MUSIC RECOMMENDATION SYSTEM

Kahaan Patel
AU1841110
Ahmedabad University
kahaan.p@ahduni.edu.in

Vidit Vaywala
AU1841128
Ahmedabad University
vidit.v@ahduni.edu.in

Hemil Shah
AU1841135
Ahmedabad University
hemil.s@ahduni.edu.in

Meet kadiya
AU1841099
Ahmedabad University
meet.k@ahduni.edu.in

I. ABSTRACT

We are trying to build a simple recommendation system using the k-nearest neighbors way of classification. After achieving our initial goal, we will be improving the model built and checking how other algorithms or hyperparameters help us in improving it further. The final part of our project would be to identify a user's mood using Support Vector Machine and then recommending songs to him/her.

II. KEY WORDS

K-nearest neighbours,Fuzzy String Matching, Support Vector Machine, Classification Algorithms, Music recommendation, CSR Matrix.

III. INTRODUCTION

Music Recommend system is a system that seeks to predict or filter preference according to the user's different choices.It learns from the user's past listening history and recommends them various songs which they would probably like to hear in the future.Collaborative filtering algorithms which predict (filtering) taste of a user by collecting preferences and tastes from many other users (collaborating) are also implemented. In this first phase of the project, we have build a recommender system using k-nearest neighbors algorithm. Our dataset consists of 2 million records containing user ids, song ids and the number of times a user has listened to a particular song. [1]

IV. LITERATURE SURVEY

A. *k NEAREST NEIGHBOURS -KNN algorithm*

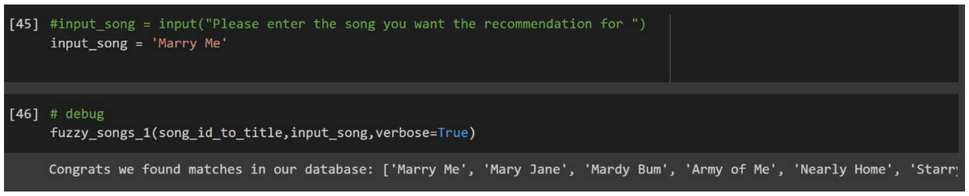
The algorithm k-nearest neighbours is used to predict a user's rating of unrated items. When KNN makes a song prediction, it calculates the ”distance” between the target song and every other song in its database. It then ranks the songs based on their distances and returns the top k as the most similar song recommendations. The k-nearest neighbours (KNN) algorithm relies on item feature similarity rather than making any assumptions about the underlying data distribution. A data point is classified in KNN by a majority vote of its neighbours, with the data point assigned to the most common class among its k-nearest neighbours, as measured by a distance function. inclass if k = 1, then simply the data point is assigned to the class of its nearest neighbor—i.e., itself. In general; to reduce the overall noise ,a large k value is more favourable. [2]

IMPLEMENTATION

1) **Collecting Data**- We have used the million song dataset that is available online for developing our model. It consists of users id, songs id ,title ,author and how many times a user has listened to a song.
2) **Filtering and Visualizing data**- Using the Pandas and the seaborn library we filtered out the dataset and then explored its characteristics after visualizing it using boxplots, histograms, etc.
3) **Using Scipy Sparse CSR Matrix**- A lot of values in put matrix are 0 and hence we are dealing with an extremely sparse matrix. We will use the CSR matrix to represent our sparse matrix.
4) **Using Fuzzy String Matching**- When a user enters a song's name partially correctly then our fuzzy matcher matches the user's input to all the songs in our database. And then it selects the movie with the highest ratio. For example, if the user enters “Mary mee” instead of “Marry me(this is the actual song name)” then the fuzzy matcher will understand that the user wanted to select the song “Marry me”.
5) **Using KNN algorithm to make predictions**- Using the scikit-learn library we used the k-nearest neighbor algorithm to find the best recommendations to a particular song that user inputted
6) **Getting the Recommendation**- Our model provides recommendations to the user based on evaluation of the k-nearest neighbors of our input song.

RESULTS

1) Fuzzy Matching - When a user enters a song's name partially correctly then our fuzzy matcher matches the user's input to all the songs in our database. And then it selects the movie with the highest ratio. For example, if the user enters ”Mary mee” instead of ”Marry me (this is the actual song name)” then the fuzzy matcher will understand that the user wanted to select the song ”Marry Me” .Hence through simple language processing we can exempt the user from entering the correct name everytime. Little mistakes will be taken care of by the fuzzy matcher. We have implemented this using the fuzzywuzzy library of python.

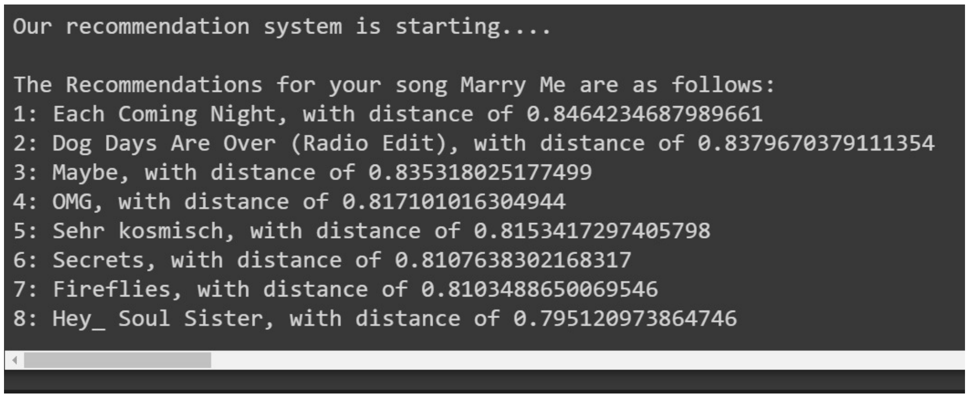


```
[45] #input_song = input("Please enter the song you want the recommendation for ")
     input_song = 'Marry Me'

[46] # debug
     fuzzy_songs_1(song_id_to_title,input_song,verbose=True)

     Congrats we found matches in our database: ['Marry Me', 'Mary Jane', 'Mardy Bum', 'Army of Me', 'Nearly Home', 'Starr
```

2) Music Recommendation Output - Using the scikit-learn library we used the k-nearest neighbor algorithm to find the best recommendations to a particular song that user inputted. Our version of KNN algorithm Calculates the Cosine distance (Cosine distance metric is used to find similarities between different documents) from the new song inputted by users to all the other songs. The songs with the higher value (nearer to 1) are considered good recommendations and our model outputs those songs in the form of recommendations to the user



```
Our recommendation system is starting....

The Recommendations for your song Marry Me are as follows:
1: Each Coming Night, with distance of 0.8464234687989661
2: Dog Days Are Over (Radio Edit), with distance of 0.8379670379111354
3: Maybe, with distance of 0.835318025177499
4: OMG, with distance of 0.817101016304944
5: Sehr kosmisch, with distance of 0.8153417297405798
6: Secrets, with distance of 0.8107638302168317
7: Fireflies, with distance of 0.8103488650069546
8: Hey_ Soul Sister, with distance of 0.795120973864746
```

CONCLUSION

In the first phase of the project we have successfully build the K-nearest neighbours model and we are getting good music recommendations as output. The future work of the project includes improving this model and checking how other algorithms or hyperparameters help us in improving it. Secondly, we will also work on the problem statement of identifying user's mood using Support Vector Machine and then recommending songs to users.

REFERENCES

[1] B. Srebrenik, "Introduction to Music Recommendation and Machine Learning," Medium, 04-Dec-2018. [Online]. Available: https://medium.com/@briansrebrenik/introduction-to-music-recommendation-and-machine-learning-310c4841b01d: :text=These

[2] "Using KNN algorithm for classification of textual documents," IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8079924/?$casa_token$=35bn2HhwmIMAAAAA

[3] C.-S. Mike Wu and D. Garg, "Movie Recommendation System Using Collaborative FilteringChing-Seh Mike Wu," IEEE Xplore. [Online]. Available: https://ieeexplore.ieee.org/document/8663822. [Accessed: 17-Mar-2021].