Spotify Playlist Popularity Prediction

Project 2 Report

I. Objective

Playlist Popularity Prediction: Can we develop a model to predict the popularity of playlists based on features such as the number of tracks, number of followers, duration, and number of edits? This could help identify key factors that contribute to playlist success and guide playlist curation strategies.

II. Data Preprocessing

We first imported essential libraries like pandas and numpy, including Seaborn to help with visualizations. Then, we loaded the data containing information about Spotify playlists into a pandas DataFrame, allowing us to efficiently manipulate the large dataset. Once loaded, we sought to uphold data integrity through removing irrelevant values and fixing any noticeable errors in the dataset. Additionally, we engaged in feature engineering, a crucial step where we created new features or transformed existing ones to enhance the predictive power of our model. Following data preprocessing, we split the dataset into training and testing sets using sklearn's train and test functions, enabling us to train our model on a subset of the data and evaluate its performance on unseen data.

III. Model Development

We began by selecting an appropriate machine learning algorithm for regression tasks, considering options including Random Forest, Extreme Gradient Boosting, Logistic Regression, Support Vector Machine, Gradient Boosting, and Decision Tree.

With the chosen algorithm in place, we then proceeded to train the model on the data, utilizing it to find optimal parameters that minimize the error between the actual and predicted values. Once trained, the model could then be employed to make predictions on the testing data using the predict method.

IV. Evaluation Steps

In analyzing the data, our group chose to focus specifically on key metrics including the playlists' modification date, number of artists, number of albums, and the total number of tracks. Utilizing this data, we developed a model designed to generate a popularity score for each playlist, thereby establishing a ranking based on this metric. Unfortunately, out of the six different ML models we tested, only one model (Random Forest) had an accuracy score of over 60%. The R^2 scores obtained for all the regression models were negative, indicating a poor fit of the models to the data. This suggests that the models were likely overfitting due to data imbalance, where certain classes or categories within the dataset were disproportionately represented. Additionally, it's possible that some input features were irrelevant to the target variable, further contributing to the poor performance of the models. Furthermore, the presence of outliers in the dataset, such as the extremely high number of followers (e.g., 1038), could have adversely affected the model's performance. These outliers should be carefully handled during preprocessing to prevent them from disproportionately influencing the model's predictions. The accuracy achieved by the models was also quite low, indicating that they were not effectively capturing the underlying patterns in the data. Even after under-sampling to address data imbalance, the maximum accuracy obtained was only 62%.

This suggests that the issue may not solely lie with data imbalance but could also be attributed to insufficient or irrelevant data.

Model	Accuracy Obtained
Random Forest	0.62
Extreme Gradient Boosting	0.52
Logistic Regression	0.5
Support Vector Machine	0.57
Decision Tree	0.55
Naive Bayes	0.48

Ultimately, the main reason behind the poor performance of the models appears to be the quality of the data itself. Insufficient data, irrelevant features, and data imbalance all contributed to the challenges encountered in developing accurate regression models for predicting playlist popularity. Addressing these issues through improved data collection, feature selection, and preprocessing techniques will be crucial for enhancing the performance of future models.