Joshua Lin, Jolene Okamoto, Son Nguyen, Ji Yoon Rhee

10 April 2022

Project: Identifying trends/patterns in genres and songs on Spotify

Phase 3: Model selection, Training, and Evaluation

What algorithms will be explored and why. You are required to compare at least three algorithms.

The algorithms we will be exploring are linear regression, decision tree regression, random forest, and k-nearest neighbor. We chose to run regression algorithms on our data set because our goals for this project are to identify the trend of any feature(s) that affect the popularity score and make predictions based on that trend. However, we also want to try the k-nearest neighbors classifier to see if we get a higher accuracy with the dataset. We are not necessarily classifying new data into specific classes but rather predicting how popular new data is based on the values of the studied features. Therefore, when we use k-nearest neighbors algorithm, which is a classifier algorithm, we will parameterize the popularity variable into a classifiable variable through “high popularity” and “low popularity” classes.

Given how each feature showed very low correlation to the popularity score of each genre, it was very difficult for us to determine which algorithms to use. Thus, we referred to previous studies to 1) confirm that their studies have shown similarly low correlations between each feature versus the popularity score and 2) get inspiration for which algorithms to use. As we have confirmed, the authors of the previous studies also struggled with the low correlations of the features to the target and had to explore through different models before settling on one. The commonly explored models were linear regression, decision tree, and random forest regression algorithms, so we decided to explore the same.

Linear regression algorithm compares a singular feature variable to the popularity score while decision tree, random forest, and KNN algorithms use a collection of features to identify the trend. The decision tree and random forest algorithms would allow us to make predictions about popularity if we were to provide our own data. The KNN algorithm would not be able to give us a specific popularity score but still should be able to predict whether the popularity score will be considered “high” or “low”.

How will the model be trained and evaluated?

For the RandomForestRegressor, the features we will explore will be numerical-acousticness, danceability, energy, duration, instrumentalness, valence, tempo, liveness, loudness, and speechiness. The target will be popularity. We can use the feature_importances_ method to find out which of the features are most important. We can evaluate the model using the test split.

What parameters will be tuned (i.e. hyper-parameter tuning)?

For the RandomForestRegressor, we can tune the parameters of n-estimators and tree max_depth by simply testing different values to see which one has the smallest mean squared error.

For the k-nearest neighbor algorithm, we can play around with different k-values to determine which k-value returns the highest accuracy of the model.

What is the measure of accuracy that you expect from this model and why?

Since we are using a RandomForestRegressor for our model to be trained and evaluated, we will use the mean squared error to measure the accuracy of our model. The higher the mean squared error, then lower the accuracy. For our KNN model, we will use the classification report to evaluate our accuracy. We are not expecting an extremely high accuracy because previous works that we looked at had accuracies of 56% and around 80%. Since the accuracy is dependent on the algorithm and sample sizes, we are expecting that the accuracies from the different algorithms will vary greatly. The dataset is also large and there are a lot of different features that can affect the popularity.