Capstone Project: Music Genre Classification Task

Mateus Silva Aragao - msa8779

Final Model

The final model selected for this project was XGBoost. More specifically, the model implemented was an XGBoost Classifier, configured as a One-vs-Rest Classifier. The model was defined with 180 estimators, a maximum depth of 25, a learning rate of 0.25, and an evaluation metric of 'mlogloss'. XGBoost is a potent gradient-boosting algorithm, renowned for its efficacy across a broad range of classification problems. Gradient boosting is a machine learning technique where new models are created that predict the residuals or errors of prior models, then added together to make the final prediction. It leverages the idea of transforming weak learners into strong learners, thereby continually improving the model's accuracy.

Compared to a Random Forest Classifier, XGBoost often demonstrates superior performance due to its ability to limit overfitting, as it incorporates a regularization term in its loss function which a random forest does not. It adds a penalty for complexity, resulting in simpler and more generalizable models.XGBoost's robustness and efficiency stem from its distinct loss function. Unlike AdaBoost, which uses an exponential loss function, XGBoost uses a more robust loss function called the 'logistic loss function.' This function is less sensitive to outliers in the data, making it less likely to overfit the training data.

In this project, the XGBoost model was trained on the entire dataset, bypassing any form of dimensionality reduction. This decision was based on prior tests showing that the model performed better without dimensionality reduction.

Evaluation Metric

The Area Under the ROC Curve (AUC-ROC) is the evaluation metric for this project. AUC-ROC is a suitable metric when dealing with imbalanced datasets, as it considers both the true positive rate and the false positive rate, providing a robust measure of model performance across all possible classification thresholds. This characteristic makes it an appropriate choice for genre classification, where the distribution of songs across genres may be imbalanced.

Model Building Process

Data Preprocessing

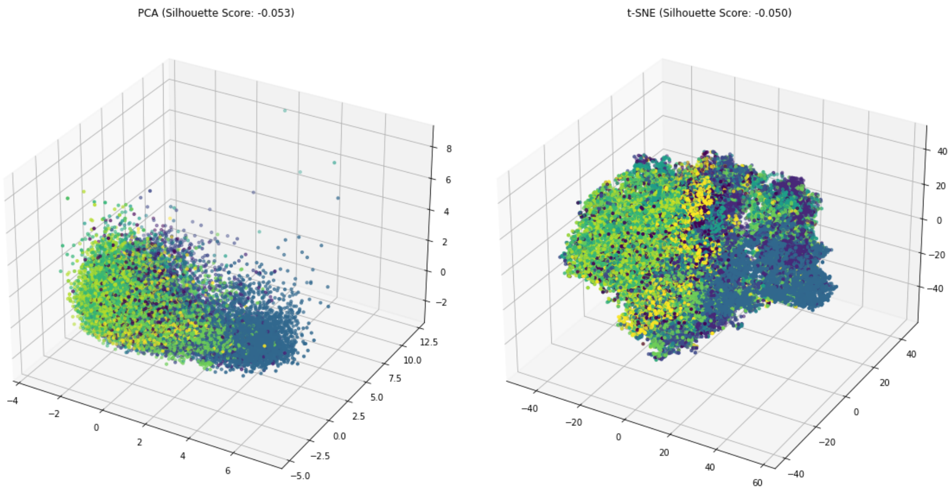
The first step in the model-building process was to preprocess the data. This involved handling missing values and scaling the features. Missing values in the dataset were handled by dropping the rows with NaNs since they constituted only a small fraction of the data. In addition, columns with '?' were identified and converted to NaNs. These values were imputed with the median value of the respective column per genre.

Feature scaling was performed using the StandardScaler. This ensured that all features had a mean of 0 and a standard deviation of 1, thereby standardizing the range of the independent features in the data.

Certain features such as 'instance_id', 'artist_name', 'track_name', and 'obtained_date' were dropped from the dataset because they were not significant for genre classification. Other features like 'key' and 'mode' were transformed into numerical values, which enhanced the model's classification performance.

Dimensionality Reduction

Dimensionality reduction strategies such as PCA and t-SNE were explored. However, none of these strategies improved the AUC-ROC score, indicating that these methods were not beneficial for this particular dataset.



Model Selection and Cross-Validation

The model testing process was a crucial part of the project. It began by experimenting with a variety of machine learning models on data preprocessed with t-SNE for dimensionality reduction. The models tested include Logistic Regression, Random Forest, Support Vector Machines, K-Nearest Neighbors, and XGBoost.

Logistic Regression, a simple and widely-used classification algorithm, was chosen due to its efficiency and ease of interpretation. However, it performed poorly, achieving only 0.305566 accuracy and an AUC-ROC score of 0.790516. This might be because logistic regression may not handle complex relationships in data well, particularly when there are interactions between features.

Random Forest, an ensemble method that constructs multiple decision trees, was chosen due to its ability to handle complex datasets and provide feature importance. It performed better than Logistic Regression with an accuracy of 0.456469 and an AUC-ROC score of 0.863476. However, it still fell short of expectations, possibly due to the high dimensionality of the data.

Support Vector Machines (SVM), known for their effectiveness in high-dimensional spaces and their adaptability through the use of different kernel functions, were also tested. Yet, SVMs achieved an accuracy of 0.433749 and an AUC-ROC score of 0.851400, which was less than ideal. SVMs can be computationally intensive, and they might not perform well when there's a lot of noise in the data, as it may lead to overfitting.
K-Nearest Neighbors, a simple algorithm that classifies a data point based on the majority class of its 'k' nearest neighbors, achieved an accuracy of 0.475609 and an AUC-ROC score of 0.827669. However, K-Nearest Neighbors can suffer from the curse of dimensionality, where the distance between neighbors becomes less meaningful in high-dimensional spaces.

XGBoost, a powerful gradient boosting algorithm, performed better than the previous models with an accuracy of 0.471309 and an AUC-ROC score of 0.873206. The effectiveness of XGBoost can be attributed to its ability to capture complex relationships in the data and its robustness against overfitting through the use of gradient boosting.

Following this, the focus was narrowed down to Random Forest and XGBoost models. These were tested using a One-vs-Rest strategy with Stratified KFold cross-validation on the non-reduced dataset. XGBoost outperformed Random Forest, yielding an accuracy of 0.457969 and an impressive AUC-ROC score of 0.936215, which was the best result so far.

Despite the promising results from XGBoost, further experimentation was conducted using dimensionality reduction techniques such as PCA, t-SNE, PCA+t-SNE, and LDA. However, none of these strategies improved the AUC-ROC score, suggesting that these methods were not beneficial for this particular dataset.
Three different neural network architectures were also experimented with. These architectures were designed with varying numbers of layers and neurons, and additional techniques like dropout and batch normalization were used to prevent overfitting and accelerate training. Despite these efforts, none of these architectures achieved satisfactory accuracy, suggesting that the complexity and high dimensionality of the data might be better suited to tree-based models like XGBoost.

Model 1 consisted of a simple feed-forward neural network with two hidden layers. The first layer had 64 neurons, and the second had 128 neurons. However, despite its simplicity, it failed to provide satisfactory accuracy.

Model 2 was a slightly larger network with more neurons in each layer (128 in the first layer and 256 in the second layer), but it also did not yield satisfactory results. This could be because while increasing the number of neurons can improve the model's capacity to learn complex patterns, it can also make it more prone to overfitting, especially if there's not enough data to adequately train the model.

Model 3 introduced more complexity by adding a dropout layer and a batch normalization layer. Dropout is a regularization technique that randomly drops out (i.e., sets to zero) several output features of the layer during training, which can help prevent overfitting. Batch normalization is a technique to provide any layer in a neural network with inputs that are zero mean/unit variance, which can help speed up learning. The model consisted of two layers with 128 neurons each, a dropout rate of 0.5, and batch normalization after the first layer. Despite these additions, Model 3 also failed to achieve satisfactory accuracy.

The XGBoost model, when employed with a One-vs-Rest strategy for the multi-class problem, yielded the best AUC-ROC score. Nonetheless, none of the models reached satisfactory accuracy, indicating the potential need for further model tuning or alternative pre-processing steps. Stratified K-Fold cross-validation was integrated into the process to ensure an unbiased representation of each genre during model training. Despite these rigorous attempts, the use of neural network models did not yield successful outcomes. The likely reasons could be the complexity and high dimensionality of the data, which might be more compatible with tree-based models like XGBoost. Additionally, it's plausible that the neural network architectures were not optimal for this specific problem, or that they were not trained sufficiently to converge to a robust solution.

Results and Discussion
The XGBoost model achieved the best AUC-ROC score of 0.94, indicating a high performance in genre classification. The model used the original set of predictors without dimensionality reduction. This suggests that the initial feature space contained important information that was key to successful genre classification and that dimensionality reduction methods like PCA and t-SNE were unable to retain this information.
Among the various challenges encountered during the project, handling missing data was a major one. Rows with NaN values were dropped, and special characters such as "?" were identified and converted to NaNs,

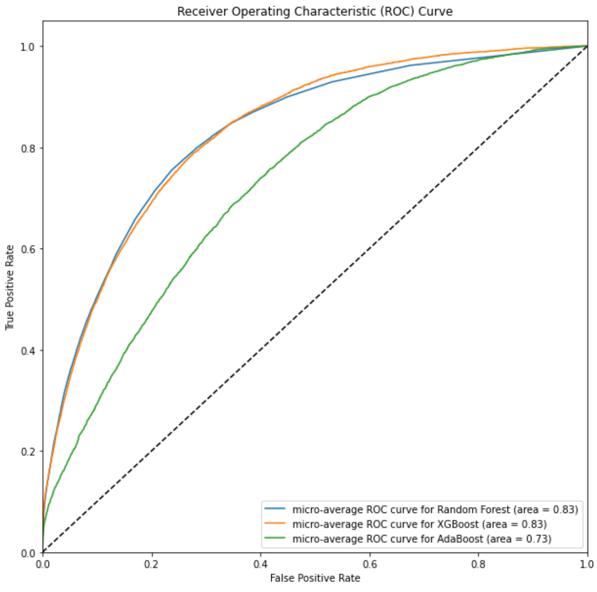
then imputed with the median value of the respective column per genre. This approach proved effective in dealing with the issue.

The feature selection process also proved to be crucial for the success of the classification task. Certain features such as 'instance_id', 'artist_name', 'track_name', and 'obtained_date' were dropped from the dataset as they were not significant for genre classification. On the other hand, features like 'key' and 'mode' were transformed into numerical values, contributing to the improved performance of the model.

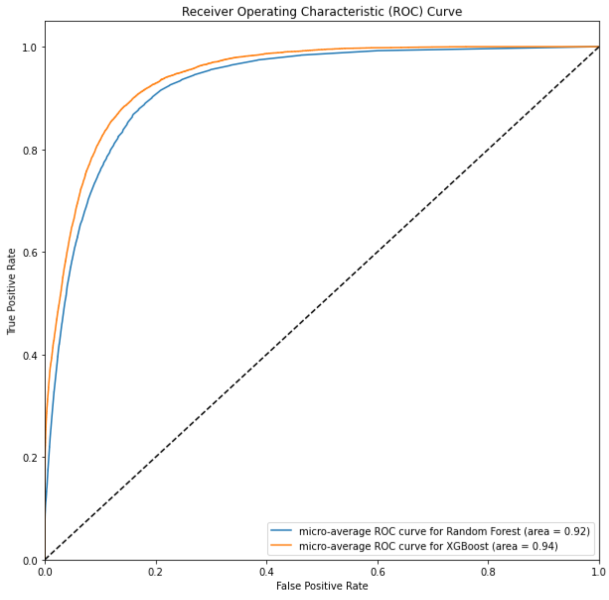
Testing different models and architectures was an integral part of the project. An array of models from traditional machine learning algorithms like Logistic Regression and Random Forest to advanced neural network architectures were tested. The best results were obtained from the XGBoost model, affirming its robustness and reliability for this classification task. The three different neural network architectures, although not as successful, provided important insights into the data's behavior and complexity.

The most important features for the XGBoost classification were 'popularity', 'danceability', 'speechiness', and 'instrumentalness'. This indicates that these features have a significant impact on the genre of a song, and thus can be the focus of future research or application in similar tasks.

In conclusion, to achieve a successful classification taking into account the AUC metric, it was fundamental to have explored different architectures and models. This project underlines the importance of detailed exploratory data analysis, rigorous preprocessing, and thoughtful model selection.



Model with Dimensionality reduction with TSNE



Model with raw data, scaled only

The final XGBoost model that was chosen uses the dataset scaled but not dimensionally reduced. This allowed the model to make use of the full feature space and achieve an AUC score of 0.94, the highest score in all the experiments. The hyperparameters were mostly set to default, with some tuning to optimize performance. This achievement illustrates the potential of XGBoost for multi-class classification problems and encourages further exploration and application of this algorithm.