Intro

With the ever-growing availability of digital music libraries and streaming platforms, the ability to automatically categorize music into distinct genres has become increasingly important. By accurately classifying genres, music genre classification can facilitate efficient music organization, recommendation systems, and personalized user experiences.

For the dataset from Kaggle, music genres were characterized by artist name, track name, popularity, acousticness, danceability, duration in ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, date obtained, and valence. To go more in-depth on some of the characteristics that may be confusing are:

- Popularity is how popular is the song
- Acousticness is a confidence measure from 0.0 to 1.0 that sees whether the track is acoustic, where 1.0 represents high confidence 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.
- Duration in ms is the time length of the song, recorded in milliseconds
- Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Elements that contribute to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
- Instrumentalness predicts whether a track contains no vocals. Sounds such as "ooh" and "aah" are treated as instrumental.
- Liveness detects the presence of an audience in the recording, determining whether the track was performed live or not.
- Loudness is the overall loudness of a track in decibels (dB).
- Mode indicates whether the song is Major or Minor
- Speechiness detects the presence of spoken words in a track.
- Valence is a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive, while tracks with low valence sound more negative.

Look at these elements, some of them are overlapping such as energy and danceability. Using traditional approaches for genre classification relied on manual annotation by human experts, which can be time-consuming, expensive, and prone to inconsistencies. By using machine learning techniques, I am hoping to understand current music recommender algorithms and apps such as Shazam that are able to easily detect what song is being played. By leveraging different feature extraction methods and models, I am hoping to learn to identify and capture intricate patterns and characteristics that define different music genres. However, developing accurate genre classification models will be a challenge due to factors such as the diversity of musical styles and potential genre fusions.

Methods

The dataset for this music genre classification task was obtained from Kaggle, consisting of 50,000 track samples with 18 feature columns spanning the years 2001 to 2005. The focus was on classifying tracks into ten genres: alternative, anime, blues, classical, country, electronic, hip-hop, jazz, rap, and rock.

Data Preprocessing:

Using the dataset, there were several issues identified in the raw dataset, including the presence of '?', empty fields, and '-1' values. These were handled by replacing '?' and '-1' with 'nan' values, which were then imputed using the KNNImputer from scikit-learn. The KNNImputer replaces missing values with the average of the five nearest neighbors, mitigating potential issues during the training process. Label encoding was applied to convert the music genres, mode, and key features into integer values, facilitating pattern recognition and classification.

The preprocessed dataset was partitioned into separate training and test sets.The train-test split was performed using a stratified approach, which preserves the class distributions in both subsets. Specifically, a 80/20 split was employed, allocating 80% of the data for training the models and reserving the remaining 20% as a held-out test set for unbiased performance evaluation.

Exploratory Data Analysis:

To gain insights into the predictor distributions, histograms were generated to visualize the scaling of each feature. This step aided in understanding the characteristics of the dataset before proceeding with feature engineering and model selection.

Feature Engineering:

Initially, features based on artist names were explored, considering their potential relevance for classifying genres like hip-hop and rap. Attention then shifted to the track name column, where specific characters were analyzed for potential patterns contributing to genre classification.

Model Selection and Evaluation:

Inspired by discussions and code from the Kaggle dataset page, several models were considered for this task:

1. Light Gradient Boosting Machine (LightGBM): Chosen for its ability to handle large datasets and high-dimensional features efficiently, which aligned with the project's dataset characteristics.
2. Extreme Gradient Boosting (XGBoost): Selected for comparison with LightGBM, as both are gradient boosting models with differences in tree traversal methods.
3. Random Forest Classifier: Included to benchmark performance against the more advanced gradient boosting models, despite its generally slower training speed.
4. K-Nearest Neighbors (KNN): Considered due to its simplicity and potential suitability for the dataset, where genres were observed to be clustered together without shuffling.

Proper evaluation metrics and cross-validation techniques were employed to assess the performance of these models objectively and select the most appropriate approach for the music genre classification task. Given the multi-class nature of the music genre classification problem, the F1-macro score was chosen as the primary evaluation metric. The F1-macro score calculates the unweighted mean of the F1-scores across all classes, placing equal importance on each genre. This metric provides insights into the model’s ability to accurately classify positive instances for every class, ensuring a comprehensive assessment of performance across the diverse set of genres.

To gain deeper insights into the models' performance, Receiver Operating Characteristic (ROC) curves and confusion matrices were also analyzed. ROC curves plot the true positive rate against the false positive rate at various classification thresholds, providing a visualization of the tradeoff between sensitivity and specificity. The area under the ROC curve (AUC) serves as a summary statistic, with higher values indicating better overall performance. Confusion matrices, on the other hand, offer a more granular view by displaying the number of correct and incorrect predictions for each class. These matrices highlight the specific genres that the models struggle with, aiding in the identification of potential areas for improvement.

By leveraging these comprehensive evaluation techniques, a thorough and multi-faceted assessment of the models' capabilities was conducted. This approach ensured that the most suitable model was selected for the music genre classification task, balancing predictive performance across all genres while accounting for potential class imbalances and misclassification patterns.

Results

During the feature engineering phase, the potential impact of artist names on music genre classification was explored. However, no strong patterns or correlations between artist names and specific genres were observed. These features did not significantly improve the model's performance and were ultimately excluded from the final feature set. The decision to remove artist name features was further reinforced by the presence of empty fields in the data, which could not be reliably imputed using the KNNImputer technique for string data, as it would likely introduce inaccuracies.

In contrast, the analysis of track names yielded promising results. A notable correlation was identified between the presence of the special character ':' in track names and the classical music genre. This pattern suggested that the track name feature could potentially contribute valuable information for distinguishing classical music from other genres. Consequently, the track name feature was retained and incorporated into the final set of features used for training the classification models.

Upon finalizing the feature set through rigorous feature engineering and selection, the focus shifted to evaluating the performance of various machine learning models for the music genre classification task:

1. KNN: The K-Nearest Neighbors (KNN) model was evaluated using the F1-macro score, which assesses the unweighted average of F1-scores across all classes. The KNN model achieved an F1-macro score of 0.5265, indicating a modest improvement over random guessing but leaving substantial room for performance enhancements in accurately classifying the diverse array of music genres. Further analysis of the Receiver Operating Characteristic (ROC) curves, Area Under the Curve (AUC) scores, and confusion matrices revealed insights into the model's strengths and weaknesses. While the overall performance was suboptimal, the KNN model demonstrated a relatively stronger capability in classifying the anime and classical music genres compared to other genres in the dataset.

```
{'knn__n_neighbors': 50, 'knn__weights': 'uniform', 'preprocessing': MinMaxScaler()}
0.5265383773007376
```

Fig 1: Best parameters and F1-macro score for KNN model

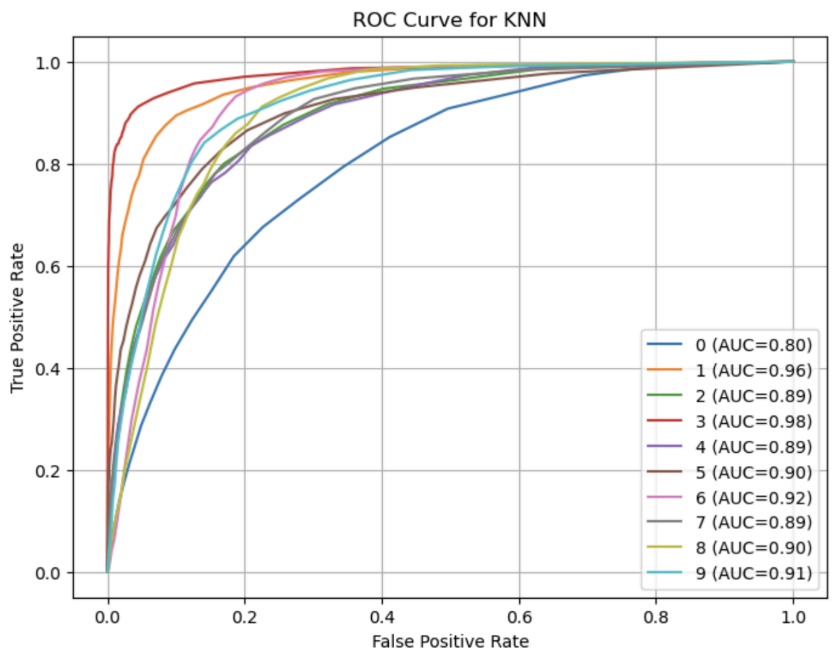

Fig 2: ROC Curve and AUC score for KNN model

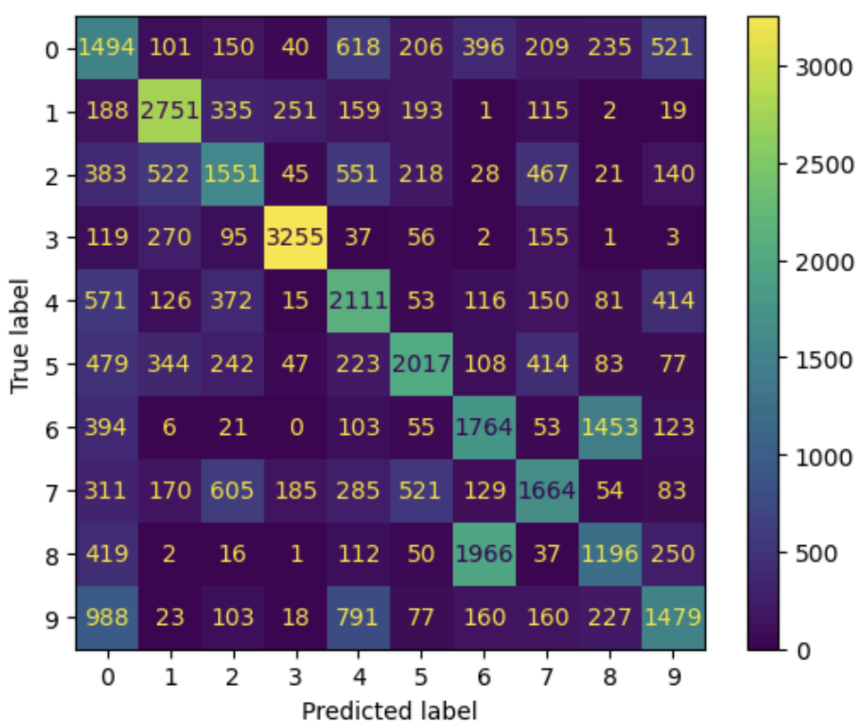

Fig 3: Confusion matrix for KNN model

2. LightGBM: The Light Gradient Boosting Machine (LightGBM) model demonstrated an improvement over the KNN model, achieving an F1-macro score of 0.5910. While this score represents a notable increase in performance, it still falls short of the desired level of accuracy for reliably distinguishing among the diverse range of music genres in the dataset. Further evaluation through the analysis of the Receiver Operating Characteristic (ROC) curves and their corresponding Area Under the Curve (AUC) scores revealed that the LightGBM model exhibited a higher level of discriminative power across all genres compared to the KNN model. Additionally, the confusion matrix provided a more granular view of the model's performance, highlighting an overall increase in the correct classification of various music genres.

```
{'lgbm__colsample_bytree': 0.7, 'lgbm__learning_rate': 0.01, 'lgbm__n_estimators': 300}
0.5910264315589083
```

Fig 4: Best parameters and F1-macro for LightGBM model

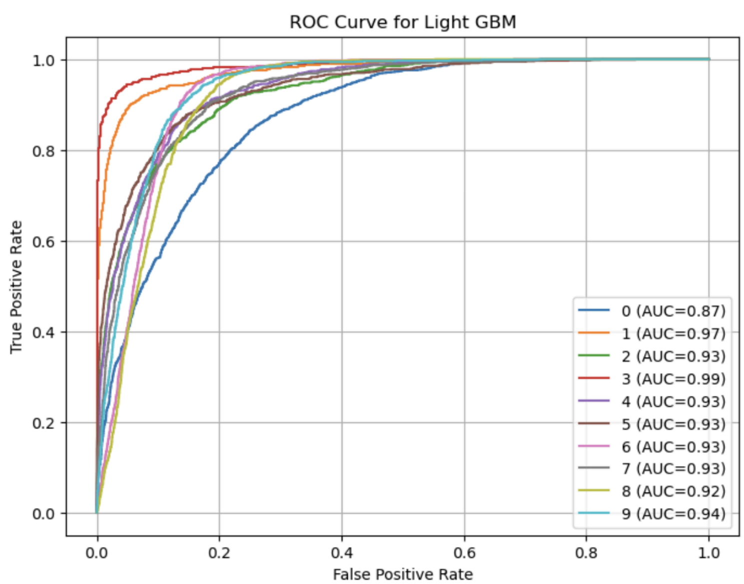

Fig 5: ROC curve and AUC score for LightGBM model

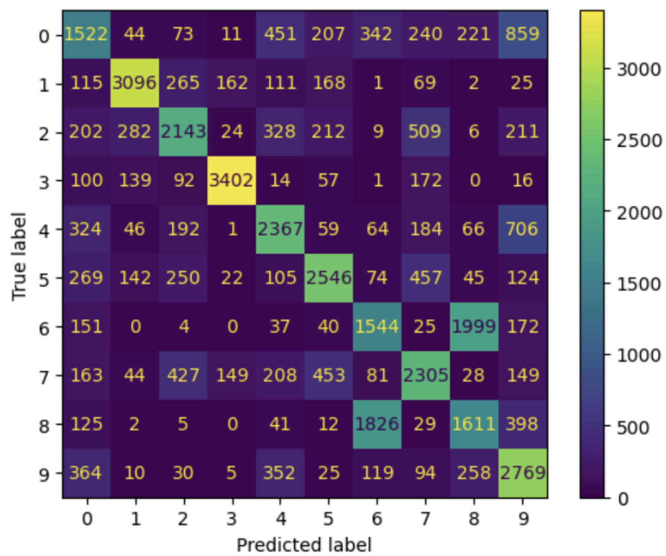

Fig 6: Confusion matrix for LightGBM model

3. XGBoost: The Extreme Gradient Boosting (XGBoost) model was evaluated alongside the other approaches. Its performance, as measured by the F1-macro score of 0.5808, was comparable to the Light Gradient Boosting Machine (LightGBM) model, albeit with a slight decrease in overall accuracy. A detailed analysis of the Receiver Operating Characteristic (ROC) curves and their corresponding Area Under the Curve (AUC) scores revealed a marginal reduction in the discriminative power of the XGBoost model compared to LightGBM across multiple genres. Furthermore, the confusion matrix provided granular insights, highlighting a moderate decline in the correct classification of certain genres when compared to the LightGBM model.

```
{'xgb__max_depth': 7, 'xgb__min_child_weight': 1, 'xgb__n_estimators': 300}
0.5808544262352711
```

Fig 7: Best Parameters and F1-Macro score for XGBoost model

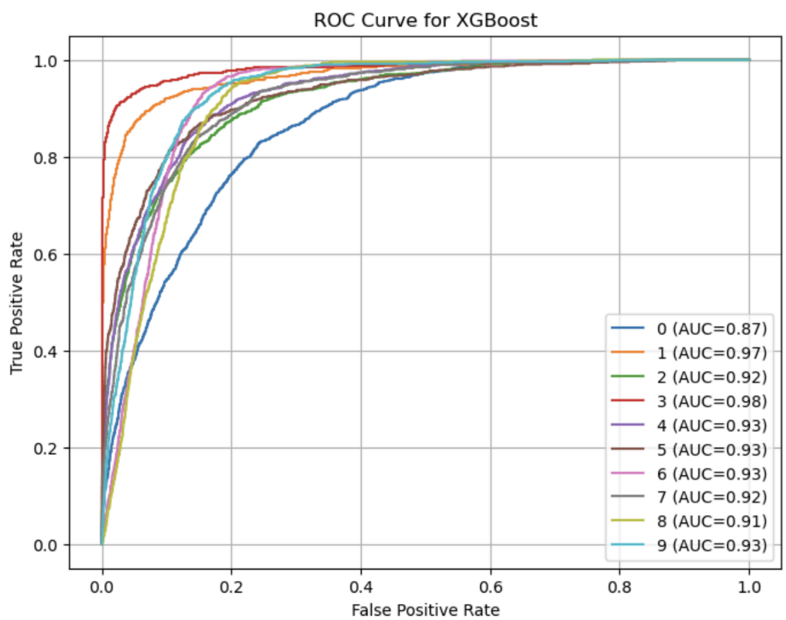

Fig 8: ROC curve and AUC score for XGBoost model

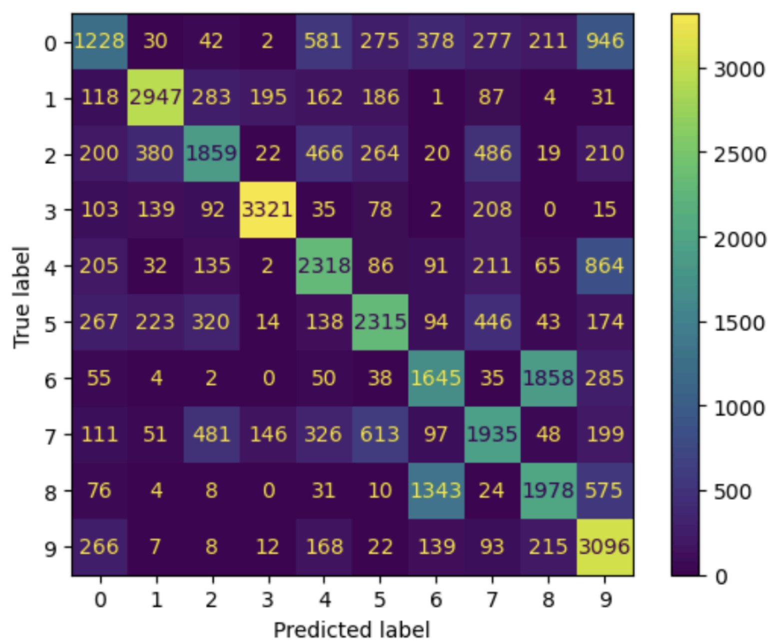

Fig 9: Confusion matrix for XGBoost model

4. Random Forest Classifier: The Random Forest Classifier model was evaluated alongside the gradient boosting and KNN models. It achieved an F1-macro score of 0.5674, outperforming the baseline KNN model but falling short of the performance exhibited by the Light Gradient Boosting Machine (LightGBM) and Extreme Gradient Boosting (XGBoost) models. Further analysis of the Receiver Operating Characteristic (ROC) curves and their corresponding Area Under the Curve (AUC) scores revealed a slight decrease in the discriminative power of the Random Forest Classifier compared to the gradient boosting approaches across multiple genres.

Additionally, the confusion matrix highlighted a modest decline in the correct classification of certain genres when compared to the LightGBM and XGBoost models.

```
{'rf__max_depth': 20, 'rf__min_samples_leaf': 4, 'rf__min_samples_split': 2, 'rf__n_estimators': 200}
0.5674902190150288
```

Fig 10: Best Parameters and F1-macro score for Random Forest model

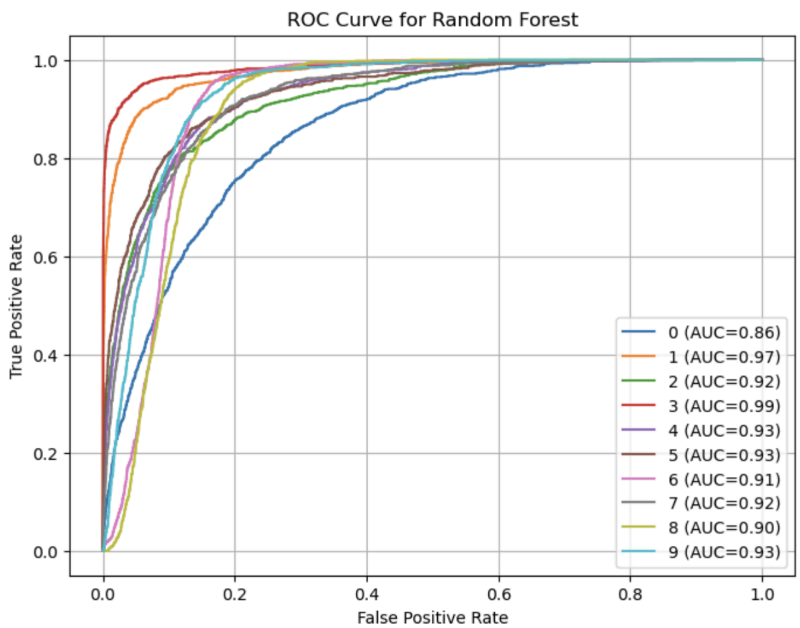

Fig 11: ROC curve and AUC score for Random Forest model

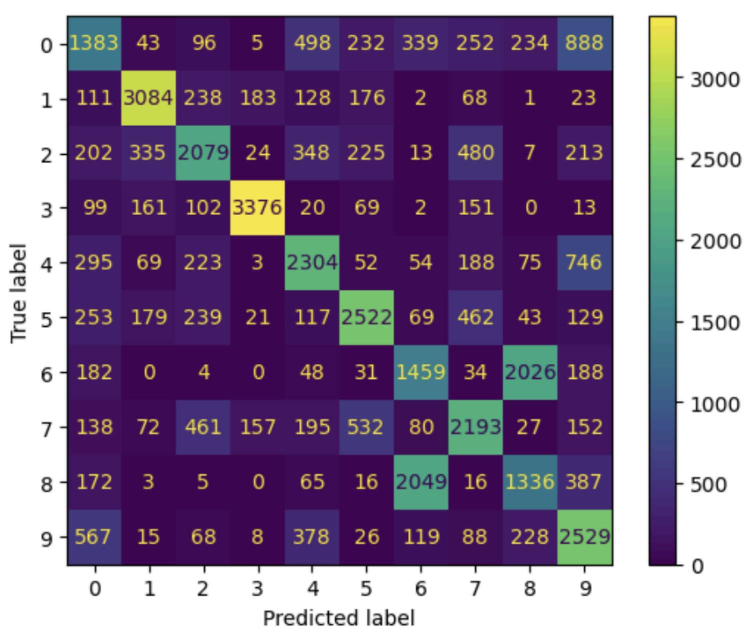

Fig 12: Confusion matrix for Random Forest model

Discussions

The music genre classification task posed significant challenges due to the diverse range of genres and the intricate patterns embedded within the audio features.

The baseline K-Nearest Neighbors (KNN) model demonstrated modest performance, with an F1-macro score of 0.5265, indicating its limited ability to effectively discriminate among the different music genres. However, the analysis of the confusion matrix and ROC curves revealed a relatively stronger capability in classifying the anime and classical genres, potentially due to their distinct feature representations aligning well with the distance-based classification approach of KNN.

Interestingly, while the LightGBM and XGBoost models demonstrated comparable overall performance with F1-macro scores of 0.5910 and 0.5808, subtle differences emerged in their genre-specific discriminative power and confusion patterns. These nuances underscore the potential trade-offs between the two gradient boosting techniques and suggest the need for further fine-tuning and optimization to fully leverage their respective strengths in this specific domain.

The Random Forest Classifier, although outperforming the KNN model, lagged behind the gradient boosting approaches in terms of overall accuracy and genre-specific classification capabilities. This finding aligns with the theoretical advantages of gradient boosting algorithms, which are known for their ability to handle high-dimensional feature spaces and capture intricate patterns more effectively than traditional tree-based models like Random Forests.

Additionally, the feature engineering process revealed the potential impact of track names, particularly the presence of certain special characters, in aiding genre classification. This finding suggests the need for further exploration and incorporation of domain-specific knowledge and audio signal processing techniques to derive more informative and discriminative features, potentially enhancing the models' performance. If more time was allocated, more feature engineering would be considered in order to improve the classification of music genres.

In summary, this study highlights the efficacy of gradient boosting models, particularly LightGBM and XGBoost, in tackling the complex music genre classification task. The comprehensive evaluation strategy employed provided valuable insights into the relative strengths and weaknesses of the different modeling approaches, guiding the selection and refinement of the most promising techniques. However, the ongoing pursuit of improved accuracy and genre-specific classification capabilities underscores the need for further research and innovation in this domain.

References

https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre/data
https://www.kaggle.com/code/stpeteishii/music-genre-xgboost-lgbm-randomforest
https://www.kaggle.com/code/heinereugene/multi-classification-comparison-with-visualization#Summary