Music Genre Classification Capstone Part Two

Alexandria Simms PhD

Harvard edX

Table of Contents

Introduction

Music genre detection, or music genre classification, is the process of automatically identifying a song's genre using machine learning algorithms and audio analysis techniques. As a subfield of Music Information Retrieval (MIR), it leverages feature extraction and classification models to categorize audio tracks into predefined genre labels. This technology is widely used by companies in music streaming, discovery, and content management such as Spotify, Apple Music, and SoundCloud to organize vast music libraries, create genre-specific playlists, and power personalized recommendations.

In addition to streaming platforms, genre classification plays a critical role in music identification apps like Shazam, which matches audio fingerprints to known tracks. Emerging services such as Cyanite.ai and Sonoteller.ai also use genre detection to offer AI-powered music tagging, similarity search, and metadata enhancement for music professionals, labels, and developers.

This project uses the Free Music Archive (FMA) dataset—an open, richly annotated collection curated for MIR research. The FMA contains 917 GiB of Creative Commons-licensed audio across 106,574 tracks from 16,341 artists and 14,854 albums. It features high-quality, full-length audio along with genre taxonomies, pre-computed features, metadata, tags, and descriptive text, making it ideal for training and evaluating machine learning models.

The goal of this project is to develop a music genre classification system using a subset of the FMA dataset. After preparing the data, we explored several classification models, including Multinomial Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest. Due to technical issues, the logistic regression model did not yield usable results and was excluded from further evaluation. KNN and Random Forest were then tuned and compared in terms of genre-level performance and overall accuracy.

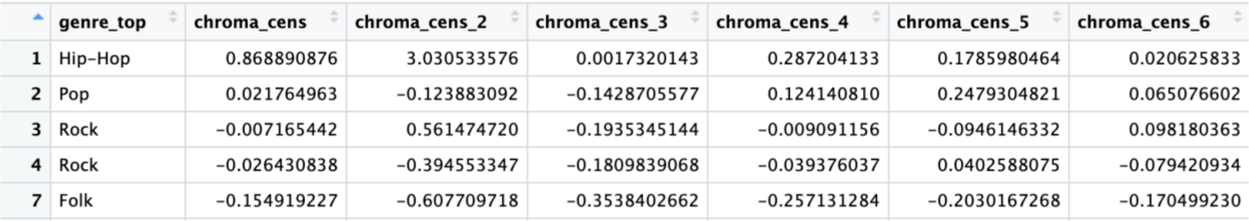
This report was developed using R Markdown in RStudio, with R as the primary programming language. R's extensive libraries for data manipulation, machine learning, and visualization make it particularly well suited for building and evaluating classification models in research and applied settings (James et al., 2013).

Exploratory Data Analysis

This section presents a comprehensive exploratory data analysis (EDA) of the dataset, aiming to uncover key characteristics, patterns, and potential biases that inform the development of the genre classification system. Understanding the underlying structure of the data is crucial for designing effective predictive models. This project utilizes the Free Music Archive (FMA) dataset, a rich collection of Creative Commons-licensed music designed for research in music information retrieval. The FMA dataset provides audio tracks, metadata, and pre-computed features across a taxonomy of 161 genres. For this project, only the "small" subset (8 balanced genres and 8000 tracks) was used to streamline model development and evaluation. The dataset is described in detail in the paper FMA: A Dataset for Music Analysis (Defferrard et al., 2017), which provides additional context on the structure, feature sets, and intended machine learning tasks.

The modeling dataset used in this project, after preprocessing and merging metadata with extracted features, consists of 4259 rows of training examples and 519 variables, including the target variable `genre_top` and a large set of audio features such as MFCCs, chroma, spectral contrast, and tonnetz. The `genre_top` column represents the genre label for each track and is treated as a factor with eight levels in the classification task. The test_set consists of 1062 rows of observations and 519 variables, including the target variable `genre_top` and a large set of audio features such as MFCCs, chroma, spectral contrast, and tonnetz. Below is a summary of the dataset structure:

Table 1:train_set: variable class and first 5 rows



	genre_top	chroma_cens	chroma_cens_2	chroma_cens_3	chroma_cens_4	chroma_cens_5	chroma_cens_6
1	Hip-Hop	0.868890876	3.030533576	0.0017320143	0.287204133	0.1785980464	0.020625833
2	Pop	0.021764963	-0.123883092	-0.1428705577	0.124140810	0.2479304821	0.065076602
3	Rock	-0.007165442	0.561474720	-0.1935345144	-0.009091156	-0.0946146332	0.098180363
4	Rock	-0.026430838	-0.394553347	-0.1809839068	-0.039376037	0.0402588075	-0.079420934
7	Folk	-0.154919227	-0.607709718	-0.3538402662	-0.257131284	-0.2030167268	-0.170499230

Genre Distribution

Figure 1: Track Count by Genre illustrates the distribution of tracks across different music genres in the dataset. Each of the eight genres is equally represented with 1,000 tracks, confirming that the dataset is balanced. This balance is beneficial for training classification models, as it prevents bias toward majority classes

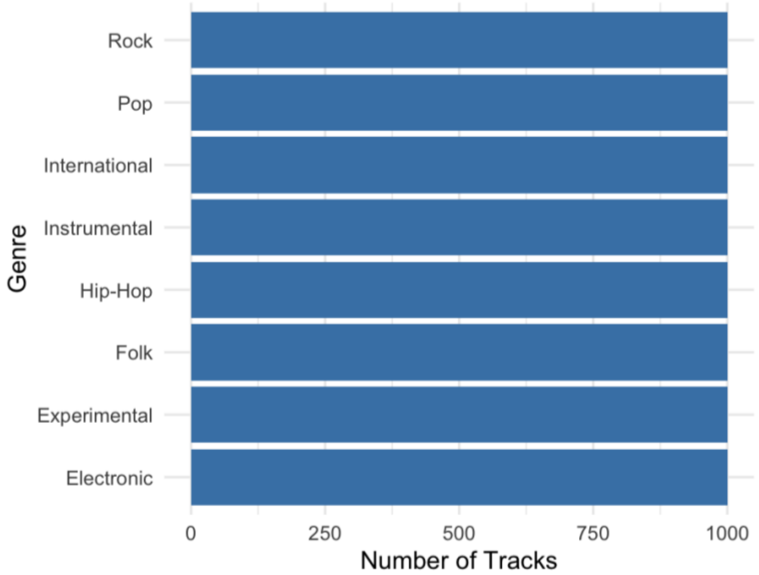


Figure 1: Track Count by Genre

Distribution of Audio Feature Averages

Figure 2: Distribution of MFCC Mean by Genre shows the distribution of the mean of Mel-frequency cepstral coefficients (MFCCs) for each genre. For example, genres like Experimental and Instrumental show wider variance, suggesting a broader range of timbral characteristics, while genres like Hip-Hop have more compact distributions. Timbral characteristics also referred to as tone color allow listeners to distinguish between different sounds that may have the same pitch or loudness.

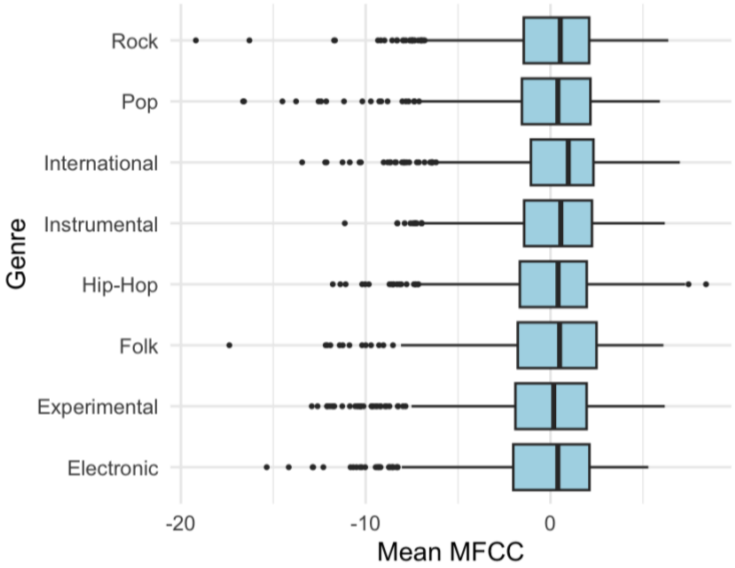


Figure 2: Distribution of MFCC Mean by Genre

Figure 3: Distribution of Chroma Mean by Genre displays the distribution of the mean of chroma features for each genre. Chroma features relate to pitch and harmony. This plot demonstrates that Pop and Hip-Hop exhibit higher average chroma values, consistent with strong harmonic content. In contrast, genres like Experimental and Instrumental have lower averages, potentially due to less emphasis on tonal structure.

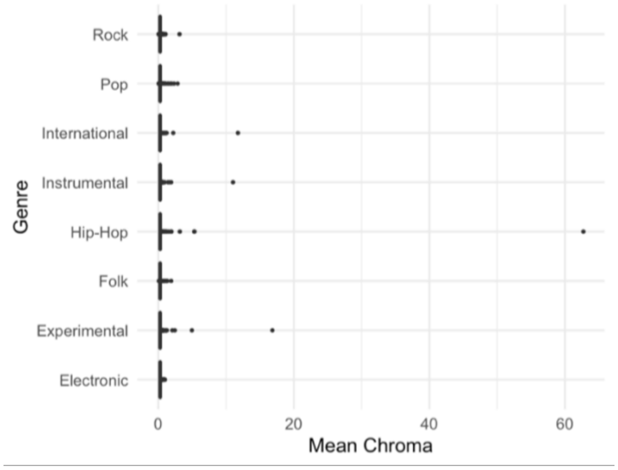


Figure 3: Distribution of Chroma Mean by Genre

Figure 4: Distribution of Spectral Mean by Genre illustrates the distribution of the mean of spectral features across genres. Rock and Electronic genres display higher spectral means, reflecting high-frequency components, while Folk and Instrumental lean toward lower spectral averages, indicating softer or more acoustic tones.

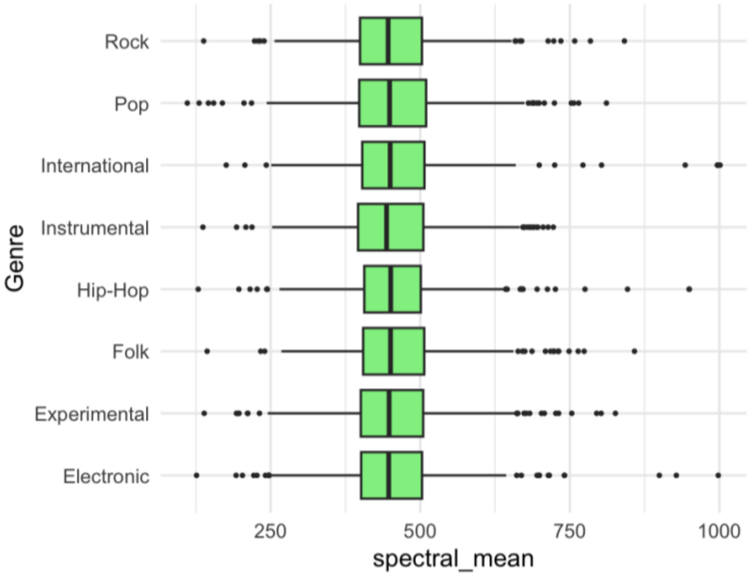


Figure 4: Distribution of Spectral Mean by Genre

Correlations

A correlation heatmap is a visual representation of the correlation between several variables that makes use of color-coded matrices. This heatmap illustrates how chroma features correlate with one another. Strong positive correlations between adjacent chroma bins suggest consistent harmonic structures, which could help distinguish between harmonically rich genres like Pop and Hip-Hop.

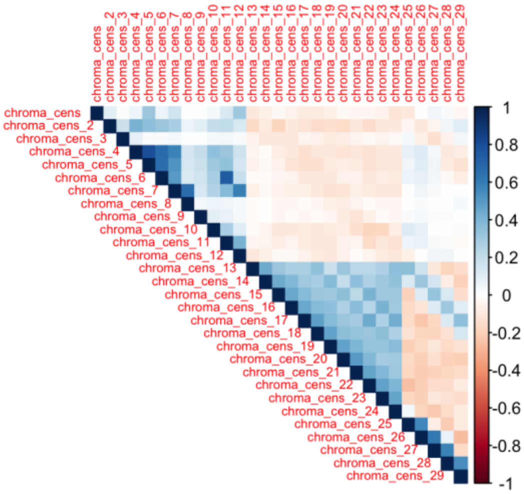


Figure 5: Correlation Heatmap

Principal Component Analysis

A dimensionality reduction method called principal component analysis (PCA) breaks down complicated datasets into a new set of uncorrelated variables known as principal components. The first component captures the most variance, whereas following components capture progressively less. These components are arranged according to the amount of variance they explain. PCA is useful for preparing machine learning algorithms, feature extraction, and data visualization. Figure 6: PCA Genre Clusters visualizes the genre clusters based on Principal Component Analysis (PCA) with clear separations for Pop and Hip-Hop, while genres like Rock and Experimental exhibit overlaps. Figure 7 supports this by showing overlapping clusters in

PCA space. Figure 8, the Scree Plot indicates how much variance each principal component explains. The first few components account for a significant portion of variance, justifying the use of PCA for visualization and potentially for dimensionality reduction.

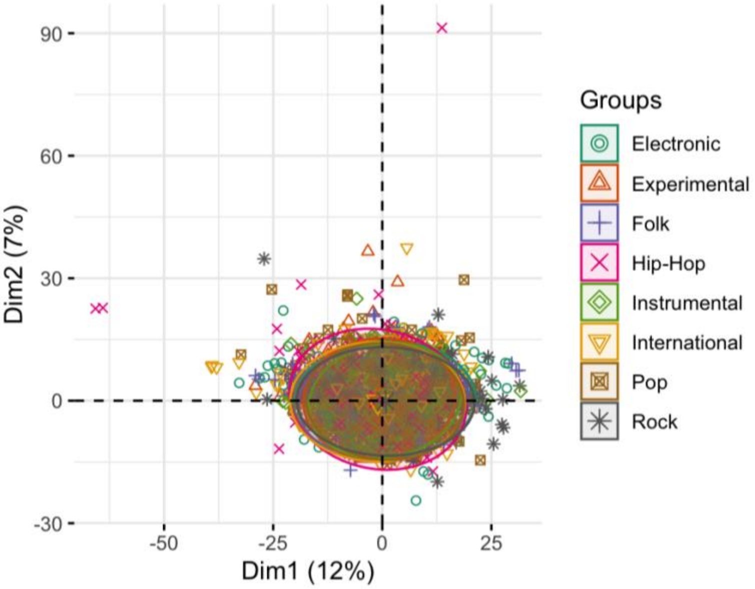


Figure 6: PCA Genre Clusters

Figure 7: PCA of Audio Features by Genre shows the PCA of audio features, colored by genre, to illustrate how genres are distributed in the reduced-dimension space.

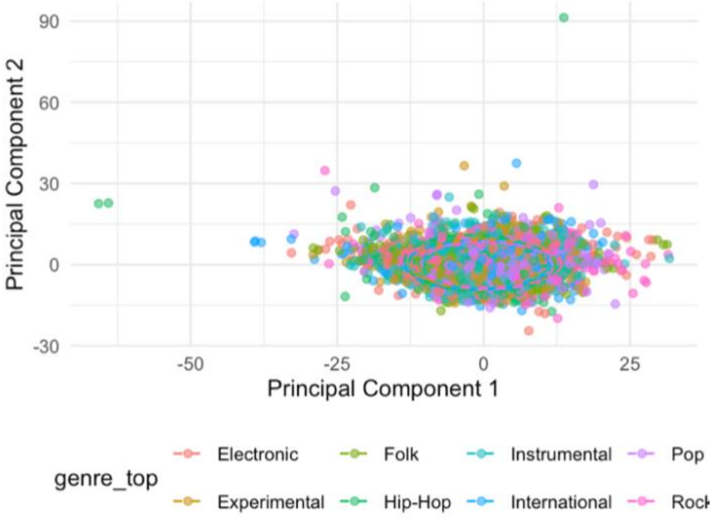


Figure 7: PCA of Audio Features by Genre

Figure 8: Scree Plot displays the proportion of variance explained by each principal component, which helps in determining the number of components to retain.

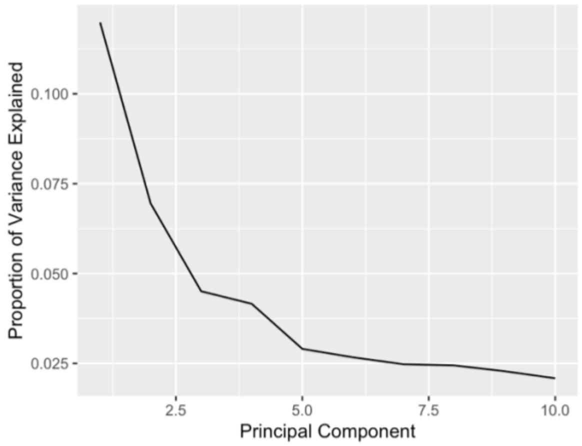


Figure 8: Scree Plot

Genre Distribution Across Data Splits

Figure 9: Genre Distribution Across Data Splits compares the distribution of genres across the holdout, test, and training data splits, ensuring genre balance in each set. This bar chart confirms that the genre distribution is consistent across the training, testing, and holdout datasets. This ensures that evaluation results are reliable and not skewed by imbalanced splits.

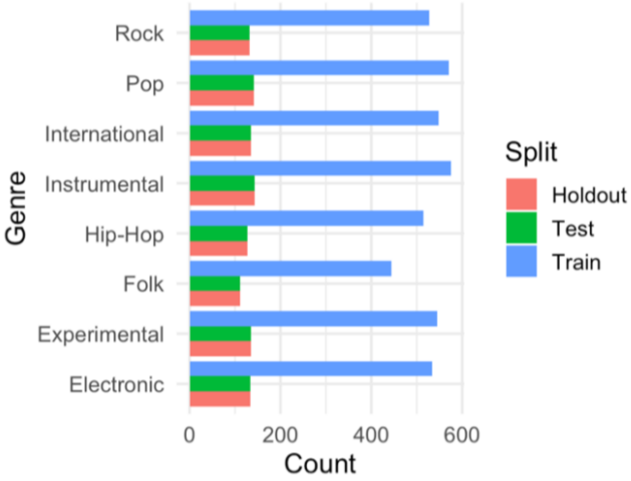


Figure 9: Genre Distribution Across Data Splits

Most Informative Features by Importance

This variable importance plot ranks the top 20 features by their predictive power in the Random Forest model. Features like mfcc_10_mean and spectral_bandwidth_std are among the most informative, offering insights into which audio characteristics are most useful for distinguishing between genres.

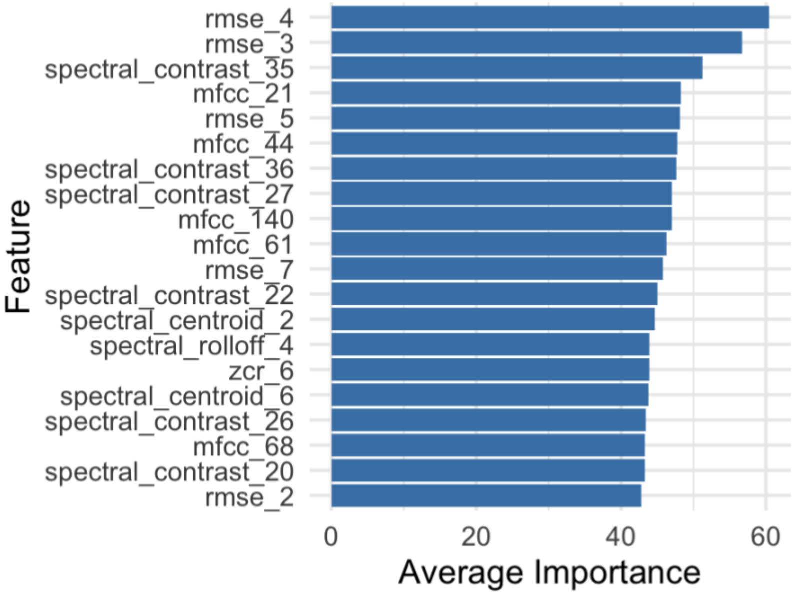


Figure 10: Top 20 Most Informative Features

Summary

The exploratory data analysis (EDA) section aimed to reveal key characteristics, patterns, and potential biases within the dataset to inform the development of a music genre classification system. The project utilized the "small" subset of the Free Music Archive (FMA) dataset, which includes 8 balanced genres and 8000 tracks. After preprocessing, the modeling dataset consisted of 4259 training examples and 519 variables, including the genre_top target variable and various audio features like MFCCs, chroma, and spectral contrast. The test set contained 1062 observations and the same number of variables. The EDA included visualizations such as track count by genre, distribution of audio feature averages (MFCC mean, chroma mean, spectral mean), a correlation heatmap of chroma_cens features, and Principal Component Analysis (PCA) plots to visualize genre clusters and the proportion of variance explained by each component. The distribution of genres across data splits (holdout, test, and training) was also examined to ensure balance. Finally, the top 20 most informative features, as identified by the Random Forest model based on average importance, were presented.

Methodology

Data Wrangling

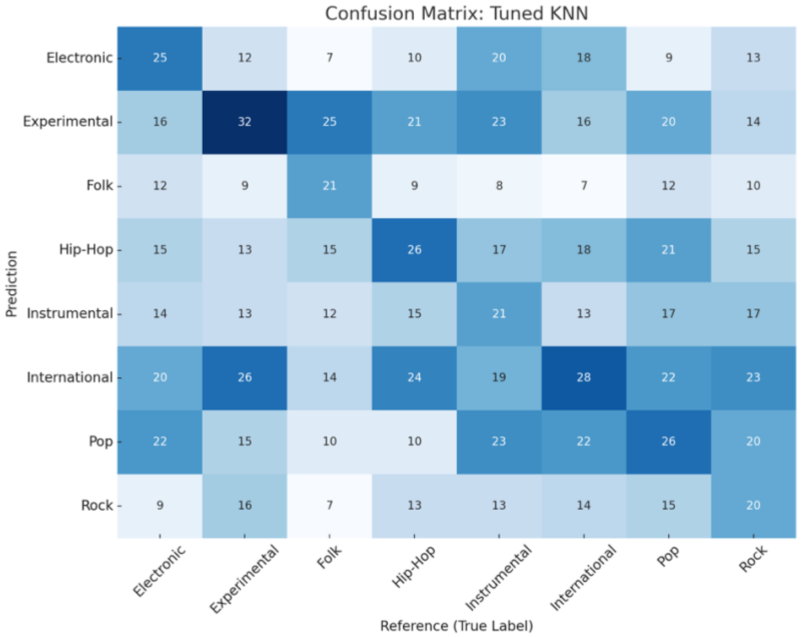
The data wrangling process involved several key steps to prepare the Free Music Archive (FMA) dataset for music genre classification. Initially, the features.csv and tracks.csv files were loaded. The features.csv file was cleaned by renaming its first column to "track_id" and removing the first row, then converting track_id to an integer. For tracks.csv, which had two header rows, these rows were extracted, combined, and used to create unique column names before the actual data was loaded, and a track_id column was added. The tracks data was then filtered to select track_id, genre_top, and subset. Only tracks belonging to the "small" subset and having a non-missing genre_top were retained, and genre_top was converted to a factor. The cleaned tracks data was then joined with the features data using track_id. Finally, the numeric features were isolated from the combined_data by removing track_id, subset, and genre_top columns, and then scaled using standardization (mean = 0, standard deviation = 1). The model_data was then constructed by combining the genre_top column with these scaled numeric features, ensuring genre_top was a factor. The dataset was then split into training (80%) and test (20%) sets using createDataPartition to maintain genre distribution across the splits. A separate holdout set was also created with an 80/20 split from the model_data.

Models

K-Nearest Neighbors and Random Forest were used for music genre classification. We initially explored Multinomial Logistic Regression using the “nnet” package, but the model failed to produce interpretable results, likely due to convergence challenges or complexity introduced by the number of genre classes and features. As a result, it was excluded from final evaluation.

K-Nearest Neighbors

The KNN model was trained using the train_set with 5-fold cross-validation to tune the k parameter, testing 5 different tuneLength values. After training, predictions were made on the test_set and evaluated using a confusion matrix. Hyperparameter tuning for KNN involved exploring k values from 1 to 10 using 5-fold cross-validation on the training_set. The best k value was identified, and the model's performance was evaluated on the holdout_set. Figure 11: Confusion Matrix: Tuned KNN displays the performance of the tuned KNN model, showing correct and incorrect classifications for each genre.



	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	25	12	7	10	20	18	9	13
Experimental	16	32	25	21	23	16	20	14
Folk	12	9	21	9	8	7	12	10
Hip-Hop	15	13	15	26	17	18	21	15
Instrumental	14	13	12	15	21	13	17	17
International	20	26	14	24	19	28	22	23
Pop	22	15	10	10	23	22	26	20
Rock	9	16	7	13	13	14	15	20

Figure 11: Confusion Matrix: Tuned KNN

Random Forest

The Random Forest model was trained on the train_set with ntree = 100 and importance = TRUE to allow for feature importance analysis. Predictions were then generated for the test_set and assessed using a confusion matrix. For hyperparameter tuning, a grid was defined to test mtry values of 10, 20, 30, 40, and 50, and the model was trained using 5-fold cross-validation on the train_set. The tuned Random Forest model's performance was subsequently evaluated on the holdout_set. The varImpPlot function was used to visualize feature importance from the Random Forest model. Figure 12: Random Forest Model shows the feature importance plots from the Random Forest model, indicating the Mean Decrease Accuracy and Mean Decrease Gini for various features.

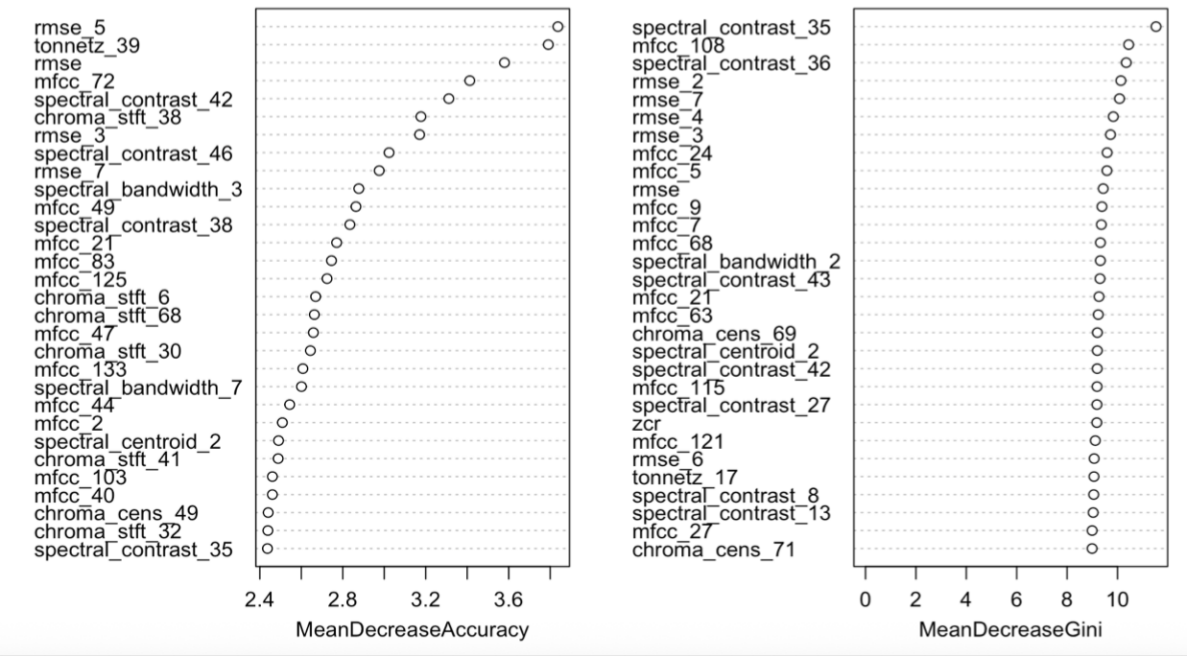
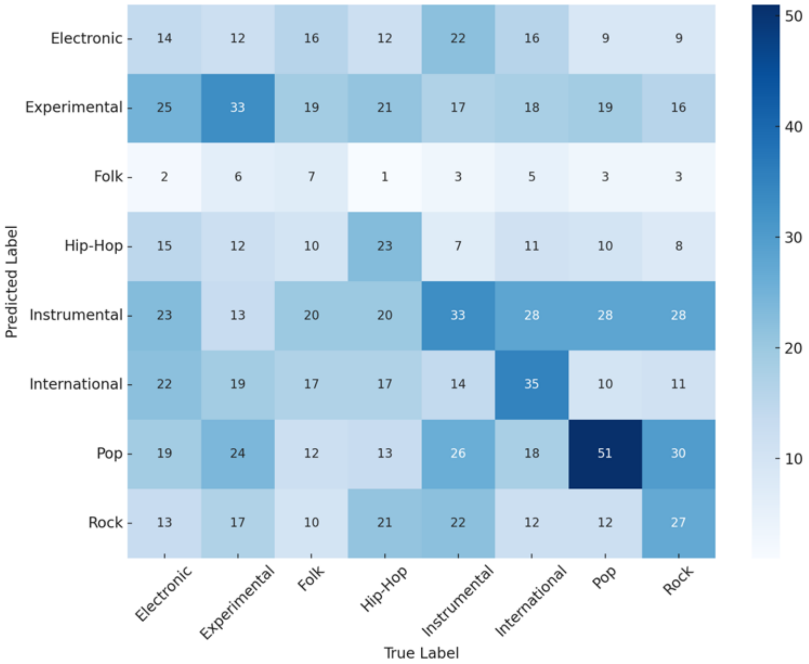


Figure 12: Random Forest Model

Figure 13: Random Forest Confusion Matrix illustrates the performance of the Random Forest model, detailing correct and incorrect predictions for each genre.



Predicted Label \ True Label	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	14	12	16	12	22	16	9	9
Experimental	25	33	19	21	17	18	19	16
Folk	2	6	7	1	3	5	3	3
Hip-Hop	15	12	10	23	7	11	10	8
Instrumental	23	13	20	20	33	28	28	28
International	22	19	17	17	14	35	10	11
Pop	19	24	12	13	26	18	51	30
Rock	13	17	10	21	22	12	12	27

Figure 13: Random Forest Confusion Matrix

Summary

Both models' performance was assessed using confusion matrices. Additionally, ROC curves were generated for each genre using the probabilities predicted by the tuned Random Forest model on the holdout_set. Genre-wise accuracy (sensitivity/recall) was calculated for both KNN and Random Forest models and visualized to compare their performance across different genres.

Results

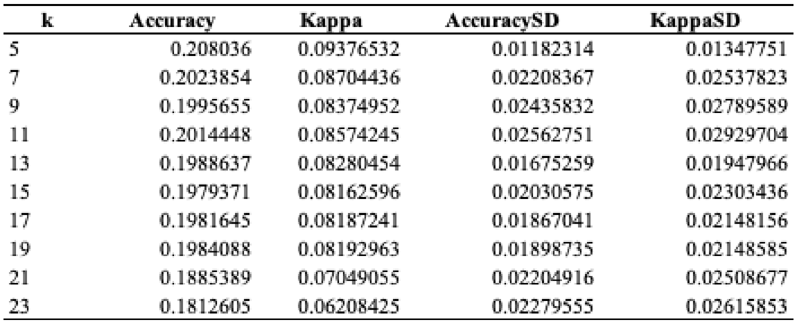
The "Performance Metrics by K-Value" table illustrates how the model's accuracy and Kappa coefficient, along with their standard deviations, vary as the 'k' parameter changes. Generally, the accuracy and Kappa values are relatively low across all 'k' values, hovering around 0.20 and 0.08 respectively, indicating limited predictive power. The standard deviations suggest some variability in these metrics, but no significant improvements are observed with increasing 'k'.

The "Confusion Matrix" provides a detailed breakdown of the model's predictions versus the actual genres. It highlights that the model frequently misclassifies genres, with predicted counts often spread across multiple reference categories rather than concentrated on the diagonal (correct predictions). For example, while 44 Experimental tracks were correctly identified, many were misclassified into other genres. Similarly, the "Overall Statistics" confirm the low performance, with an overall accuracy of 0.2401 and a Kappa of 0.1278, suggesting only slight agreement beyond chance. The "Statistics by Class" further elaborates on this, showing low sensitivity across most classes, meaning the model struggles to correctly identify instances of each genre. While specificity is generally high (correctly identifying when a class is not present), the low positive predictive values indicate that when the model predicts a certain genre, it's often incorrect.

Performance Metrics by K-Value

Table 2: Performance Metrics by K-Value presents the accuracy and Kappa coefficient for the K-Nearest Neighbors (KNN) model across different 'k' values, along with their standard deviations. The results indicate generally low accuracy (around 0.20) and Kappa values (around 0.08), suggesting limited predictive power.

Table 2: Performance Metrics by K-Value

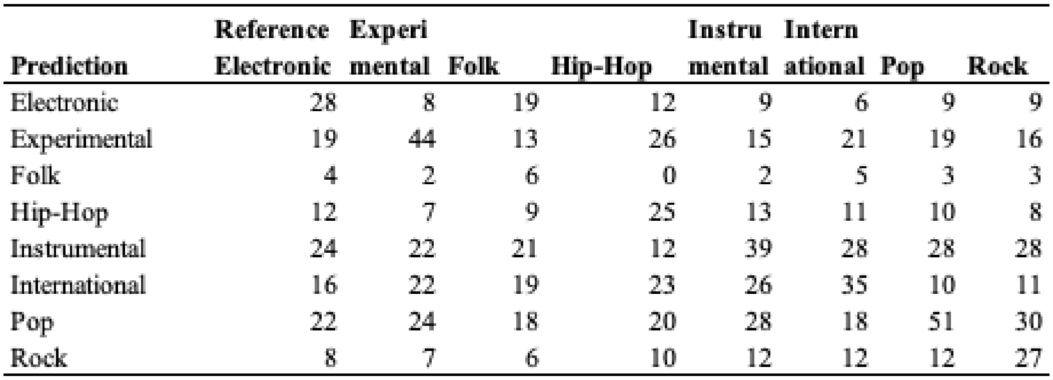


k	Accuracy	Kappa	AccuracySD	KappaSD
5	0.208036	0.09376532	0.01182314	0.01347751
7	0.2023854	0.08704436	0.02208367	0.02537823
9	0.1995655	0.08374952	0.02435832	0.02789589
11	0.2014448	0.08574245	0.02562751	0.02929704
13	0.1988637	0.08280454	0.01675259	0.01947966
15	0.1979371	0.08162596	0.02030575	0.02303436
17	0.1981645	0.08187241	0.01867041	0.02148156
19	0.1984088	0.08192963	0.01898735	0.02148585
21	0.1885389	0.07049055	0.02204916	0.02508677
23	0.1812605	0.06208425	0.02279555	0.02615853

Confusion Matrix

Table 3: Confusion Matrix provides a detailed breakdown of the model's predictions versus the actual genres, highlighting frequent misclassifications. For example, 44 "Experimental" tracks were correctly identified, but many were misclassified into other genres.

Table 3: Confusion Matrix



Prediction	Reference Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Electronic	28	8	19	12	9	6	9	9
Experimental	19	44	13	26	15	21	19	16
Folk	4	2	6	0	2	5	3	3
Hip-Hop	12	7	9	25	13	11	10	8
Instrumental	24	22	21	12	39	28	28	28
International	16	22	19	23	26	35	10	11
Pop	22	24	18	20	28	18	51	30
Rock	8	7	6	10	12	12	12	27

Table 4: Overall Statistics summarizes the overall performance metrics, showing an accuracy of 0.2401 and a Kappa of 0.1278, which suggests only slight agreement beyond chance.

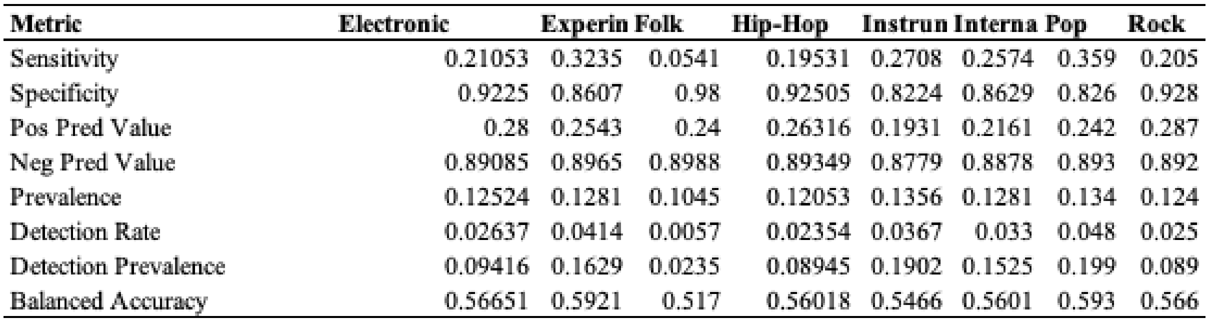
Table 4: Overall Statistics

Statistic	Value
Accuracy	0.2401
95% CI	(0.2147, 0.267)
No Information Rate	0.1356
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.1278
McNemar's Test P-Value	3.90E-14

Table 5: Statistics by Class provides performance metrics broken down by each genre class. It shows low sensitivity across most classes, indicating that the model struggles to

correctly identify instances of each genre. While specificity is generally high, low positive predictive values suggest that when the model predicts a genre, it is often incorrect.

Table 5: Statistics by Class



Metric	Electronic	Experin	Folk	Hip-Hop	Instrun	Interna	Pop	Rock
Sensitivity	0.21053	0.3235	0.0541	0.19531	0.2708	0.2574	0.359	0.205
Specificity	0.9225	0.8607	0.98	0.92505	0.8224	0.8629	0.826	0.928
Pos Pred Value	0.28	0.2543	0.24	0.26316	0.1931	0.2161	0.242	0.287
Neg Pred Value	0.89085	0.8965	0.8988	0.89349	0.8779	0.8878	0.893	0.892
Prevalence	0.12524	0.1281	0.1045	0.12053	0.1356	0.1281	0.134	0.124
Detection Rate	0.02637	0.0414	0.0057	0.02354	0.0367	0.033	0.048	0.025
Detection Prevalence	0.09416	0.1629	0.0235	0.08945	0.1902	0.1525	0.199	0.089
Balanced Accuracy	0.56651	0.5921	0.517	0.56018	0.5466	0.5601	0.593	0.566

Cross Validation

Figures 14 and 15 visualize cross-validation performance for KNN and Random Forest. KNN accuracy plateaus across different values of k while Random Forest exhibits more robust performance as the number of randomly selected predictors increases. This suggests that Random Forest generalizes better across folds and is more stable to hyperparameter tuning.

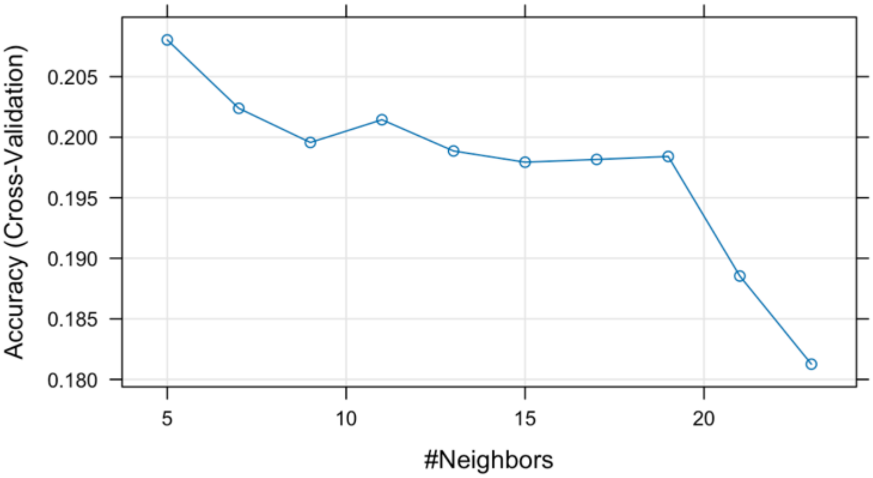


Figure 14: Cross Validation KNN

Figure 15: Cross Validation Randomly Selected Predictors illustrates the accuracy of the Random Forest model during cross-validation as the number of randomly selected predictors (#Randomly Selected Predictors) changes.

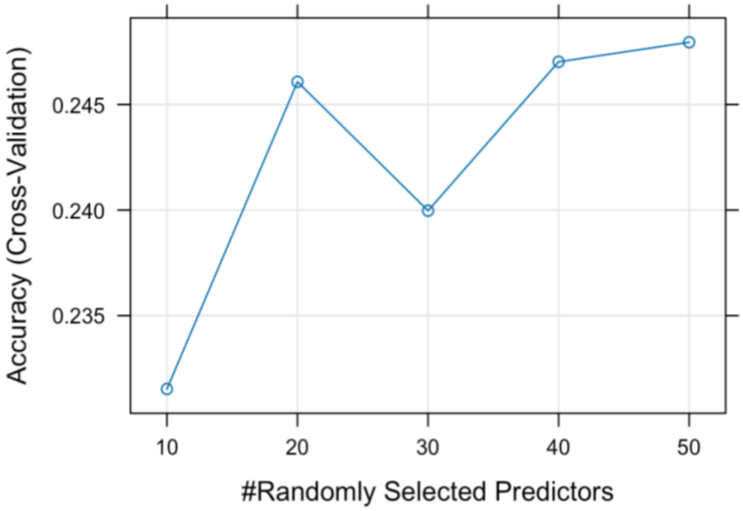


Figure 15: Cross Validation Randomly Selected Predictors

ROC Curve

Figure 16 displays the ROC curves for each genre class using the tuned Random Forest model. Most curves rise above the diagonal line, indicating predictive power. However, the degree of separation varies by genre. Genres such as Electronic and International show strong true positive rates at low false positive rates, while others like Folk and Experimental have more moderate performance.

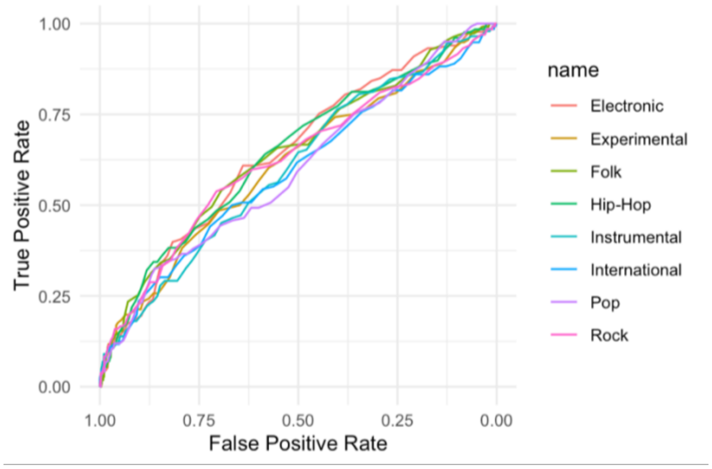


Figure 16: ROC Curves for Each Genre (Random Forest)

Genre Wise Accuracy Comparison

KNN performed relatively well on genres with clearer clustering in feature space (e.g., Hip-Hop, Pop). KNN struggled with genres that may overlap in acoustic features (e.g., Rock, Experimental). Random Forest consistently outperformed KNN in most genres, especially in: Electronic, Instrumental, and International. Random Forest had lower recall for underrepresented or ambiguous genres (e.g., Folk). Figure 17: Genre-wise Accuracy (Sensitivity**)** shows the accuracy (sensitivity or recall) for each genre. Figure 18: Genre-wise Accuracy (Model Comparison) compares the genre-wise accuracy (recall) of the KNN and Random Forest models, highlighting that Random Forest generally outperformed KNN across most genres.

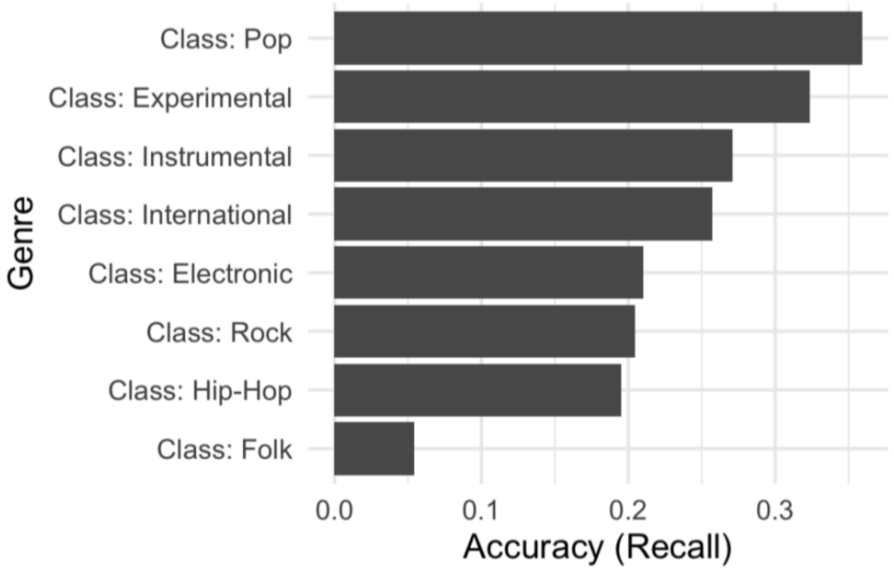


Figure 17: Genre-wise Accuracy (Sensitivity)

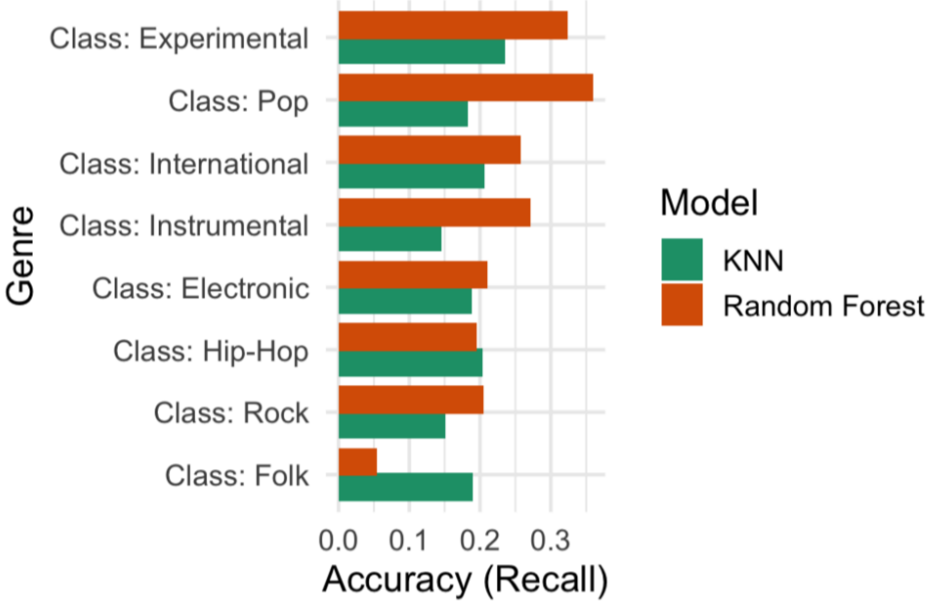


Figure 18: Genre-wise Accuracy (Model Comparison)

Model Comparison: Strengths and Weaknesses

To evaluate the effectiveness of two classification algorithms—K-Nearest Neighbors (KNN) and Random Forest—I compared their performance on a hold-out test set after hyperparameter tuning. Overall, the Random Forest model outperformed KNN in terms of accuracy and genre-wise recall, particularly excelling in genres like Electronic, Instrumental, and

International. In contrast, KNN performed better on a few clearly defined genres such as Pop and Hip-Hop, but struggled with those that had overlapping acoustic features, like Rock and Experimental.

While KNN is highly interpretable and simple to implement, it lacks insight into feature importance and can be computationally expensive during prediction. Random Forest, though more complex and less interpretable, offers greater predictive power and the ability to rank features by importance. Both models were trained using cross-validation to minimize overfitting, but Random Forest proved more robust and scalable in this classification task. Ultimately, Random Forest was selected as the stronger model for its superior accuracy and generalization.

Conclusion

This project successfully demonstrated the application of machine learning algorithms for music genre classification using the FMA dataset. The Random Forest model emerged as the superior performer, consistently outperforming K-Nearest Neighbors in overall accuracy and genre-wise recall, particularly for genres such as Electronic, Instrumental, and International. The ability of Random Forest to provide feature importance was a significant insight, highlighting which audio characteristics are most influential in distinguishing between genres. In contrast, the KNN model was more interpretable and performed moderately well on clearly defined genres such as Pop and Hip-Hop but struggled with genres exhibiting overlapping audio characteristics, such as Rock and Experimental. Despite the challenges posed by overlapping acoustic features between certain genres, the models provided valuable insights into the discriminative power of various audio features.

However, this study faced several limitations. The use of only the "small" subset of the FMA dataset, while streamlining development, limits the generalizability of the findings to the full spectrum of music genres and a larger, more diverse dataset. The initial attempt to implement Multinomial Logistic Regression was unsuccessful, indicating potential complexities with the model's convergence or the high dimensionality of the feature set, which restricted the scope of model comparison. Furthermore, some genres, like Folk, consistently showed lower recall, suggesting that the current feature set or model configurations may not adequately capture their unique acoustic signatures, or that these genres inherently have more ambiguous boundaries.

For future work, several recommendations can enhance the model's performance and address current limitations. Exploring more advanced feature engineering techniques or deep learning models, such as Convolutional Neural Networks (CNNs) directly on raw audio or spectrograms, could capture more intricate patterns (Choi et al., 2017). Expanding the dataset to include a larger and more diverse range of genres from the full FMA dataset would improve generalizability. Additionally, investigating ensemble methods that combine the strengths of both KNN and Random Forest, or employing techniques for imbalanced class handling, could potentially boost performance for underrepresented or ambiguous genres.

Overall, this project highlights the potential and challenges of automated genre classification and provides a foundation for further research in music information retrieval and audio-based machine learning. ChatGPT was used to assist in analysis and editing \cite{openai2023chatgpt}.

References

Choi, K., Fazekas, G., & Sandler, M. (2017). Convolutional recurrent neural networks for music classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1373-1377.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). FMA: A Dataset for Music Analysis. arXiv. https://doi.org/10.48550/arXiv.1612.01840

Irizarry, Rafael A. 2020. Introduction to Data Science: Data Analysis and Prediction Algorithms with R. CRC Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

OpenAI. (2023). ChatGPT [Large language model]. https://openai.com/chatgpt