Spotify Song Popularity

Jessica Hamilton, Reagan Matlock, Bailey Perosa, Chidi Henry Ukaegbu

***Abstract* – This proposal details what our project is about, our methodology, and our expected outcomes. The goal of the project is to classify song popularity based on known song attributes.**

I. INTRODUCTION

The primary objective of this project is to classify song popularity based on song attributes such as danceability, acousticness, and loudness. Popularity in this project will be explained on a scale from 0-100 and defined by the total number of plays a song has had and how recent those plays are (songs with more current plays will be classified as more popular than songs that were played a lot in the past). The motivation for this project is curiosity surrounding what song attributes most contribute to popularity. This knowledge can be used by radio stations and DJs to better serve a general audience. Artists could also use this information to make creative decisions for the success of future works. From a business perspective, this information can also promote a positive consumer experience.

II. DATA

The data being used in this project comes from the Spotify Web API. The data contains almost 175,000 rows and 19 columns of song attributes (acousticness, artists, danceability, duration ms, energy, explicit, id, instrumentalness, key, liveness, loudness, mode, name, popularity, release date, speechiness, tempo, valence, and year). All columns except for artists, id, name, and release date are numeric. Of these numeric columns, most contain values between 0-1 while only six columns contain values greater or less than 1 (duration ms, key, loudness, popularity, tempo, and year). The artists column contains a list of all the artists featured on a song. The id column contains a unique identifier for each song in the dataset. The name and release date columns provide the full name of each song as well as the month, day, and year it was released.

The Popularity column was used to classify the songs in the data by creating groups based on the numeric value of Popularity (e.g., 0-24 = not popular, 25-49 = somewhat popular, 50-74 = popular, 75-100 = very popular). The id column was removed since it was not needed as an explanatory variable in the models. The other numeric columns were also discretized during data cleaning to help the models run more efficiently.

The first bit of data cleaning completed, before discretization, was to group the year variable by decade. A new decade variable, created from the year each song was released, was utilized in place of year during all analyses. The 2010s had the most songs of all the decades recorded in the data and the 1920s had the least. The next step in the data cleaning process was to discretize into five groups Tempo, Acousticness, Danceability, Energy, Instrumentalness, Key, Loudness, Speechiness, and Valance based on the distributions of their original values. The Explicit, Key, and Mode columns were also converted into factors to create more

categorical columns. A log transformation was then applied to the Duration MS column to remove its heavy skew. Lastly, a binary Multi_artist column was created from the original Artists column to classify songs as having one artist or multiple. Overall, the majority of songs had a single artist.

III. TEAM RESPONSIBILITIES

Jessica will be responsible for importing and cleaning the data (discretizing columns, getting rid of unwanted columns, etc.). She will also code, run, validate, and gather insights for the naïve model.

Bailey will be responsible for coding, running, validating, and gathering insights for the decision tree and random forest models. She will also set up the final report and presentation templates.

Reagan will be responsible for coding, running, validating, and gathering insights for the support vector machine and XG CatBoost. He will also create the final summary table for all the models.

Henry will be responsible for coding, running, validating, and gathering insights for the k-nearest neighbors model. He will also summarize variable importances from all models and provide the insights for the most important factors for classifying popularity.

All members will work on data visualization, the final report, and the final presentation.

IV. TIMELINE OF MILESTONES

Below shows a general timeline for the project:

Week 1 (Oct. 11): Jessica will import and clean the data, as well as split the data into training and holdout sets. The rest of the team will familiarize themselves with the data and add additional cleaning if necessary.

Week 2 (Oct. 18): Jessica will code the naïve model, Bailey will code the decision tree and random forest models, Reagan will code the boosted tree and randomized logistic regression models, and Henry will code the k-nearest neighbor model.

Week 3 (Oct. 25): Every member will work on testing their models on the holdout set, gathering insights, and possibly tuning the models.

Week 4 (Nov. 1): Reagan will create the summary table of models, and pick the best model using the 1 standard deviation rule. Henry will summarize all the variable importances and choose the top factor that impact classifying song popularity.

Week 5 (Nov. 8): All members will review the insights (which model is best, why, and which factors are most important). All members will create visualizations of the top factors, most likely 1 factor per member. Bailey will create the template for the final presentation and report.

Week 6 (Nov. 15): All members will work on and finish the final presentation and final report.

V. MODEL BUILDING

A. Naïve:

Post-discretization the naïve model predicts the most common popularity level for all records.

B. Decision Tree:

The decision tree model splits and classifies data based on variable levels. It is an inexpensive model to construct, and it has easy interpretability since the tree can be visualized. This model also makes it easy to see which variables are most important. The decision tree uses the impurity of the data to construct the tree and split the data into smaller subsets. One disadvantage of this model is that it can easily overfit to the training data.

C. Random Forest:

The random forest model builds a collection of decision trees that are as uncorrelated as possible, then gathers the average of the predictions from the different trees. This method tends to overfit less and is much more robust to outliers than the decision tree since an average is being taken. One disadvantage of the random forest compared to the decision tree is the random forest can be slower than the decision tree.

D. K-Nearest Neighbors (KNN):

The k-nearest neighbor model uses k closest points for classification performance. This model classifies new data based on how similar it is to already classified data. The model requires three inputs: previously classified data, a distance methodology, and the value of k. Choosing the value of k is critical for this model. If the value of k is too small then the model will be too sensitive to noise points. This model is also relatively simple to use and requires a small amount of calculation time.

E. Support Vector Machine (SVM):

SVM is a supervised learning model that can efficiently perform linear and non-linear classifications. SVM maps training examples to points in space between categories and then uses these gaps to make predictions on new data. SVM maps training examples to points in space between categories and then uses these gaps to make predictions on new data.

F: XG CatBoost:

The XG CatBoost model uses a gradient boosting framework which attempts to solve for categorical features using a permutation driven alternative compared to the classic algorithm. CatBoost uses ordered boosting to overcome overfitting.

VI. RESULTS AND INSIGHTS

A. Naïve:

The most common discretized popularity level was “Not Popular” (level 0). The accuracy of predicting this level for each song was 47%. This accuracy, as expected, was the lowest among all models tested.

B. Decision Tree:

The decision tree model had an accuracy of 76%, the second highest among all the models tested. The variables that were most important to this model were decade, duration, and key.

C. Random Forest:

The accuracy for the random forest model was 66%, the second worst accuracy found among the tested models. Decade, duration, and acousticness were the top three most important variables for this model.

D. K-Nearest Neighbors (KNN):
The KNN model had an accuracy of 75%, the third highest among all the models tested.

E. Support Vector Machine (SVM):
The model accuracy for the SVM was 70%, the third lowest accuracy among all six models tested.

F: XG CatBoost:
The XG CatBoost model had an accuracy of 77%. This accuracy was the highest achieved during the project. The most important variables for this model were decade, duration, and instrumentalness. The CatBoost model appeared to handle the different categorical variables better than the other models.

Overall, the team achieved the main goal of beating the naïve model accuracy of 47%. The decade and duration variables were found to be the most important. The XG CatBoost model performed the best on the test data and is therefore the model the team recommends.

VII: CHALLENGES

Overall, this project contained only a few hiccups. This first challenge encountered was that the cross-validation attempted was not set up correctly. The second challenge was model tuning. While model tuning was attempted during the modeling process, unexpected errors were produced that the team could not overcome within the project timeframe. The third challenge came during our insights stage with variable importances. The team was unable to compute variable importance for two of the six models (KNN and SVM) due to complexities of those two models. The last main challenge faced in this project was with multiple prediction responses. As mentioned above, the popularity variable was discretized into four groups. The six models used, along with most models, struggle handling anything outside of binary classification.