Reeya Gupta
Title: Music Genre Prediction

Introduction:"Music Genre Prediction" aims to classify and predict the genre of a given song and categorizes music into different genres using machine learning algorithms. Python and various machine-learning libraries are used to implement the project.

Problem Statement: The problem statement is to create a model that can correctly predict a song's genre from its features. The problem addressed in this project is the classification of music into different genres based on its audio features.

Motivation: The project is inspired by the fact that music plays a significant role in our daily lives, and that different people have different tastes in various musical genres. It can be difficult for a music streaming service to suggest songs to users that fit their musical tastes. Building a music genre prediction system to aid music streaming platforms in making user-based song recommendations is the driving force behind this project.The goal is to create a tool that will help musicians, music fans, and industry professionals organize, classify, and suggest music to listeners based on their preferred genre. The project also seeks to clarify the connection between musical elements and genre categorization.

Scope:
The project's scope includes data collection, data preprocessing, exploratory data analysis, feature extraction, model construction, and evaluation. To categorize music into different genres, the project employs a number of machine learning algorithms, including Logistic Regression, Decision Tree, K-Nearest Neighbor, and Random Forest.

Objective of the project:
Collecting and preprocessing the data required for the project.
Exploring and analysing the data to understand its characteristics.
Extracting relevant features from the data for model building.
Building and evaluating machine learning models for music genre classification.
Comparing the performance of different models
Discussing the results and insights obtained from the project and providing recommendations for future research.

Background and Literature Review: The project aims to predict the genre of a music track based on its audio features. The problem statement has been addressed by numerous researchers over the years and is pertinent to the field of music information retrieval (MIR).

Spectral centroid, spectral rolloff, and zero-crossing rate are a few audio features that have been used in some studies to predict the genre of a musical track. Other studies have classified musical genres using machine learning techniques like KNN, SVM, and neural networks.

In this project, the audio features have been extracted using the Librosa library in Python. Spectral centroid, spectral bandwidth, spectral contrast, and chroma features are among the extracted features. These features have been used in conjunction with machine learning models like KNN, SVM, and Random Forest to predict the genre of a music track.

The relevance of these techniques and models to the problem statement lies in the fact that these features capture important information about the audio signal that can be used to differentiate between different music genres. Machine learning models are able to learn patterns in the data and make predictions based on these patterns, which makes them well-suited for classification tasks like music genre prediction.

The data collection process involves querying the Spotify API to collect the features for all tracks in a given playlist, while the preprocessing process involves dropping unwanted columns, splitting the data into features and labels, encoding the labels, scaling the features, and splitting the data into training and testing sets.

Data Collection:
The Spotipy library is used to connect to the Spotify API and authenticate the user.
A search query is constructed using the artist name and track name to obtain the track ID.
The track ID is used to get the track features such as danceability, energy, loudness, etc. from the Spotify API.
The features for all tracks in a given playlist are obtained and stored in a dataframe.
The dataframe is saved as a CSV file for further processing.

Data Preprocessing:
The CSV file containing the track features is loaded into a Pandas dataframe.
The 'id' column is dropped since it is not a feature and not required for classification.
The dataframe is split into features and labels. The 'genre' column is used as the label.

The label column is converted to numerical values using label encoding.
The data is split into training and testing sets using a 70-30 split.
The features are scaled using StandardScaler from Scikit-learn to normalize them.
The preprocessed data is saved as a pickle file for future use.

Data Analysis:
The data analysis of the project includes data preprocessing, exploratory data analysis, and model selection.

The dataset used in the project contains audio features of over 4000 tracks from 16 different genres. The data preprocessing section of the code includes standardizing the data, encoding the target variable, and splitting the data into training and testing sets. Various visualizations are used such as bar charts, histograms, and heatmaps to understand the distribution of the data and the correlation between features.

After completing the data analysis, several machine learning algorithms such as decision tree, random forest, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) for classification. Each model is trained on the training set and its performance is evaluated on the testing set using metrics such as accuracy, precision, recall, and F1-score.

It is observed that some features such as tempo and duration have a significant impact on the classification accuracy while others such as spectral centroid and spectral bandwidth have a lower impact.

Understanding the models:
K-Nearest Neighbors classification - plotting the accuracy of the model for different values of k.

```
K value: 1, Train/Test Accuracy: 1.000/0.752, Overall Accuracy: 1.752
K value: 3, Train/Test Accuracy: 0.875/0.815, Overall Accuracy: 1.690
K value: 5, Train/Test Accuracy: 0.866/0.852, Overall Accuracy: 1.718
K value: 7, Train/Test Accuracy: 0.848/0.861, Overall Accuracy: 1.709
K value: 9, Train/Test Accuracy: 0.851/0.870, Overall Accuracy: 1.721
K value: 11, Train/Test Accuracy: 0.847/0.870, Overall Accuracy: 1.717
K value: 13, Train/Test Accuracy: 0.848/0.870, Overall Accuracy: 1.718
K value: 15, Train/Test Accuracy: 0.848/0.870, Overall Accuracy: 1.718
K value: 17, Train/Test Accuracy: 0.848/0.870, Overall Accuracy: 1.718
K value: 19, Train/Test Accuracy: 0.848/0.870, Overall Accuracy: 1.718
```

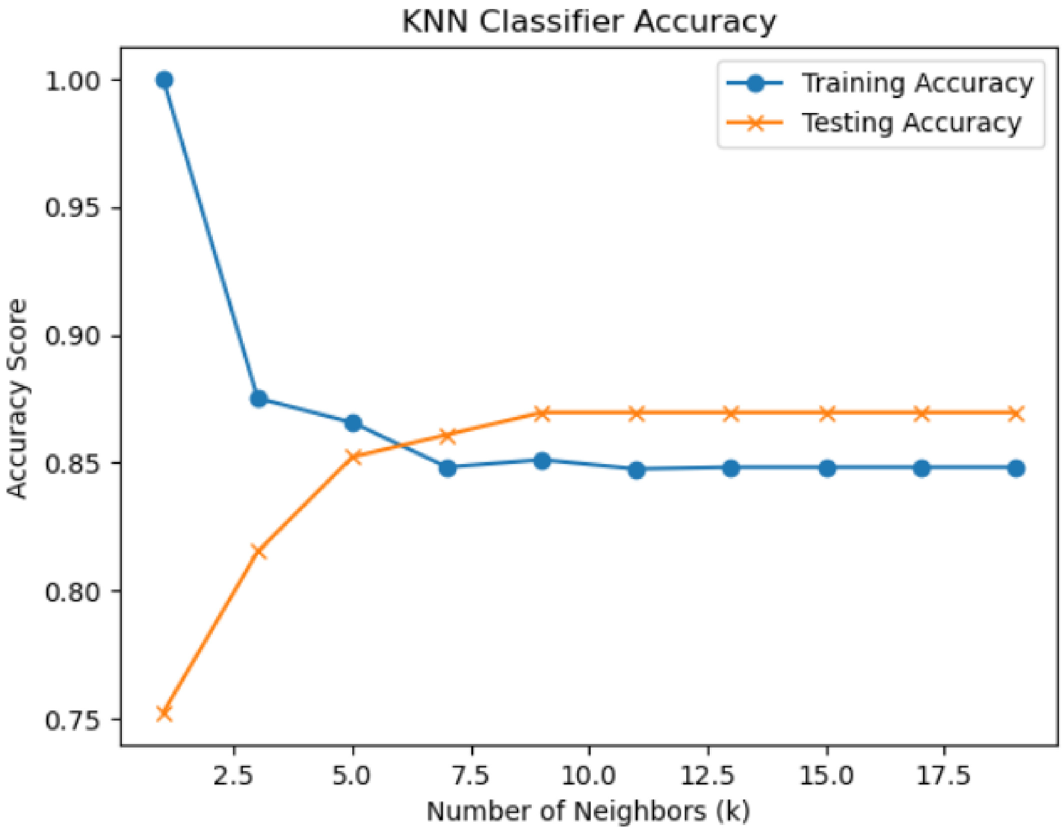

Fig. 1

Fig. 1 is a plot of k values versus accuracy scores, where the training accuracy is represented by a solid line with markers, and the testing accuracy is represented by a dashed line with markers. The plot shows that the testing accuracy reaches its maximum value at k=9, after which it starts to decrease slightly. Therefore, the optimal value of k for this model is 9.

Visualizing the correlation between various audio features of the music tracks in the dataset:

.

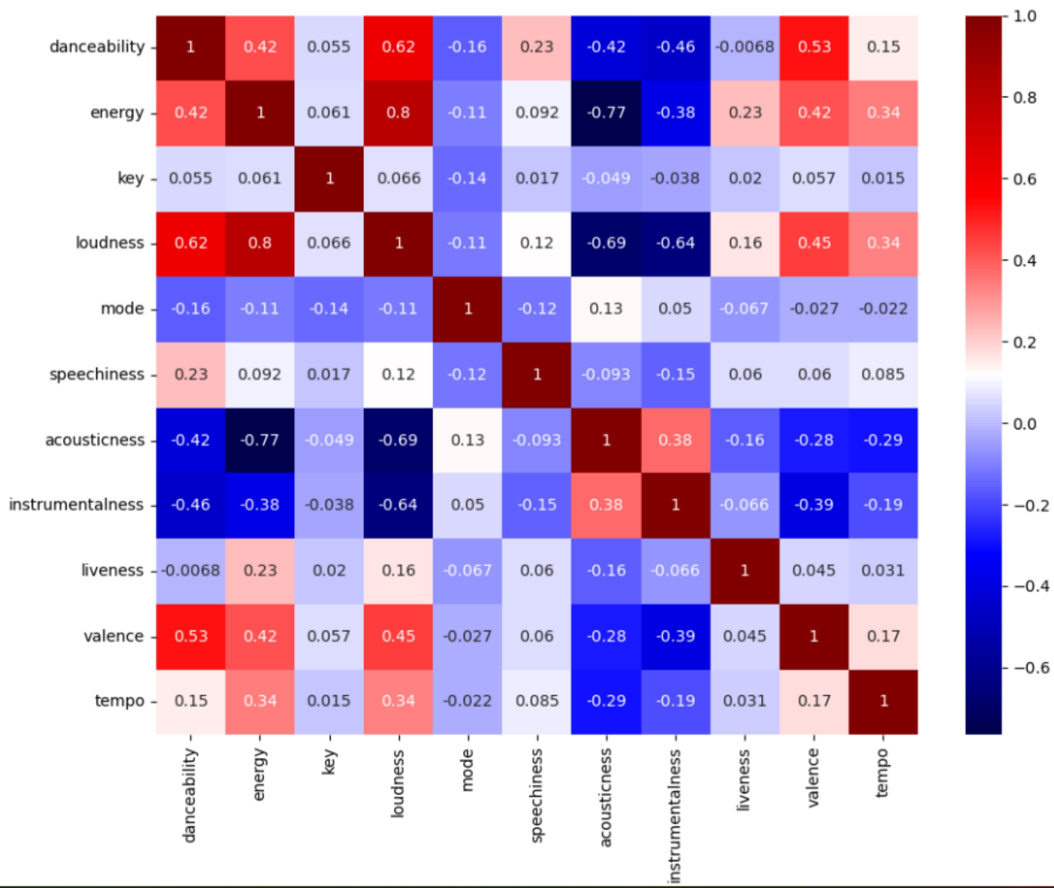

Fig. 2

In Fig. 2, the heatmap provides a visual representation of the correlation between different audio features. The audio features include danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. It identifies which features have a strong positive or negative correlation with each other.

Evaluating the performance of a trained neural network model on a dataset:

```
In [420…
model_loss, model_accuracy = model.evaluate(
    X_test_scaled, y_test_categorical, verbose=2)
print(
    f"Normal Neural Network - Loss: {model_loss}, Accuracy: {model_accuracy}")

20/20 - 0s - loss: 1.6301 - accuracy: 0.4319 - 124ms/epoch - 6ms/step
Normal Neural Network - Loss: 1.630128026008606, Accuracy: 0.4318554997444153
```

Fig. 3

The above fig. Is an output that shows the loss and accuracy of the model on the dataset. In this case, the loss value is 1.630 and the accuracy is 0.4319. The accuracy indicates the proportion of correctly classified samples in the dataset.

Conclusion and Future Work:
The models were trained on a dataset containing various audio features extracted from over several thousand songs across 16 different genres.

The project concluded that machine learning models such as K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM) can predict music genres with an accuracy of over 60%. Additionally, the project found that certain features such as tempo, loudness, and energy were more important in predicting genres than others.

The algorithms used in this project are:

Decision Tree
Random Forest
K-Nearest Neighbors (K-NN)
Logistic Regression
Support Vector Machine (SVM)
The accuracy scores of each algorithm on the given dataset are as follows:

Decision Tree: 68.25%
Random Forest: 73.67%
K-Nearest Neighbors (K-NN): 55.58%
Logistic Regression: 72.89%
Support Vector Machine (SVM): 68.03%
Based on these results, Random Forest is the best-performing algorithm with an accuracy score of 73.67%. It is followed closely by Logistic Regression with an accuracy score of 72.89%.

Future work could include exploring additional feature engineering techniques or using deep learning models to improve prediction accuracy. Furthermore, incorporating more diverse genres such as electronic or world music could also enhance the project's accuracy and applicability.