Music Genre Classifier

Abbiram Ramanathan (a6ramana)
Pranavan Pirahalathan (ppirahal)
Rishan Ratnasivam (r2ratnas)
Sathursan Santhirakumar (s5santhi)
Suvasan Krishnasamy (s63krish)

Objective

For our project, we chose to perform an empirical evaluation where we collected data and proposed a classifying solution. We created a music genre classifier that leverages machine learning techniques to determine the genre of a given song. Currently, Spotify does not track genre information on a song-by-song basis and only displays genres for each artist. Our project aims to offer users a reliable tool for automatically classifying the genres of their songs on the Spotify platform.

Motivation

1 Objectivity

The classification of music genres is inherently subjective and complicated by the evolution of genres over time. Having a model that can use machine learning would make this process much more objective.

2 Education

Being able to classify diverse music into smaller genres will enhance the listeners' experiences by helping them discover new music and will serve as a valuable educational tool for studying music theory through genre analysis, which promotes a higher appreciation and understanding of different cultures of music.

3 Artist Exposure

With a music genre classifying machine learning model, users will be able to find new artists that classify
under their favourite genres. Smaller artists benefit by being able to be found under similar genres to bigger artists, increasing their exposure and their audience.

Data Collection

Identify Playlists

Identified **56 different** playlists on Spotify, consisting of **9** playlists across **6** genres. Ensured minimal overlap between playlists to provide a diverse selection of songs within each genre.

Connect to Spotify API

Used the API to download **music metadata** (danceability, acousticness, etc.), artist genres, and **concurrently downloaded** audio MP3 files using the fetched URL for a 30-second snippet of each song.

Extract Audio Features

Extracted the mean and standard deviation of **advanced audio features** (such as MFCC, Tonal, Chroma, Zero Crossing Rate, and other spectral audio metrics) using the **Librosa** audio library.

Our Raw Dataset

Merged the Spotify metadata dataset with the advanced audio features dataset to generate a final raw dataset consisting of **~15,000 rows** and **85 unique predictors.**

Data Preparation

Clean Data Points

Dropped rows with missing **preview_urls** or null values and deduplicate any songs that appear across multiple different playlists

Encode Features

Encoded categorical features using various techniques: Key was **One Hot Encoded**, Artist Genre was **Count Vectorized**, and Song Genre was **Label Encoded**.

Remove Outliers

Removed data across all columns where the values are greater than or less than the mean plus **3.5 times the standard deviation**

Data Preparation

Data Partitioning

Partitioned dataset containing 3316 observations using an **80% train and 20% test split**. Stratified the partitions to ensure the distribution of each class is consistent between train and test sets.

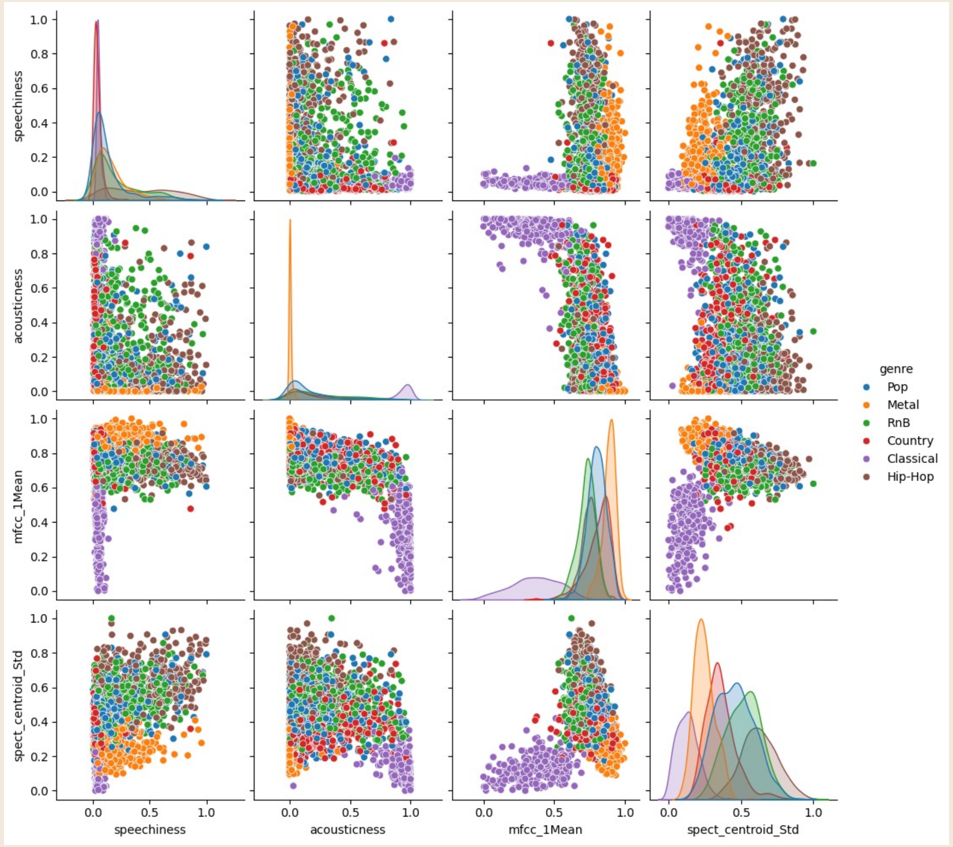
Feature Scaling

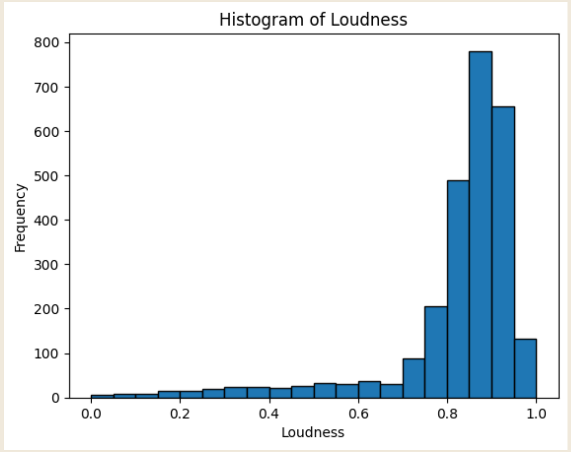
Applied a Min-Max Normalization to 81 continuous features to rescale between 0 and 1. As most features do not have a normal distribution, the **MinMaxScaler from the sklearn package** was an appropriate choice for feature scaling.

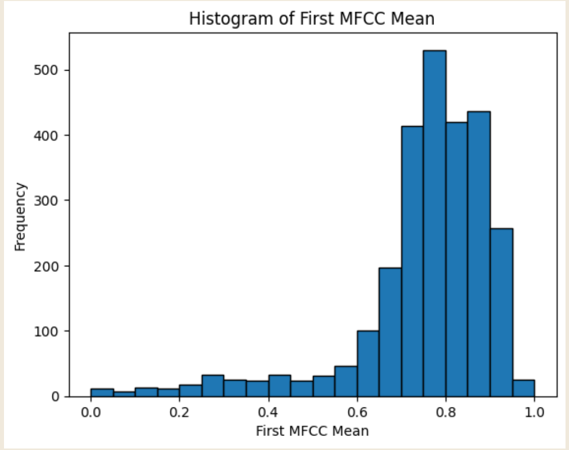
Data Visualization

Plotted a **pairplot using the seaborn package** to compare distributions of 4 different features (speechiness, acousticness, mfcc_1 mean and spect_centroid_std) across all classes

Data Visualization







Machine Learning Methods

Random Forest

Uses an ensemble learning method by building multiple decision trees and combining their results to determine the classification of data. **Great for capturing non-linear and complex relationships.**

Gradient Boosting

Uses an ensemble learning technique where the results of multiple weak decision tree learners are combined and optimized based on errors from previous iterations. **Great for non-linearity and complex trends.**

Support Vector Machine

Uses kernel tricks by converting non-linear lower dimension space to a higher-dimension space. **Supports non-linearity** and **great for feature sets that are not high relative to dataset size**.

Training Procedure

1 Feature Selection

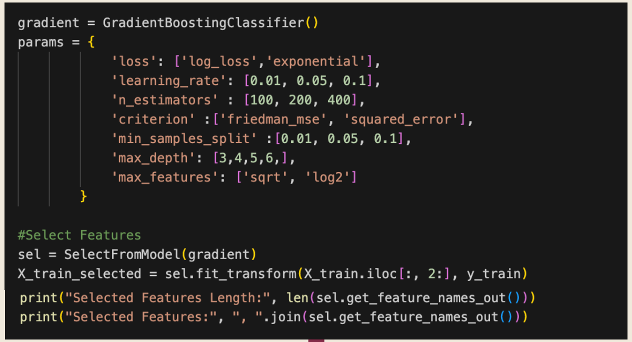
Used a base classification model and selected the best features using the model's feature importance scores. This feature selection method provides us just as much cross-validated accuracy as Stepwise Selection and Recursive Feature Elimination, while being significantly more computationally efficient.

2 Hyperparameter Tuning

Performed hyperparameter tuning on a base model with the selected features using an exhaustive 5-fold cross-validated grid search. The parameter grid values were selected based on the most commonly used space of values for each hyperparameter.

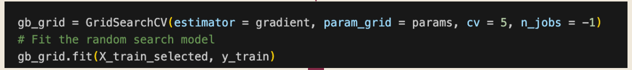
3 Optimal Model Training

Trained our final model for the associated ML method using the optimal set of features and hyperparameters found from feature selection and hyperparameter tuning.

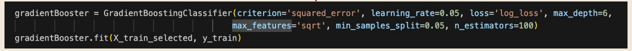


```
gradient = GradientBoostingClassifier()
params = {
            'loss': ['log_loss','exponential'],
            'learning_rate': [0.01, 0.05, 0.1],
            'n_estimators' : [100, 200, 400],
            'criterion' :['friedman_mse', 'squared_error'],
            'min_samples_split' :[0.01, 0.05, 0.1],
            'max_depth': [3,4,5,6,],
            'max_features': ['sqrt', 'log2']
        }

#Select Features
sel = SelectFromModel(gradient)
X_train_selected = sel.fit_transform(X_train.iloc[:, 2:], y_train)
print("Selected Features Length:", len(sel.get_feature_names_out()))
print("Selected Features:", ", ".join(sel.get_feature_names_out()))
```



```
gb_grid = GridSearchCV(estimator = gradient, param_grid = params, cv = 5, n_jobs = -1)
# Fit the random search model
gb_grid.fit(X_train_selected, y_train)
```

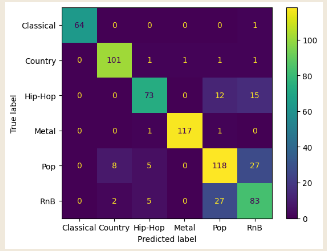


```
gradientBooster = GradientBoostingClassifier(criterion='squared_error', learning_rate=0.05, loss='log_loss', max_depth=6,
                                             max_features='sqrt', min_samples_split=0.05, n_estimators=100)
gradientBooster.fit(X_train_selected, y_train)
```

Model Results

Confusion Matrix | Selected Features | Best Hyper Parameters

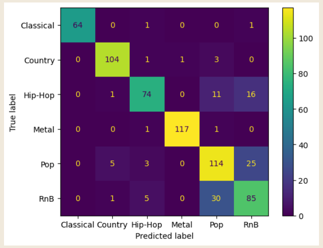
Random Forest



33 Features: danceability, energy, loudness, speechiness, acousticness, mfcc_1Mean, mfcc_2Mean, mfcc_3Mean, mfcc_3Std, mfcc_4Std, mfcc_5Std, mfcc_6Std, mfcc_7Std, mfcc_8Mean, mfcc_8Std, mfcc_9Std, mfcc_10Std, mfcc_11Std, zero_cross_Mean, spect_centroid_Mean, spect_centroid_Std, spect_contrast_Mean, spect_bw_Mean, spect_rolloff_Mean, spect_rolloff_Std, artist_genre_classical, artist_genre_contemporary, artist_genre_country, artist_genre_hip, artist_genre_hop, artist_genre_metal, artist_genre_pop, artist_genre_rap

'max_depth': 9,
'max_features': 'auto',
'min_samples_split': 0.01,
'n_estimators': 150

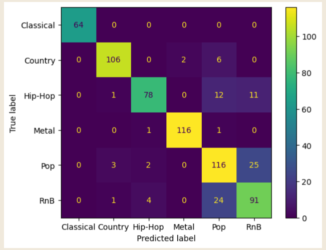
Gradient Boosting



18 Features: danceability, energy, loudness, speechiness, acousticness, mfcc_1Mean, mfcc_3Mean, mfcc_5Std, mfcc_7Std, spect_centroid_Std, spect_contrast_Mean, artist_genre_contemporary, artist_genre_country, artist_genre_hip, artist_genre_hop, artist_genre_metal, artist_genre_pop, artist_genre_rap

'criterion': 'friedman_mse',
'learning_rate': 0.01,
'loss': 'log_loss',
'max_depth': 6,
'max_features': 'log2',
'min_samples_split': 0.1,
'n_estimators': 400

SVM

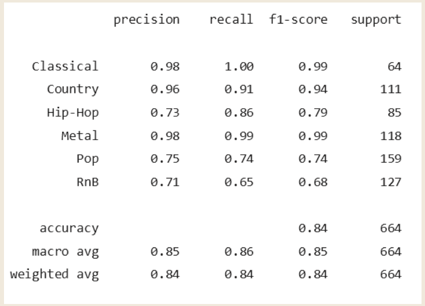


44 Features: mode, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, mfcc_1Mean, mfcc_1Std, mfcc_2Std, mfcc_3Mean, mfcc_4Std, mfcc_5Std, mfcc_6Mean, mfcc_7Mean, mfcc_7Std, mfcc_8Mean, mfcc_9Mean, mfcc_10Std, mfcc_12Mean, chroma_2Mean, chroma_3Mean, chroma_5Std, chroma_6Mean, chroma_6Std, chroma_8Mean, chroma_9Mean, chroma_9Std, chroma_10Mean, chroma_11Std, chroma_12Std, tonal_4Std, tonal_5Std, zero_cross_Mean, spect_centroid_Std, spect_contrast_Mean, spect_contrast_Std, artist_genre_contemporary, artist_genre_country, artist_genre_metal, artist_genre_pop

‘C’: 1
‘gamma’: 1

Conclusion

Performance of the Models

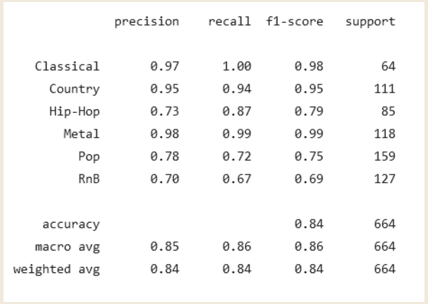


```
              precision    recall  f1-score   support

   Classical       0.98      1.00      0.99        64
     Country       0.96      0.91      0.94       111
     Hip-Hop       0.73      0.86      0.79        85
       Metal       0.98      0.99      0.99       118
         Pop       0.75      0.74      0.74       159
         RnB       0.71      0.65      0.68       127

    accuracy                           0.84       664
   macro avg       0.85      0.86      0.85       664
weighted avg       0.84      0.84      0.84       664
```

Random Forest

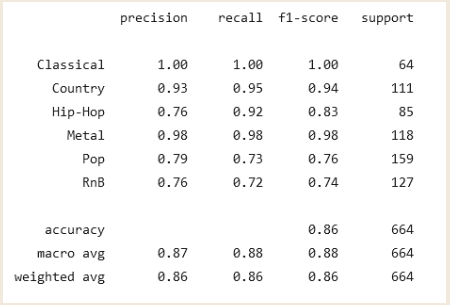


```
              precision    recall  f1-score   support

   Classical       0.97      1.00      0.98        64
     Country       0.95      0.94      0.95       111
     Hip-Hop       0.73      0.87      0.79        85
       Metal       0.98      0.99      0.99       118
         Pop       0.78      0.72      0.75       159
         RnB       0.70      0.67      0.69       127

    accuracy                           0.84       664
   macro avg       0.85      0.86      0.86       664
weighted avg       0.84      0.84      0.84       664
```

Gradient Boosting



```
              precision    recall  f1-score   support

   Classical       1.00      1.00      1.00        64
     Country       0.93      0.95      0.94       111
     Hip-Hop       0.76      0.92      0.83        85
       Metal       0.98      0.98      0.98       118
         Pop       0.79      0.73      0.76       159
         RnB       0.76      0.72      0.74       127

    accuracy                           0.86       664
   macro avg       0.87      0.88      0.88       664
weighted avg       0.86      0.86      0.86       664
```

SVM

Key Takeaways

- SVM is our best model overall, with a better accuracy and f1-score across the different genres
- As seen in our confusion matrix, Pop and RnB were the most difficult genre to identify by our models
- The biggest improvement in SVM over the other two models is in its precision and recall of Hip-Hop, Pop and RnB.

Bibliography

1. *API reference*. scikit. (n.d.). https://scikit-learn.org/stable/modules/classes.html
2. McFee, B., Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, … Waldir Pimenta. (2023). librosa/librosa: 0.10.1 (0.10.1). Zenodo. https://doi.org/10.5281/zenodo.8252662
3. Ndou, N., Ajoodha, R., & Jadhav, A. (2021). Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches. 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 1–6. doi:10.1109/IEMTRONICS52119.2021.9422487
4. Spotify. (n.d.). Spotify Web API. Web API | Spotify for Developers. https://developer.spotify.com/documentation/web-api