MUSIC RECOMMENDATION SYSTEM AND GENRE PREDICTION

COURSE PROJECT REPORT
18CSE398J -Machine Learning - Core Concepts with Applications

(2018 Regulation)

III Year/ VI Semester

Academic Year: 2022 -2023 (EVEN)

By

ARYAMAN ADIVYA SINGH (RA2011003011266)
ADITYA SUYASH (RA2011003011261)

Under the guidance of
DR. VADIVU G.
Professor
Department of Data Science and Business Systems

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Kattankulathur, Kancheepuram
MAY 2023

ABSTRACT

The aim of this project is to develop a recommendation system for music that suggests songs to users based on their listening history and various factors such as energy, danceability, key, tempo, etc. The system uses machine learning techniques to analyze the user's listening behavior and preferences to generate personalized recommendations.

The dataset used for training the model consists of music metadata such as genre, duration, time signature which includes various features such as tempo, key, energy, genre, duration, time signature etc, which are used to represent the characteristics of each song. The listening history is used to capture the user's musical preferences and behavior.

The proposed system is a useful tool for music lovers who are looking for personalized song suggestions based on their preferences and listening history. With the increasing popularity of music streaming services, such recommendation systems can enhance the user experience and help users discover new music they may enjoy.

Additionally, Genre prediction is the task of automatically identifying the genre of a given piece of text or media. It plays an important role in various fields, such as music and movie recommendation systems, content tagging, and digital marketing. This task can be approached through different methods, such as machine learning algorithms and natural language processing techniques.

INTRODUCTION

Music recommendation systems are becoming increasingly popular as a way to help users discover new music that they might enjoy. These systems use machine learning algorithms to analyze user listening habits and recommend songs that are similar to their preferences. The project at hand is a machine learning-based music recommendation system that uses song metadata and user listening history to make personalized song recommendations to users.

The system uses collaborative filtering, a popular machine learning technique, to identify similarities between users and make recommendations based on listening history. Additionally, the system also takes into account song metadata such as key, genre, danceability, mode, and tempo to further improve the accuracy of recommendations. By combining user listening history with song metadata, the system can provide more personalized and accurate recommendations.

The primary goal of the project is to enhance the user experience by providing accurate and engaging recommendations that cater to individual user preferences. The system aims to increase user engagement and satisfaction with the music platform by offering recommendations that are tailored to their unique taste in music. The use of machine learning and song metadata is expected to improve the accuracy of recommendations and provide a more engaging listening experience for users.

DATASET

Dataset Link: https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify

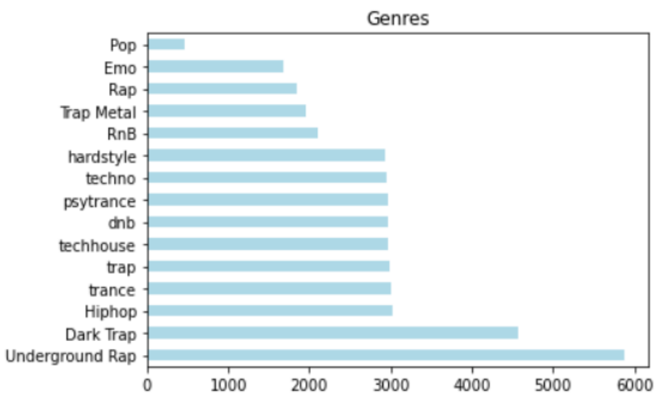
The project uses the Dataset of Songs on Spotify from Kaggle. It houses a full list of genres that contain Trap, Techno, Techhouse, Trance, Psytrance, Dark Trap, DnB (drums and bass), Hardstyle, Underground Rap, Trap Metal, Emo, Rap, RnB, Pop and Hiphop.

# danceability	# energy	# key	# loudness	# mode	# speechiness	# acousticness	# instrumentalness	# liveness	# valence
danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence
0.07 – 0.99	0 – 1	0 – 11	-33.4 – 3.15	0 – 1	0.02 – 0.95	0 – 0.99	0 – 0.99	0.01 – 0.99	0.02 – 0.99
0.831	0.8140000000000001	2	-7.364	1	0.42	0.0598	0.0134	0.0556	0.389
0.7190000000000001	0.493	8	-7.23	1	0.0794	0.401	0.0	0.11800000000000001	0.124
0.85	0.893	5	-4.783	1	0.0623	0.0138	4.14e-06	0.37200000000000005	0.0391
0.47600000000000003	0.7809999999999999	0	-4.71	1	0.10300000000000001	0.0237	0.0	0.114	0.175
0.7979999999999999	0.624	2	-7.667999999999999	1	0.293	0.217	0.0	0.166	0.591
0.721	0.568	0	-11.295	1	0.414	0.0452	0.212	0.128	0.109
0.718	0.6679999999999999	8	-4.162	1	0.13699999999999998	0.0254	0.0078	0.124	0.038
0.6940000000000001	0.711	8	-5.525	1	0.221	0.0397	0.0	0.11199999999999999	0.28300000000000003
0.774	0.7509999999999999	1	-2.445	1	0.198	0.0614	0.0	0.0728	0.18899999999999997
0.893	0.907	11	-10.405999999999999	1	0.36700000000000005	0.152	0.0311	0.5579999999999999	0.302
0.8640000000000001	0.365	8	-10.219	1	0.0655	0.187	0.0	0.11599999999999999	0.0478
0.736	0.932	1	-3.7260000000000004	1	0.271	0.146	0.0025	0.182	0.18
0.825	0.7609999999999999	8	-5.388999999999999	1	0.10400000000000001	0.0111	0.00359	0.33399999999999996	0.161
0.767	0.5760000000000001	10	-9.683	0	0.256	0.145	2.61e-06	0.0968	0.187
0.765	0.726	5	-5.58	1	0.191	0.0077	0.0	0.619	0.27
0.617	0.541	6	-4.113	1	0.78	0.125	0.0	0.369	0.43

The various factors in consideration include danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence.
By analysing these factors and comparing with other data values in the dataset, the model attempts to predict realistic song suggestions.

Meanwhile for Genre Prediction, the same factors are used to predict genre which plays a crucial role in music recommendation systems.The dataset contains a variety of song genres to train onto as mentioned below:

Underground Rap	5875
Dark Trap	4578
Hiphop	3028
trance	2999
trap	2987
techhouse	2975
dnb	2966
psytrance	2961
techno	2956
hardstyle	2936
RnB	2099
Trap Metal	1956
Rap	1848
Emo	1680
Pop	461



METHODS

MUSIC RECOMMENDATION SYSTEM

DATA PREPROCESSING

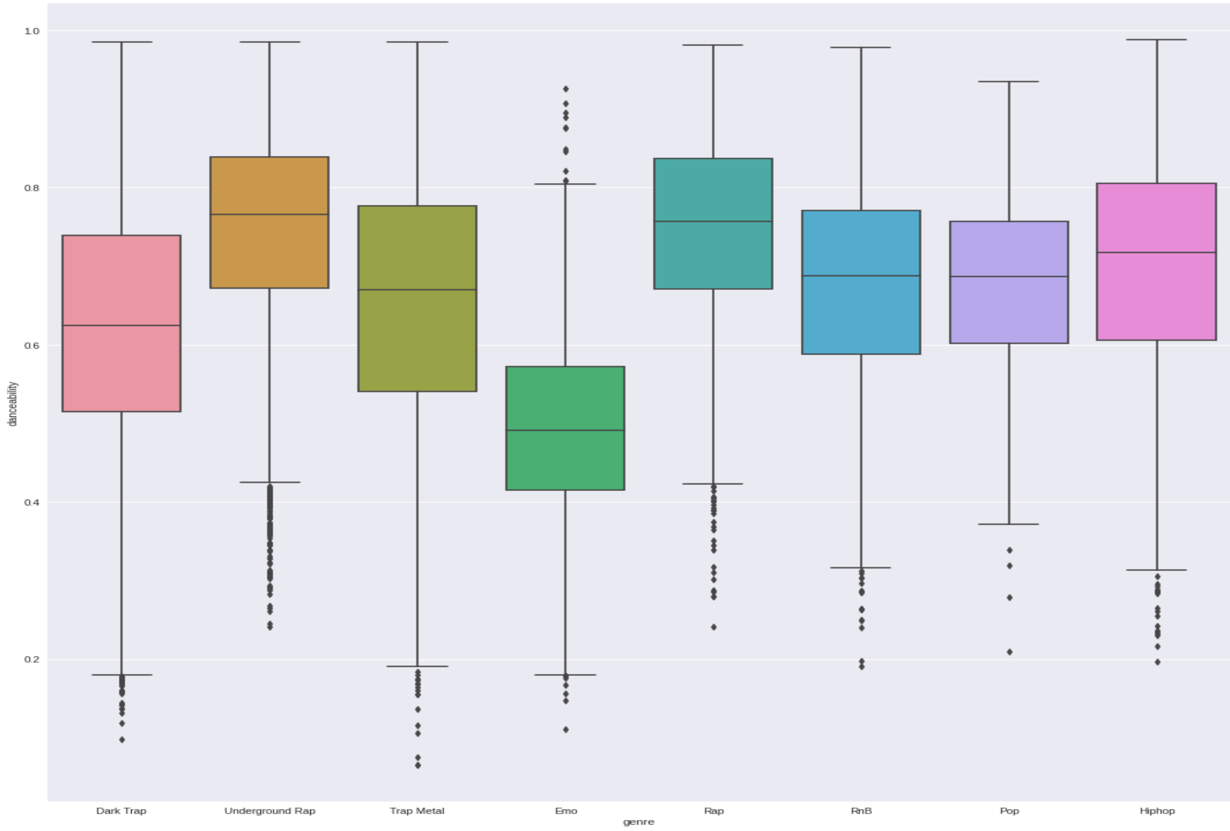
Data pre-processing involves cleaning, transforming, and organising raw data in order to prepare it for analysis.
Basic Data Preprocessing and Preparation is done to remove unwanted columns from data set that may cause

EDA (Exploratory Data Analysis)

Exploratory data analysis (EDA) is an approach to analyzing data that involves investigating and summarizing the main characteristics of the data to gain insights and generate hypotheses. EDA is typically performed at the beginning of a data analysis process and involves techniques such as visualization, summary statistics, and data cleaning. The main goals of EDA are to understand the distribution and relationships between variables, detect outliers and anomalies, and identify patterns and trends in the data.

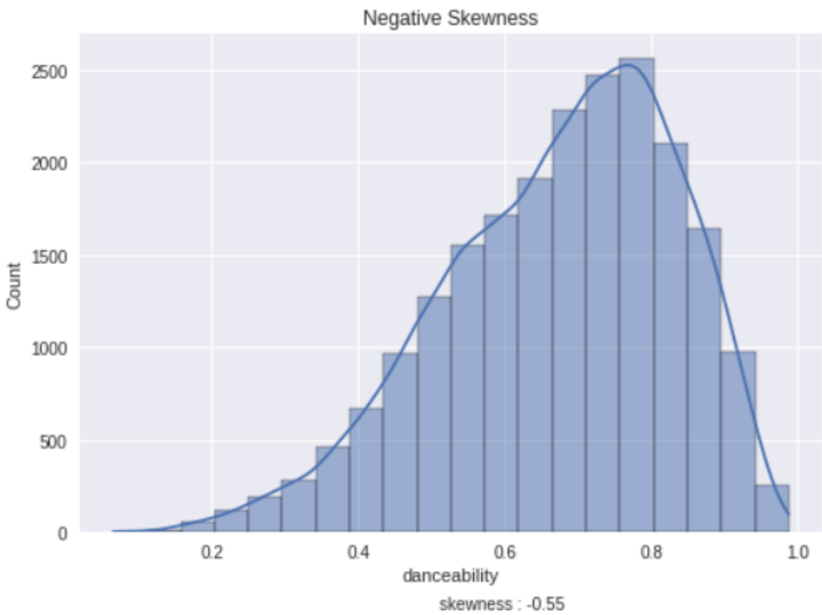
Box Plots

Box plots are a powerful tool in exploratory data analysis (EDA) that provide a visual representation of the distribution and spread of a dataset. A box plot displays the median, quartiles, and outliers of a dataset, making it easy to identify skewness, variability, and potential outliers. By comparing multiple box plots, analysts can quickly identify differences and similarities between groups or datasets. Box plots can also be used to identify patterns and trends in time-series data, as well as to compare distributions across different levels of a categorical variable.



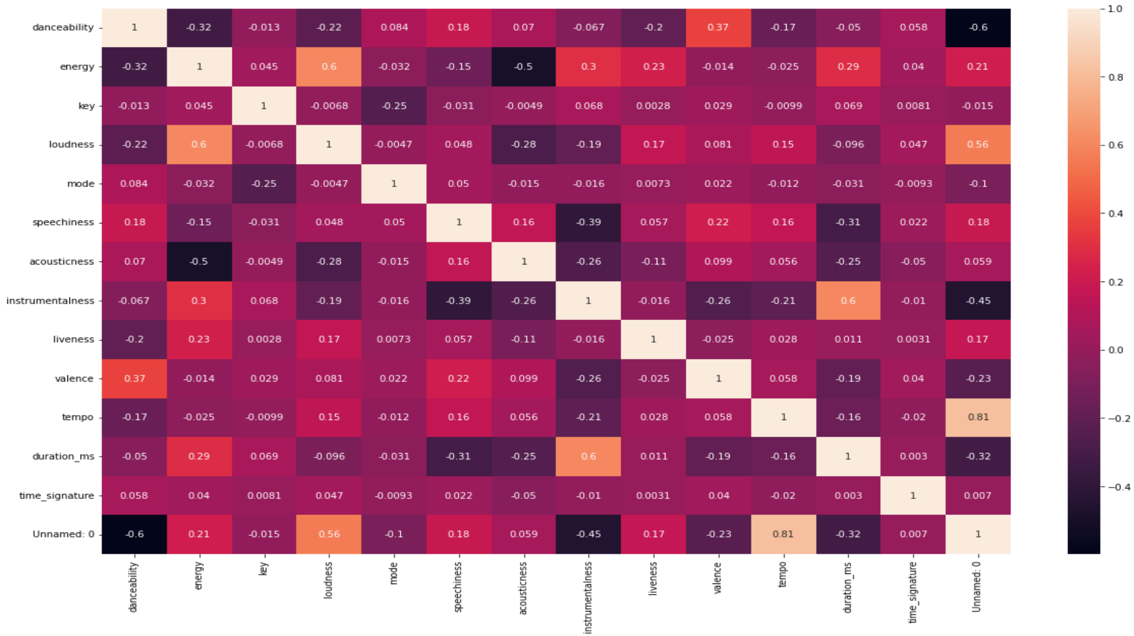
Skewness Tests

Skewness tests are necessary because they help us to understand the shape of a distribution and identify departures from normality. Skewness is a measure of the asymmetry of a distribution, with positive skewness indicating a longer tail to the right and negative skewness indicating a longer tail to the left. Skewed distributions can have a significant impact on statistical analyses and can lead to biased results if not taken into account.



Correlation matrix

Correlation matrix is necessary while building machine learning models because it helps us to identify the relationships between the features (variables) in the dataset. In machine learning, we typically have a large number of features, and it can be difficult to identify which features are most relevant to the target variable. Correlation matrix provides a quantitative measure of the strength and direction of the linear relationship between pairs of features, which can help us to identify which features are highly correlated with each other and which features are most strongly correlated with the target variable.



MACHINE LEARNING MODEL BUILDING

Song Recommendation System

In a music recommendation system, one of the key tasks is to find songs or artists that are similar to the user's preferred music. To accomplish this task, various machine learning models can be used, such as the linear kernel model, Euclidean distance model, and cosine similarity model. These models are often used in conjunction with a normalized matrix that represents the musical attributes of each song or artist.

Linear kernel model:
The linear kernel model is a type of Support Vector Machine (SVM) that calculates similarity between songs based on the relationship between their attributes. In this model, the similarity between songs is calculated using a dot product, which measures the similarity between the attributes of two songs. The normalized matrix is used to calculate the dot product and to ensure that the data is scaled consistently.

```
recommendation(songName)

69                     Magazine
2166                   Magazine
1548                    Enemies
1621                       Yeah
353                        Mrs.
1260    2020 PHARMACY FREESTYLE
620               Bladed Choppa
856                 Dead Angels
1341                 Devil/skin
2112                        H2O
Name: song_name, dtype: object
```

Euclidean distance model:
The Euclidean distance model calculates the distance between songs by measuring the distance between their attributes. The normalized matrix is used to calculate the Euclidean distance, which is the straight-line distance between two points in a multi-dimensional space. This model can be used to find songs that are similar to each other based on their musical attributes.

```
recommendation(songName,model =euclidian )

2887                              Intro
2754                          sacrifice
1367    SadlyThatsJustTheWayThingsAre
2971                             Shiver
3015                   End of Broadcast
2890                          Interlude
2744             last night (glo remix)
718                 Doesithurttoloveme?
3025                            Wayward
1067                             Wither
Name: song_name, dtype: object
```

Cosine similarity model:

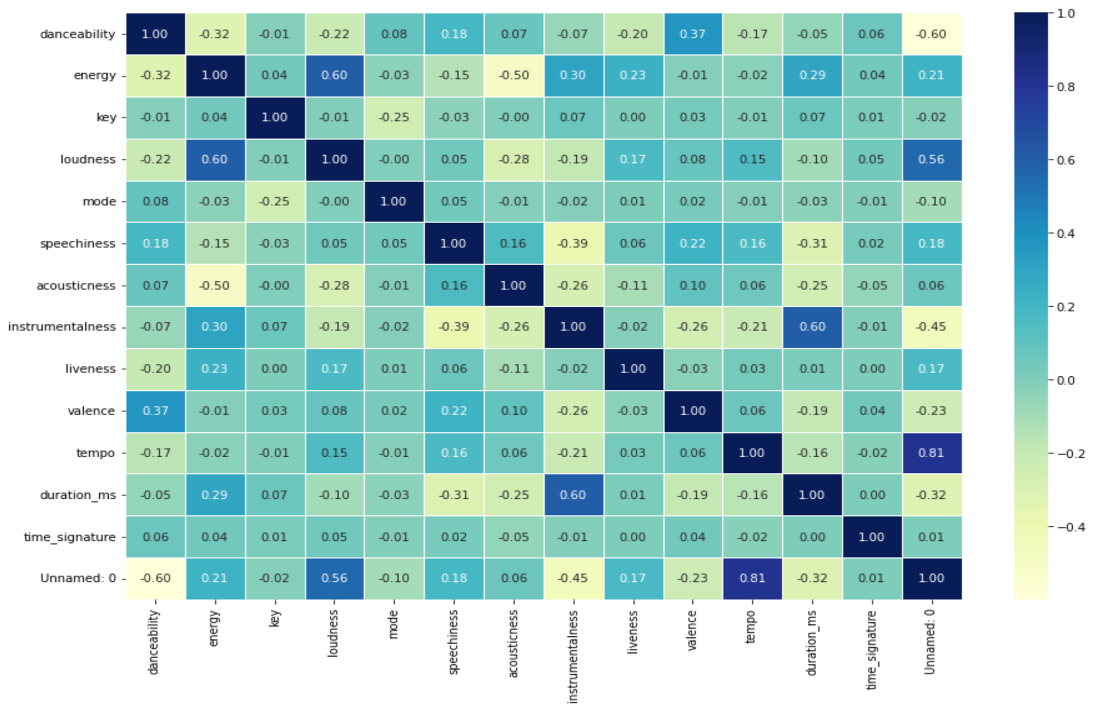
The cosine similarity model is a measure of similarity between two non-zero vectors, which measures the cosine of the angle between them. In the music recommendation system, the cosine similarity model can be used to measure the similarity between songs based on their musical attributes. The normalized matrix is used to calculate the cosine similarity, which can be used to find songs that are similar to each other based on their musical

```
recommendation(songName,model =consine )

586                 Florida Thang
2044                Florida Thang
2276                           41
1474                 Jeffer Drive
2224                 Jeffer Drive
187                        PRBLMS
3224                       Narshe
2929                 Become Again
1581                   Goth Bitch
629      Case 19 (feat. 6ix9ine)
Name: song_name, dtype: object
```

Genre Prediction

Genre prediction is a common task in machine learning that involves predicting the genre of a given piece of media, such as a song or a movie. There are several machine learning algorithms that can be used for genre prediction, including logistic regression, K-nearest neighbors (KNN), and random forest.



Logistic regression is a supervised learning algorithm that is used for binary classification tasks, where the target variable has two possible outcomes. In genre prediction, logistic regression can be used to predict whether a given piece of media belongs to a particular genre or not. Logistic regression works by estimating the probability of the target variable based on the input features, and then making a binary prediction based on a threshold value.

K-nearest neighbors (KNN) is a non-parametric algorithm that is often used for classification and regression tasks. KNN works by finding the k nearest neighbors to a given data point based on a distance metric, and then making a prediction based on the class of the majority of the nearest neighbors. In genre prediction, KNN can be used to identify similar pieces of media based on their genre, and then make a prediction based on the most common genre among the k-nearest neighbors.

Random forest is an ensemble learning algorithm that is often used for classification and regression tasks. Random forest works by creating multiple decision trees on randomly sampled subsets of the data, and then aggregating the predictions of the individual trees to make a final prediction. In genre prediction, random forest can be used to identify the most important features for predicting genre, and then make a prediction based on the collective predictions of the decision trees.

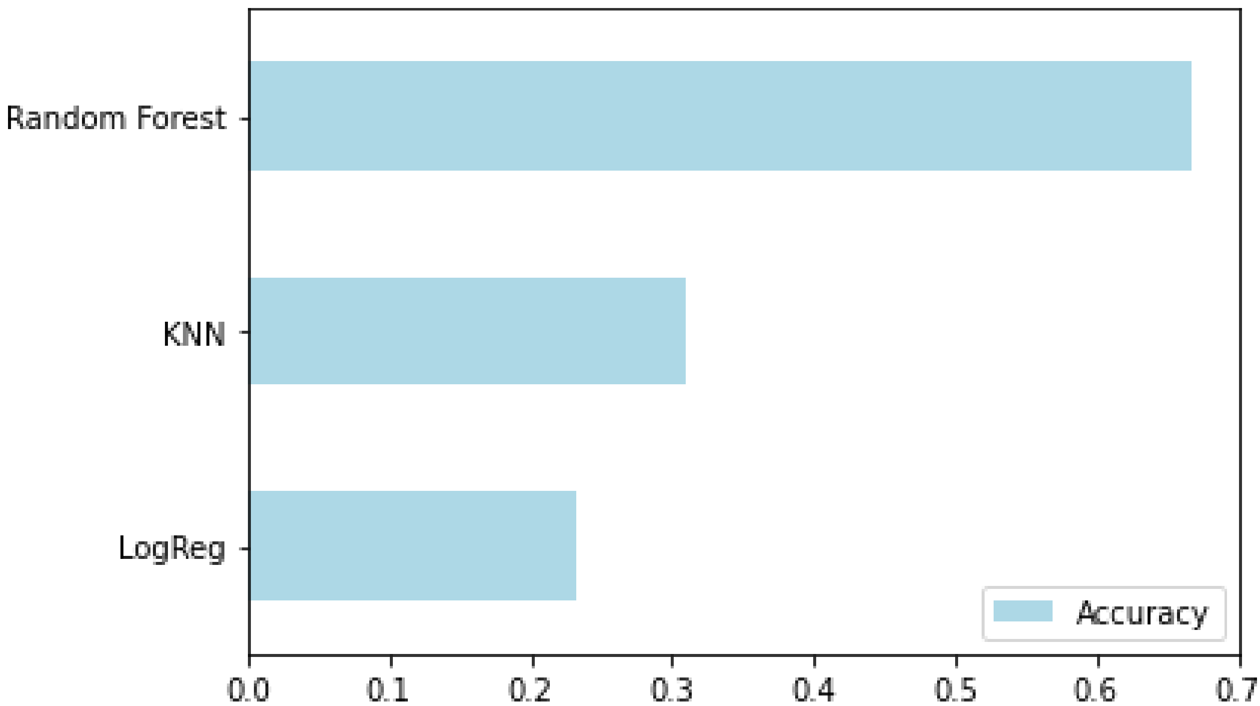
When using these algorithms for genre prediction, it is important to pre-process the data appropriately and choose an appropriate evaluation metric to assess their performance. Additionally, hyperparameter tuning can be used to optimize the performance of these algorithms on a given dataset.

EXPERIMENTS AND RESULTS

The experiments that were conducted to evaluate the performance of this recommendation system include:

Feature importance analysis: This experiment involves analyzing the importance of different song metadata features (key, genre, danceability, mode, tempo) in the ML model. This can help identify which features are most important for making accurate recommendations and can guide future feature engineering efforts

Accuracy comparison: The below mentioned graph depicts how the models created for genre prediction perform in predicting the genres with Random forest showing the highest accuracy for predictions. Even though the Random Forest has the highest score, it is not able to reach a 90% accuracy due to common causes of overfitting and they cannot be avoided as they have a tendency of causing high variance and high test errors.



CONCLUSION AND FUTURE WORK

The use of song metadata in conjunction with user listening history can significantly improve the accuracy of song recommendations. This approach provides a more personalised and tailored experience to users, which can increase user satisfaction and engagement with the music platform.

Collaborative filtering can be an effective method to make recommendations based on user listening history. However, the ML model that takes into account song metadata features can potentially provide even more accurate recommendations.

Future work could focus on improving the ML model by incorporating additional features or data sources. For example, sentiment analysis of song lyrics could be used to identify songs with similar themes or moods, or user feedback on recommended songs could be used to improve the model's performance.

The ultimate goal of these improvements is to enhance the user experience and increase user engagement with the music platform. By providing accurate and personalized recommendations, users are more likely to spend more time on the platform, discover new music, and ultimately enjoy their listening experience more.

In conclusion, the ML project using song metadata and user listening history to make song recommendations has shown promising results for improving the user experience. Future work can focus on incorporating additional features and data sources to further improve the accuracy of recommendations and increase user satisfaction. The end goal is to create a more personalized and enjoyable listening experience for users.

REFERENCES

1. “Dataset of songs in Spotify” - https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify
2. "Collaborative Filtering Recommender Systems" by Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl (2001)
3. "Content-Based Recommendation Systems" by Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich (2010)
4. "Music Recommendation and Discovery in the Long Tail" by Chris Anderson (2004)
5. "Music information retrieval" edited by Zbigniew W. Ras and Alicja Wieczorkowska (2010)
6. "Deep Learning for Music Recommendation: Challenges and Opportunities" by Yi-Hsuan Yang, Yi-An Chen, and Zhe-Cheng Fan (2019)
7. "A Survey on Music Recommendation Systems" by Jonghyun Kim and Yejin Jang (2016)
8. "The Million Song Dataset" by Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere (2011)
9. "The Spotify Dataset" by Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M. Schmidt, and Xavier Serra (2017)
10. "Last.fm Dataset" by Last.fm (2014)
11. "A Survey on Deep Learning Techniques for Music Generation" by Yongqiang Li, Li Li, Ying Zhang, Jing Li, and Dongxin Liu (2020)
12. "The Music Genome Project" by Pandora (2008)
13. "Music Recommendation Using Collaborative Filtering and Word Embedding" by Jun Wang, Yun Zhang, Jiaxiang Wu, and Chengzhong Xu (2019)
14. "Multi-armed Bandit for Music Recommendation" by Daniel Oskarsson and Vasilis Vryniotis (2019)
15. "Music Emotion Recognition: State of the Art and Future Prospects" by Wenqiang Lei, Tao Liu, and Jianhua Tao (2020)
16. "Exploiting User Reviews for Music Genre Classification Using Deep Convolutional Neural Networks" by Saumitra Mishra, Oded Nov, and Luis Gravano (2018)
17. "A Hybrid Approach for Music Recommendation based on Collaborative Filtering and Audio Content Analysis" by Jaesik Choi, Kyogu Lee, and Juhan Nam (2014)
18. "A Comparative Study of Collaborative Filtering Algorithms for Music Recommendation" by Wei-Nan Zhang, Jian Xu, Yun He, and Xiangyang Xue (2008)
19. "A Hybrid Method of Music Recommendation Using Content-Based and Collaborative Filtering" by Youngwook Kim, Suhyun Choi, and Chae-Jung Park (2015)
20. "Music Recommendation Using Tree-Based Collaborative Filtering with Graph Embedding" by Jun Wang, Yu Zhang, and Chengzhong Xu