Music Genre Prediction Based on Audio Features

Problem Setting

Today we see hyper-personalization in every product in the market. The music industry does not lag, with music streaming services like Soundcloud and Spotify always looking for ways to increase user engagement with their products. This is where genre classification acts as a boon helping them identify a user's preferred genres and use it to produce recommendations by finding similar tracks and artists. Additionally, they can also identify what other genres the user may find interesting by studying behavior patterns of similar types of users.

Problem Definition

Through this project we aim to accurately identify the genre a song belongs to based on its audio features and identify relevant audio features for purpose of this classification. To begin we apply data processing techniques to the dataset and use visualization to see the distribution of the audio features. We then input audio features as predictors to our classification model to correctly identify the genre of the song.

Data Sources

The dataset has been sourced from Kaggle

https://www.kaggle.com/vicsuperman/prediction-of-music-genre

Data Description

The dataset contains 18 columns and around 50000 rows. The columns are namely: instance_id, artist_name, track_name, popularity, acousticness, danceability, duration_ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, obtained_date, valence, music_genre. There are 4 categorical columns ,1 date column and the rest are numerical columns. The list of genres present in the genre column are 'Electronic', 'Anime', 'Jazz', 'Alternative', 'Country', 'Rap', 'Blues', 'Rock', 'Classical', 'Hip-Hop'.

Data Mining Task

Data Collection and Processing

The total number of records in the dataset is 50,005 records. It has a total of 18 variables. The data types of these variables are listed below.

```
#   Column           Non-Null Count  Dtype
---  ------          --------------  -----
 0   instance_id      50000 non-null  float64
 1   artist_name      50000 non-null  object
 2   track_name       50000 non-null  object
 3   popularity       50000 non-null  float64
 4   acousticness     50000 non-null  float64
 5   danceability     50000 non-null  float64
 6   duration_ms      50000 non-null  float64
 7   energy           50000 non-null  float64
 8   instrumentalness 50000 non-null  float64
 9   key              50000 non-null  object
 10  liveness         50000 non-null  float64
 11  loudness         50000 non-null  float64
 12  mode             50000 non-null  object
 13  speechiness      50000 non-null  float64
 14  tempo            50000 non-null  object
 15  obtained_date    50000 non-null  object
 16  valence          50000 non-null  float64
 17  music_genre      50000 non-null  object
```

There are 5 records in the dataset which show null values for all variables. These records are removed from the dataset.

Numerical Variables:

We observe that variable duration_ms which represents duration of track in milliseconds contains negative values. To overcome this, we replace all '-1' with Nan values, then calculate the mean of other duration_ms records and use this mean to impute the missing records. Tempo has been wrongly misclassified as a categorical variable and has records which are missing and populated with '?'. This is imputed as, first replace all '?' with Nan values, then calculate the mean

of other tempo records, use this mean to impute the missing records and change the data type of tempo variable from object to float.

Categorical Variables:

'Artist name' has around 5% values missing in the dataset. These values cannot be imputed. There is also no major impact foreseen on the model due to this variable, the same applies for track name. 'Obtained date' has only 5 unique ids, which will not provide any significant contribution to the model. The 'instance id' is only an index and therefore will not provide any impact on the model. Therefore drop both of these columns.

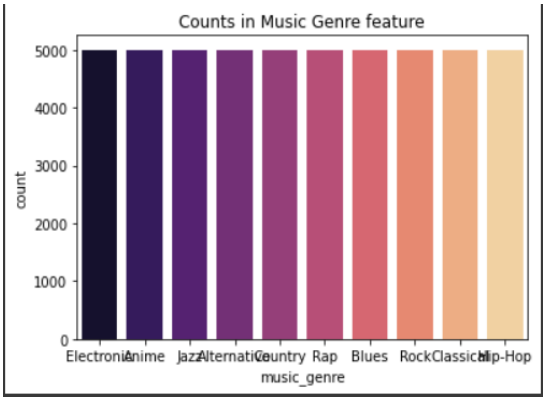
Encoding categorical variables for use in model

The technique used for modeling is one hot encoding. This technique creates dummy columns for each unique value in the variable and assigns binary code to it. This technique is applied to the Mode and Key variables which have 2 and 12 distinct values respectively.

Data Exploration

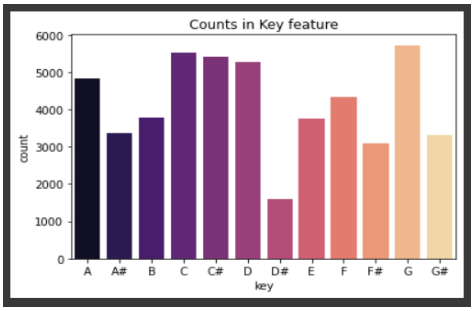
Understanding Genre:

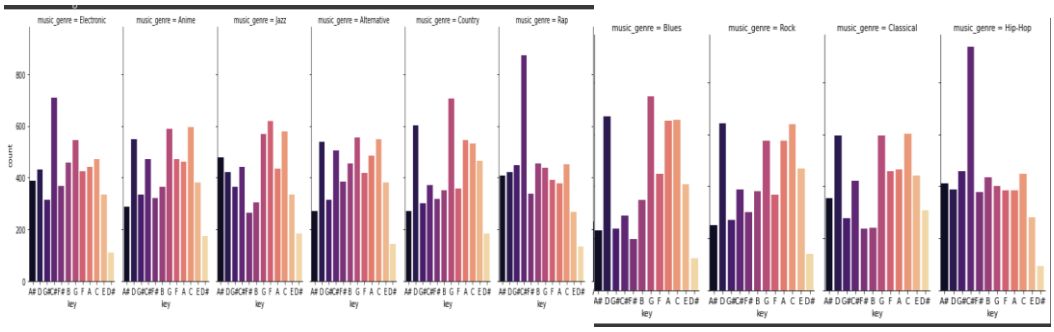
All the genres are well and equally represented, this will help avoid an unbalanced model.



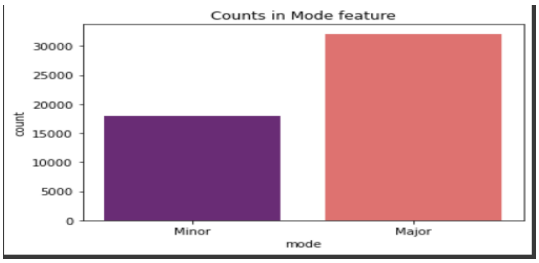
To understand the impact of each of the features on the Music Genre, by first performing univariate analysis and then bivariate analysis with the Music genre.

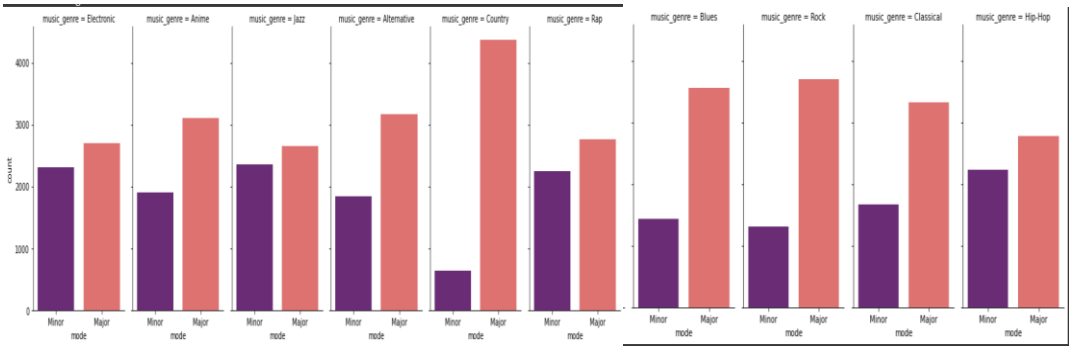
- Keys: The most frequent Key is *G* followed by *C*. *D#* has only 1500 records. There is a significant difference in keys for different genres, thus this feature after one hot encoding can be used.



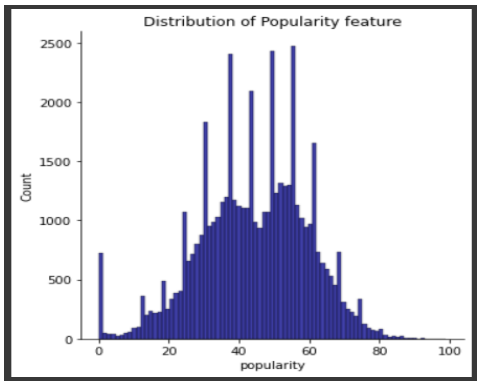


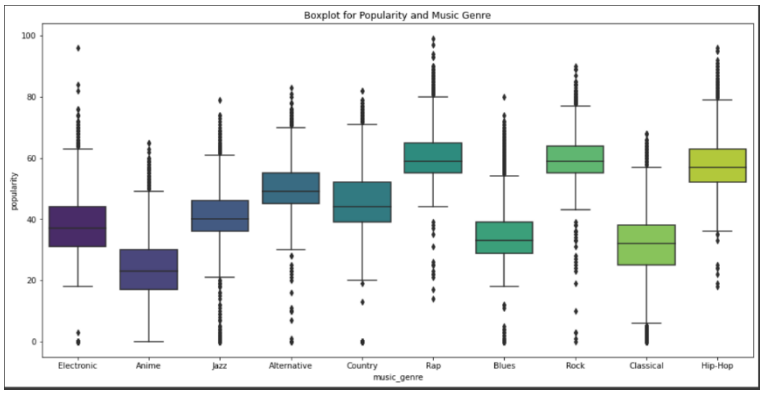
- Mode: Most songs (or melodies) are in the "major" end of the musical spectrum (see below).



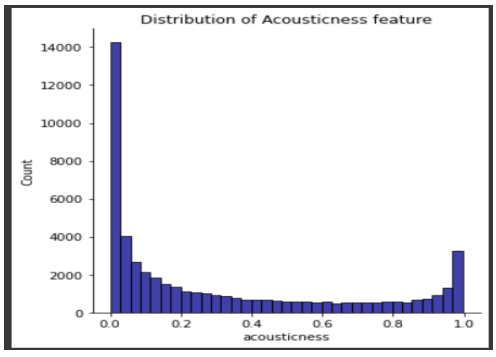


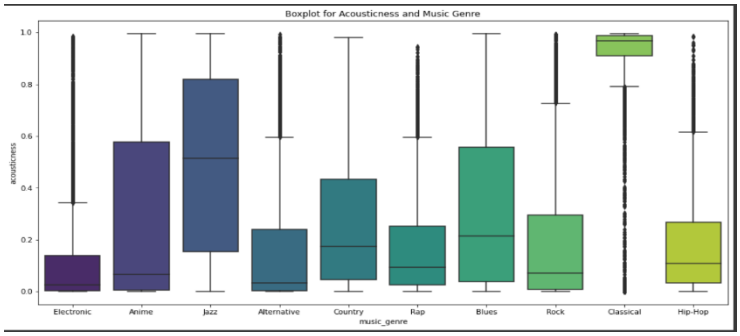
- Popularity: This feature shows a varied spread of distributions for the different genres. This will help in improving the classification. Rap, Hip-Hop and Rock are highly popular genres, while Anime, Blues and Classical are the least popular. The other 4 genres are somewhere in between.



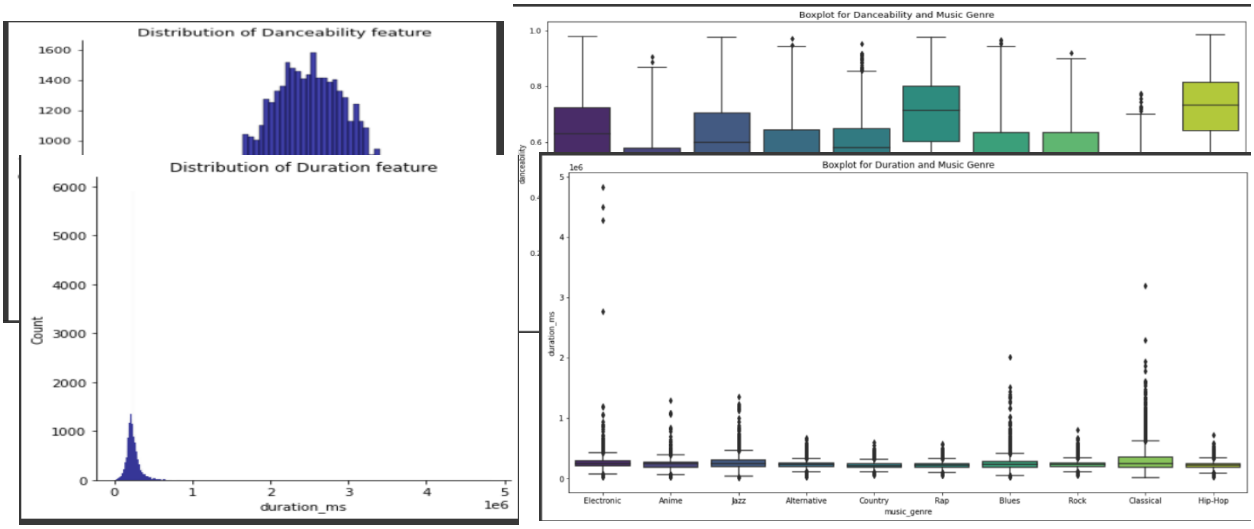


- Acousticness: Classical music is easily distinguishable as compared to other genres. Jazz also shows unique behavior.



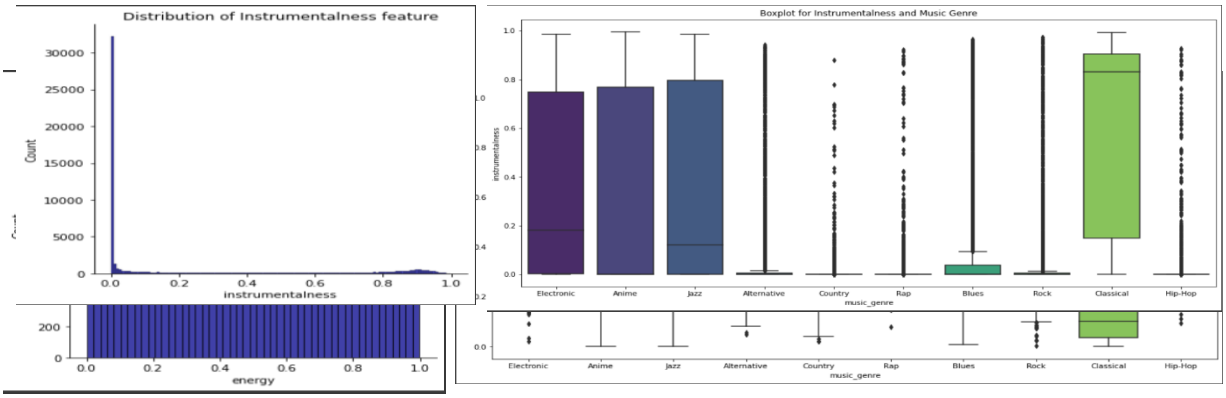


- Danceability: Classical music is once again different from other genres. Rap and Hip-Hop show similar behavior.

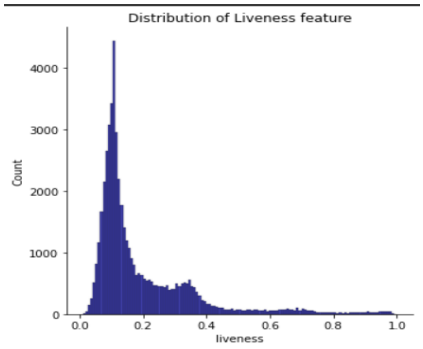


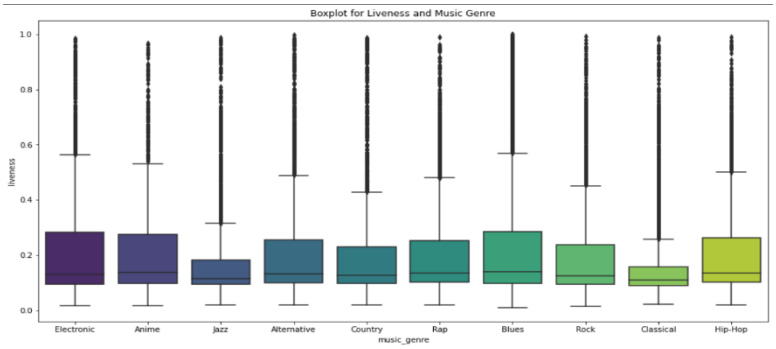
- Duration Milliseconds: Duration is highly skewed. It also shows extreme outliers which might impact model performance
- Energy: Energy shows a good distribution for the different genres. Classical is again more differentiable than the rest. Anime also shows a differentiable pattern.

- Instrumentalness: Instrumentalness shows a large number of 0.0 entries that indicates missing values rather than real data points. These records account for 1/3rd of the data. Imputing these may not give us the correct outcome. Therefore we can discard this feature.

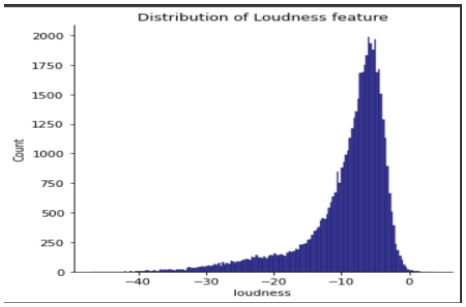


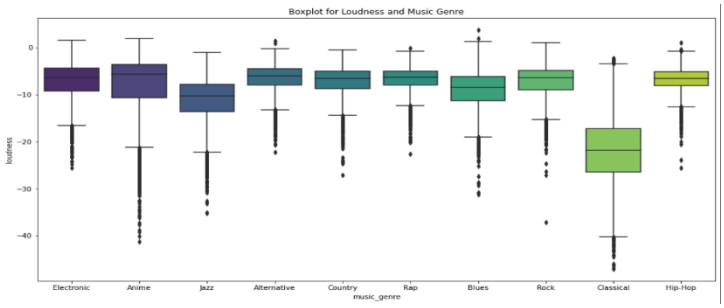
- Liveness: The distributions seem similar for all genres, so this feature will likely not contribute much to the classification. We can try running the model without incorporating the feature.



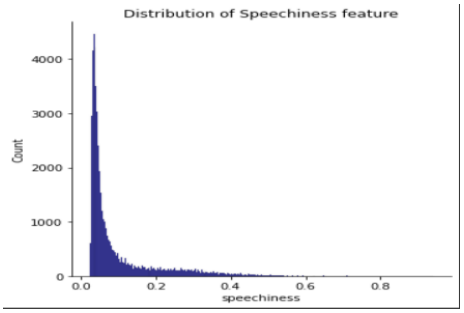


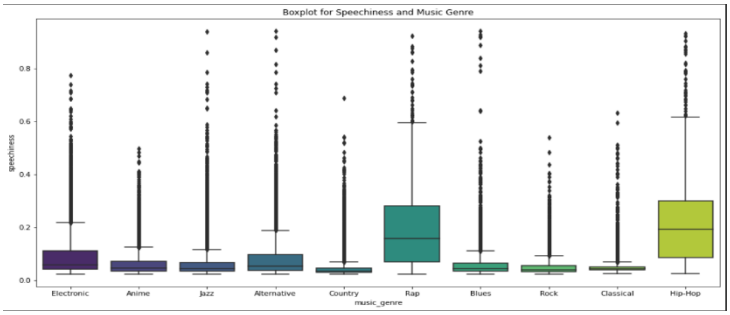
- Loudness: Apart from classical, we notice differentiation in the distribution of Blues and Jazz genres as well.



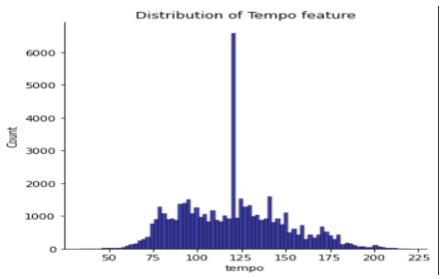


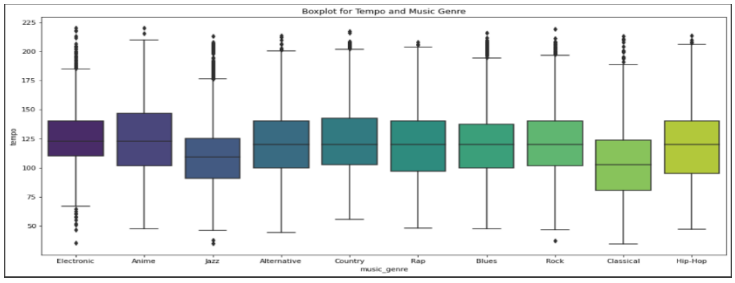
- Speechiness: Rap and Hip-Hop show the most differentiation in this feature.



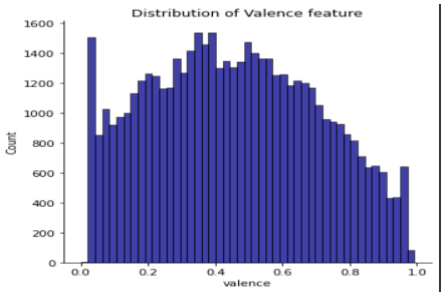


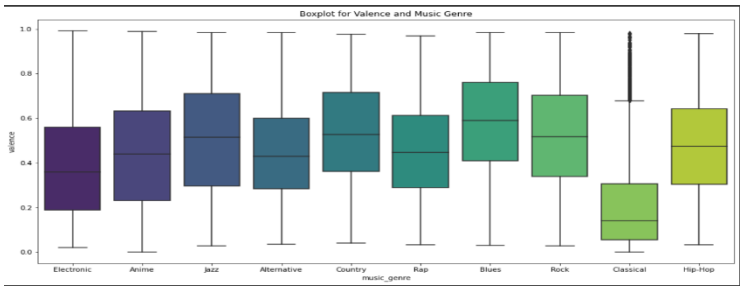
- Tempo: We do not see significant variation between genres. We can consider dropping this feature.



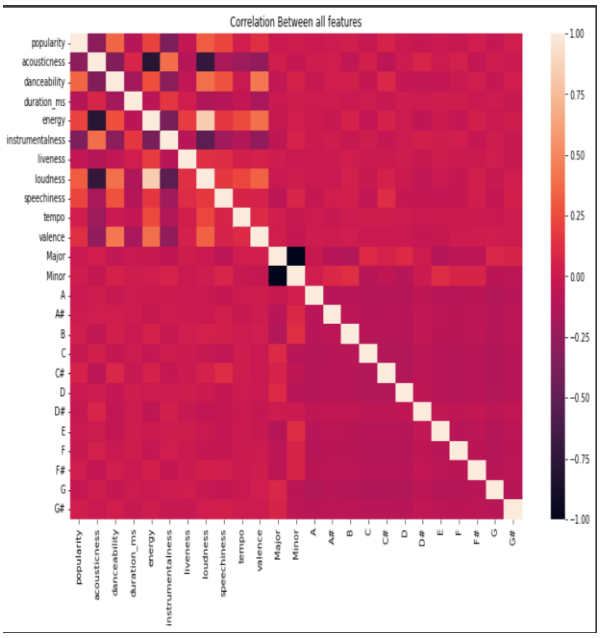


- Valence: Electronic, blues, and classical music have the highest valence difference among the genres.





Correlation between features: We observe high correlation between Energy, Acousticness and loudness. These can be useful through PCA to reduce Dimensionality of our dataset.



Data Mining Models/Methods

Model Exploration and Model Selection

Classification techniques need to be used to predict the music genre as the target column(music_genre) is a categorical column. The music genres are categorized as Electronic, Anime, Jazz, Alternative, Country, Rap, Blues, Rock, Classical, Hip-Hop.

Classification is data mining technique that categorizes data into a class or category. The different classification algorithms that could be used are as follows:

- Decision Tree:
    - Decision tree is a tree-structured classifier where internal nodes represent the features of data, branches represent decision rules, leaf node represents outcome.
    - It shows, all possible solutions to a problem based on the given conditions
    - Advantage of decision tree is that is does requires less effort in data preparation while preprocessing and no requirement of normalization/scaling.
    - Disadvantages of the technique include longer time to train the model and complexities in execution when small change is made to the data can lead to a large change in the structure of the decision tree causing instability.
- K-Nearest Neighbors:
    - K-NN takes the similarity between the new case or data and available cases. It then puts the new case into the category that is most like the available categories.
    - It basically classifies the new cases by a majority vote of its k neighbors. The distance is measured by a distance function like Euclidean, Manhattan, Minkowski, and Hamming.
    - The advantage of K-NN is that easiest to implement with lesser requirement for hyperparameter tuning.
    - It gets slightly slower as the number of predictors/independent variables increase.
- Naïve Bayes:
    - Naïve Bayes is based on *Bayes' theorem*. In other words, a **Naive Bayes classifier** assumes that that every feature is independent of each other and that all the features contribute equally to the outcome.
    - It is mainly used when a high-dimensional training dataset.is requires like *text classification*.
    - Naïve bayes technique is extremely fast compared to other methods and requires a small amount of training data to estimate the parameters.
    - The main disadvantage of this technique is that it is hard to find a set of entirely independent set of variables.

- Logistic Regression:
    - Logistic Regression predicts the probability of occurrence of an event by fitting data to a *log function***.** It tries to find a best-fitting relationship between the dependent variable and a set of independent variables. The values obtained always lie within 0 and 1 since it predicts the probability.
    - It helps understand the impact of multiple independent variables on a single outcome variable.
    - It is a very simple and efficient technique.
    - The major disadvantage of logistic regression is that it is difficult to handle large number of categorical columns. It moves forward with an assumption that data is free of missing values and predictors are independent of each other.
- Support Vector Machine:
    - SVM creates a decision boundary that divides n-dimensional space into classes by large gaps. New data points are then mapped into the same space, and their categories are predicted according to the side of the gap they fall into.
    - SVM works very well with high dimensionality spaces and when there is a clear line of separation between classes.
    - It does not perform well when target classes are overlapping and with large datasets.

The K-NN and Decision tree techniques were implemented on the standardized dataset before and after PCA. It was observed that PCA reduced the accuracy of the models.

Implementation of Selected Models

Model Name	Parameters	Accuracy
Decision Trees	Criterion : Gini	44.07
	Criterion : Entropy	43.56
	Criterion : Entropy Depth : 11	52.34
	Criterion : Entropy Depth : 11 Random State : 8	52.43
KNN	N = 5	45.47

	N= 25	48.25
	N= 35	47.91
Random Forest	N_estimators : 25 Criterion : Entropy max_depth : 13 Random State=0	55.45
	N_estimators : 30 Criterion : Entropy Max_depth :13 Random State : 0	56.07
	N_estimators:100 Criterion: Entropy max_depth : 13 Random State : 0	56.36
Logistic Regression	Penalty : L2 Solver : lbfgs	52.8
	Penalty : L2 Solver : liblinear	50.66
SVM	Default parameters	56.04

Performance Evaluation

We observed very poor performance from all models in our previous step. In this milestone we took following steps to improve this performance.

1. Eliminate features having low impact on classification
   On the basis of visualization created in previous milestone we decided to retain only the following features: Popularity, Danceability, Energy, Liveness, loudness, tempo, valence and ignore the rest.
   - DecisionTreeClassifier(criterion='entropy',max_depth = 11, random_state=0) : Accuracy : 46.71
   - KNeighborsClassifier(n_neighbors=10) : Accuracy : 46.58
   - RandomForestClassifier(n_estimators= 100,criterion = 'entropy',max_depth=13, random_state=0) : Accuracy : 49.4

- LogisticRegression(penalty='l2',solver='liblinear',random_state=0): Accuracy : 44.69
- SVC() Accurcay : 51.59

  We observe that there is no improvement in accuracy by selecting only optimal features.

2. Eliminate Features based on feature selection
   We use the selectKbest function for this which uses f_regression score. It gives us 5 best features as following : 'popularity' 'acousticness' 'energy' 'loudness' 'speechiness'. We tested this only on two models:
   - SVC() Accuracy : 49.1%
   - LogisticRegression() : 43.97%

   Clearly this approach is also not improving the model performance.

3. Grouping together Genres that have similar accuracy performance
   The third approach we tried is to group similar genres together. This grouping was done basis the visualization in previous milestones and also taking into account the classification performance of model for each of these genres. We created following groups: 'All', 'Anime', 'Classical', 'Rap and Hip-Hop'

   a. SVC()

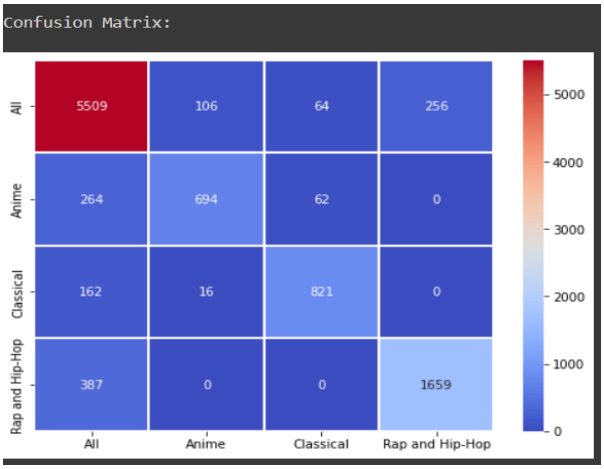
```
Train accuracy score: 87.26%
Test accuracy score: 86.83%

Classification Report for the test set:

                precision    recall  f1-score   support

           All       0.87      0.93      0.90      5935
         Anime       0.85      0.68      0.76      1020
     Classical       0.87      0.82      0.84       999
Rap and Hip-Hop       0.87      0.81      0.84      2046

      accuracy                           0.87     10000
     macro avg       0.86      0.81      0.83     10000
  weighted avg       0.87      0.87      0.87     10000
```



b. RandomForestClassifier(n_estimators= 100,criterion = 'entropy',max_depth=13, random_state=0)

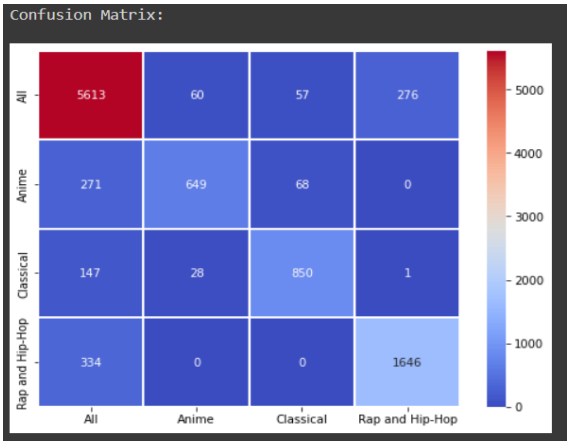
```
Train accuracy score: 93.64%
Test accuracy score: 87.58%

Classification Report for the test set:

                precision    recall  f1-score   support

           All       0.88      0.93      0.91      6006
         Anime       0.88      0.66      0.75       988
     Classical       0.87      0.83      0.85      1026
Rap and Hip-Hop       0.86      0.83      0.84      1980

      accuracy                           0.88     10000
     macro avg       0.87      0.81      0.84     10000
  weighted avg       0.88      0.88      0.87     10000
```



c. LogisticRegression()

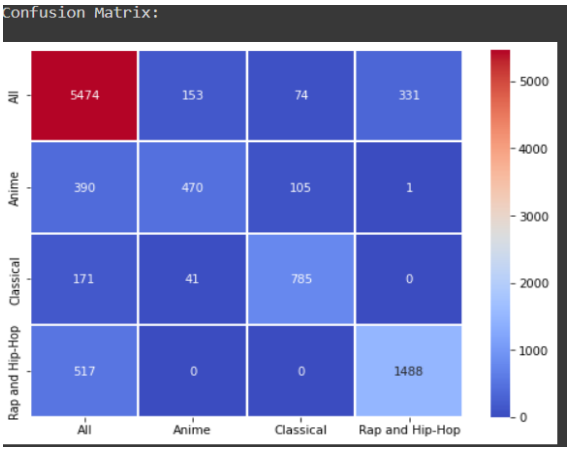
```
Train accuracy score: 81.86%
Test accuracy score: 82.17%

Classification Report for the test set:

                precision    recall  f1-score   support

           All       0.84      0.91      0.87      6032
         Anime       0.71      0.49      0.58       966
     Classical       0.81      0.79      0.80       997
Rap and Hip-Hop       0.82      0.74      0.78      2005

      accuracy                           0.82     10000
     macro avg       0.79      0.73      0.76     10000
  weighted avg       0.82      0.82      0.82     10000
```



d. DecisionTreeClassifier(criterion='entropy',max_depth = 11, random_state=0)

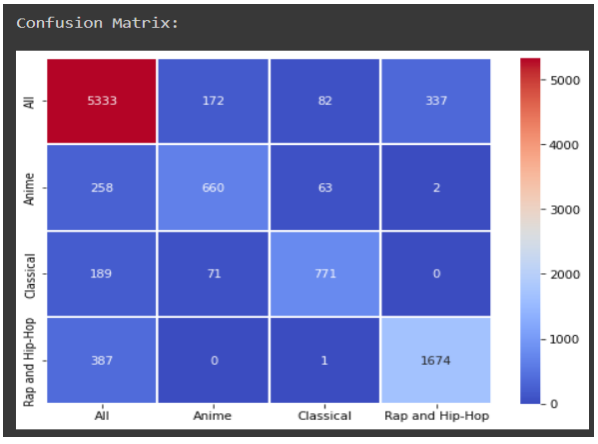
```
Train accuracy score: 90.03%
Test accuracy score: 84.38%

Classification Report for the test set:

                precision    recall  f1-score   support

           All       0.86      0.90      0.88      5924
         Anime       0.73      0.67      0.70       983
     Classical       0.84      0.75      0.79      1031
Rap and Hip-Hop       0.83      0.81      0.82      2062

      accuracy                           0.84     10000
     macro avg       0.82      0.78      0.80     10000
  weighted avg       0.84      0.84      0.84     10000
```



e. KNeighborsClassifier(n_neighbors=10)

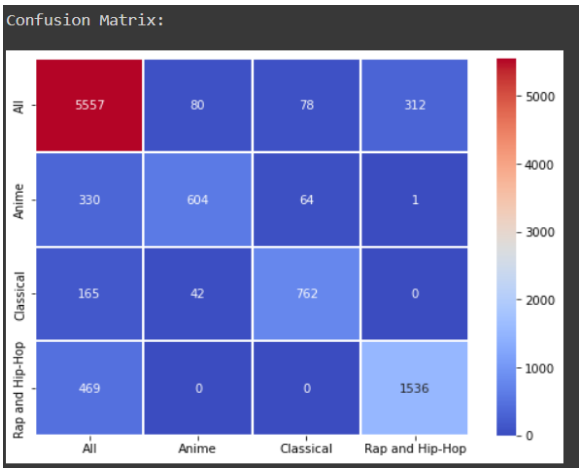
```
Train accuracy score: 86.78%
Test accuracy score: 84.59%

Classification Report for the test set:

                precision    recall  f1-score   support

           All       0.85      0.92      0.89      6027
         Anime       0.83      0.60      0.70       999
     Classical       0.84      0.79      0.81       969
Rap and Hip-Hop       0.83      0.77      0.80      2005

      accuracy                           0.85     10000
     macro avg       0.84      0.77      0.80     10000
  weighted avg       0.84      0.85      0.84     10000
```



There is a significant improvement in the classification report once we group the genres together. Random Forest and SVM give us the best performance.

Model Name	Parameters	Accuracy
Decision Trees	Criterion : Gini max_depth : 11 Random State : 0	84.38
KNN	N = 10	84.59
Random Forest	N_estimators:100 Criterion: Entropy max_depth : 13 Random State : 0	87.58
Logistic Regression	Default parameters	82.17
SVM	Default parameters	86.83