# Group Project: Spotify Track Classification

Course: CS 488

Sem: Spring 2022

Stage: 5

Date: 11/20/2022

Group: Kerry Forsythe,

Roderick Hutterer,

David Torres

## Motivation

Data is fundamental to the operation of modern cloud-based services. In addition to organizing their catalogs of products or materials, companies have the opportunity to collect data from their users’ interactions with the app. Understanding who, what, when, where, and why can help a company better serve their customers through analysis such as classifying products, identifying purchase trends, identifying patterns in users’ past behavior, predict their future behavior, identify commonalities between users, and recommending products and services.

## Problem

In this project, we propose to data mine a sub-set of the Spotify music catalog to provide descriptive and predictive analytics. The Spotify songs attributes used in the analysis are:

*Table 1: Spotify Attributes*

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Energy | The energy of a song, the higher the value, the more energetic. |
| Danceability | The higher the value, the easier it is to dance to this song. |
| Loudness | The higher the value, the louder the song (in dB). |
| Valence | The higher the value, the more positive mood for the song. |
| Duration | the higher the value, the longer the duration of the song. (in ms) |
| Acousticness | The higher the value the more acoustic the song is. |
| Instrumentalness | the higher the value, the less likely the track contains vocals |
| Liveliness | the higher the value, the more likely an audience in the recording |
| Tempo | estimated tempo of a track in beats per minute (BPM). |
| Key | estimated overall key of the track |
| Mode | Indicates the modality (major or minor) of a track; Major = 1 and minor = 0. |
| Time signature | estimated overall time signature of a track |
| Explicit | contains explicit lyrics (binary) |
| Release Year | The year each song was released. |
| Genre | songs were selected from 8 Spotify genres |
| Popularity | The higher the value the more popular the song is. |

Can these attributes be used to predict the popularity of a song? The project will focus on using classification algorithms to predict the popularity of eight genres of Spotify tracks based on the song based on the characteristics of the tracks.

## Data Collection

To collect the data, [Spotipy](https://spotipy.readthedocs.io/en/master/), a lightweight Python library for the Spotify Web API was utilized. After installing Spotipy and obtaining authentication ids from [Spotify for Developers](https://developer.spotify.com/), code was written in Jupyter Notebooks to connect and collect tracks from Spotify. The team explored several methods of collecting tracks including using popular words from songs, using billboard charts, and using user playlists, before settling on a stratified genre and year selection. Using Spotify search(), the Spotify song database was queried to find songs that met the following criteria:

* Genre – the track belonged to one of the 8 specified genres – pop, rock, rap, country, classical, jazz, techno, and world.
* Year – the track belonged to the specified year – 2012-2022
* Market – the track was available to the “US”

A maximum of 1000 distinct tracks were selected for each genre and year. Ten years (2012-2022) of tracks were collected in total producing up to 10,000 observations per genre and a total of ﻿79,379 records:

*Table 2: Track counts by genre*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Classical | Country | Jazz | Pop | Rap | Rock | Techno | World | **Total** |
| # | 9403 | 10000 | 9994 | 10000 | 10000 | 10000 | 9986 | 9996 | **79379** |

For each of these tracks, the Spotipy audio\_features() method was used to collect audio attributes including genre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and time signature. After extracting the release date from the album information, some initial columns were dropped from the data, and each genre was saved to a CSV file. These files serve as a large dataset of tracks that can be randomly sampled to create representative samples for further analysis.

## Summary Statistics

Summary statistics were obtained for the whole dataset.

*Table 3: Classification descriptive attribute*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | count | mean | std | min | max |
| popularity | 79379 | 42.49 | 16.98 | 0 | 100 |

*Table 4: Model attribute descriptive statistics*

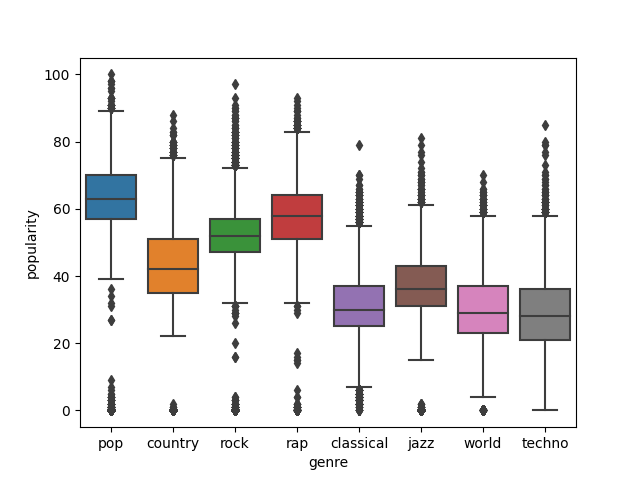
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| attribute | count | mean | std | min | max |
| duration\_ms\_x | 79379 | 246734.2 | 126266.3 | 30026 | 5237295 |
| explicit | 79379 | 0.2 | 0.4 | 0.0 | 1.0 |
| popularity | 79379 | 42.5 | 17.0 | 0.0 | 100.0 |
| album\_type | 79379 | 0.8 | 0.5 | 0.0 | 2.0 |
| release\_year | 79379 | 2017.5 | 2.9 | 2013.0 | 2022.0 |
| danceability | 79379 | 0.6 | 0.2 | 0.0 | 1.0 |
| energy | 79379 | 0.6 | 0.3 | 0.0 | 1.0 |
| key | 79379 | 5.3 | 3.6 | 0.0 | 11.0 |
| loudness | 79379 | -9.7 | 6.5 | -59.5 | 1.6 |
| mode | 79379 | 0.6 | 0.5 | 0.0 | 1.0 |
| speechiness | 79379 | 0.1 | 0.1 | 0.0 | 1.0 |
| acousticness | 79379 | 0.3 | 0.4 | 0.0 | 1.0 |
| instrumentalness | 79379 | 0.3 | 0.4 | 0.0 | 1.0 |
| liveness | 79379 | 0.2 | 0.2 | 0.0 | 1.0 |
| valence | 79379 | 0.4 | 0.2 | 0.0 | 1.0 |
| tempo | 79379 | 120.0 | 29.6 | 0.0 | 248.0 |
| time\_signature | 79379 | 3.9 | 0.4 | 0.0 | 5.0 |
| genre\_numeric | 79379 | 4.5 | 2.3 | 1.0 | 8.0 |

The following statistics are some of the more prominent and significant attributes in the dataset provide a basis on which to compare and evaluate the different genres. Danceability and energy are very similar, as expected, and will likely be good metrics for determining which songs belong in the pop, rap, techno, and jazz genres. Speechiness, unexpectedly, has a very low mean indicating that it is likely that a significant number of songs have average or minimal vocals.

## Data Preprocessing

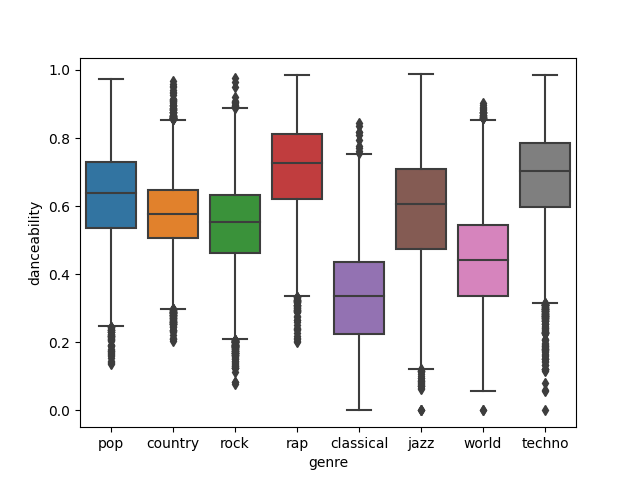
The analysis started with eight different datasets with about 10,000 songs each. Then, they were all compiled into one large dataset while removing the redundant information, such as disc number, id, track number, album type, album name, is playable, and linked from. Next, various histograms and boxplots were created to try to find the most interesting information about the data. For instance, we found that pop music tends to be the most popular out of the eight chosen genres as shown below:

*Figure 1: Boxplot of popularity vs genre*



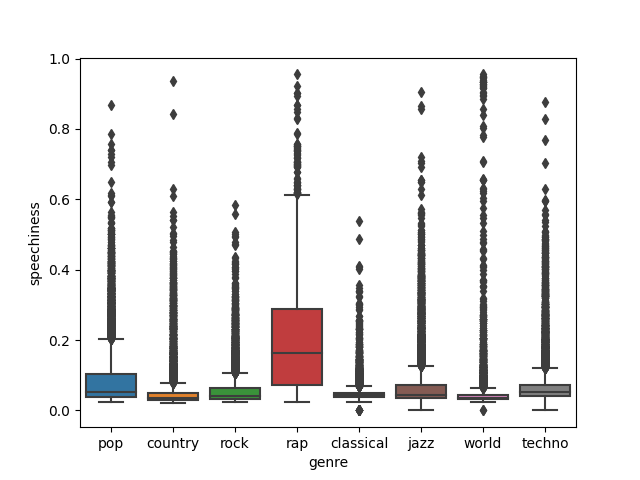
As expected, the least popular would be classical, techno, and world music. So far, the data that was collected seems to be indicative of current music. Additionally, we found that the danceable music is rap, pop, jazz, and techno:

*Figure 2: Boxplot of danceability vs genre*



Another important correlation we found (albeit a rather obvious one) was that rap music tends to be the most vocal type, while classical music is the least as shown below:

*Figure 3: Boxplot of speechiness vs genre*



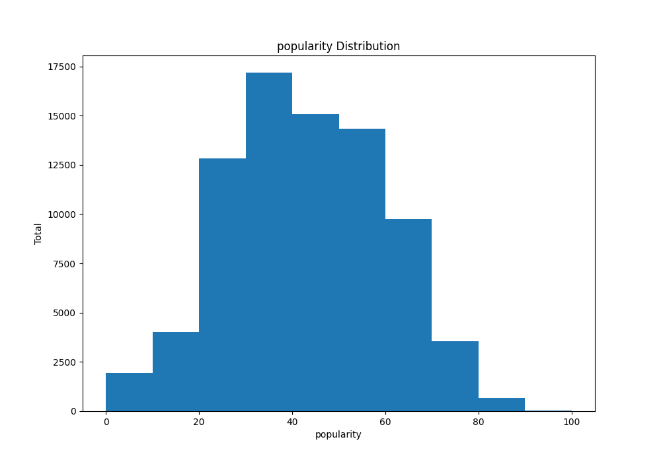
We expected the rap and classical genres to be the most distinct from one another among all the genres. The boxplots shown so far have all supported this notion and the same can be seen from looking closely at the summary statistics of the two genres (Figure 5).

*Figure 5: statistics for Rap and Classical*

|  |  |
| --- | --- |
| *Rap* | *Classical* |
| Graphical user interface, text  Description automatically generated | Graphical user interface, text, chat or text message  Description automatically generated |

These boxplots show a distinctness between these genres of music, indicating that it may be possible to predict the genres of Spotify tracks. Another important piece of information is that for all eight genres, popularity is most dense in the 30-60 range (on a scale of 0-100) while still having a good amount of variability:

*Figure 6: Histogram of popularity values*

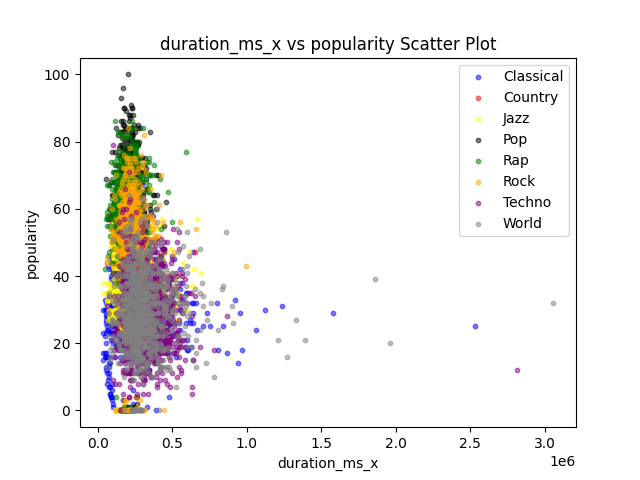
This indicates that our sample data may be useful enough to predict popularity based on all other attributes.

After plotting these initial graphs, a Pearson correlation matrix was created using the attributes of the songs. Some of the highest correlations were between: (popularity, explicit), (popularity, instrumentalness), (valence, danceability), (acousticness, loudness), (genre, duration), (genre, energy), (genre, loudness), and (genre, acousticness).

*Figure 7: Correlation matrix heatmap*Chart

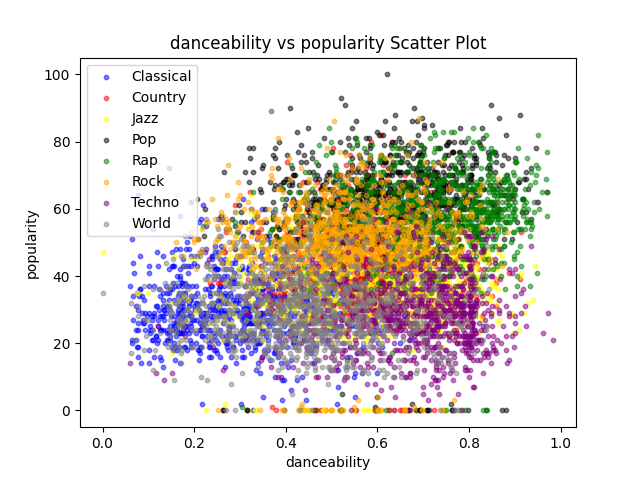
Description automatically generated with medium confidence

After this analysis, a stratified sample of about 8,000 songs was taken from the dataset (grouped by popularity). This was used to generate scatter plots in order to find any correlation between popularity and genre. One of the correlations found was techno tends to have longer duration and lower popularity than rock music:

*Figure 8: scatter plot of duration vs popularity*

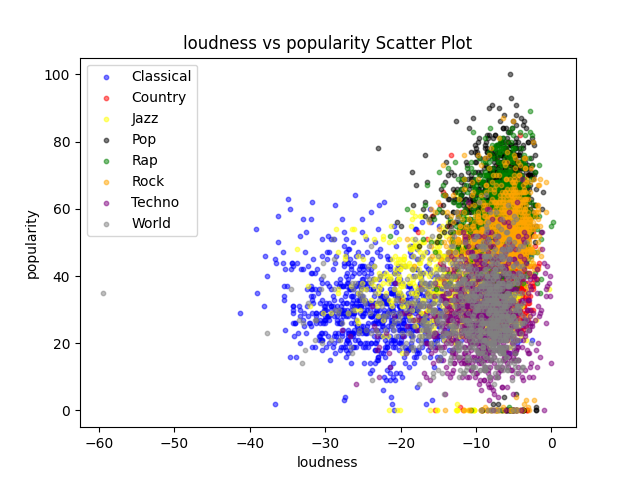
It can also be noticed that classical has much lower popularity and danceability than rap music:

*Figure 9: scatter plot of danceability vs popularity*



Another good point to notice is that loudness is more closely related to genre than popularity. In the plot below, it is shown that even though pop, rock, rap, techno and world music generally have the same loudness, but techno and world music have much lower popularity:

*Figure 10: scatter plot of loudness vs popularity*



With many more scatter plots that visualize the similarity and dissimilarity between popularity, genres, and their attributes, we will be able to train and run models using all of this important information. These scatter plots show once more that our data is concrete and that the genres are distinct enough to develop training models with good accuracies that may predict genre or popularity.

**Binary classification**

In order to predict the popularity of songs based on their song features and attributes, we first used the mean popularity to determine what is popular. In other words, anything with less than the mean popularity will be said to be unpopular, and popular otherwise. For a second splitting choice, we used the mean plus the standard deviation. So, anything less than the mean + standard deviation is said to be unpopular. Anything greater or equal to the mean + standard deviation is labeled popular. Once these values were chosen, we transformed the popularity column in the dataset to be a binary classification, unpopular (0) and popular (1).

## Classification Exploration

Ten Classification Algorithms were run using 5-fold cross-validation on a genre-stratified sample of 10% of the full dataset for the two binary splitting choices for popularity. Using the first splitting choice of Popular = 1 if popularity >= mean (popularity), Bagging, AdaBoost, and Random Forest produced the best accuracy results with 85-87% accuracy. The other 7 models ranged from 55-80% accuracy.

Using a second splitting choice of popular =1 if the popularity >= mean(popularity) + standard deviation(popularity) improved the performance for 9 out of 10 models. Bagging, AdaBoost, and Random Forest remained the 3 most accurate models with 2-3% increases in accuracy to 87-88%. The other 7 models' accuracy ranges from 74-83% accuracy with increases from -3% to 18%. Since this popularity choice resulted in better accuracies, the mean+standard deviation popularity was used for all subsequent runs.

*Table 5: Accuracy 10 models, 2 splitting points*

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | | |
| **Classifier** | **pop>=mean** | **pop>=mean+std** | **improvement** |
| **Bagging** | 0.848829 | 0.867977 | 0.019148 |
| **AdaBoost** | 0.843536 | 0.874024 | 0.030488 |
| **Random Forest** | 0.861299 | 0.880322 | 0.019023 |
| **Logistic Regression** | 0.661376 | 0.824642 | 0.163266 |
| **SVM** | 0.646889 | 0.826910 | 0.180021 |
| **K-Means** | 0.610105 | 0.788865 | 0.17876 |
| **Decision Tree** | 0.794533 | 0.834467 | 0.039934 |
| **MLP Neural** | 0.499745 | 0.809648 | 0.309903 |
| **Gaussian Process** | 0.570798 | 0.773243 | 0.202445 |
| **Quadratic Discriminant** | 0.774373 | 0.740489 | -0.033884 |

*Figure 11: Accuracy 10 models, 2 splitting points*

Chart, bar chart

Description automatically generated

*Figure 12: Efficiency of 10 models*

Chart

Description automatically generated

The very next thing that was tried was using a Gaussian Naïve Bayes Classifier model. This model was run using the 5-Fold Validation on the entire dataset, which yielded an average accuracy of ~73% using pop>=mean+std. Based on these results, further experimentation and analysis focused on the ensemble classifiers – Bagging, AdaBoost, and Random Forest with popular =1 for popularity >= mean + standard deviation.

The Bagging Classifier was able to provide promising results using 5-Fold Validation. It boasted an average accuracy of 88.79% with a standard deviation of 0.00208 of the accuracies, when trained using the entire dataset. Of course, this wasn’t achieved without experimenting with the setting of this classifier. Using the default settings on the classifier, an average accuracy of about 86-87% was achieved. But what proved to be the best method was setting it’s oob\_score to true. Doing this improved the average accuracy to about 88.79%. With such a low deviation, this method so far seemed suitable for predicting the popularity of Spotify songs from these 8 genres.

*Table 6: Bagging Classifier Model statistics*

|  |  |
| --- | --- |
| **Average Accuracy** | 0.8879 |
| **Standard Deviation** | 0.00208 |
| **Weighted Average Recall** | 0.89 |
| **Weighted Average F1** | 0.88 |

*Table 7: Bagging Classifier - Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 1.0 | 1.0 | 1.0 | 52254 |
| **1** | 1.0 | 0.99 | 0.99 | 11250 |
| **Testing** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.95 | 0.93 | 13103 |
| **1** | 0.72 | 0.60 | 0.65 | 2772 |

As for the AdaBoost Classifier, its accuracy showed good results as well with an average accuracy of 88.3% (using 5-Fold Validation on the full dataset). Its standard deviation of the accuracies was 0.0024. These optimized results were achieved by keeping all of the settings of the classifier as default since any change resulted in degradation of its statistics. While this classifier had reasonable results, both its accuracies and standard deviation were higher (marginally) than those of the Bagging Classifier. Thus, it was clear that of these two, the Bagging Classifier was the more desirable model to use.

*Table 8: AdaBoost Classifier Model Statistics*

|  |  |
| --- | --- |
| **Average Accuracy** | 0.883 |
| **Standard Deviation** | 0.0024 |
| **Weighted Average Recall** | 0.88 |
| **Weighted Average F1** | 0.0.88 |

*Table 9: AdaBoost Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.95 | 0.93 | 52254 |
| **1** | 0.70 | 0.60 | 0.65 | 11250 |
| **Testing** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.94 | 0.93 | 13102 |
| **1** | 0.69 | 0.60 | 0.64 | 2772 |

The Random Forest Classifier consistently provided the highest accuracy, so it was run on the full data set. It yielded a training accuracy of 99.7% and a testing accuracy of 89.9%. It had a standard deviation of its accuracies of 0.00218. These results were achieved by turning on the oob\_score and setting the criterion to use Gini Index. These were the most optimized setting using this classifier. Here are the statistics:

*Table 10: Random Forest Classifier Model statistics*

|  |  |
| --- | --- |
| **Average Accuracy** | 0.8974 |
| **Standard Deviation** | 0.00218 |
| **Weighted Average Recall** | 0.90 |
| **Weighted Average F1** | 0.89 |

The tables and chart provide the confusion matrix, precision, recall, and F1 score, feature importance for the classification, and accuracy by genre.

*Table 11: Random Forest Confusion matrix*

|  |  |  |
| --- | --- | --- |
| **Training** | **0** | **1** |
| **0** | 52193 | 43 |
| **1** | 148 | 11119 |
| **Testing** | 0 | 1 |
| **0** | 12563 | 558 |
| **1** | 1053 | 1702 |

*Table 12: Random Forest Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 1.0 | 1.0 | 1.0 | 52236 |
| **1** | 1.0 | 0.99 | 0.99 | 11267 |
| **Testing** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.96 | 0.94 | 13121 |
| **1** | 0.75 | 0.90 | 0.68 | 2755 |

*Figure 13: Random Forest Feature Importance*

Chart, bar chart

Description automatically generated

*Figure 14: Random Forest Accuracy by Genre*

Chart, bar chart

Description automatically generated

After running these ensemble models using 5-Fold Validation, the models were run using warm start, where the full dataset was partitioned into 5 partitions, used four of them to train and one of them to test. Due to the singular nature of the AdaBoost Classifier, warm start could not be run on it. So, warm start was run on the Bagging and Random Forest Classifiers.

*Table 13: Bagging – Warm Start*

|  |  |
| --- | --- |
| **Accuracy** | 0.8944 |
| **Weighted Average Recall** | 0.89 |
| **F1** | 0.89 |

*Table 14: Bagging – Warm Start – Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.93 | 0.97 | 0.95 | 52234 |
| **1** | 0.85 | 0.67 | 0.75 | 11270 |
| **Testing** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.96 | 0.94 | 13123 |
| **1** | 0.75 | 0.59 | 0.66 | 2752 |

*Table 15: Random Forest – Warm Start*

|  |  |
| --- | --- |
| **Accuracy** | 0.8956 |
| **Weighted Average Recall** | 0.90 |
| **F1** | 0.89 |

*Table 16: Random Forest -warm - Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Training** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.93 | 0.98 | 0.95 | 52234 |
| **1** | 0.87 | 0.67 | 0.75 | 11270 |
| **Testing** | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.92 | 0.96 | 0.94 | 13123 |
| **1** | 0.76 | 0.58 | 0.66 | 2752 |

As can be seen in the tables above, the Bagging Classifier improves using warm start by about 0.6%, but the Random Forest gets worse using warm start by about 0.2%. Overall, it seems the top-performing classification model will always be the Random Forest Classifier using 5-Fold Validation. This model consistently has the highest accuracy (89.74%), the some of the top precision, recall, and f1 scores among all of the classification models used so far. In conclusion, the preferred model to predict the popularity of Spotify songs from the chosen eight genres is the Random Forest Classifier using 5-Fold Validation using the parameters: n\_estimators=100, criterion=’gini’, bootstrap=True, oob\_score=True, warm\_start=False, max\_depth=None, max\_samples=None.

## Billboard Spotify Dataset

Another Spotify dataset was obtained using the Billboard charts to select Spotify tracts from 2011-2022. Four billboard charts were selected- three from genres that matched the original dataset (Pop, Country, Rap) and one new genre (Alternative) generating 1669 tracks. Not surprisingly, the billboard dataset has a mean popularity of 67.7%, which is 25.2% higher than the original set. However, the splitting value from the original set was used to label the new data as popular or unpopular.

*Figure 15: Histogram of Billboard popularity values*Chart, histogram

Description automatically generated

*Table 17: Billboard – classification descriptive statistics*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **attribute** | **count** | **mean** | **std** | **min** | **max** |
| popularity | 1669 | 67.7 | 13.8 | 0 | 98 |

*Table 18: Billboard - model descriptive statistics*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **attribute** | **count** | **mean** | **std** | **min** | **max** |
| duration\_ms\_x | 1669 | 215011.9 | 69787.1 | 52114 | 2399056 |
| release\_year | 1669 | 2015.1 | 7.6 | 1956 | 2022 |
| danceability | 1669 | 0.6 | 0.1 | 0 | 0.963 |
| energy | 1669 | 0.7 | 0.2 | 0.00342 | 0.991 |
| key | 1669 | 5.3 | 3.6 | 0 | 11 |
| loudness | 1669 | -6.2 | 2.5 | -30.704 | -0.91 |
| mode | 1669 | 0.7 | 0.5 | 0 | 1 |
| speechiness | 1669 | 0.1 | 0.1 | 0 | 0.567 |
| acousticness | 1669 | 0.2 | 0.2 | 3.81E-06 | 0.988 |
| instrumentalness | 1669 | 0.0 | 0.1 | 0 | 1 |
| liveness | 1669 | 0.2 | 0.1 | 0.0219 | 0.963 |
| valence | 1669 | 0.5 | 0.2 | 0 | 0.982 |
| tempo | 1669 | 123.3 | 29.6 | 0 | 210.1 |

The new dataset did not have a column for time signature. From Figure, 12, time signature was the least important attribute for the model, so it was removed from both data sets without significantly affecting the accuracy of the model for the original dataset (89.9%). The correlation matrix for the new data (Figure 15) set also generally shows weaker correlations compared to the original dataset (Figure 7).

*Figure 16: Correlation Matrix Heatmap*Chart

Description automatically generated

After training the random forest algorithm on the original Spotify dataset, the model was used to predict the popularity of the Billboard dataset. The accuracy of the model dropped significantly to 62.1%.

*Table 19: Billboard – Random Forest*

|  |  |
| --- | --- |
| **Accuracy** | 0.6189 |
| **Weighted Average Recall** | 0.62 |
| **F1** | 0.64 |

*Table 20: Billboard - Random Forest - Precision, recall, F1 score*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1** | **Total** |
| **0** | 0.37 | 0.97 | 0.54 | 384 |
| **1** | 0.98 | 0.51 | 0.68 | 1285 |

The accuracy of the Rap genre remained the highest at 82.8%, followed by Pop at 72.7%, and Country at 64.1%. Not surprisingly, the Alternative accuracy was the lowest at 31.9%. This makes sense since this genre was not included in the original training set and suggests that the model will be less accurate for genres that are not in the model.

*Figure 17: Billboard Accuracy by Genre*

Chart, bar chart

Description automatically generated

This was to be expected using a model that had been trained without that particular genre. So naturally, our next step would be to prove that our model could be trained using the new ‘alternative’ genre and still provide desirable results. Here is what we found when running the model on the entire dataset:

*Table 21: Random Forest Classifier Model statistics using ‘alternative’ genre in model training*

|  |  |
| --- | --- |
| **Average Accuracy** | 0.8965 |
| **Standard Deviation** | 0.00236 |
| **Weighted Average Recall** | 0.89 |
| **Weighted Average F1** | 0.89 |

As seen in the table above, the results of our most favorable model, the Random Forest Classifier, provides nearly exactly the same results as before adding a new genre. What we learned from this, however, is that our model is very versatile, and if trained using any combination of genres, it will yield very reliable results.

**Conclusion**

In our problem to find a reliable classification model in order to predict the popularity of any given Spotify song, we believe to have found a suitable model to use on at least our eight different genres of music chosen for our original data set. Our most suitable model for this problem ended up being a Random Forest Classifier with the parameters: n\_estimators=100, criterion=’gini’, bootstrap=True, oob\_score=True, warm\_start=False, max\_depth=None, max\_samples=None. Though, we chose our model based on our original data set, which was a set of Spotify songs from a list of genres (country, rap, jazz, pop, world, rock, techno and classical) and from a time frame of 2012-2022. In turn, we are limited to say what our prediction model is useful for. In other words, we are completely comfortable saying that our model is ~90% accurate when used on any tracks from this list of genres and time frame. But when we start to introduce new datasets with different genres and statistics, we begin to lose confidence in our classification model. With our new data set collected from Billboard, we found that the accuracies for the genres that we had previously trained and test in our original data only saw an average loss of ~5% accuracy. But for the newly introduced genre, which was ‘alternative’, the accuracy was drastically lower than the rest (by ~30-40% on average). So, to see how versatile our classifier could be, we retrained our Random Forest Classifier using the new genre. We saw the same high prediction accuracies we had seen before, thus, proving that our chosen model is capable of being trained with different genres while retaining reliability.

In conclusion, training a Random Forest Classifier using the previously specified parameters gives very reliable results when training it using our original list of music genres. It can also be reliable when trained using different groups of genres as proven before. On average, we see an accuracy of ~90%, even when using different genres to train the model. Of course, this model is only useful when it is trained with the exact genres intended to have their popularities predicted. If a different genre of music is every introduced into the prediction stage of this model, it must first be used to trained into the model using warm start, by retraining the entire model, or by creating an entirely new Random Forest Classifier using the new subset of genres. By collecting very useful and meaningful data, we were able to create a binary classification model that accurately predicts the popularity of Spotify songs from a given set of genres from the last 10 years.