Project L: Milestone 2

Ilona Demler, Daniela Garcia, Kayla Manning, and Saul Soto
November 30, 2021

This report will include the following: a description of the data, EDA, our research question, and an initial baseline model. The materials used to generate this report can be found on our GitHub Repository.

1 DATA DESCRIPTION

1.1 RAW DATA

This project will utilize the Million Playlist Dataset, which consists of a sample of 1 million public playlists from over 4 billion public playlists on Spotify. Consisting of over 2 million unique tracks from nearly 300,000 artists, the dataset contains public playlists created by Spotify users between January 2010 and October 2017.

The dataset, sourced from AIcrowd, contains the following columns:

- `pid` - integer - playlist id - the MPD ID of this playlist. This is an integer between 0 and 999,999.
- `name` - string - the name of the playlist
- `description` - optional string - if present, the description given to the playlist. Note that user-provided playlist descriptions are a relatively new feature of Spotify, so most playlists do not have descriptions.
- `modified_at` - seconds - timestamp (in seconds since the epoch) when this playlist was last updated. Times are rounded to midnight GMT of the date when the playlist was last updated.
- `num_artists` - the total number of unique artists for the tracks in the playlist.

- `num_albums` - the number of unique albums for the tracks in the playlist
- `num_tracks` - the number of tracks in the playlist
- `num_followers` - the number of followers this playlist had at the time the MPD was created. (Note that the follower count does not include the playlist creator)
- `num_edits` - the number of separate editing sessions. Tracks added in a two-hour window are considered to be added in a single editing session.
- `duration_ms` - the total duration of all the tracks in the playlist (in milliseconds)
- `collaborative` - boolean - if true, the playlist is a collaborative playlist. Multiple users may contribute tracks to a collaborative playlist.
- `tracks` - an array of information about each track in the playlist. Each element in the array is a dictionary with the following fields:
- `track_name` - the name of the track
- `track_uri` - the Spotify URI of the track
- `album_name` - the name of the track's album
- `album_uri` - the Spotify URI of the album
- `artist_name` - the name of the track's primary artist
- `artist_uri` - the Spotify URI of the track's primary artist
- `duration_ms` - the duration of the track in milliseconds
- `pos` - the position of the track in the playlist (zero-based)

Playlists are sampled with randomization and anonymized to protect user privacy. The dataset is manually filtered to maximize quality and to remove offensive content, and some playlists have fictitious tracks added as noise. Therefore, the playlists in this dataset are not representative of the overall population of playlists on Spotify.

- Original Data Site and Data Download
- Downloaded Sliced and Joined Raw Data.

1.2 Processed Data

To examine the data and build our model, we first joined the 1000 data frame slices into one aggregate data frame (this code can be viewed in the `join_data.ipynb` notebook in our GitHub Repo). From there, we sampled 100,000 observations from the 1,000,000. Next, we cleaned the data and generated new predictors.

We created the binary variables `collaborative` and `has_description`, which indicate whether the playlist is collaborative and/or has a description, respectively.

Next, we created the binary variable `popular_name`, which indicates whether each playlist name contains a word listed as one of the most popular in the `stats.txt` file provided in the data download. We chose to generate this variable to generalize the playlist names variable, as there are too many unique playlist names to reveal any significant trends. This allows us to use the names as a more effective predictor.

Then, we iterated through each playlist and counted the total number of popular songs and artists (according to the `stats.txt` file) and saved the counts in variables `total_popular_tracks` and `total_popular_artists`.

In the end, our cleaned dataset contains the following variables, which we can now use to conduct EDA:

- `popular_name`
- `has_description`
- `num_artists`
- `num_albums`
- `num_tracks`
- `duration_ms`
- `collaborative`
- `total_popular_tracks`
- `total_popular_artists`
- `num_followers` (outcome variable)

2 Exploratory Data Analysis

2.1 Outcome Variable

First, we need to determine whether the response variable should be continuous or categorical. To do this, we first examine the distribution of playlist followers in the training dataset:

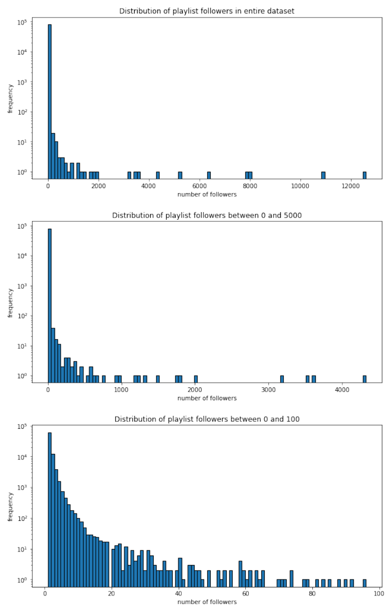


Figure 2.1: Distribution of playlist followers in training dataset over all ranges, for 0-5000 followers, and 0-100 followers. Y-axis is log-scaled. The overwhelming majority of playlists have under 20 followers, but there are peaks at higher ranges as well, which suggests that we could use a categorical outcome variable.

Based on these histograms, the data appears to follow a very right-skewed distribution, with the large majority of playlists having fewer than 20 followers.

We considered translating `num_followers` into a binary categorical variable with a threshold of 20, with playlists with 20+ followers being considered "popular". However, there were very few playlists with greater than 20 followers, which raises concerns about overfitting the "popular" category. For our baseline model, we decided to stick with regression.

2.2 Outcome Variable vs Each Predictor

Before constructing our model, we explored the distributions of our different variables. First, we generated visualizations that treated our response variable as categorical, classifying a playlist as popular or unpopular based on if it fell above or below 20 followers. The below plots reveal that popular and unpopular playlists follow the same general distributional shape for each of the predictors. However, the distributions of the popular playlists tend to have much greater variance. On the other hand, the unpopular playlists tend to fall on the extremely low end of the spectrum. Most of these values are clustered near 0 for continuous predictors or in the 0 category for binary predictors. This likely results from people creating playlists but not fully developing them. These incomplete, small playlists have few albums, few tracks, few artists, and therefore few followers.

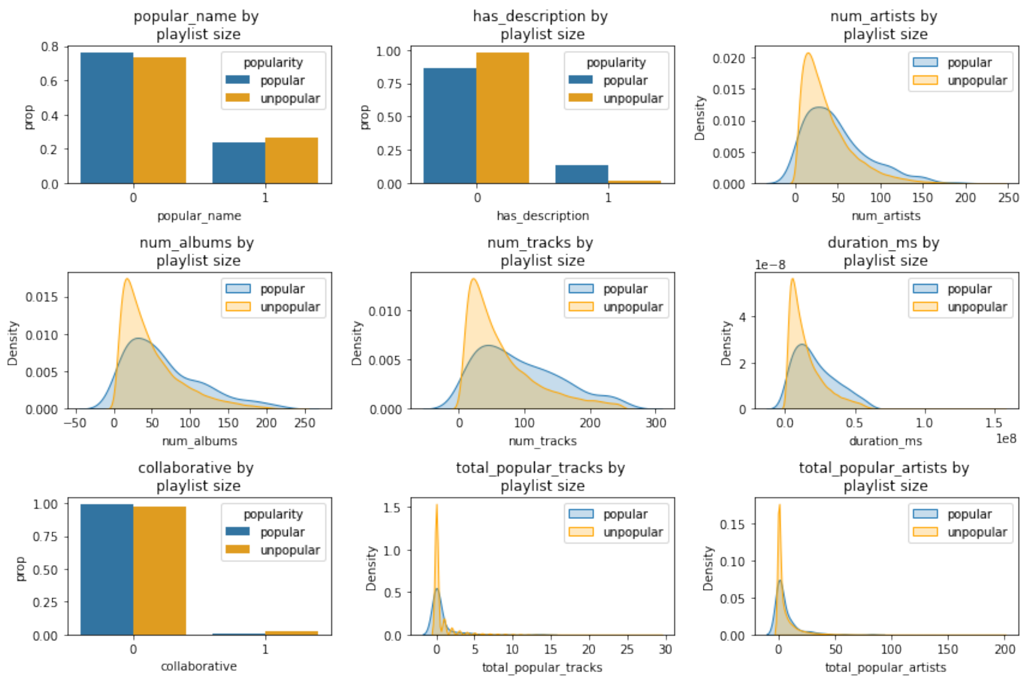


Figure 2.2: Comparing predictors to categorical outcome variable.

We then generated a second set of plots, treating the number of followers as a continuous variable. These plots map the number of followers along the y-axis, rather than treating them as two separate distributions. We glean similar results from these plots. Overall, there are very few observations with large numbers of followers. However, the pattern emerges that collaborate playlists and those that have descriptions or popular names tend to have a greater number of followers on average.

Based on the scatterplots, we see a negative trend between the follower count an the number of artists, playlist duration, total popular tracks, and total popular artists.

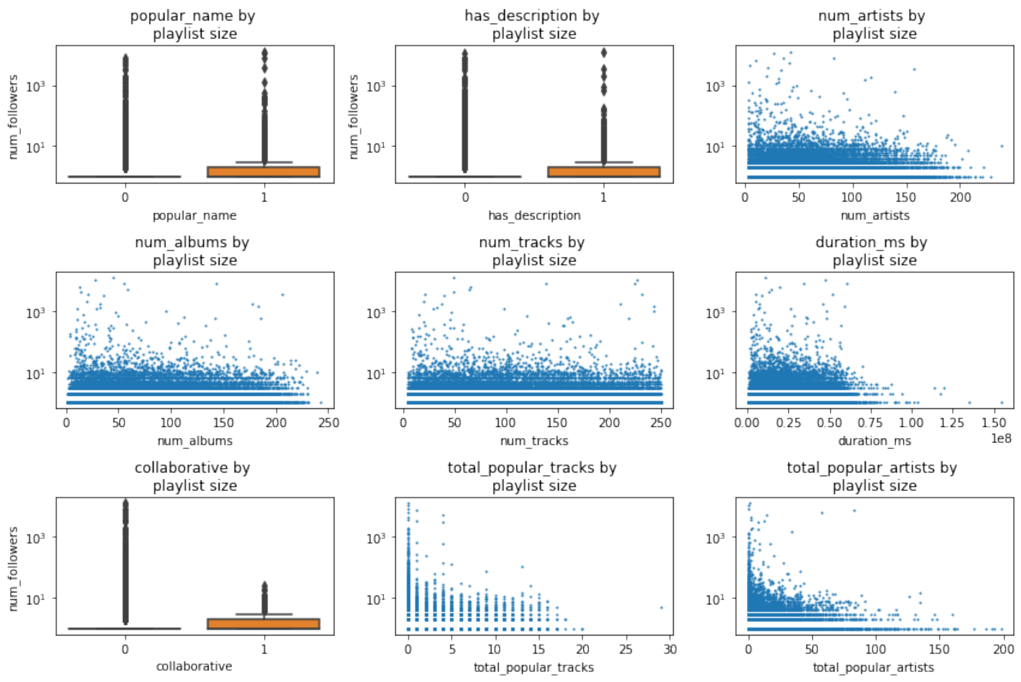
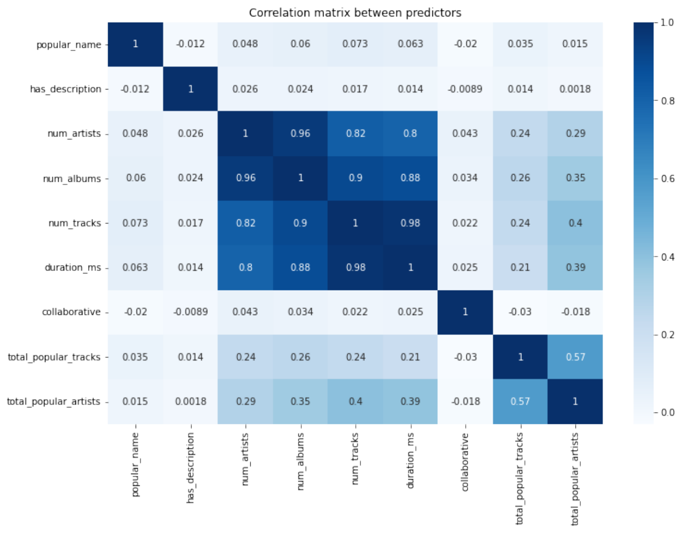


Figure 2.3: Comparing predictors to continuous outcome variable.

2.3 Collinearity between Predictors

After conducting some baseline plots, we assessed the potential collinearity between predictors. The below plot reveals strong correlation between several of the predictors.



	popular_name	has_description	num_artists	num_albums	num_tracks	duration_ms	collaborative	total_popular_tracks	total_popular_artists
popular_name	1	-0.012	0.048	0.06	0.073	0.063	-0.02	0.035	0.015
has_description	-0.012	1	0.026	0.024	0.017	0.014	-0.0089	0.014	0.0018
num_artists	0.048	0.026	1	0.96	0.82	0.8	0.043	0.24	0.29
num_albums	0.06	0.024	0.96	1	0.9	0.88	0.034	0.26	0.35
num_tracks	0.073	0.017	0.82	0.9	1	0.98	0.022	0.24	0.4
duration_ms	0.063	0.014	0.8	0.88	0.98	1	0.025	0.21	0.39
collaborative	-0.02	-0.0089	0.043	0.034	0.022	0.025	1	-0.03	-0.018
total_popular_tracks	0.035	0.014	0.24	0.26	0.24	0.21	-0.03	1	0.57
total_popular_artists	0.015	0.0018	0.29	0.35	0.4	0.39	-0.018	0.57	1

Figure 2.4: Correlation matrix of predictors in train dataset. Values close to 1 indicate high collinearity. From the correlation matrix, we can denote 6 strong correlations: ***num_tracks*** and ***duration_ms*** (0.98), ***num_albums*** and ***num_artists*** (0.96), ***num_albums*** and ***num_tracks*** (0.91), ***duration_ms*** and ***num_albums*** (0.9), ***num_tracks*** and ***num_artists*** (0.83), ***duration_ms*** and ***num_artists*** (0.81)

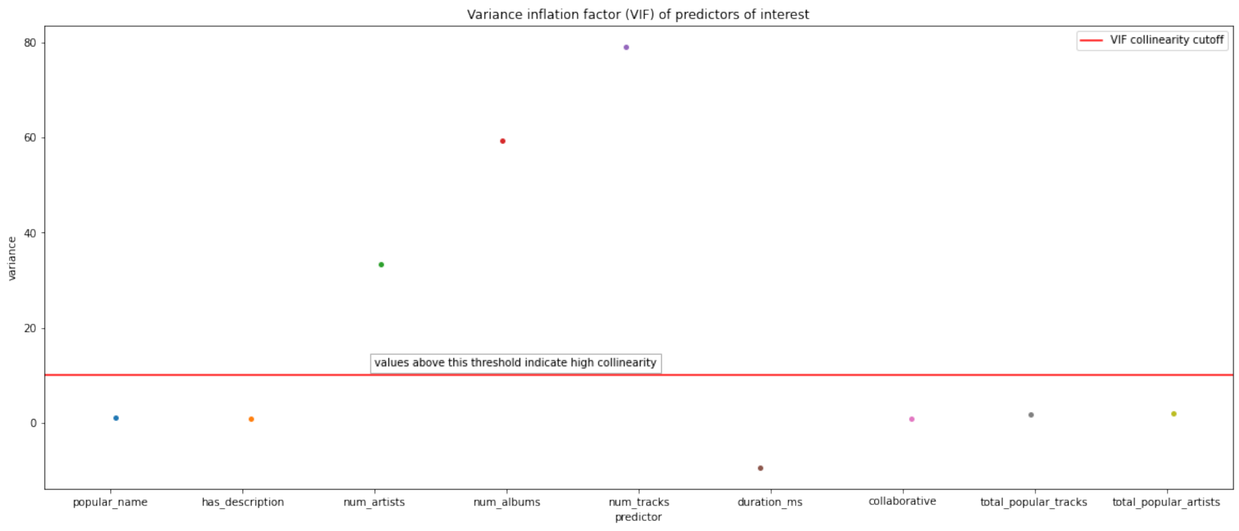


Figure 2.5: Variance inflation factor (VIF) plot showing the amount of variance on the regression coefficients due to collinearity. VIF values above 10 were considered to be caused by high collinearity. Four predictors lied above this threshold, and those with the highest correlation between other predictors (in ascending order) are: ***num_artists***, ***num_albums***, ***duration_ms***, ***num_tracks***

.

A lot of the correlation between the predictors make sense, as the more tracks you have the longer your playlist will be. Furthermore, since the number of albums and number of artists are not highly correlated with one another, we should expect to see that there are multiple albums coming from the same artist (or vice versa) in some of the playlists. When we go on to perform our regression, we will have to consider the relationship between these four predictors as they can skew our results to the extreme.

2.4 Missing Data

Since this dataset is processed in advance there is no missingness in any of our tables. However, the playlist description fields (denoted by the *description* variable) are sometimes empty, because user-provided description fields are a relatively new feature on Spotify. We don't anticipate this variable will contribute significantly to our models.

3 Research Question

What drives popularity (quantified by the number of followers) for a Spotify playlist?

Our predictors of interest include whether or not the playlist name includes commonly used words; whether or not the playlist has a description; how many artists, albums, and tracks the playlist includes; the duration of the playlist, whether or not the playlist is collaborative; and how many commonly played songs and artists are featured in the playlist.

Our ultimate goal is to build an effective regression model using some, if not all, of these predictors, to predict the number of followers of a playlist. We can then discuss what number of followers would deem a playlist to be "popular".

4 Initial Baseline Model

Since we are working with a continuous outcome (`num_followers`), we decided to build a regression model.

For our baseline model, we chose a singular `DecisionTreeRegressor` with a `max_depth` of 3. This value was chosen fairly naively, and we plan to perform cross-validation on future iterations of the model to tune this parameter. We used all of the predictors in the dataset for this model.

From this baseline model, we can then using parameter tuning, bootstrapping, bagging, and boosting approaches to improve the predictive quality.

As a result, our `DecisionTreeRegressor` model is as follows, with a training MSE of 6253.19 and a test MSE of 10681.29:

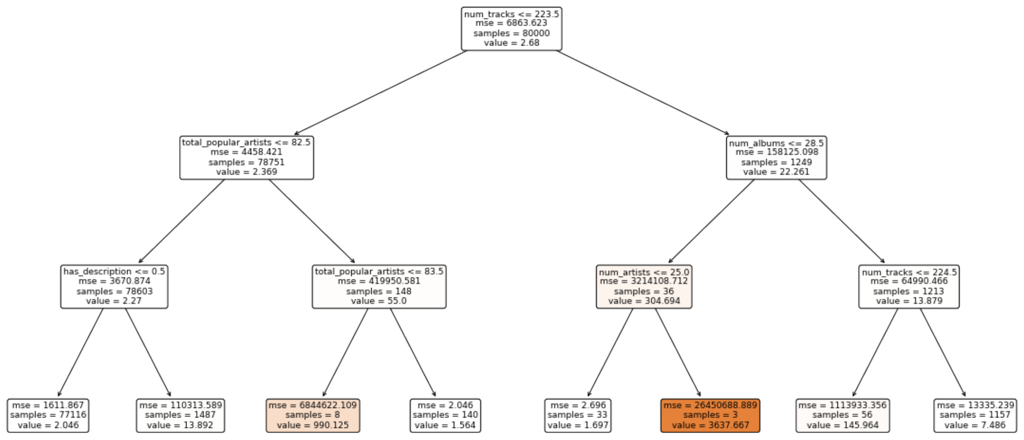


Figure 4.1: `DecisionTreeRegressor` plot showing the decision boundaries for the baseline model.