Google Colab Link

https://colab.research.google.com/drive/1ZTfXwiXmU8QUsm5grfNST__Cq-U6Ilsg

Motivation

Automatically tagging music by genre is critical for enhancing user experience on streaming platforms. It enables better content organization, accurate search results, and personalized music recommendation has greater accuracy per user. The motivation behind this project stems from the growing need for scalable, intelligent systems that are capable of categorizing music libraries without relying on error-prone labeling. However, genre classification presents a technical challenge, that is that the acoustic features that distinguish genres are often subtle enough to overlap with one another. This blurs the lines between genres that are closely related, such as hip-hop and rap. This made the problem both technically interesting and valuable for real-world applications in music streaming and other recommendation systems.

Prior Research

Music Genre Classification using Machine Learning Techniques

Hareesh Bahuleyan, and other collaborators, explored music genre predictability by using machine learning in a study titled “Music Genre Classification Using Machine Learning Techniques” (2018). Their primary objective was to improve music recommendation systems by enhancing user experience through better content organization. Their plan for this was to do so by improving automated music genre tagging.

For their findings, they used audio features as their inputs and compared their results by using SVM, KNN, and deep learning CNN. They found that KNN performed the best for their study due its ability to capture acoustic patterns, which is similar to why we chose KNN as one of our algorithms to use. They also highlighted the fact that there was a challenge with accurate predictions due to overlapping genres.

Goals

This project aims to accurately classify songs into their respective genres by leveraging audio features through comparative analysis of multiple machine learning models. For this, we will:

- Implement and compare three classifiers:

    - Random Forest - Handling non-linear relationships
    - SVM - For high-dimensional feature separation
    - KNN - For ease of interability and pattern detection
- Optimize performance by tuning hyperparameters (such as n_estimators for Random Forest, C for SVM, n_neighbors for KNN) using cross-validation
- Evaluate models with a strong focus on precision to minimize songs being assigned to the wrong genre, which will ensure reliable predictions for music recommendations systems.

This approach will identify the most effective model for automated genre tagging while maintaining interpretability and scalability.

Dataset

Source: kaggle music genre dataset

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

Contents: Each row corresponds to a song and its features

- Instance_id - A unique identifier for each track in the dataset. Used to distinguish between records.
- Artist_name - The name of the artist or band who created the track.
- Track_name - The title of the track or song.
- Popularity - A score (0–100) reflecting how popular the track is, based on factors like total streams and recent user activity on platforms like Spotify.
- Acousticness - A confidence measure from 0.0 to 1.0 of whether the track is acoustic. A higher value indicates a greater likelihood that the track is acoustic.
- Danceability - Describes how suitable a track is for dancing, based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A higher score (0.0 to 1.0) indicates more danceable tracks.
- Duration_ms - The total duration of the track, measured in milliseconds.
- Energy - A measure from 0.0 to 1.0 representing intensity and activity. Energetic tracks feel fast, loud, and noisy. For example, metal would score high on energy, while a Classical music would score low.
- Instrumentalness - Predicts whether a track contains no vocals. The closer the value is to 1.0, the more likely it is that the track is instrumental. Values above 0.5 are intended to represent instrumental tracks, but this isn't guaranteed.

- Key - The pitch class of the track, represented as an integer from 0 (C) to 11 (B). It does not indicate major or minor; that's covered by Mode.
- Liveness – Detects the presence of an audience in the recording. A value above 0.8 indicates a strong likelihood that the track was performed live.
- Loudness – The overall loudness of a track in decibels (dB). Typically ranges between -60 and 0 dB. Loudness is averaged across the entire track and is useful for normalization.
- Mode – Indicates the modality (major or minor) of the track: 1 is major, 0 is minor. Major is typically associated with happier-sounding music, and minor with sadder or more dramatic music.
- Speechiness – Detects the presence of spoken words in a track. A value closer to 1.0 indicates more speech-like content. Values above 0.66 are likely speech (e.g., talk shows), values between 0.33 and 0.66 may contain both music and speech (e.g., rap), and below 0.33 are likely music.
- Tempo – The estimated overall tempo of a track in beats per minute (BPM). This represents the speed or pace of the music.
- Obtained_date – The date the data or track information was retrieved. This may not affect model performance directly but could be used for time-based filtering.
- Valence – A measure of musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g., happy, cheerful, euphoric), while those with low valence sound more negative (e.g., sad, depressed, angry).
- Music_genre – The target variable for our classification task, representing the genre assigned to the track.

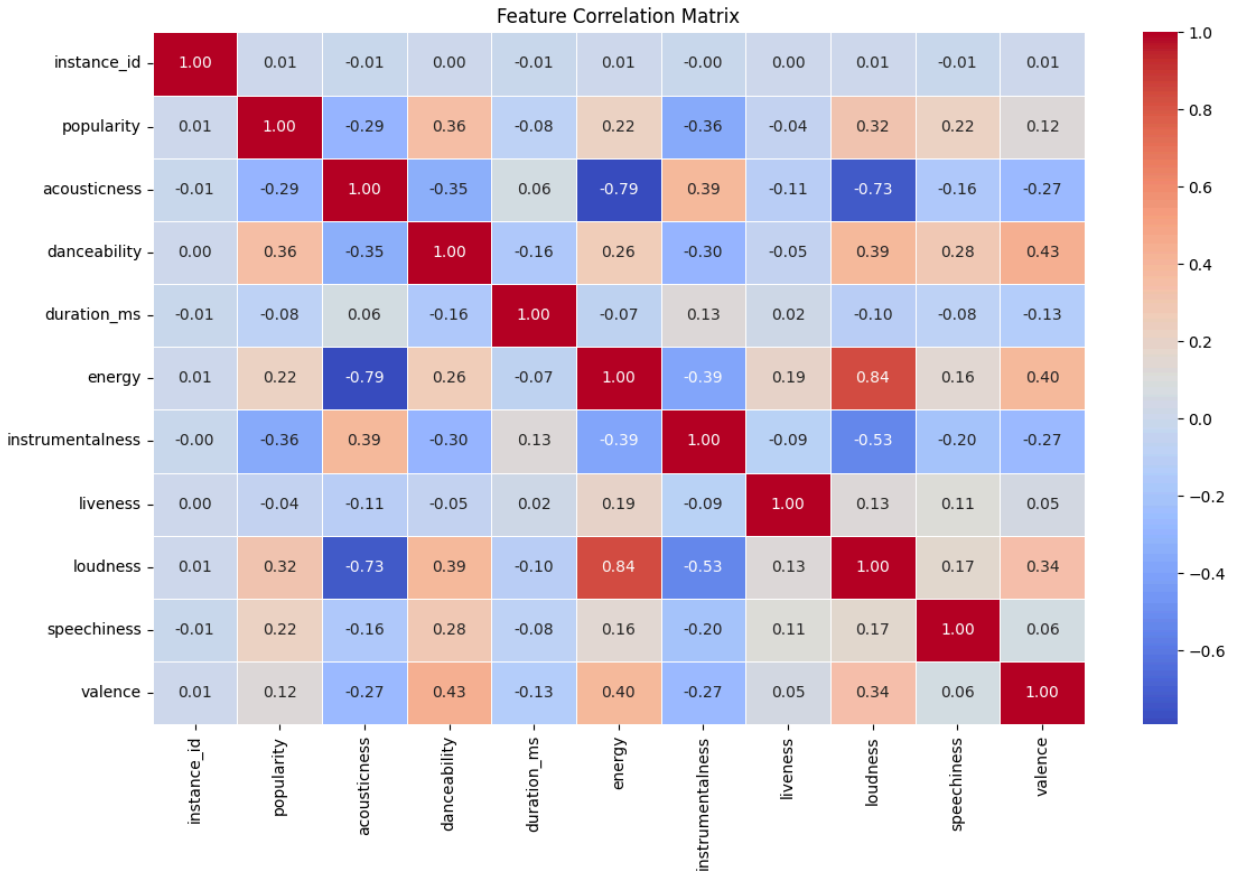
Exploratory Data Analysis

Data Overview:

- Contains 50,000 songs with 5,000 songs per genre.
- Missing/Invalid data
    - 5 rows with no data at all
    - Handled data with negative duration
    - Dropped rows containing ‘?’ for tempo

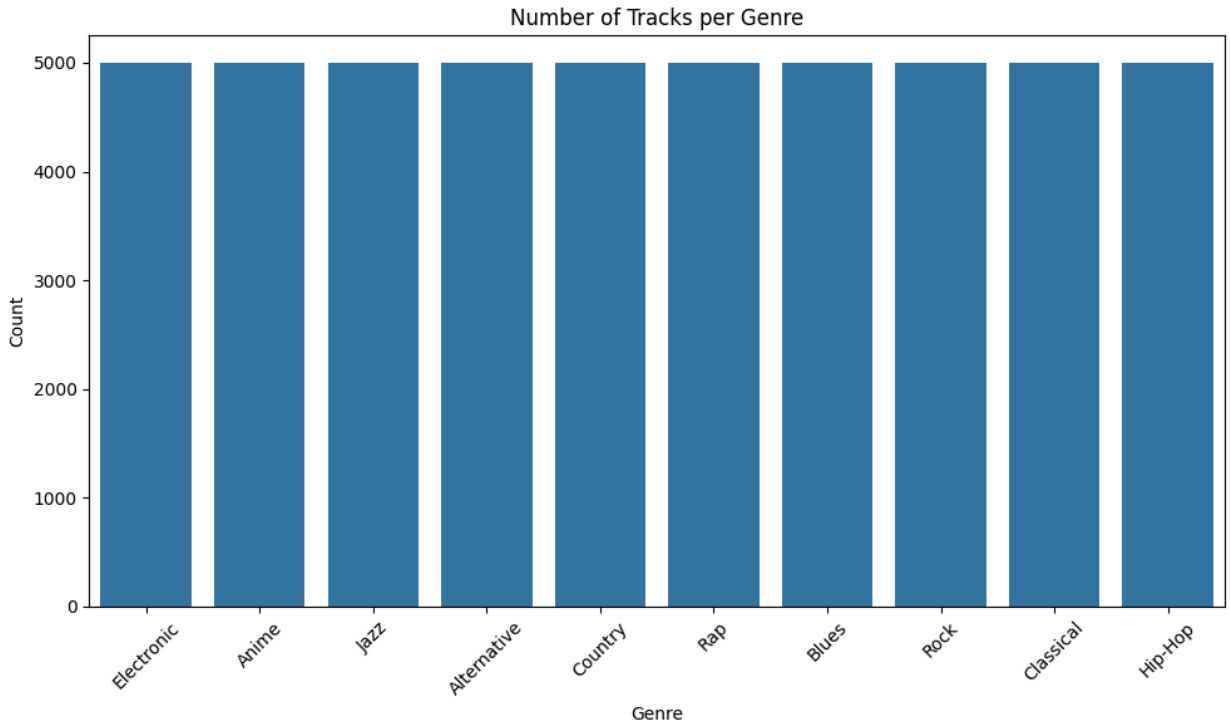
Correlation matrix:

Reveals relationships among features (e.g. energy & loudness)



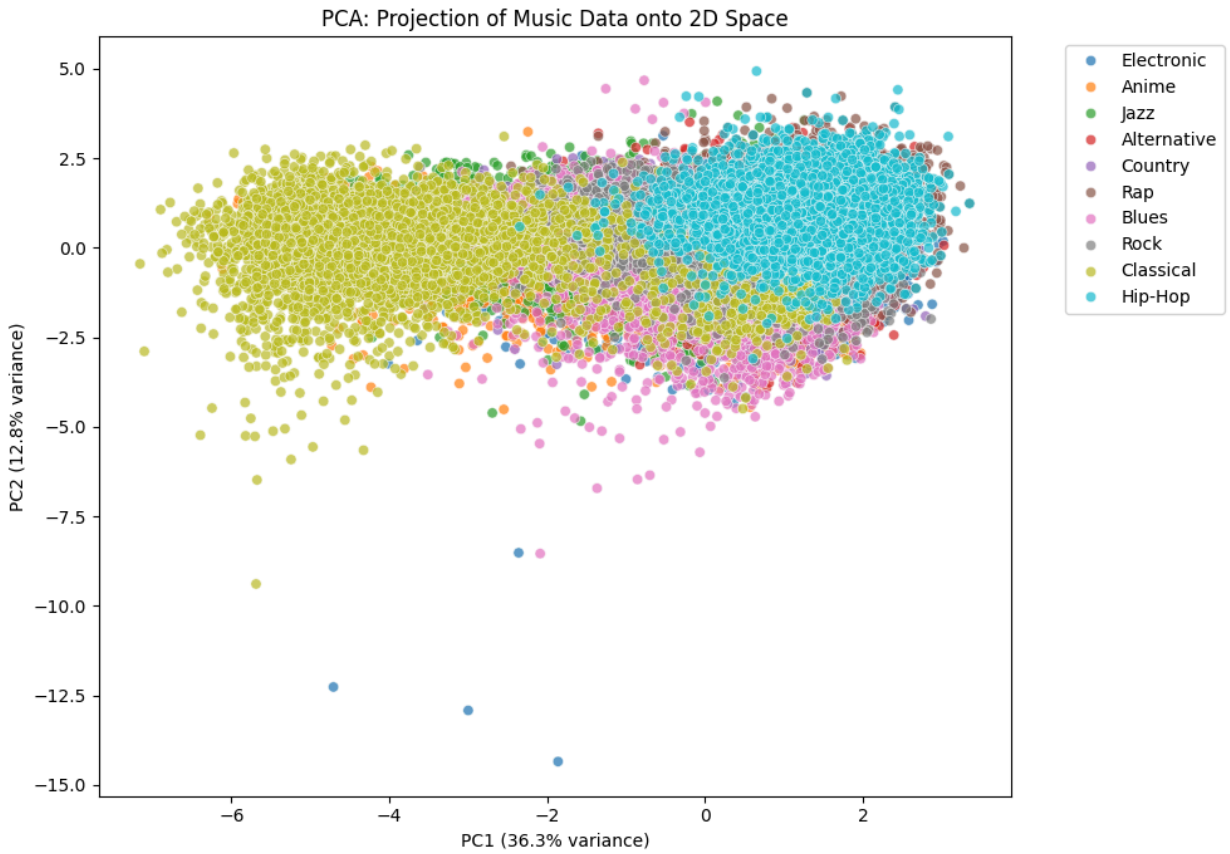
Genre distribution:

Showing potential class imbalance



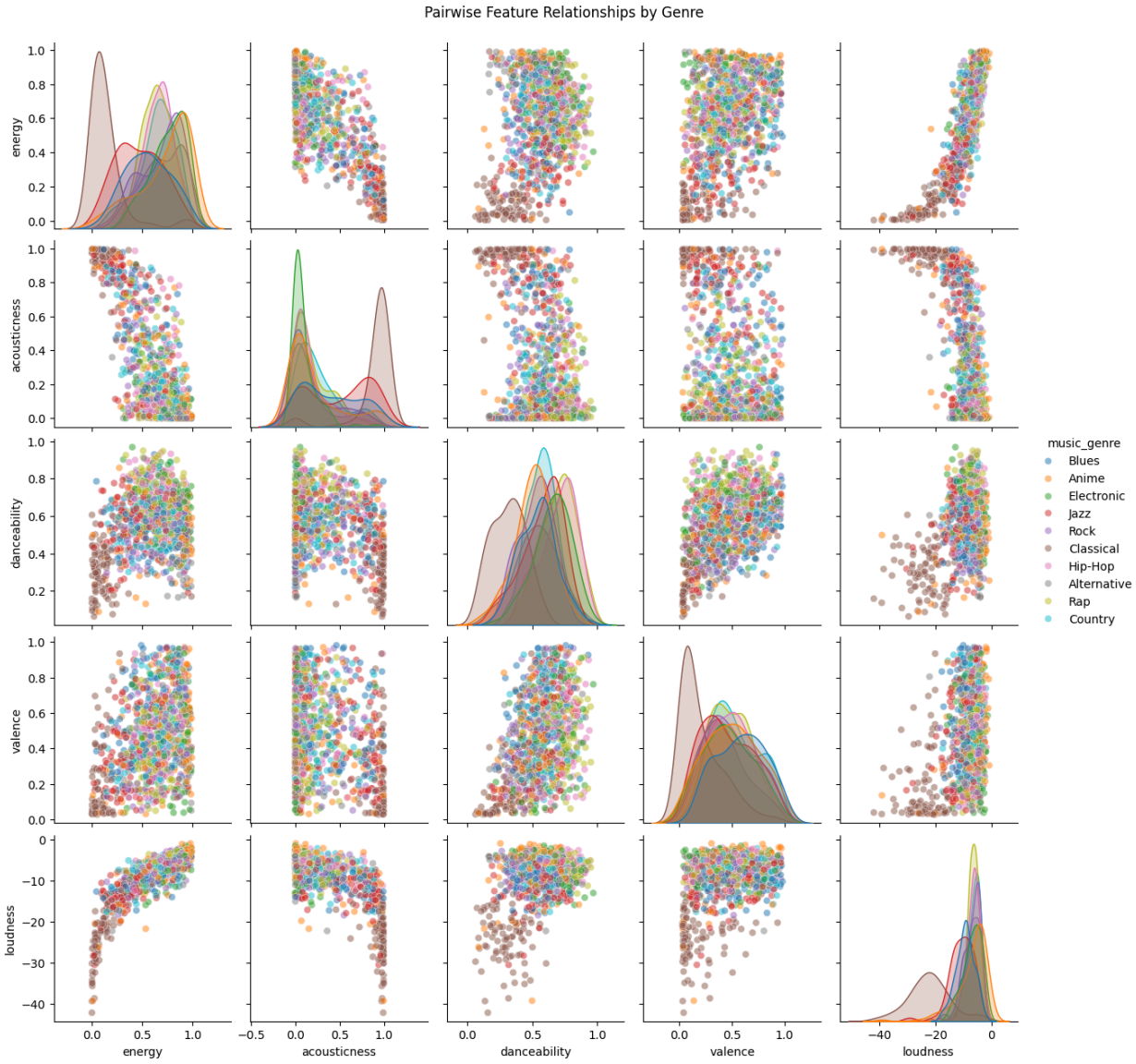
Principal Component Analysis:

Shows how well genres can be separated based on audio features (note that only hip-hop and classical are easily distinguishable clusters)



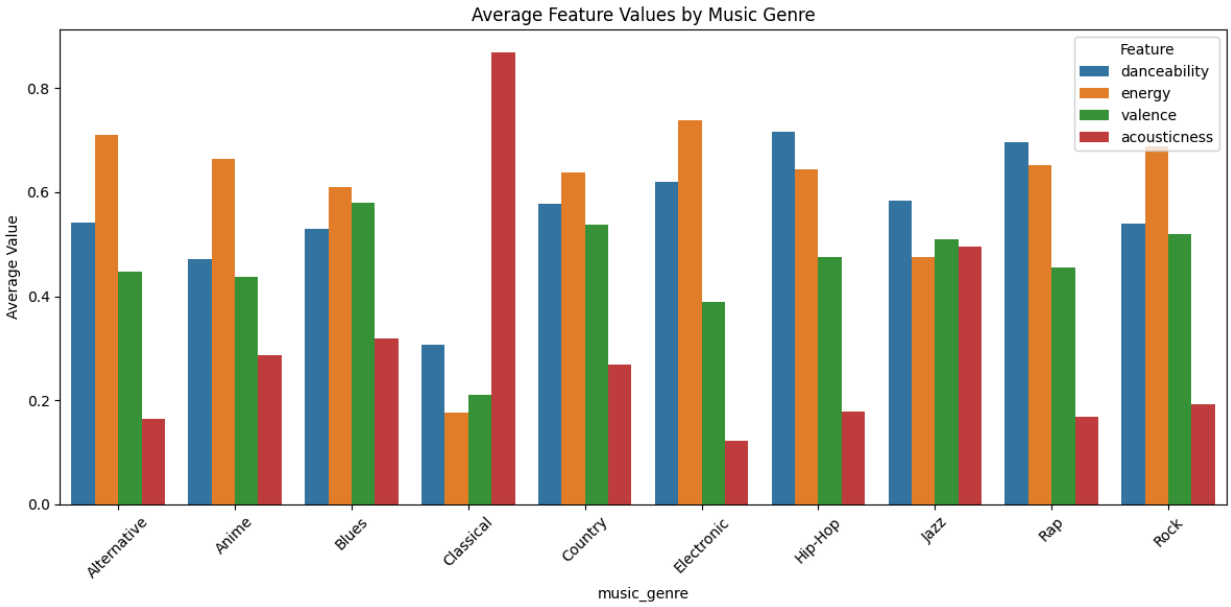
Pairwise Feature Plots:

Visualize relationships between features and their distribution across genres. Note that the linear patterns shown in loudness/energy and other plots. This pattern typically indicates correlation between two features.



Bar Plot:

Shows average values for selected features, allows us to see what features may be similar/different across genres.



Pre-processing

- Feature Encoding:
  - Applied label encoding to categorical variables (key, mode, genre)
  - Preserved all original features after testing showed reduced performance when removing low-importance features
- Data Cleaning:
  - Removed null and invalid entries
  - Conducted initial testing before merging similar genres to maintain evaluation consistency
- Dataset Preparation:
  - Implemented stratified 80/20 train-test split
  - Performed genre merging only after establishing baseline model performance

Algorithms

Random Forest: The Random Forest classification model was chosen for its ability to handle non-linear data, its ability to maintain stable and accurate performance under various conditions (like noise), and its built-in feature importance. This algorithm is able to handle mixed data types, like the ones found in our dataset. It avoids heavy preprocessing, and is very easy to interpret. However, Random Forest is prone to overfitting without tuning, and performs slowly on larger data. We did see some overfitting in our training data, but the model performed well enough on the testing data.

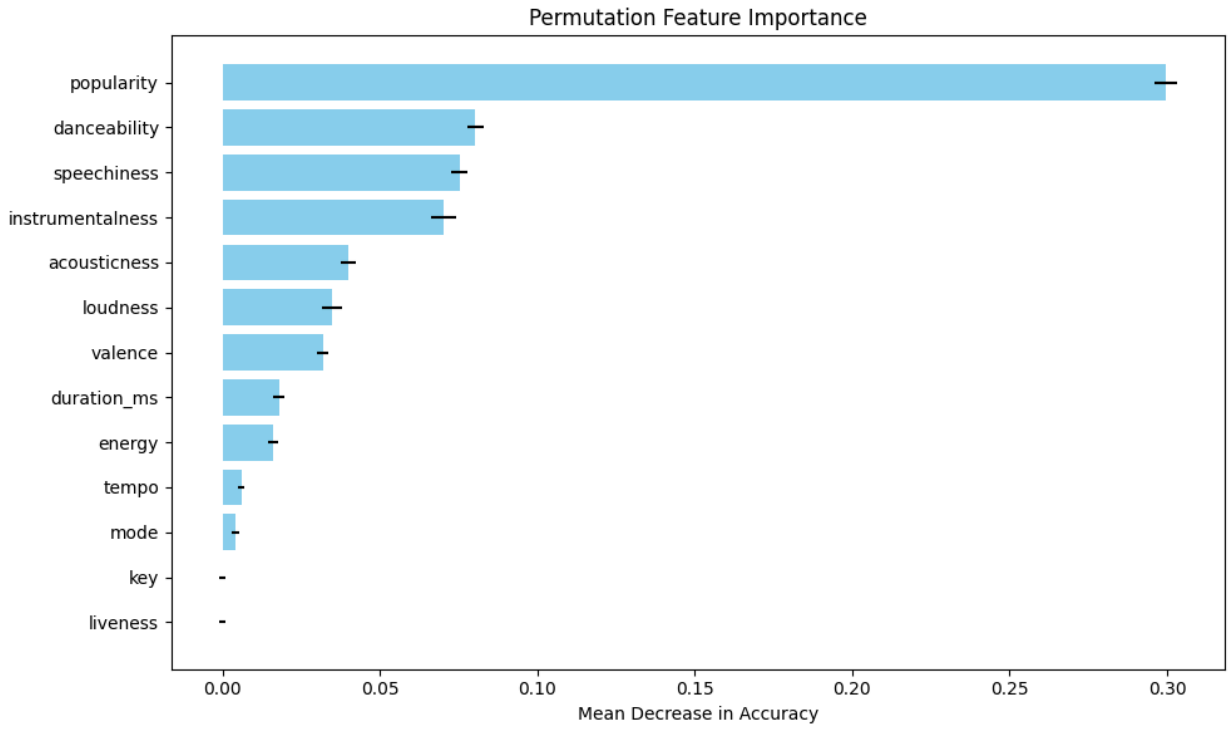
SVM: Support Vector Machine classification model was chosen because of its incredible ability to classify data. The data in this research is not the commonly linear data seen in SVM's, but SVM luckily has the ability to create a higher dimensional space, the kernel trick. The data is heavy text classification, making SVM efficient in analyzing the data. SVM are less likely to overfit and generalize well even with data the model had not seen before. The biggest pro of SVM is its ability to successfully learn from datasets with many features. The model handles linear and nonlinear data and can be used for both classification and regression tasks. However, this algorithm does require careful tuning of hyperparameters and performs poorly with imbalance datasets unless the class weights are adjusted.

KNN: Due to the nature of the data, K-Nearest Neighbors was chosen since we could treat the data as points in a multi-dimensional space. Unlike Linear Regression, KNN does not assume linear separability, which is common in audio data where some genres may overlap in their features. As a non-parametric method, KNN does not require training since it automatically infers patterns from the data. its ease of interpretability. Decisions are based on nearest neighbors, which makes it easy to explain why a song was classified as a certain genre. In addition, because music genre prediction is essentially multiclass, KNN supports multiclass without having to make modifications. Despite the pros of this algorithm, its biggest con is that it is sensitive to irrelevant data or redundant features. It needs feature scaling and selection. In addition, it will perform poorly with imbalance data unless weights are used.

Experiments

Plots:

- Confusion matrix for each model
- Feature importance bars (tested with all features & feature removals)



Grid search:

- 5 fold CV

```
# Define the parameter grid
param_grid = {
    'n_estimators': [100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5],
    'min_samples_leaf': [1, 2],
}

grid_search = GridSearchCV(
    estimator=rf,
    param_grid=param_grid,
    cv=5,  # 5-fold cross-validation
    scoring='accuracy',
    verbose=2
)

# Run grid search
grid_search.fit(X_train_reduced, y_train)
```

Train/Test split:

- Trained models on X_train and evaluated on X_test (80/20 split)

Analysis:

- Accuracy
- Precision
- Recall
- F1-score

```
Accuracy: 0.7187
Precision (weighted): 0.7197
Precision (macro): 0.7283
              precision    recall  f1-score   support

       Anime       0.83      0.75      0.78       813
   Classical       0.84      0.83      0.83       807
     Country       0.64      0.50      0.57       810
  Electronic       0.68      0.53      0.60       806
     Hip-Hop       0.83      0.89      0.85      1624
        Jazz       0.65      0.69      0.67      1622
        Rock       0.62      0.71      0.66      1630

    accuracy                           0.72      8112
   macro avg       0.73      0.70      0.71      8112
weighted avg       0.72      0.72      0.72      8112
```

Error analysis:

- Genres like country and electronic had lower recall and precision, likely due to feature overlap or genre ambiguity

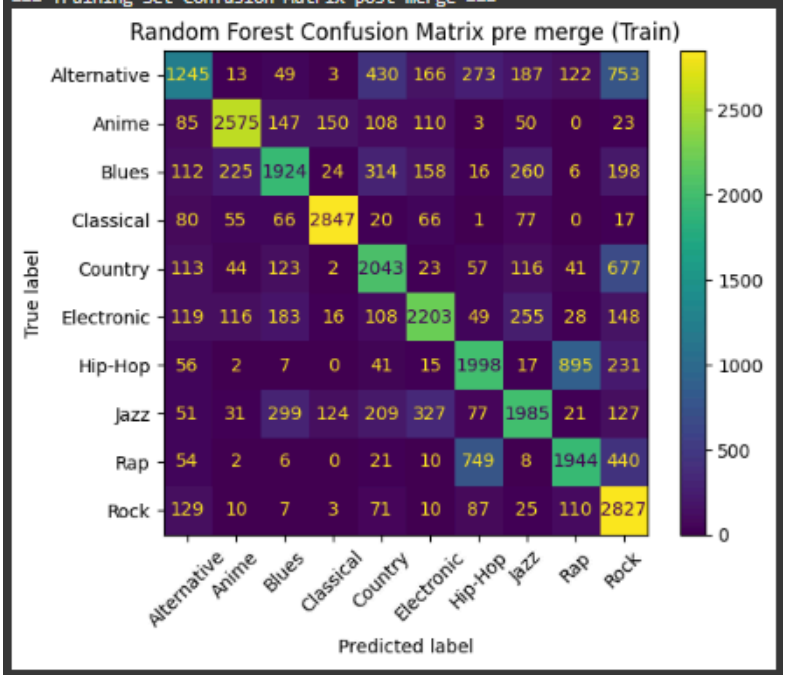
Result Analysis

Random Forest:

Random Forest performed the best of all three models. You can see below that early in our experiment we did testing and training on the original dataset with no changes. The model did not perform very well under these circumstances and struggled in both the training and testing sets. After the initial tests, we noticed through a confusion matrix that a large number of classifications for certain genres like rock/alternative and rap/hip-hop were being mistaken for one another. As a result, we decided to merge these genres as they have very similar audio features and often can be classified as one another in the real world, hence the confusion. It's clear that this had a positive effect on both the testing and training sets as the precision increased significantly, with an increase from 58% accuracy to 72% on the test set as well as a large increase from 68% to 99% in the training set. The confusion matrix showed that some genres were still struggling to be properly classified but overall the model performed well.

Training data pre-genre merge

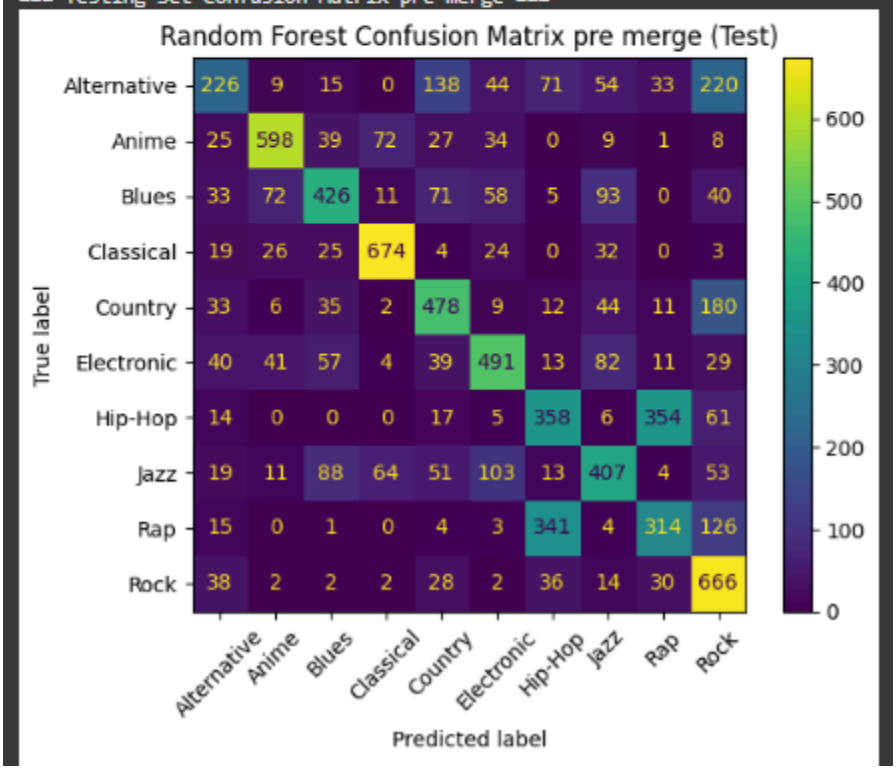
```
=== Training Set Classification Report post merge ===
              precision    recall  f1-score   support

 Alternative       0.61      0.38      0.47      3241
       Anime       0.84      0.79      0.81      3251
       Blues       0.68      0.59      0.64      3237
   Classical       0.90      0.88      0.89      3229
     Country       0.61      0.63      0.62      3239
  Electronic       0.71      0.68      0.70      3225
     Hip-Hop       0.60      0.61      0.61      3262
        Jazz       0.67      0.61      0.64      3251
         Rap       0.61      0.60      0.61      3234
        Rock       0.52      0.86      0.65      3279

    accuracy                           0.67     32448
   macro avg       0.68      0.67      0.66     32448
weighted avg       0.68      0.67      0.66     32448


=== Training Set Confusion Matrix post merge ===
```



Testing data pre-genre merge (note the high amount of confusion between rap/hip hop and rock/alternative)

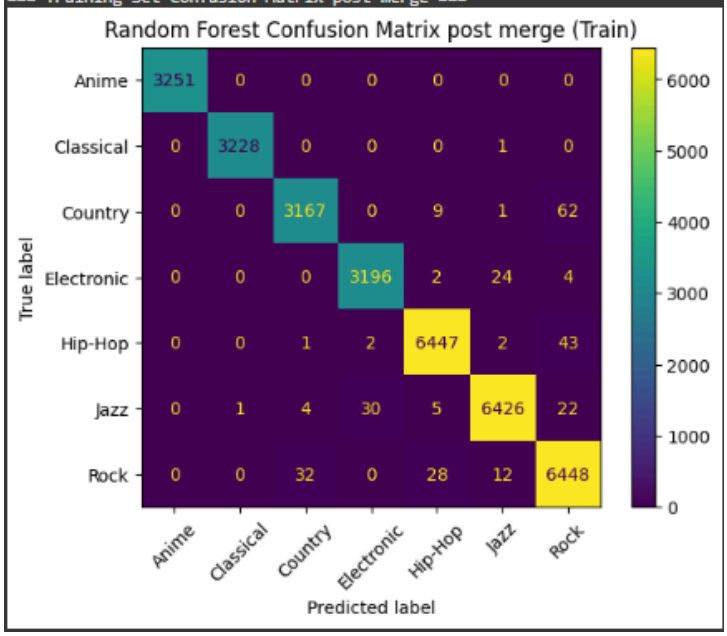
```
=== Testing Set Classification Report pre merge ===
              precision    recall  f1-score   support

 Alternative       0.49      0.28      0.36       810
       Anime       0.78      0.74      0.76       813
       Blues       0.62      0.53      0.57       809
   Classical       0.81      0.84      0.82       807
     Country       0.56      0.59      0.57       810
  Electronic       0.64      0.61      0.62       807
     Hip-Hop       0.42      0.44      0.43       815
        Jazz       0.55      0.50      0.52       813
         Rap       0.41      0.39      0.40       808
        Rock       0.48      0.81      0.60       820

    accuracy                           0.57      8112
   macro avg       0.58      0.57      0.57      8112
weighted avg       0.58      0.57      0.57      8112
```

```
=== Testing Set Confusion Matrix pre merge ===
```



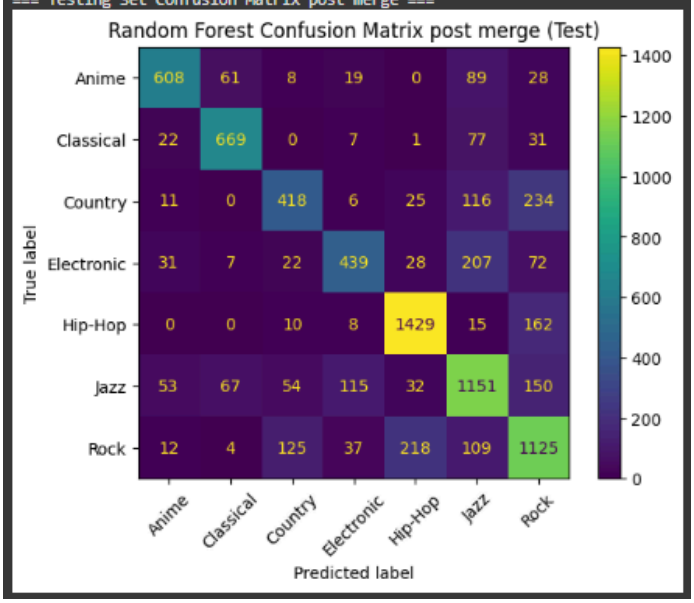
Training data post-genre merge

```
=== Training Set Classification Report post merge ===
              precision    recall  f1-score   support

       Anime       1.00      1.00      1.00      3251
   Classical       1.00      1.00      1.00      3229
     Country       0.99      0.98      0.98      3239
  Electronic       0.99      0.99      0.99      3226
     Hip-Hop       0.99      0.99      0.99      6495
        Jazz       0.99      0.99      0.99      6488
        Rock       0.98      0.99      0.98      6520

    accuracy                           0.99     32448
   macro avg       0.99      0.99      0.99     32448
weighted avg       0.99      0.99      0.99     32448


=== Training Set Confusion Matrix post merge ===
```



Testing data post-genre merge

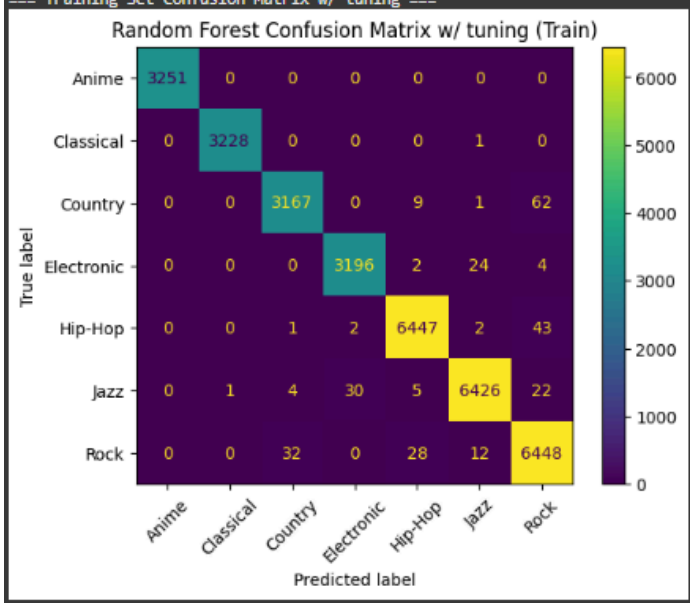
```
=== Testing Set Classification Report post merge ===
              precision    recall  f1-score   support

       Anime       0.82      0.75      0.78       813
   Classical       0.83      0.83      0.83       807
     Country       0.66      0.52      0.58       810
  Electronic       0.70      0.54      0.61       806
     Hip-Hop       0.82      0.88      0.85      1624
        Jazz       0.65      0.71      0.68      1622
        Rock       0.62      0.69      0.66      1630

    accuracy                           0.72      8112
   macro avg       0.73      0.70      0.71      8112
weighted avg       0.72      0.72      0.72      8112

=== Testing Set Confusion Matrix post merge ===
```



Training data post-genre merge w/ tuning

```
=== Training Set Classification Report w/ tuning ===
              precision    recall  f1-score   support

       Anime       1.00      1.00      1.00      3251
   Classical       1.00      1.00      1.00      3229
     Country       0.99      0.98      0.98      3239
  Electronic       0.99      0.99      0.99      3226
     Hip-Hop       0.99      0.99      0.99      6495
        Jazz       0.99      0.99      0.99      6488
        Rock       0.98      0.99      0.98      6520

    accuracy                           0.99     32448
   macro avg       0.99      0.99      0.99     32448
weighted avg       0.99      0.99      0.99     32448
```

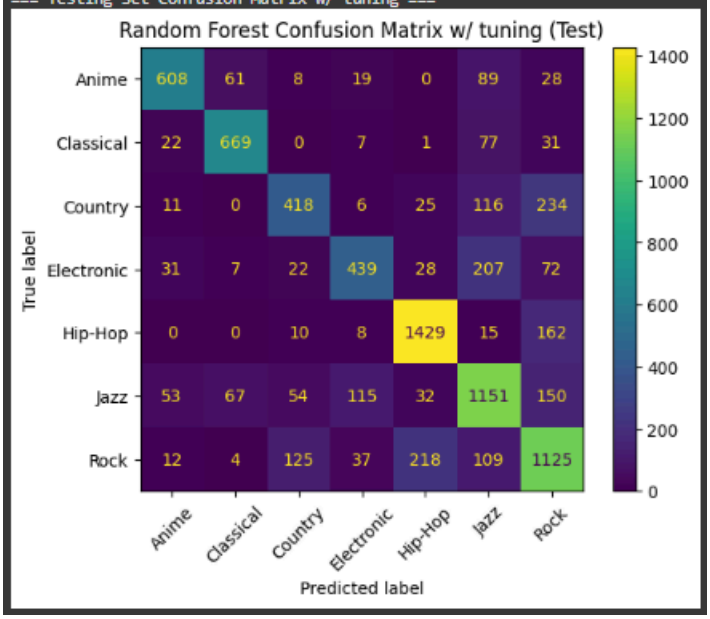
```
=== Training Set Confusion Matrix w/ tuning ===
```



Testing data post-genre merge w/ tuning

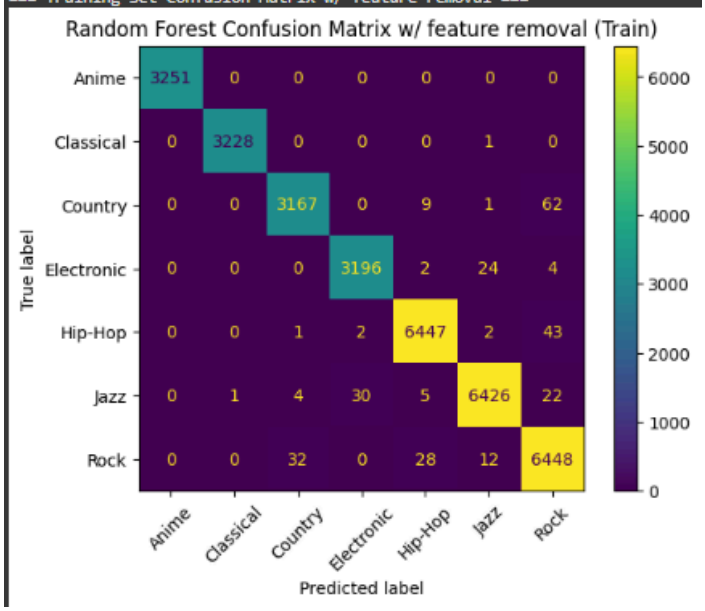
```
=== Testing Set Classification Report w/ tuning ===
              precision    recall  f1-score   support

       Anime       0.82      0.75      0.78       813
   Classical       0.83      0.83      0.83       807
     Country       0.66      0.52      0.58       810
  Electronic       0.70      0.54      0.61       806
     Hip-Hop       0.82      0.88      0.85      1624
        Jazz       0.65      0.71      0.68      1622
        Rock       0.62      0.69      0.66      1630

    accuracy                           0.72      8112
   macro avg       0.73      0.70      0.71      8112
weighted avg       0.72      0.72      0.72      8112
```

```
=== Testing Set Confusion Matrix w/ tuning ===
```



Training data w/ Feature selection

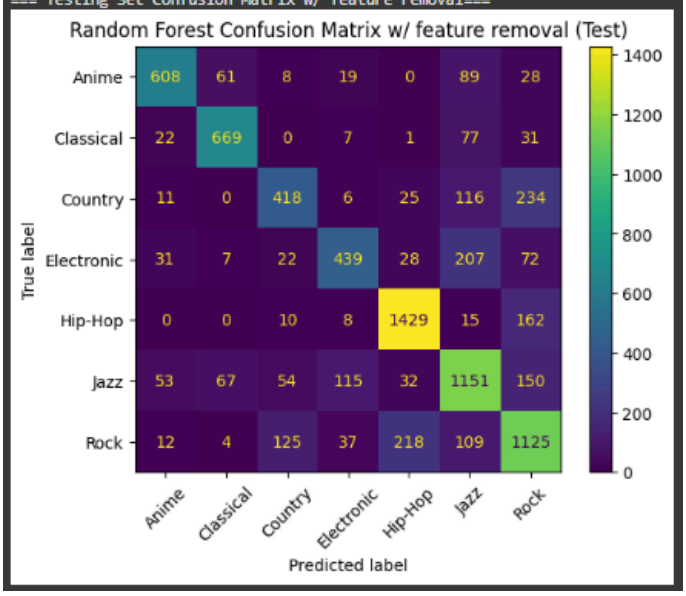
```
=== Training Set Classification Report w/ feature removal ===
              precision    recall  f1-score   support

       Anime       1.00      1.00      1.00      3251
   Classical       1.00      1.00      1.00      3229
     Country       0.99      0.98      0.98      3239
  Electronic       0.99      0.99      0.99      3226
     Hip-Hop       0.99      0.99      0.99      6495
        Jazz       0.99      0.99      0.99      6488
        Rock       0.98      0.99      0.98      6520

    accuracy                           0.99     32448
   macro avg       0.99      0.99      0.99     32448
weighted avg       0.99      0.99      0.99     32448


=== Training Set Confusion Matrix w/ feature removal ===
```



Testing data w/ Feature selection

```
=== Testing Set Classification Report w/ feature removal===
              precision    recall  f1-score   support

       Anime       0.82      0.75      0.78       813
   Classical       0.83      0.83      0.83       807
     Country       0.66      0.52      0.58       810
  Electronic       0.70      0.54      0.61       806
     Hip-Hop       0.82      0.88      0.85      1624
        Jazz       0.65      0.71      0.68      1622
        Rock       0.62      0.69      0.66      1630

    accuracy                           0.72      8112
   macro avg       0.73      0.70      0.71      8112
weighted avg       0.72      0.72      0.72      8112
```

```
=== Testing Set Confusion Matrix w/ feature removal===
```

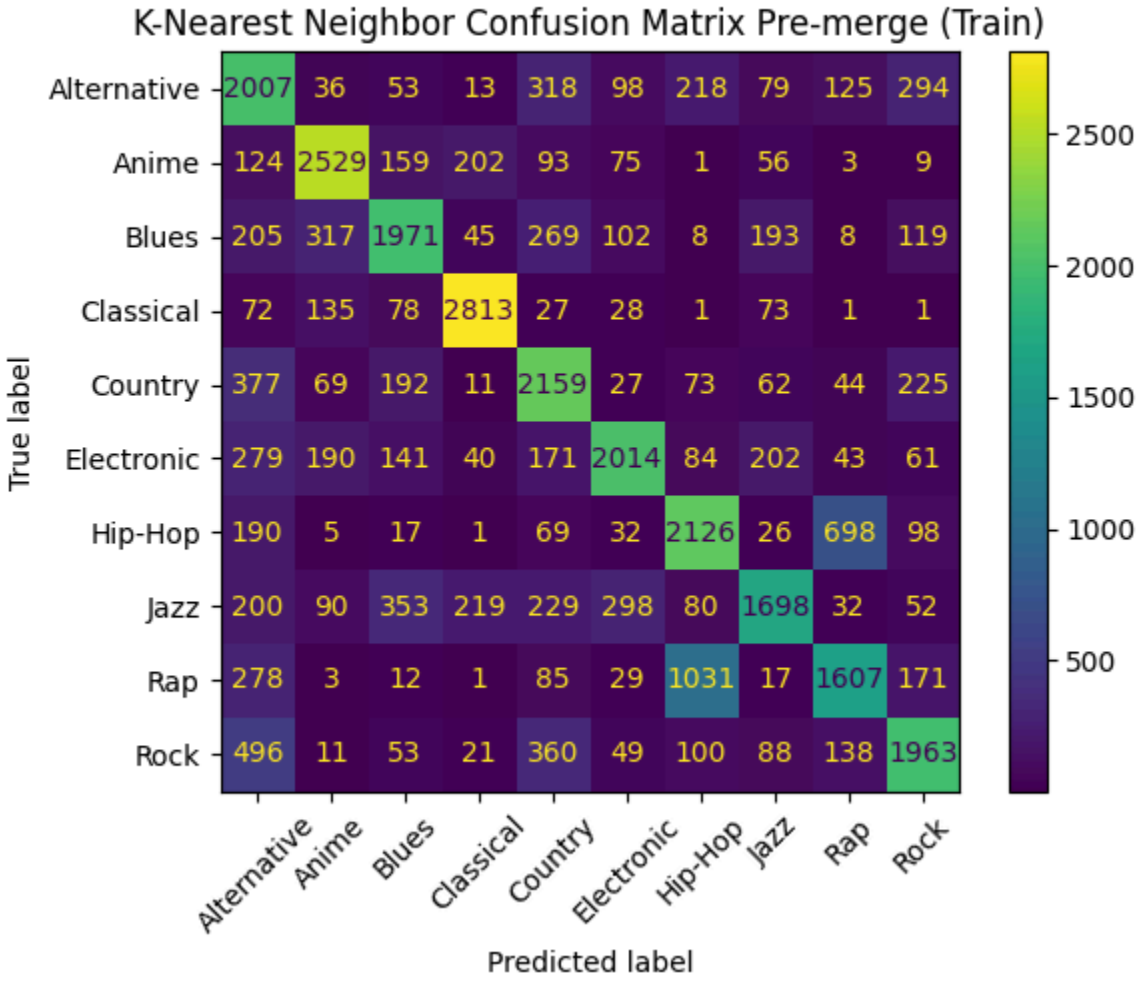


K-Nearest Neighbors

Although Random Forest performed the best, K-Nearest Neighbors came to close with precision and accuracy. In the earlier experiment, the training data showed an accuracy of 64% and the testing accuracy came out to be 47% which was a poor result. Much like what was found in the Random Forest outcome, we could see that genres with similarities in their audio features were getting mixed in together. Hip-hop and Rap were our main culprits in getting mixed up. Due to this, we went ahead and merged the two genres together to prevent conflicting and poor results. Once the genres were merged, we went ahead and created a new report and confusion matrix, which showed the training and testing data both had higher accuracy and precision points. Once the data was merged, we went ahead and tuned the data to be sure to get the best results and found that our accuracy and precision continued to improve overall. Although the testing data never went above 70% in accuracy, the confusion matrix visualization shows that the genre predictability is possible with more refinements and KNN could be an algorithm to keep working with.

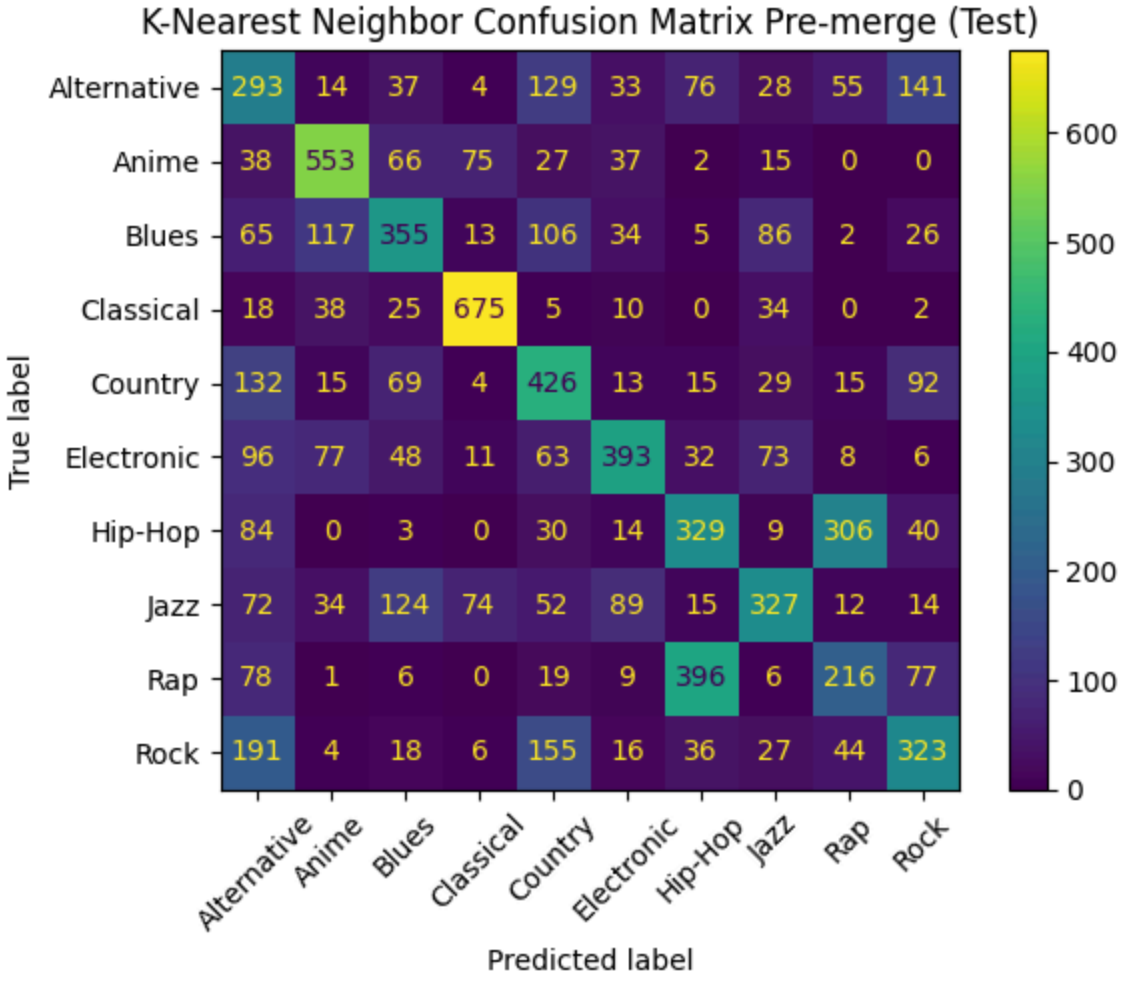
Training data pre-genre merge

```
Accuracy: 0.6437
Weighted Precision: 0.6514
Macro Precision: 0.6515
```



Testing data pre-genre merge

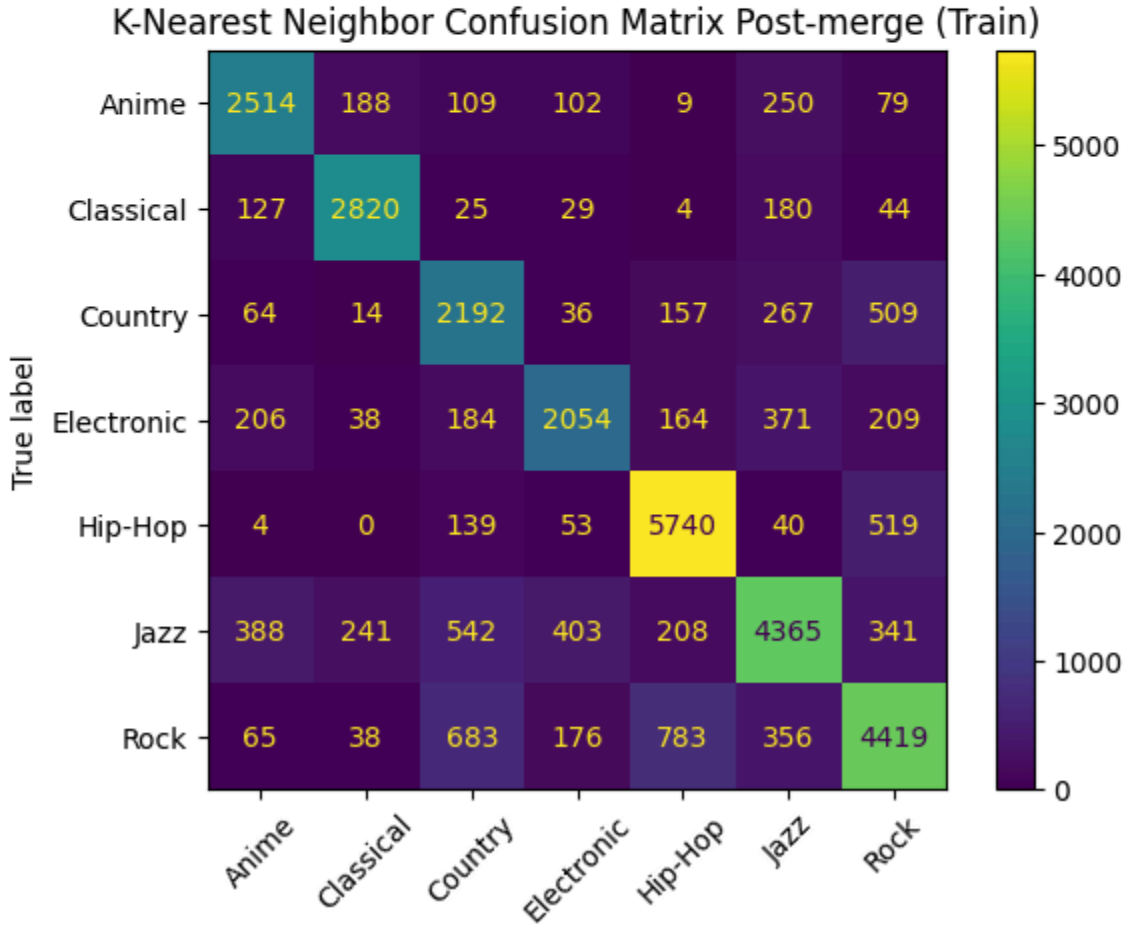
```
Accuracy: 0.4795
Weighted Precision: 0.4860
Macro Precision: 0.4861
Text(0.5, 1.0, 'K-Nearest Neighbor Confusion Matrix Pre-merge (Test)')
```



Training data post-genre merge

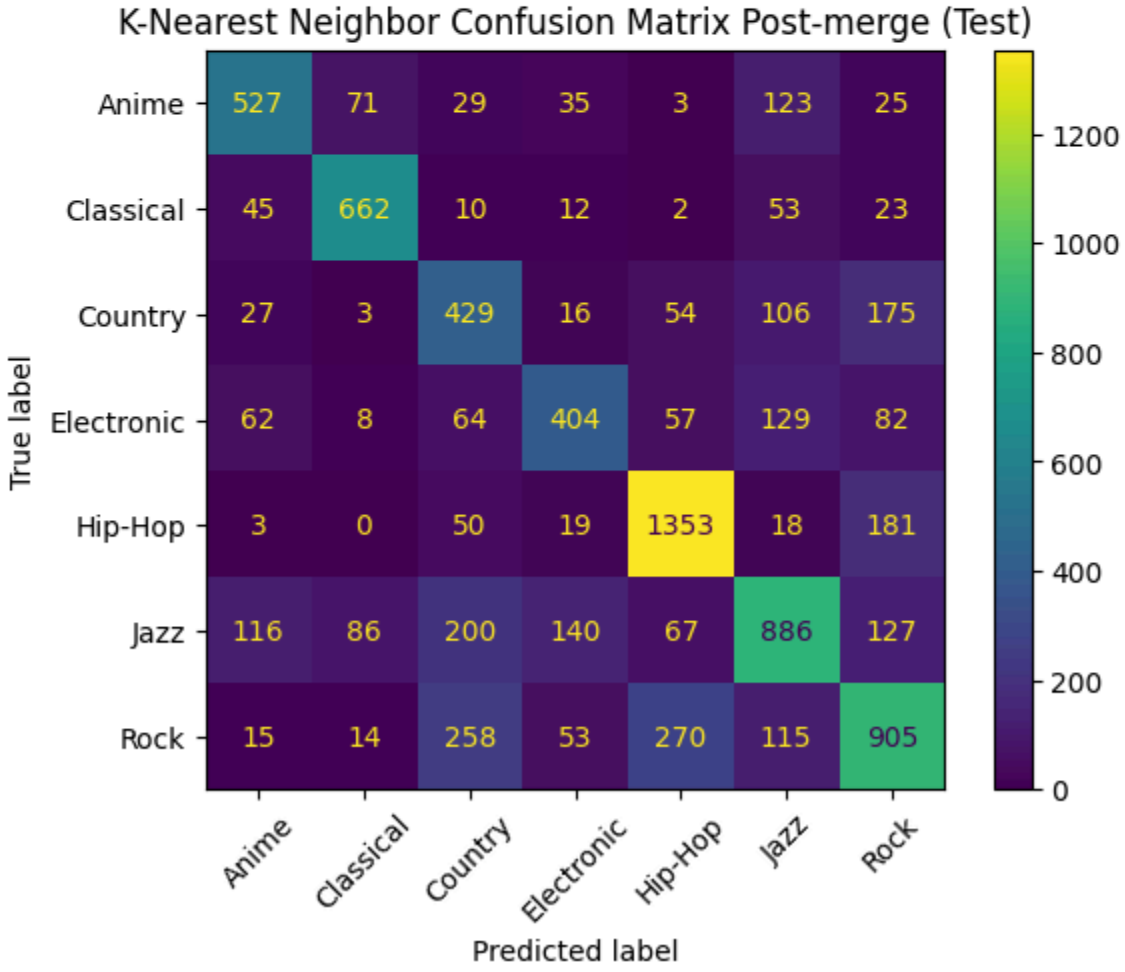
```
Training Classification Report (Post-Merge):
              precision    recall  f1-score   support

       Anime       0.75      0.77      0.76      3251
   Classical       0.84      0.87      0.86      3229
     Country       0.57      0.68      0.62      3239
  Electronic       0.72      0.64      0.68      3226
     Hip-Hop       0.81      0.88      0.85      6495
        Jazz       0.75      0.67      0.71      6488
        Rock       0.72      0.68      0.70      6520

    accuracy                           0.74     32448
   macro avg       0.74      0.74      0.74     32448
weighted avg       0.74      0.74      0.74     32448
```

```
<Figure size 1000x1000 with 0 Axes>
```



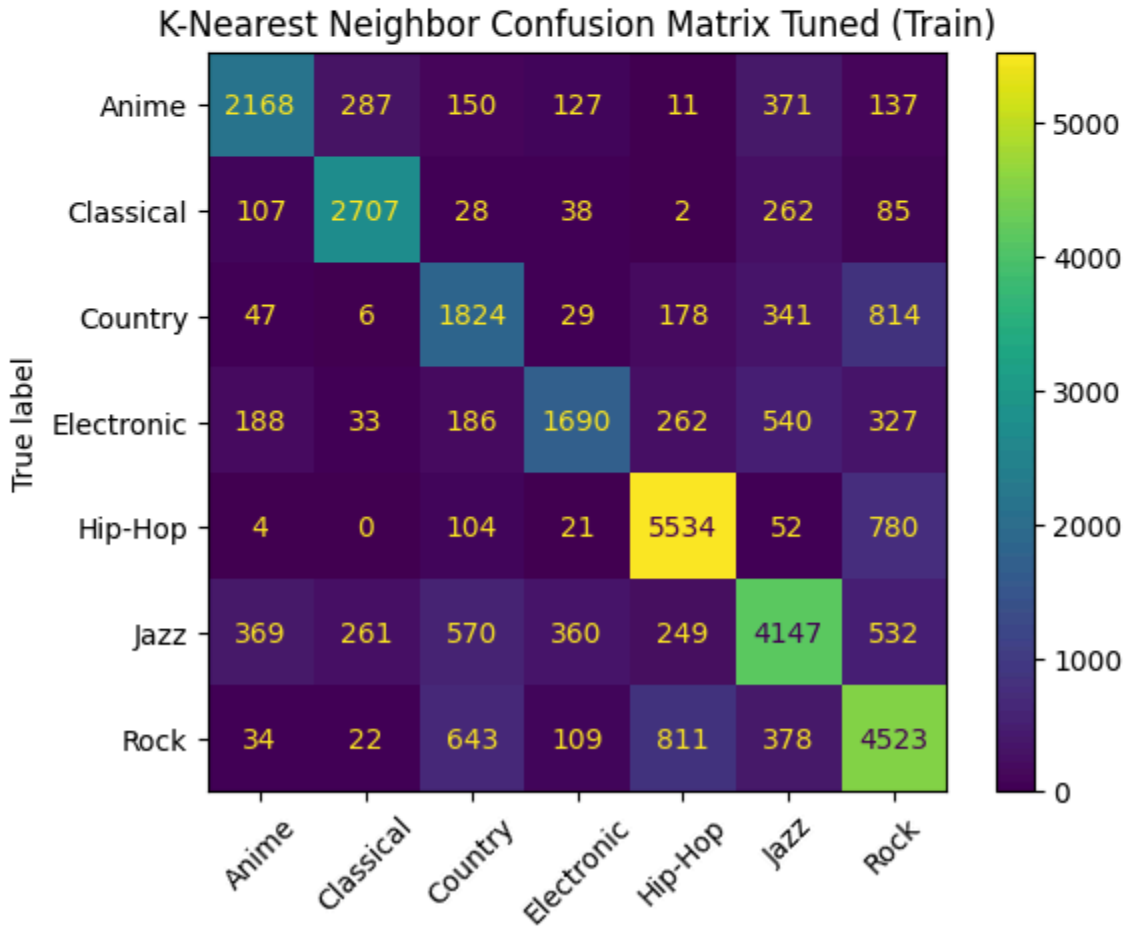
Testing data post-genre merge

```
Test Classification Report (Post-Merge):
              precision    recall  f1-score   support

       Anime       0.66      0.65      0.66       813
   Classical       0.78      0.82      0.80       807
     Country       0.41      0.53      0.46       810
  Electronic       0.59      0.50      0.54       806
     Hip-Hop       0.75      0.83      0.79      1624
        Jazz       0.62      0.55      0.58      1622
        Rock       0.60      0.56      0.57      1630

    accuracy                           0.64      8112
   macro avg       0.63      0.63      0.63      8112
weighted avg       0.64      0.64      0.64      8112

<Figure size 1000x1000 with 0 Axes>
```



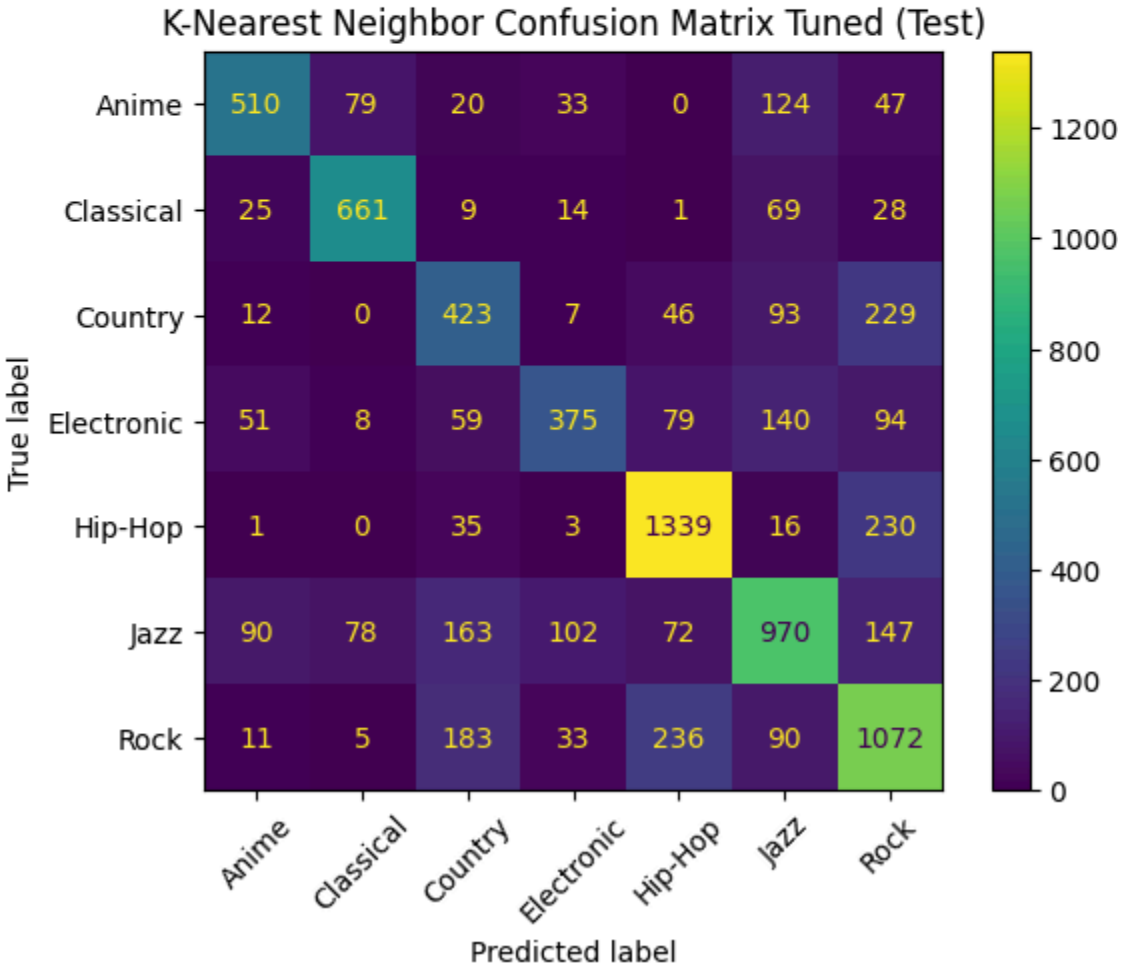
Training data post-genre merge w/ tuning

```
Best number of neighbors (k): 20

Tuned Training Classification Report:
              precision    recall  f1-score   support

       Anime       0.74      0.67      0.70      3251
   Classical       0.82      0.84      0.83      3229
     Country       0.52      0.56      0.54      3239
  Electronic       0.71      0.52      0.60      3226
     Hip-Hop       0.79      0.85      0.82      6495
        Jazz       0.68      0.64      0.66      6488
        Rock       0.63      0.69      0.66      6520

    accuracy                           0.70     32448
   macro avg       0.70      0.68      0.69     32448
weighted avg       0.70      0.70      0.69     32448
```



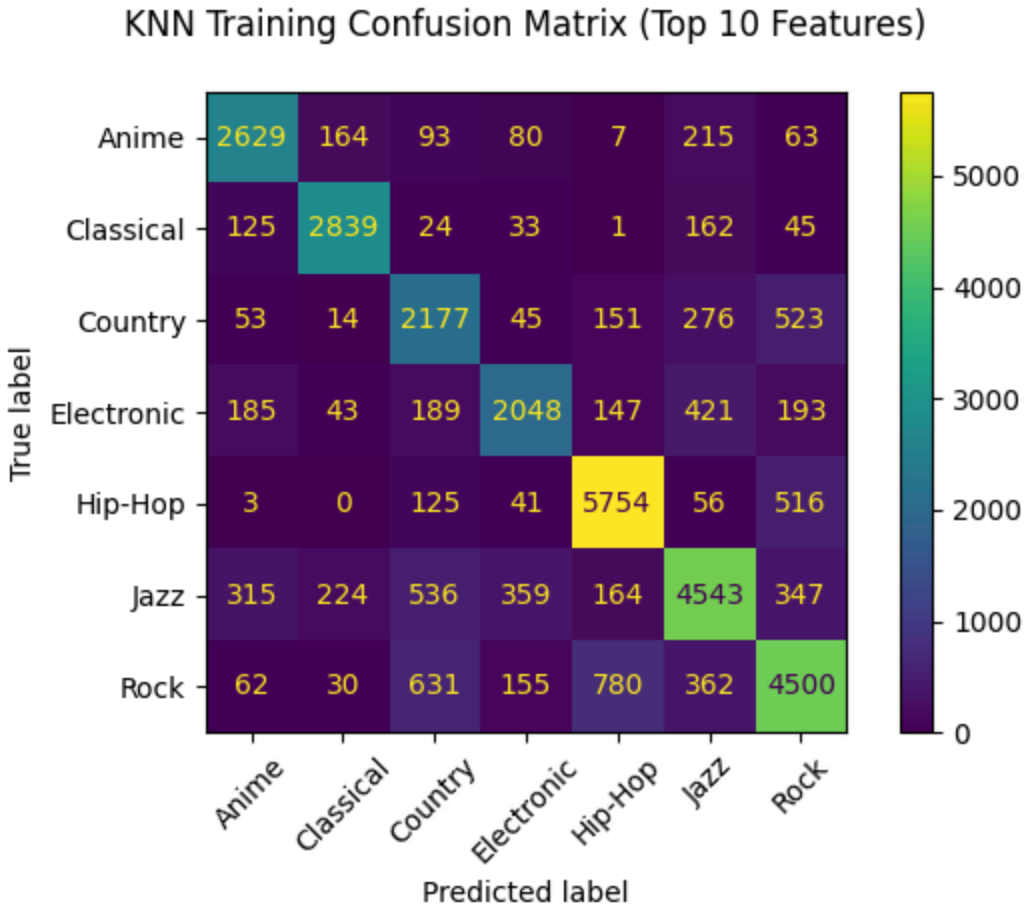
Testing data post-genre merge w/tuning

```
Tuned Test Classification Report:
              precision    recall  f1-score   support

       Anime       0.73      0.63      0.67       813
   Classical       0.80      0.82      0.81       807
     Country       0.47      0.52      0.50       810
  Electronic       0.66      0.47      0.55       806
     Hip-Hop       0.76      0.82      0.79      1624
        Jazz       0.65      0.60      0.62      1622
        Rock       0.58      0.66      0.62      1630

    accuracy                           0.66      8112
   macro avg       0.66      0.64      0.65      8112
weighted avg       0.66      0.66      0.66      8112
```



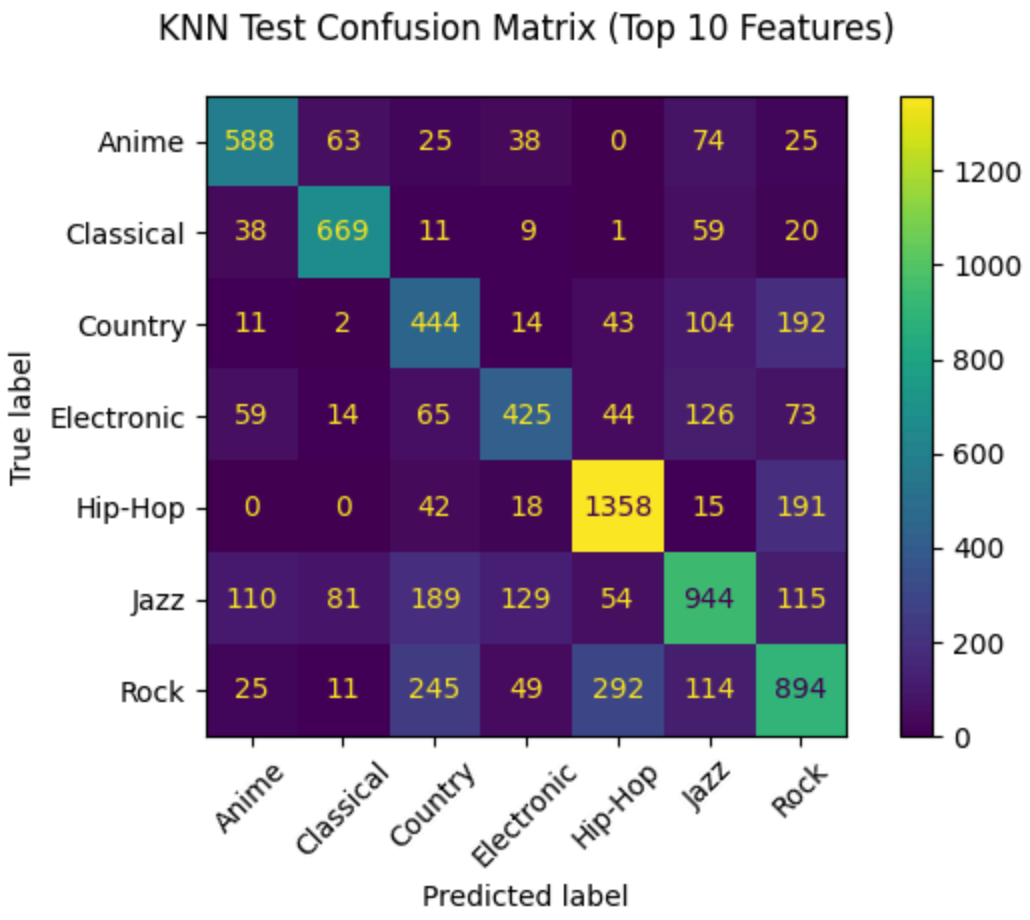
Training with feature selection

```
Classification Report (Training):
              precision    recall  f1-score   support

       Anime       0.78      0.81      0.79      3251
   Classical       0.86      0.88      0.87      3229
     Country       0.58      0.67      0.62      3239
  Electronic       0.74      0.63      0.68      3226
     Hip-Hop       0.82      0.89      0.85      6495
        Jazz       0.75      0.70      0.73      6488
        Rock       0.73      0.69      0.71      6520

    accuracy                           0.75     32448
   macro avg       0.75      0.75      0.75     32448
weighted avg       0.76      0.75      0.75     32448

<Figure size 1200x800 with 0 Axes>
```



Testing with feature selection

```
Classification Report (Test):
              precision    recall  f1-score   support

       Anime       0.71      0.72      0.72       813
   Classical       0.80      0.83      0.81       807
     Country       0.43      0.55      0.48       810
  Electronic       0.62      0.53      0.57       806
     Hip-Hop       0.76      0.84      0.80      1624
        Jazz       0.66      0.58      0.62      1622
        Rock       0.59      0.55      0.57      1630

    accuracy                           0.66      8112
   macro avg       0.65      0.66      0.65      8112
weighted avg       0.66      0.66      0.65      8112

<Figure size 1200x800 with 0 Axes>
```



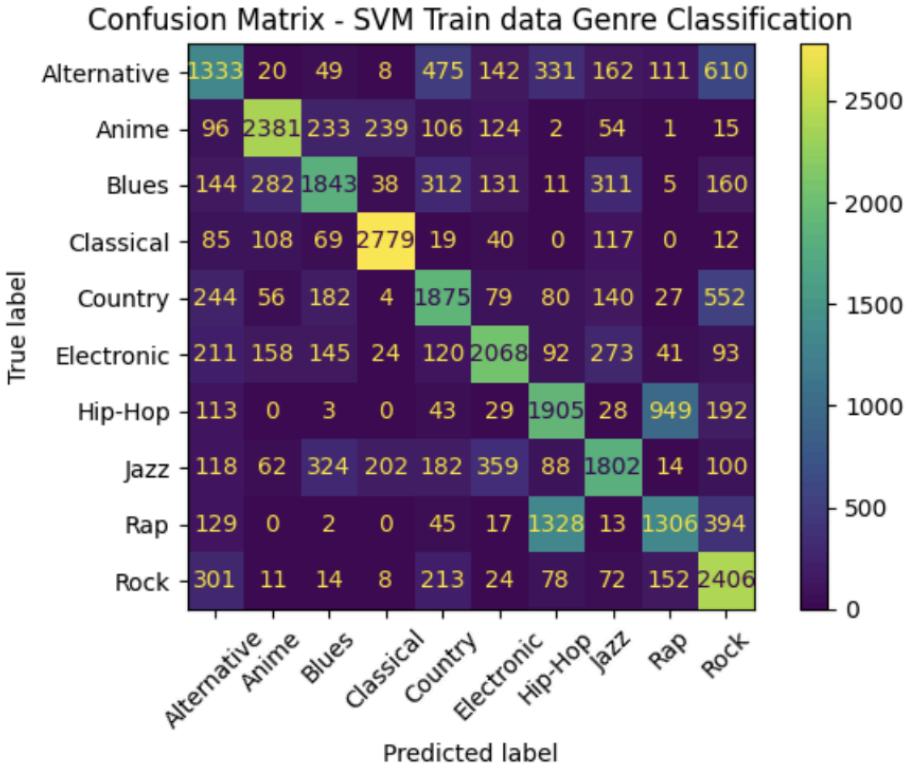
Support Vector Machine

Originally the support vector machine had not performed the best due to not having yet merged a couple of features. After merging features, the results SVM gave

improved tremendously. Before the merging of certain features or tuning the hyperparameters the macro and weighted average of precision in the training data was 61%, that is a lot better than the 52%. The testing data had a precision of 57%, a bit lower. After merging data, the train precision increased to 73% and the testing data to 71%.The merging of similar genres due to the keys and musical notes improved the results drastically. The merging of the similar genre, created a genre with a larger dataset, making it easier for SVM to analyze. SVM works well with large amounts of complex data. After tuning the hyperparameters, the precision of the training dataset was similar to the results after merging. Continuing to adjust the hyperparameters and possibly merging music genres may achieve a higher precision in being able to differentiate music genres.

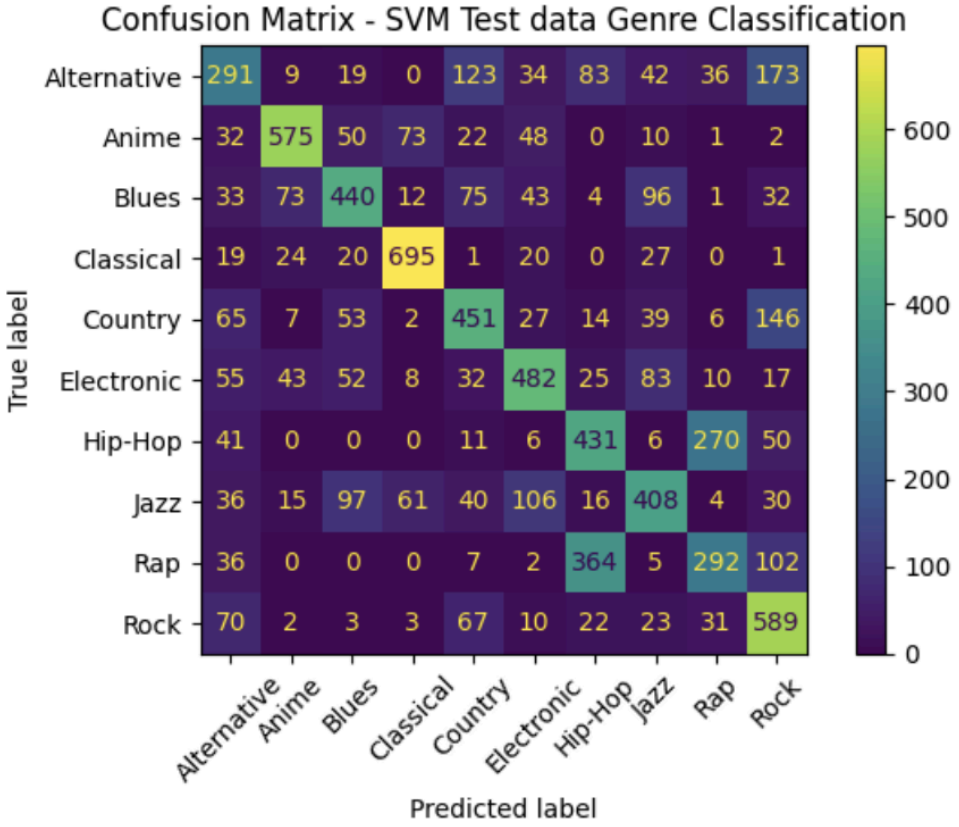
Training data before merging genres

```
Accuracy:  0.6070636094674556
Precision:  0.6101141553506138
Classification report for train data:
               precision    recall  f1-score   support

  Alternative       0.48      0.41      0.44      3241
        Anime       0.77      0.73      0.75      3251
        Blues       0.64      0.57      0.60      3237
    Classical       0.84      0.86      0.85      3229
      Country       0.55      0.58      0.57      3239
   Electronic       0.69      0.64      0.66      3225
      Hip-Hop       0.49      0.58      0.53      3262
         Jazz       0.61      0.55      0.58      3251
          Rap       0.50      0.40      0.45      3234
         Rock       0.53      0.73      0.62      3279

     accuracy                           0.61     32448
    macro avg       0.61      0.61      0.61     32448
 weighted avg       0.61      0.61      0.61     32448
```



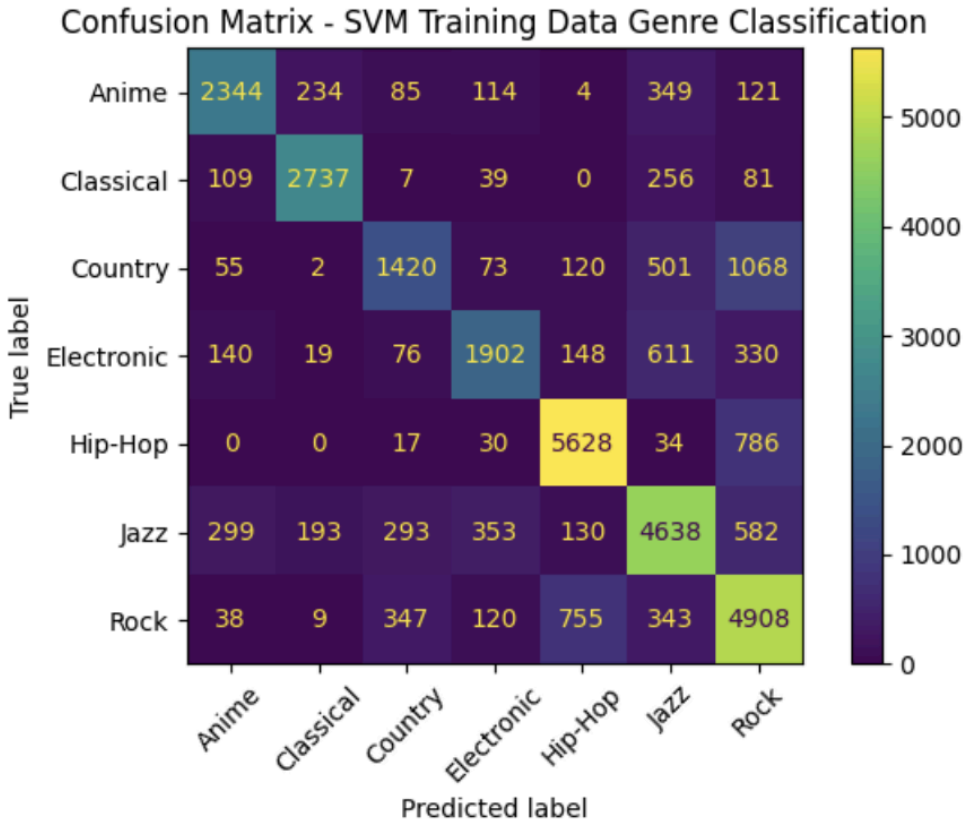
Testing data before merging genres

```
Accuracy:  0.5737179487179487
Precision:  0.5738960268103982
Classification report for test data:
              precision    recall  f1-score   support

 Alternative       0.43      0.36      0.39       810
       Anime       0.77      0.71      0.74       813
       Blues       0.60      0.54      0.57       809
   Classical       0.81      0.86      0.84       807
     Country       0.54      0.56      0.55       810
  Electronic       0.62      0.60      0.61       807
     Hip-Hop       0.45      0.53      0.49       815
        Jazz       0.55      0.50      0.53       813
         Rap       0.45      0.36      0.40       808
        Rock       0.52      0.72      0.60       820

    accuracy                           0.57      8112
   macro avg       0.57      0.57      0.57      8112
weighted avg       0.57      0.57      0.57      8112
```


Confusion Matrix - SVM Test data Genre Classification
Alternative 291 9 19 0 123 34 83 42 36 173
Anime 32 575 50 73 22 48 0 10 1 2
Blues 33 73 440 12 75 43 4 96 1 32
Classical 19 24 20 695 1 20 0 27 0 1
Country 65 7 53 2 451 27 14 39 6 146
Electronic 55 43 52 8 32 482 25 83 10 17
Hip-Hop 41 0 0 0 11 6 431 6 270 50
Jazz 36 15 97 61 40 106 16 408 4 30
Rap 36 0 0 0 7 2 364 5 292 102
Rock 70 2 3 3 67 10 22 23 31 589
True label
Alternative
Anime
Blues
Classical
Country
Electronic
Hip-Hop
Jazz
Rap
Rock
Predicted label
600
500
400
300
200
100
0

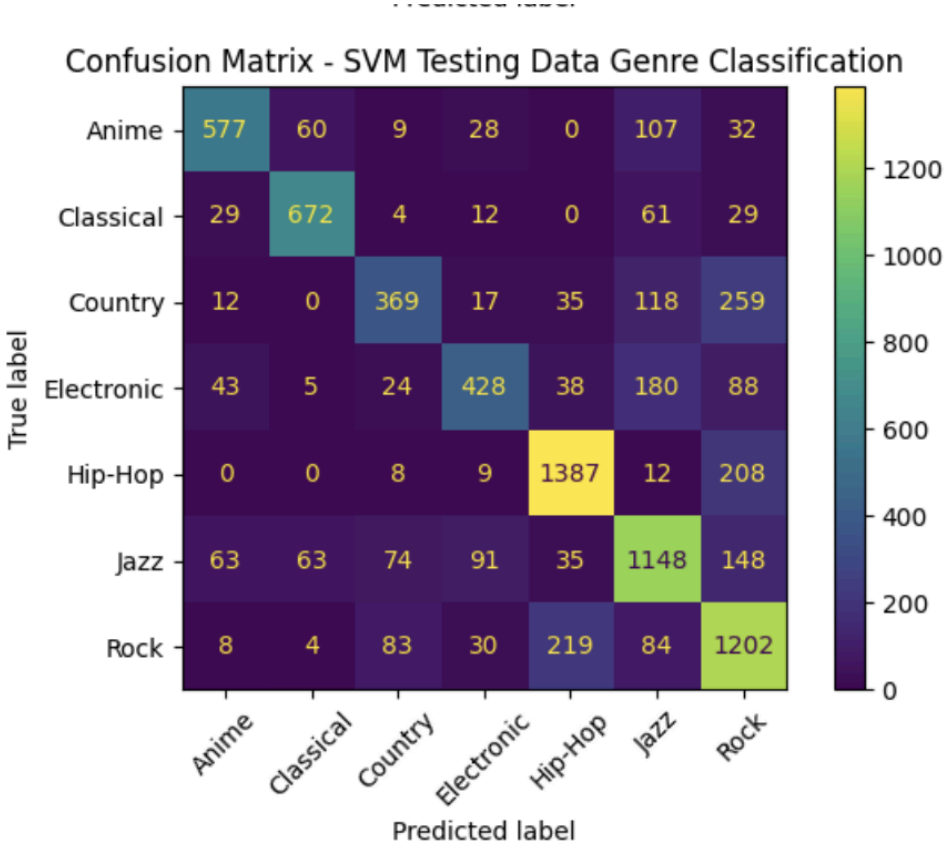
Training data after merging genres

```
Train Accuracy: 0.7266
Test Accuracy: 0.7129
Test Weighted Precision: 0.7149
Test Macro Precision: 0.7226

Train Set Classification report:
              precision    recall  f1-score   support

       Anime       0.79      0.72      0.75      3251
   Classical       0.86      0.85      0.85      3229
     Country       0.63      0.44      0.52      3239
  Electronic       0.72      0.59      0.65      3226
     Hip-Hop       0.83      0.87      0.85      6495
        Jazz       0.69      0.71      0.70      6488
        Rock       0.62      0.75      0.68      6520

    accuracy                           0.73     32448
   macro avg       0.73      0.70      0.71     32448
weighted avg       0.73      0.73      0.72     32448
```



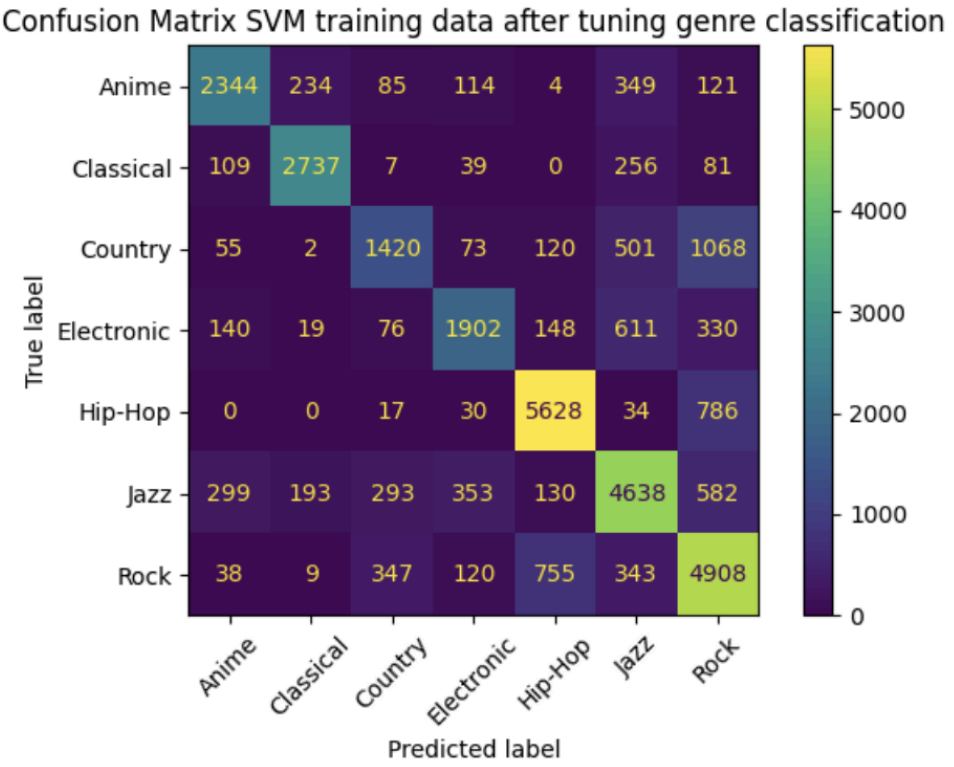
Testing data after merging genres

```
Test Set Classification report:
              precision    recall  f1-score   support

       Anime       0.79      0.71      0.75       813
   Classical       0.84      0.83      0.83       807
     Country       0.65      0.46      0.53       810
  Electronic       0.70      0.53      0.60       806
     Hip-Hop       0.81      0.85      0.83      1624
        Jazz       0.67      0.71      0.69      1622
        Rock       0.61      0.74      0.67      1630

    accuracy                           0.71      8112
   macro avg       0.72      0.69      0.70      8112
weighted avg       0.71      0.71      0.71      8112
```



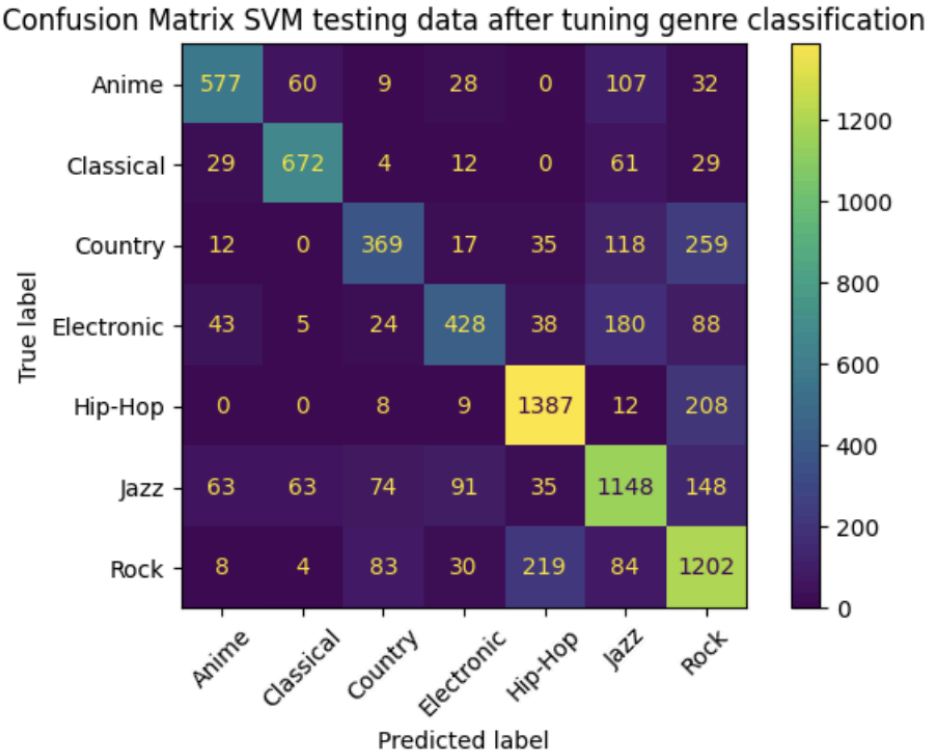
Training data after merge w/tuned hyperparameters

```
Tuned Train Accuracy: 0.7266
Tuned Train Data Weighted Precision: 0.7280
Tuned Train Data Precision: 0.7342

Train Classification Report:
              precision    recall  f1-score   support

       Anime       0.79      0.72      0.75      3251
   Classical       0.86      0.85      0.85      3229
     Country       0.63      0.44      0.52      3239
  Electronic       0.72      0.59      0.65      3226
     Hip-Hop       0.83      0.87      0.85      6495
        Jazz       0.69      0.71      0.70      6488
        Rock       0.62      0.75      0.68      6520

    accuracy                           0.73     32448
   macro avg       0.73      0.70      0.71     32448
weighted avg       0.73      0.73      0.72     32448
```

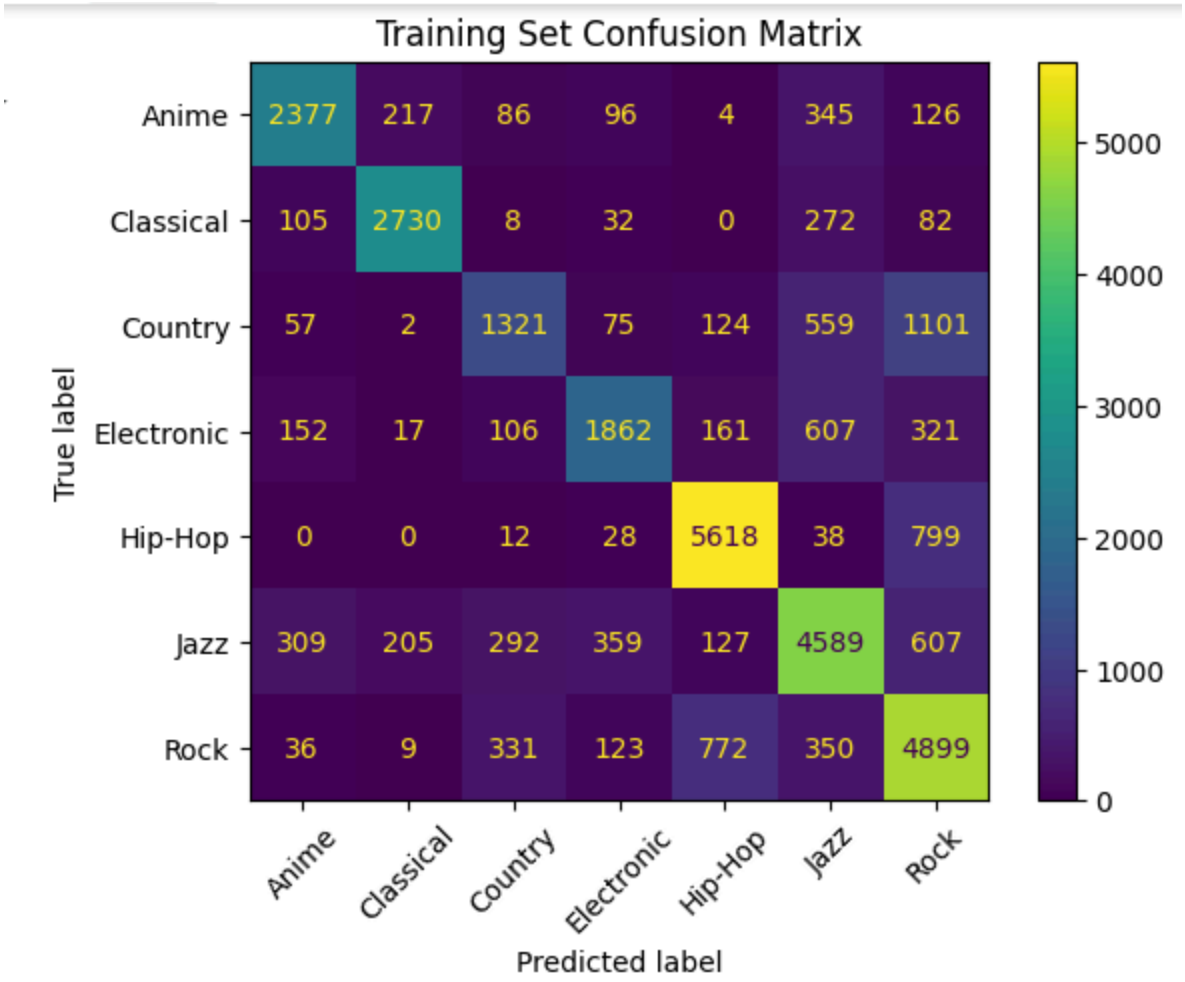


Testing data after merge w/tuned hyperparameters

```
Best SVM parameters: {'kernel': 'rbf', 'gamma': 'auto', 'C': 1}
Tuned Test Accuracy: 0.7129
Tuned Test Data Weighted Precision: 0.7149
Tuned Test Data Precision: 0.7226

Train Classification Report:
              precision    recall  f1-score   support

       Anime       0.79      0.71      0.75       813
   Classical       0.84      0.83      0.83       807
     Country       0.65      0.46      0.53       810
  Electronic       0.70      0.53      0.60       806
     Hip-Hop       0.81      0.85      0.83      1624
        Jazz       0.67      0.71      0.69      1622
        Rock       0.61      0.74      0.67      1630

    accuracy                           0.71      8112
   macro avg       0.72      0.69      0.70      8112
weighted avg       0.71      0.71      0.71      8112
```



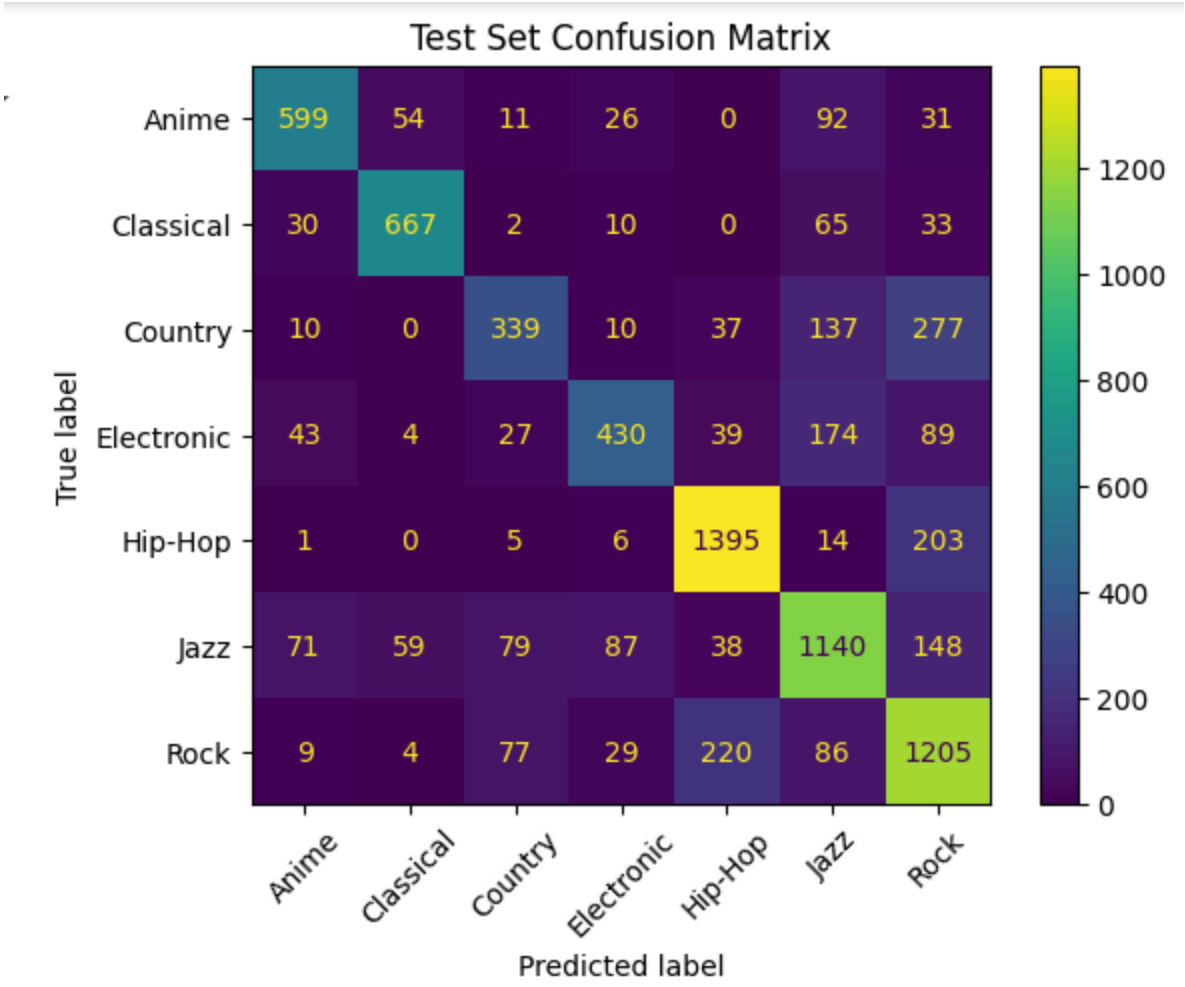
Training data feature selection



Training Set Classification Report:

	precision	recall	f1-score	support
Anime	0.78	0.73	0.76	3251
Classical	0.86	0.85	0.85	3229
Country	0.61	0.41	0.49	3239
Electronic	0.72	0.58	0.64	3226
Hip-Hop	0.83	0.86	0.84	6495
Jazz	0.68	0.71	0.69	6488
Rock	0.62	0.75	0.68	6520
accuracy			0.72	32448
macro avg	0.73	0.70	0.71	32448
weighted avg	0.72	0.72	0.72	32448

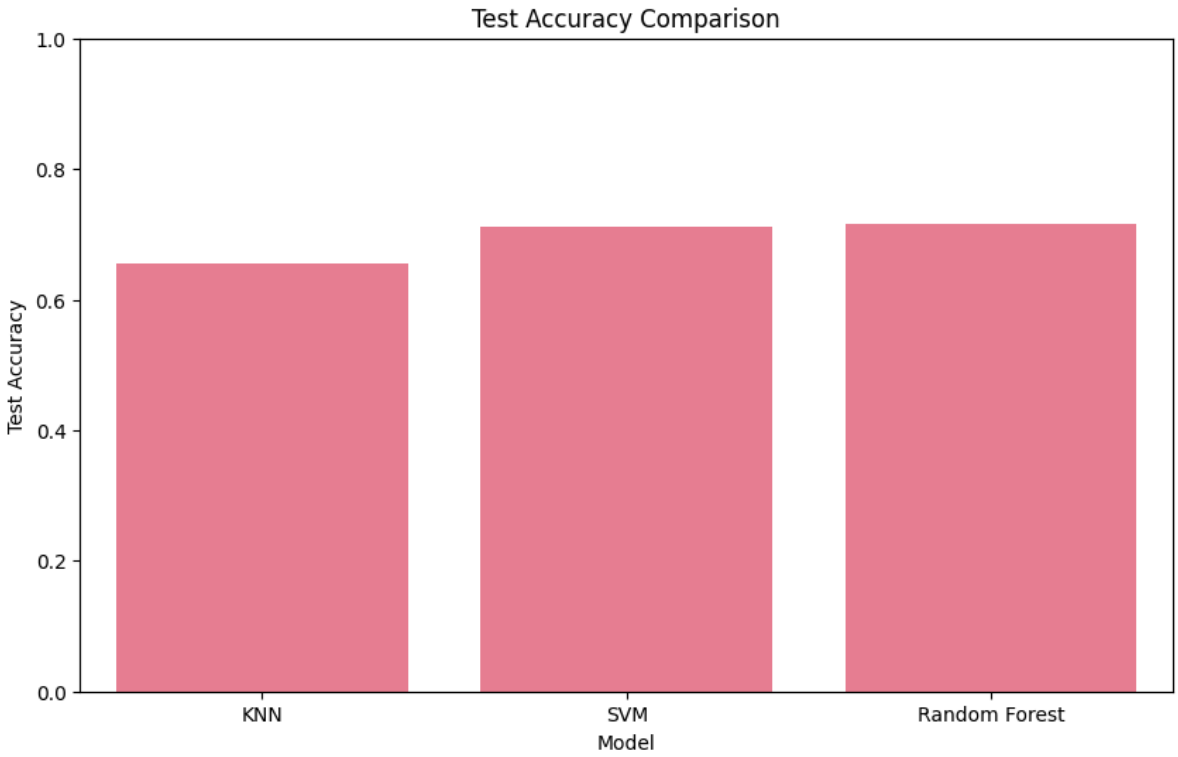
Testing data feature selection

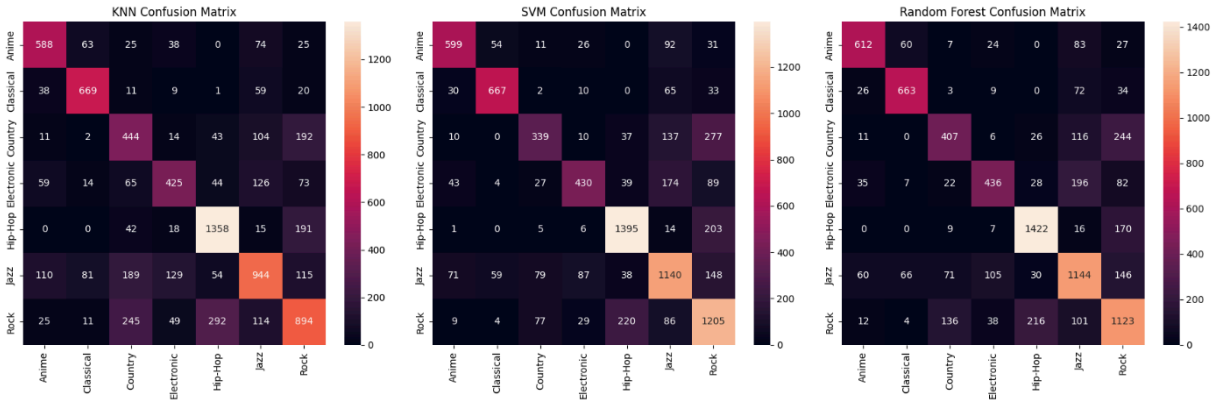


Test Set Classification Report:

	precision	recall	f1-score	support
Anime	0.79	0.74	0.76	813
Classical	0.85	0.83	0.84	807
Country	0.63	0.42	0.50	810
Electronic	0.72	0.53	0.61	806
Hip-Hop	0.81	0.86	0.83	1624
Jazz	0.67	0.70	0.68	1622
Rock	0.61	0.74	0.67	1630
accuracy			0.71	8112
macro avg	0.72	0.69	0.70	8112
weighted avg	0.71	0.71	0.71	8112

Model Comparison






```
=== Comparative Analysis Summary ===
| Model         |   Train Accuracy |   Test Accuracy |   Training Time (s) |   Prediction Time (s) |   Avg F1-Score |
|:--------------|-----------------:|----------------:|--------------------:|----------------------:|---------------:|
| KNN           |         0.754777 |        0.656065 |           0.0974333 |               2.89201 |       0.652263 |
| SVM           |         0.721031 |        0.711908 |             43.7759 |               12.9628 |       0.699219 |
| Random Forest |         0.991217 |        0.715853 |             15.4878 |               0.29692 |       0.707584 |

Key Observations:
1. Accuracy:
   - Best Test Accuracy: 0.72 by Random Forest
   - Most Overfit: 0.28 difference (KNN)

2. Speed:
   - Fastest Training: 0.10s (Random Forest)
   - Fastest Prediction: 0.30s (Random Forest)

3. F1-Score:
   - Best Balanced Performance: 0.71 (SVM)
```

The three algorithms showed distinct trade-offs between accuracy, speed, and generalization. Random Forest achieved the highest test accuracy, demonstrating strong predictive power while maintaining fast prediction times. However, the near-perfect training accuracy is an indication of significant overfitting.

SVM actually came to be the runner-up algorithm for our project. It had the second-highest test accuracy at a staggering 71% and had the best F1-Score, which showed that it handled class imbalances very well. We do believe that there was moderate overfitting due to the training accuracy being at 72.1%.

Finally, KNN surprisingly performed the weakest out of the three algorithms we chose. It has the lowest test accuracy score being 65%. In addition, not only were the prediction speeds slow, but there was overfitting happening as well with this algorithm (75% training accuracy vs 65% test).

Overall, all three algorithms ended up not being terrible choices for our goal. Random Forest is a great option for those who find prediction speed is the most critical aspect, SVM is great for class balance, and KNN needs improvement. If we had more time, we would like to work further to find better tuning parameters and do better feature engineering for more accuracy and precision.

Contribution

Garrett Donohue, Random Forest Classifier

Terra Simon, K-Nearest Neighbors

Yailin Carreno, Support Vector Machine