mp3
786	0.508618	Checkie_Brown_-_04_-_Hippie_Bulle_-_Stoned_Fun...

Figure 13: Recommendation Based on cosine similarity

Finally we are making an end-to-end pipeline for better user experience that is UI for user in which he/she can select a particular song and our system will recommend songs based on their input.

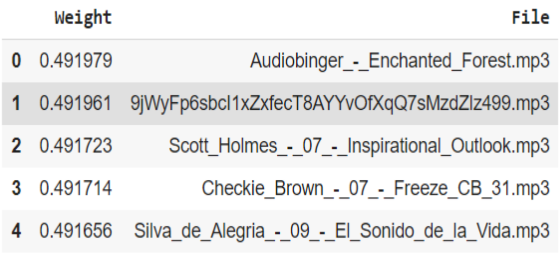
5. EVALUATION

In order to evaluate the performance of each model described above using each classifier and recommenders, we used different matrics such as accuracy score, precision



	Manhattan Distance	File
0	9634.725222	song_rock26.mp3
1	9634.711696	song_folk7.mp3
2	9634.695896	song_folk19.mp3
3	9634.685807	fcFBh9648gcBxN2f811b1b8PMenVcZIU42Ph0vp2.mp3
4	9634.683553	song_hip42.mp3

Figure 14: Recommendation Based on manhattan distance



	Weight	File
0	0.491979	Audiobinger_-_Enchanted_Forest.mp3
1	0.491961	9jWyFp6sbcl1xZxfecT8AYYvOfXqQ7sMzdZlz499.mp3
2	0.491723	Scott_Holmes_-_07_-_Inspirational_Outlook.mp3
3	0.491714	Checkie_Brown_-_07_-_Freeze_CB_31.mp3
4	0.491656	Silva_de_Alegria_-_09_-_El_Sonido_de_la_Vida.mp3

Figure 15: Recommendation Based on knn

and recall.
We have tested on different-different models for getting result more accurately based on that metrics. Till now, we got highest accuracy using KNN for genre prediction as well as for recommendation of song based on k-nearest neighbour's distance.

5.1 How system performs on New Data

We are extracting the same relevant features from the input music file also that we have extracted from training data instead of relying on lyrics or music metadata and then we are finding the similarity between the current dataset and new music test file. Therefore, our model is able to recommend the songs even though the input file is not present in our dataset i.e., **our system can handle cold start problem.**.

5.2 Analysis of Result

We are getting better result in case of KNN compared to other classifiers and recommenders. In KNN, when we make a predction about a song, the algorithm will calculate the distance between the target song and every other song form the dataset. Then KNN uses a voting based technique to find the rank for every song compared to target song and then make a prediction. It is better as it does not rely on any assumption and make prediction based on similarity only.

6. CONTRIBUTION

We have divided the dataset into smaller chunks of 20s each using pydub which is further used to match the test song. The motivation behind this is to gather as much information possible from each song. The division of the dataset in chunks enables more overlapping of input song with a song dataset.

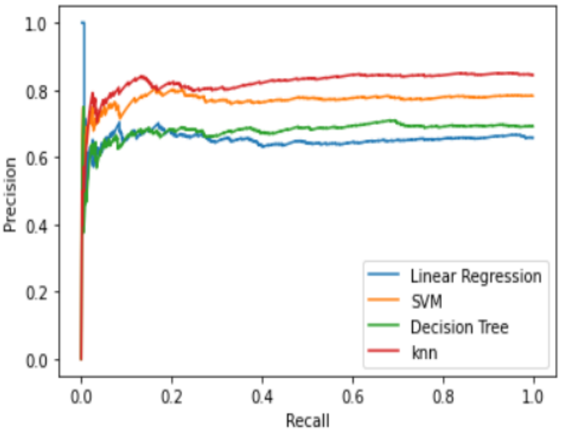


Figure 16: Precision Recall curve

For recommending songs we have also used DTW (dynamic time warping), in time series analysis,DTW is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed. DTW has been applied to temporal sequences of video, audio, and graphics data . DTW is a very efficient algorithm which can be used to calculate optimal scores between the songs, thus efficient recommandation.

Majority of papers that we read were using frequency-domain features for the any dataset but we are also combining time-domain features with frequency domain features for the same to intensify the performance or result for the data.

7. CONCLUSION

Accuracy of classification by different genres and different machine learning algorithms is varied so prediction of genre varies for some songs. In this work, the task of music recommendation is done using the manually created dataset after feature extration. Here, we use the five algorithms namely logistic regression, a support-vector machine, a decision tree, k-nearest neighbor (k-NN), and at last neural network for the genre prediction task. We have used different genres such as rock, hiphop, pop, folks and RnB.
Then we are using cosine similarity, euclidean distance and knn for the recommendation task.
According to our estimations as given above, knn has best performances in prediction as well as in recommendation so it performs far better than other classifiers and other recommenders.
Finally we are making an end-to-end pipeline for better user experience that is UI for user in which he/she can select a particular song and our system will recommend songs based on their input.

8. FUTURE WORK

1. Collection of more data which may improve performance of used models.
2. Expanding the dataset to include more genres and sub genres which would increase the capacity to recommend greater varieties of songs.
3. Analysing the effect of presence and absence of vocals on the recommendation.

9. REFERENCES

[1] J. Fang, D. Grunberg, S. Lui, and Y. Wang, "Development of a music recommendation system for motivating exercise," in *2017 International Conference on Orange Technologies (ICOT)*, 2017, pp. 83–86.

[2] B. McFee, L. Barrington, and G. Lanckriet, "Learning content similarity for music recommendation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2207–2218, 2012.

[3] D. Kim, K. Kim, K. Park, J. Lee, and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, 2007, pp. 399–403.

[4] M. Soleymani, A. Aljanaki, F. Wiering, and R. C. Veltkamp, "Content-based music recommendation using underlying music preference structure," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, 2015, pp. 1–6.

[5] P. J. Rentfrow, L. R. Goldberg, and D. J. Levitin, "The structure of musical preferences: A five-factor model." *Journal of Personality and Social Psychology*, vol. 100, no. 6, pp. 1139–1157, 2011. [Online]. Available: https://doi.org/10.1037/a0022406

[6] S. D. Teh Chao Ying and L. N. Abdullah, "Lyrics-based genre classification using variant tf-idf weighting schemes," *Journal of Applied Sciences*, vol. 15, pp. 289–294, 2015. [Online]. Available: https://scialert.net/abstract/?doi=jas.2015.289.294

[7] J. V. Balen, "Automatic recognition of samples in musical audio," 2011.

[8] P. Lamere and E. Pampalk, "Social tags and music information retrieval." vol. 37, 06 2008, p. 24.