Multi-Classification of Music Genre

Mingjun Ma
Brown University DSI
https://github.com/MingjunMa/final_project

Introduction

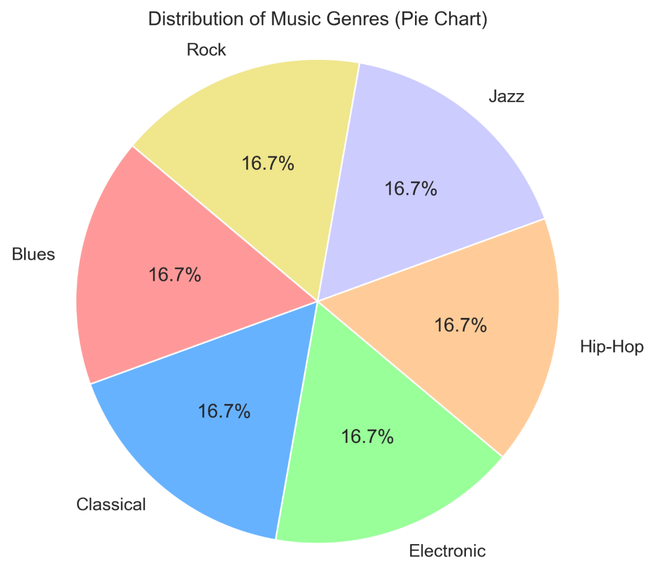
Classifying music into genres has traditionally been subjective, posing challenges in music recommendation and academic research[1]. Especially in an era where more and more new songs are being created, the lines are starting to blur for many music genres. With advancements in machine learning and data analytics, we now have the opportunity to approach music genre classification in a more objective and data-driven manner.

Spotify, an online music streaming platform, provides developers with a wealth of quantitative data for each song, including artist_name, track_name, popularity, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence, duration (ms), key, mode, tempo, obtained_date, and music_genre[2]. From these, we selected 3,000 data via Spotify's API, which covered six music genres: blues, classical, electronic, hip-hop, jazz, and rock[3]. By employing advanced data analysis and machine learning techniques to classify the music genres, we aim to overcome the limitations of subjective genre categorization, enhancing user experience in digital music platforms and providing valuable insights for musicologists and the music industry.

EDA

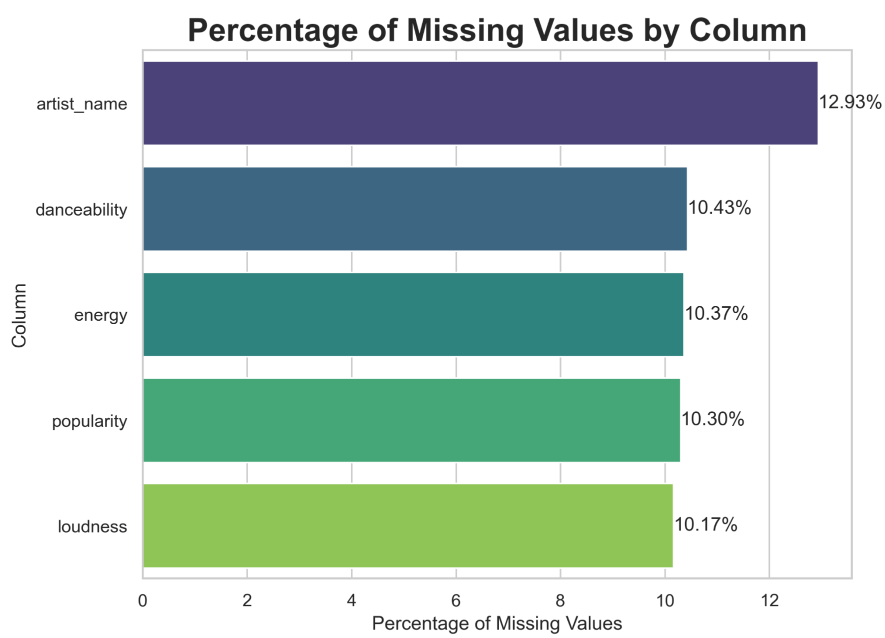
We did the descriptive analysis with several visuals to understand the data we would analyze further.

We first created a pie chart about the distribution of the target variable(fig.1.). From the pie chart, we can see that the selection of music genres is balanced, with approximately 500 data points per genre.



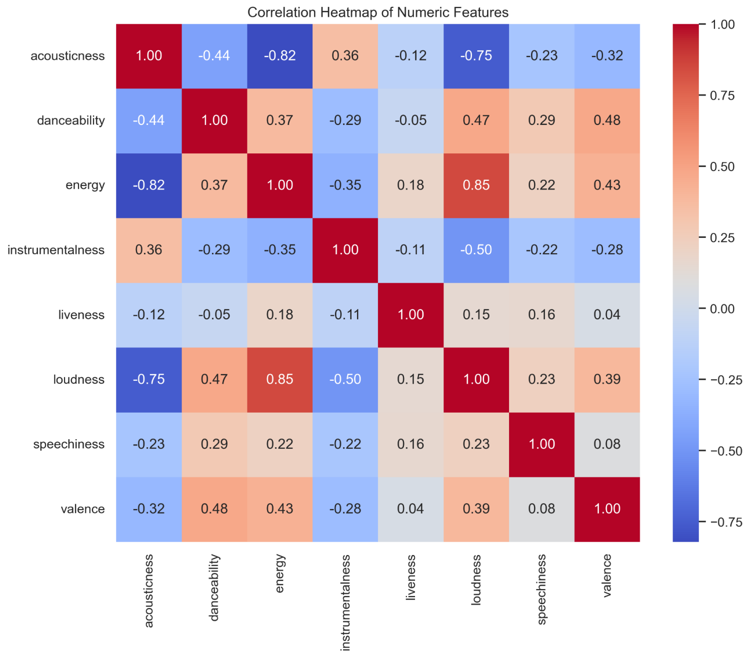
(fig.1. Pie chart of distribution of music genres)

We also created a bar chart to examine the missing values in the dataset(fig.2.). From the figure, we can see that there is missing data in five variables: airtist_name, danceability, energy, popularity, and loudness, where the missing values account for about 10% of each variable.



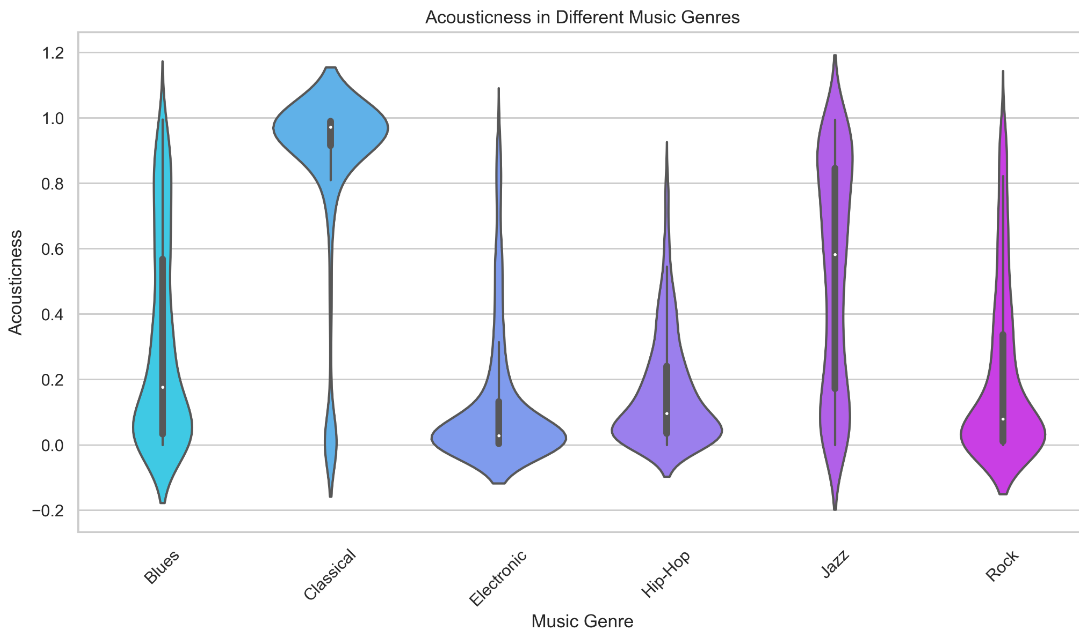
(figure.2. Bar Plot of percentage of missing values by features)

To check the relationship between features, we created a correlation heatmap(fig.3.). The heatmap shows that there's a strong positive correlation between the energy, which represents a perceptual measure of intensity and activity, and the loudness of a track in decibels (dB) (r=0.85). Also, we can see a strong negative relationship between the acousticness and those two features (r=-0.82, r=-0.75), indicating that a song high on acousticness usually has lower energy and loudness.



(figure.3. Correlation Heatmap of Numeric Features)

Moreover, we provided a violin plot that visualizes the distribution of acousticness across different music genres (fig.4.). The classical genre shows a wide distribution of acousticness, with a substantial body at higher acousticness values, suggesting that classical music generally has more acoustic properties than other genres. The blues and jazz genres have a narrower distribution centered around a medium level of acousticness. This may reflect a balance between electronic and acoustic instruments typically found in blues and jazz music. The other electronic, hip-hop, and rock genres have many tracks with lower acousticness, implying they often contain more electronic or amplified sounds.



(figure.4. Violin plot of acousticness in different music genres)

Methods

We can start to build our machine-learning pipelines with a clearer understanding of the data. We will try four different classifiers: logistic regression, random forest, SVC, and XGboost Classifier.

First, we dropped features without predictive powers, including artist_name, track_name, and obtained_date.

Then, since the target variable is balanced, we did a basic train-test split where 20% of the data was used for testing, and the remaining 80% was used for cross-validation. To increase the reproducibility of the splitting, we set a random seed to be 10.

Next, we want to preprocess our data. For the categorical feature, "mode" represents a track's modality (major or minor), and we use the one-hot encoder to encode it. The keys of each song can then be thought of as ordinal variables, and we encoded the keys using the ordinal encoder. For the rest of the numerical features, we used the standard scaler. For the missing values, we used the IterativeImputer.

After preprocessing, we tried the RandomizedSearchCV with n_iter=100 and the GridSearchCV with KFold=4. Unlike the GridSearchCV, which exhaustively tries all possible combinations, RandomizedSearchCV works by randomly selecting combinations of hyperparameters for a model from specified ranges. After comparing, we noticed that for our dataset and model, the

RandomizedSearchCV has a similar performance but requires significantly less runtime than the GridSearchCV.

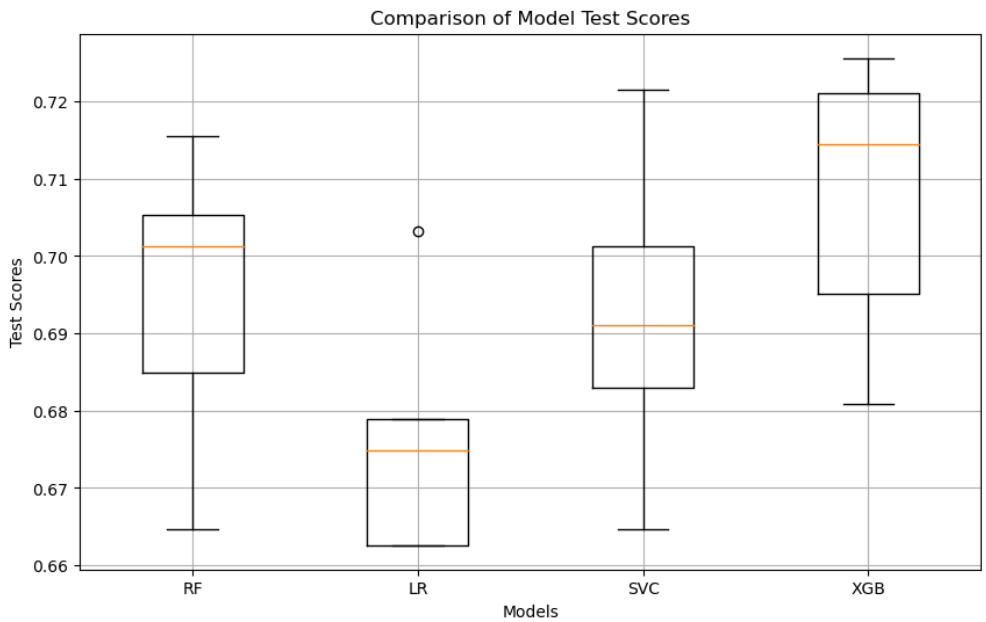
The following table shows the hyperparameters we tried to tune for each classifier (fig.5.). We choose accuracy as the evaluation metric since the target variables are distributed balanced.

Logistic Regression	C': [0.01, 0.1, 1, 10, 100], penalty': ['l2'], solver': ['saga']
Random Forest	n_estimators': [100, 200, 300], max_features': ['auto', 'sqrt'], max_depth': [5,10,20,30,100], min_samples_split': [2, 5, 10], min_samples_leaf': [1, 2, 4], bootstrap': [True, False]
SVC	C': [0.1, 1, 10], kernel': ['linear', 'rbf', 'poly'], gamma': ['scale', 'auto']
XGBoost Classifier	n_estimators': [10000], learning_rate': [0.01, 0.1, 0.2], max_depth': [3, 6, 10], subsample': [0.7, 0.8, 0.9], colsample_bytree': [0.7, 0.8, 0.9]

(figure.5. Hyperparameter tuning)

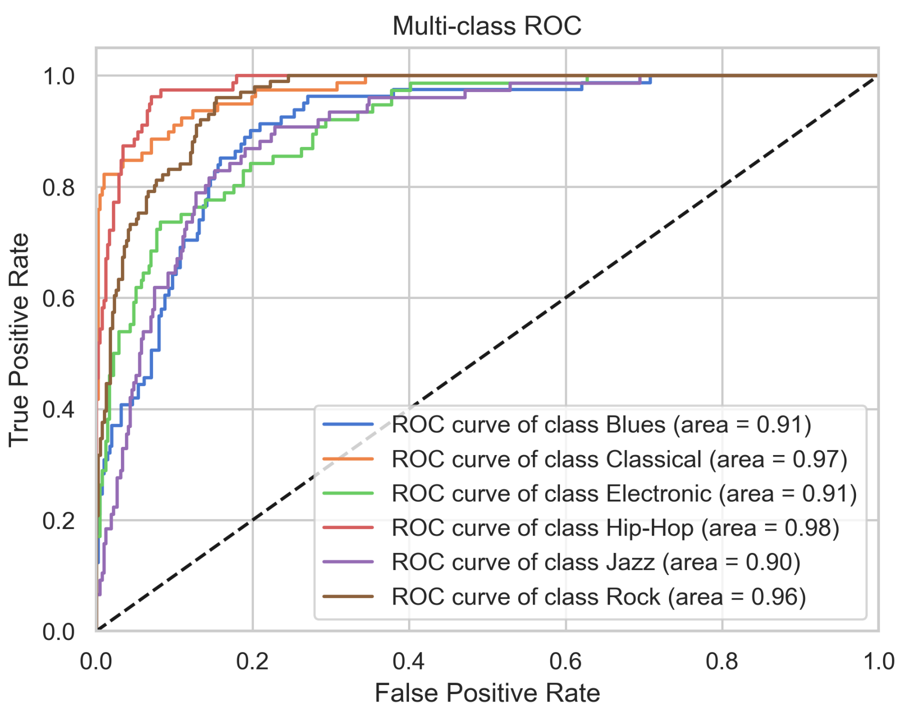
Results

We repeated the cross-validation for ten different random states for each classifier, and the boxplot below shows the distribution of accuracy scores for each model (fig.6.). As we can see, the average accuracy score for the models is around 0.7, while the baseline accuracy is about 0.167. The XGBoost classifier has a higher average accuracy than the other models. On the other hand, the Logistic regression model has the lowest accuracy score on average, but it also has the lowest standard deviation on accuracy. We can conclude that the XGBoost Classifier is more predictive than the rest of the models.



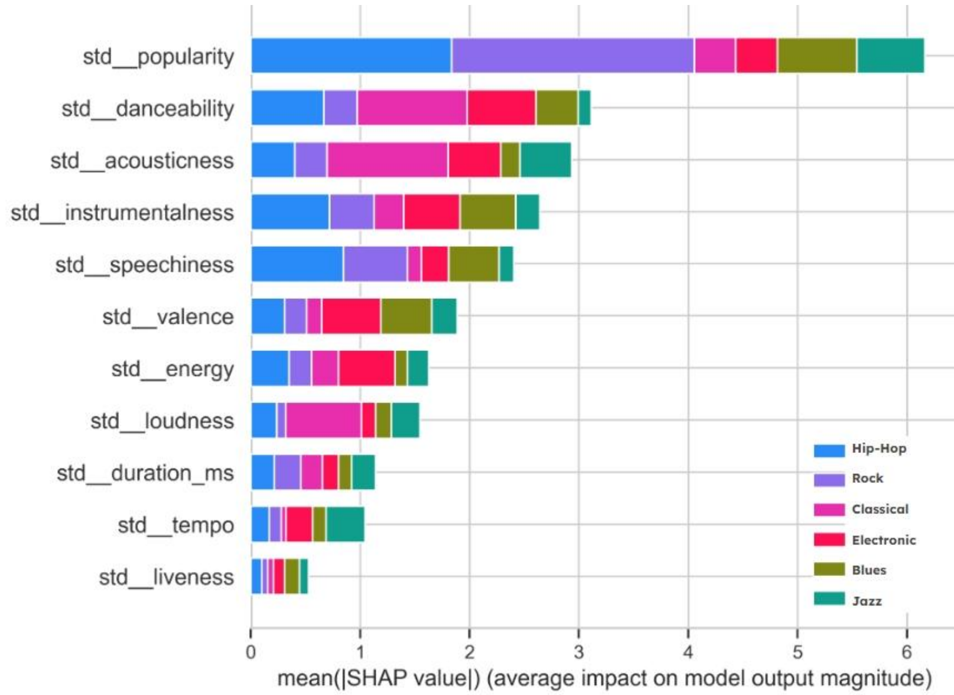
(figure.6. Boxplot of Accuracy for each model)

To check the model's ability to classify different music genres even further, we chose one of the best-performing models of random forest to plot a multi-class ROC plot (fig.7.). The plot shows that the AUC value is greater than 0.9 for all music genres. Also, the model did a great job predicting hip-hop, classical, and rock music, as the AUC value for those three genres is greater than 0.95.



(figure.7. Multi-class ROC curve for best random forest model)

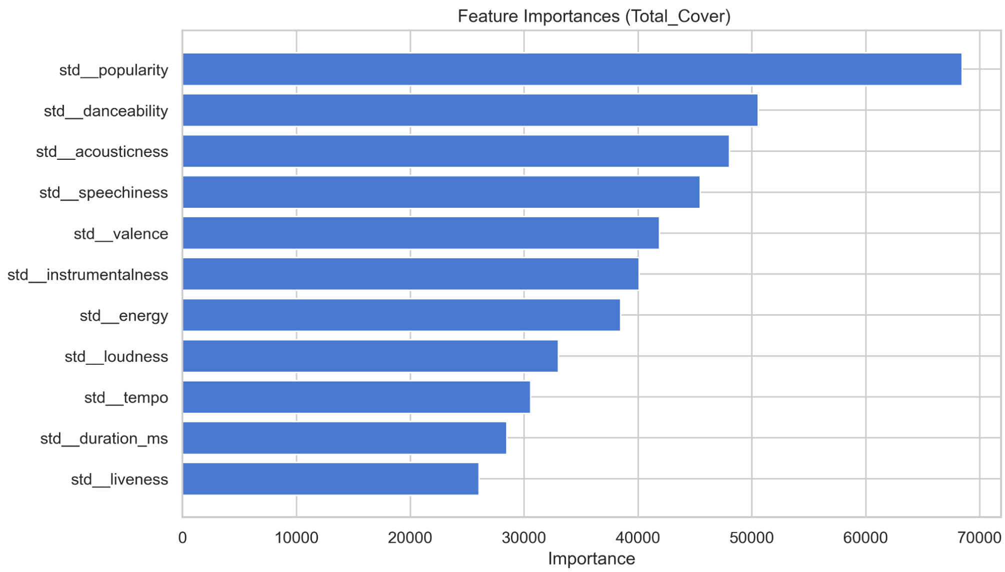
After evaluating the model's overall performance, let's examine which variables are essential to the prediction aspect. The plot below shows the feature importance by mean SHAP value (fig.8.). From the plot, we can see that the popularity, danceability, and acousticness are the top-3 most globally important features. On the other hand, the liveness, temp, and duration are relatively not important predictors.



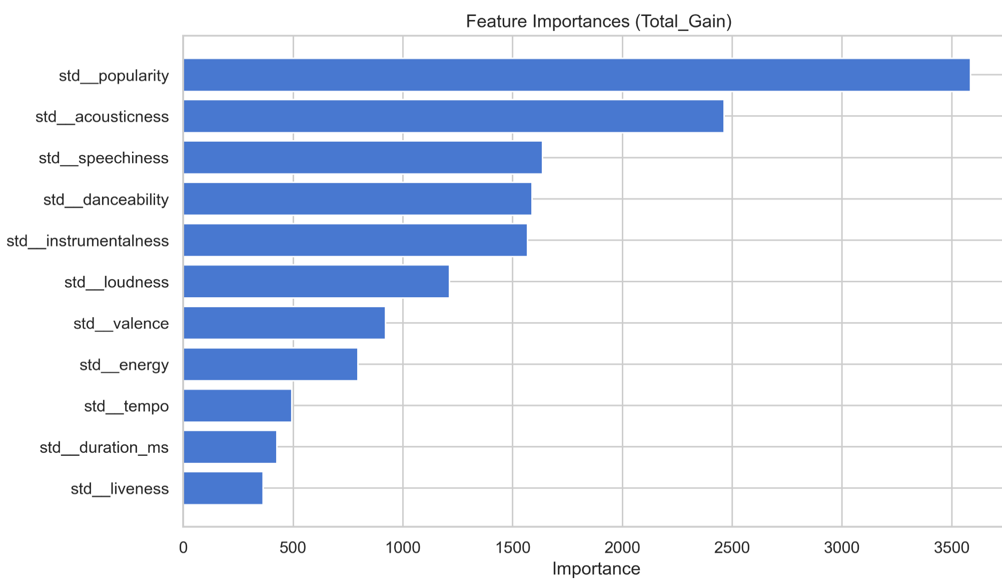
(figure.8. Feature Importance by SHAP values)

Moreover, we plotted two additional plots of importance using XGBoost's two different metrics: Total_Cover(fig.9.) and Total_Gain(fig.10.). Although these plots give different results using different metrics, we can still summarize that popularity, danceability, and acousticness are the most important features for the prediction. At the same time, the liveness, duration, and temp are less important features.

Several potential reasons could explain the results we get. The popularity often reflects how well the public receives a song and can correlate with specific genre trends. For instance, some genres may be more mainstream or commercially successful, and songs within these genres have higher popularity scores. Since the current cultural zeitgeist can influence popularity, it can strongly indicate genre. The acousticness means whether a track was made with acoustic instruments or has an acoustic sound. Different genres have varying degrees of acousticness; for example, folk or classical music tends to have higher acousticness than electronic or rock music. This feature can significantly influence genre classification as it is directly related to the instrumental makeup of a track.

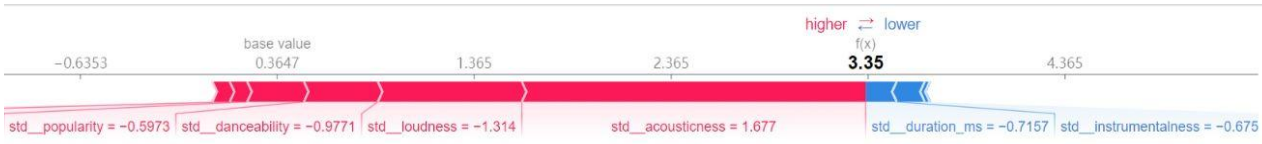


(figure.9. Feature Importance by XGBoost Total_Cover)



(figure.10. Feature Importance by XGBoost Total_Gain)

Below is the SHAP force plot for a data point being predicted as classical music with high probability(fig.11.). We can see the high value of acousticness and low value of loudness and danceability contribute positively to the prediction.



(figure.11. SHAP forceplot for index 0)

And for a data point unlikely to be predicted as classical music(fig.12.), the low value of acousticness and high value of popularity and loudness contributed negatively to the prediction.

(figure.12. SHAP forceplot for index 100)

Outlook

We can do several things to improve the model's applicability and performance.

We can increase the variety of music genres. As the variety increases, the model will be more challenging to categorize. We can observe how the model performs under complex tasks and use this to adjust our model further. On the other hand, we can use the K-Nearest Neighbors classification,CNN, and LGBM models to classify the data[4]. By comparing the performance between different models, we can choose the more suitable model for our dataset.

Reference

[1]https://scholarworks.calstate.edu/downloads/73666b68n
[2]https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation
[3]https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre/data
[4]https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn