Shreya Jakhar Choudhary

Fundamental of Machine Learning- Capstone Project Report

The first step towards the goal to find the genre of the song using different features shared in the dataset was to clean the data to create a better analysis. I loaded the file and replaced '?' with NaN. By standardizing the missing value indicators, I ensured that subsequent steps in data processing treat all missing values consistently, which is crucial for accurate imputation. This helps in applying imputation techniques later. The rows that were missing critical information like 'track_name' or 'artist_name' are dropped, as these features are essential for identifying unique tracks and cannot be imputed. Then I performed some numeric conversion and imputation. Median is chosen because it is robust to outliers, which are common in data like popularity or loudness. Then I did standardization of the numerical features, which were *numerical = ['popularity', 'acousticness', 'danceability', 'duration_ms', 'energy', 'instrumentalness', 'liveness', 'loudness', 'speechiness', 'tempo', 'valence'].* The StandardScaler is applied to numerical features to normalize their distribution. Categorical variables like 'key' and 'mode' are transformed using one-hot encoding. This was particularly important for categorical data in the music dataset where numerical encoding could mislead the model to assume a natural ordering. The 'music_genre' column is transformed from text labels into numerical labels using LabelEncoder. This is necessary as most machine learning algorithms require numerical input. Outputs a mapping of music genres to their corresponding numerical labels. This is useful for interpretation of the model's predictions.Features that are left as object types and are not encoded could either be irrelevant, contain free-form text, or have high cardinality (many unique values), which might lead to overfitting or inefficient learning. Removing them simplifies the model and focuses learning on the most impactful data.

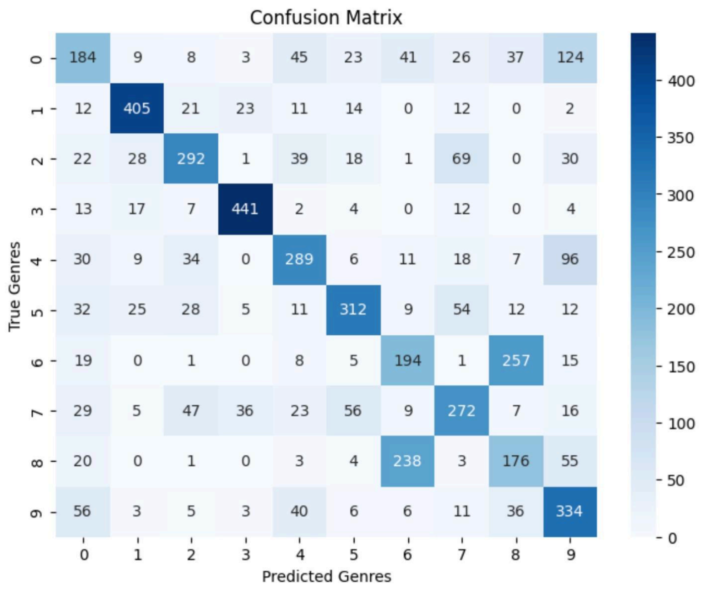
Then, when making the decision of the type of classification model to use, I first started by looking at the type of information that the dataset contained. I used a pipeline for preprocessing and model training and it was a strategic choice due to its ability to streamline the workflow, ensure reproducibility, and prevent data leakage. By encapsulating the entire process—from data transformation through one-hot encoding and scaling, to the final model training—into a coherent sequence of operations, pipelines helped me in simplifying the development process and enhancing the accuracy of model evaluation. Following that I used XGBoost for the classification task. As I knew that XGBoost has a robust performance and its efficiency in handling both numerical and categorical data. Since in the dataset we had both, the numerical and the categorical data, I first thought of this approach. **The accuracy that I got using the XGBoost classification model was 0.58.** I used the accuracy, classification report, and confusion matrix as it provides a comprehensive view of model performance. Accuracy helped me get certain with my choice of the model and the classification report, and the confusion matrix gave me the details which were necessary for further analysis. I visualized the confusion matrix using a heatmap, which makes it easier to see where the model is confusing one genre for another. This is crucial for understanding the model's behavior in a multi-class classification problem. The diagonal entries in the matrix represent the number of correct predictions for each genre. Higher numbers on the diagonal, relative to the off-diagonal entries in the same row, indicate better performance for that specific genre. For example, genre 1 (405 correct predictions), genre 3 (441 correct predictions), and genre 7 (272 correct predictions) show strong classification performance. **Genres 1, 3, 5, and 7 are relatively well-classified.** The number of true positives (correct predictions) is substantially higher than misclassifications for these genres, which indicates that the model features and the algorithm are effectively capturing the characteristics of these genres. The model frequently confuses genre 0 with genre 9, and genre 6 with genre 9. This might suggest a similarity in features or insufficient distinguishing features between these genres. The confusion matrix shows that your model is performing competently with certain genres very well identified, while highlighting areas where performance could be enhanced

Then I calculated the ROC score. The ROC-AUC scores range from **0.86 to 0.98** across different classes. These scores indicate the model's ability to distinguish between each specific genre and all other genres. Higher scores (like 0.98 for genres labeled as 1 and 3) suggest excellent model performance for those genres, where it effectively differentiates that genre from the others. A lower score (0.86 for genre 0) indicates a relatively weaker, but still good, performance for that particular genre. **The overall AUC score of 0.9303 suggests that, on average, the model has a very high probability of correctly classifying a randomly chosen positive instance (true genre) as more likely to be positive than a randomly chosen negative instance (any other genre).** This high score reflects the strong predictive power of your model across all genres, implying that it generally performs well in differentiating between genres. The results that I obtained are particularly encouraging for a multi-class classification problem like this one, which can be complex due to the subtle differences in features between different genres of music. T**he high ROC-AUC scores across most genres suggest that the features used, and the model's structure (XGBoost in this case), is effective in capturing the nuances required to differentiate between these genres.** In summary, your ROC-AUC results confirm that the model is robust and performs effectively across a diverse set of music genres. This makes it a reliable tool for automated music genre classification.

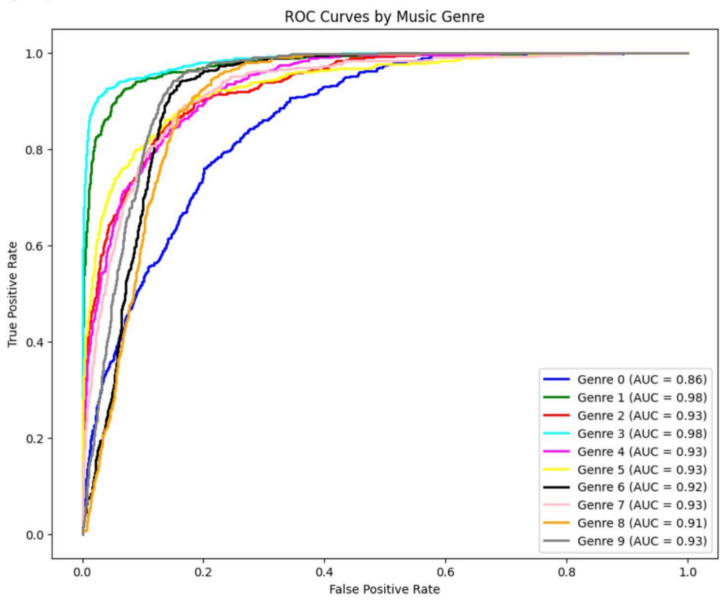
```
Accuracy: 0.5798
Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.37      0.40       500
           1       0.81      0.81      0.81       500
           2       0.66      0.58      0.62       500
           3       0.86      0.88      0.87       500
           4       0.61      0.58      0.60       500
           5       0.70      0.62      0.66       500
           6       0.38      0.39      0.38       500
           7       0.57      0.54      0.56       500
           8       0.33      0.35      0.34       500
           9       0.49      0.67      0.56       500

    accuracy                           0.58      5000
   macro avg       0.58      0.58      0.58      5000
weighted avg       0.58      0.58      0.58      5000
```

```
Genre 0 AUC Score: 0.86
Genre 1 AUC Score: 0.98
Genre 2 AUC Score: 0.93
Genre 3 AUC Score: 0.98
Genre 4 AUC Score: 0.93
Genre 5 AUC Score: 0.93
Genre 6 AUC Score: 0.92
Genre 7 AUC Score: 0.93
Genre 8 AUC Score: 0.91
Genre 9 AUC Score: 0.93
```

```
Overall AUC: 0.93
```



	0	1	2	3	4	5	6	7	8	9
0	184	9	8	3	45	23	41	26	37	124
1	12	405	21	23	11	14	0	12	0	2
2	22	28	292	1	39	18	1	69	0	30
3	13	17	7	441	2	4	0	12	0	4
4	30	9	34	0	289	6	11	18	7	96
5	32	25	28	5	11	312	9	54	12	12
6	19	0	1	0	8	5	194	1	257	15
7	29	5	47	36	23	56	9	272	7	16
8	20	0	1	0	3	4	238	3	176	55
9	56	3	5	3	40	6	6	11	36	334


ROC Curves by Music Genre
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Genre 0 (AUC = 0.86)
Genre 1 (AUC = 0.98)
Genre 2 (AUC = 0.93)
Genre 3 (AUC = 0.98)
Genre 4 (AUC = 0.93)
Genre 5 (AUC = 0.93)
Genre 6 (AUC = 0.92)
Genre 7 (AUC = 0.93)
Genre 8 (AUC = 0.91)
Genre 9 (AUC = 0.93)