Interaction Effects and Selecting Regression Models of Taylor Swift Song Popularity

Halle Schneidewind
Interaction Effects and Selecting Regression Models of Taylor Swift
Song Popularity

An Honors Thesis submitted in partial fulfillment of the requirements for Honors
Studies in Industrial Engineering

By
Halle Schneidewind
Spring 2023

The University of Arkansas
College of Engineering
Department of Industrial Engineering

Thesis Advisor: Justin Chimka, Ph.D.
Thesis Reader: Xiao Liu Ph.D.
Abstract
Understanding music popularity and what drives it is important not only for artists but for other individuals who are financially tied to music sales including producers, writers, and record labels. Studies have been done to define how a song’s popularity can be measured, what attributes or features are drivers for popularity, and to what extent can a song’s popularity even be predicted. This paper takes two linear regression approaches to predicting the popularity of a Taylor Swift song on Spotify based on auditory features the Spotify API estimates and historic popularity of songs on Spotify. One model takes into consideration interacting predictors to divide the data into four different subsets. Another model uses backward elimination to generate one model that describes the whole dataset. Based on Bayesian Information Criteria, the collection of four models does a better job predicting a song’s popularity compared to the backward elimination model. Additionally, both models found a song’s acousticness and release year as the two most important predictors of popularity.
Introduction and Related Work

Music has become a major influence both culturally and monetarily. In 2021, the music industry grossed $25.9 billion globally [1]. On average, people spend about 2.5 hours daily listening to music [2]. In the US alone in 2020, there were 877.2 billion audio streams [3]. Despite the high number of listeners and streams, songs are not getting equal market share; 13,521 or about 0.022% of songs account for about half of the 877.2 billion streams [3]. Accompanying the high competition, it can often be expensive to release a song. According to NPR, the cost of writing, producing, recording, and marketing a hit song could be as high as a million dollars [4]. With the high cost to release music, the ability to accurately predict a song’s success prior to release is valuable. This information could help labels and artists determine which songs to release, market heavily, and prioritize as singles.

Various methods have been used to model the popularity of songs. One approach is using machine learning techniques to predict the success of future music based on the performance and attributes of historical song performance. In a study that samples 18,000 songs released between 1957 – 2020, researchers used auditory features and lyrical sentimentality to model a song’s popularity [5]. The authors used Spotify’s popularity metric and converted it to a 0/1 binary variable with songs having a popularity greater than 66.5 assigned 1 and 0 otherwise. They found a random forest model to have the highest accuracy among support vector matrix, k-nearest neighbors, logistic regression, naive Bayes, and decision tree models. In the random forest model, tempo, loudness, acousticness, and liveness were most important in determining popularity while genre, key, and mode were the least important. Another study pulls more than 170,000 songs from Spotify and uses auditory features to predict a song’s popularity [6]. These authors found the release year for a song alone has a high correlation with its popularity. In a random forest model, they found acousticness to be the most important feature in determining popularity.

Most of the current research takes an aggregate of music across genres and artists to predict hits. However, this paper takes a different approach. It focuses on one artist and their entire discography to predict the popularity of songs on Spotify based on auditory features. Taylor Swift provides a good basis to create predictive modeling because of her great popularity and notoriety and large catalogue that spans across many years and genres. In this paper, the dependent variable is the popularity value Spotify assigns, and independent variables are Spotify features. Spotify data were used for this analysis because of the relative accessibility and ease of extracting the data. In addition, with 31% of the online streaming market share and 433 million monthly active listeners globally [7], listening trends on Spotify can be representative of listening trends in the music industry as a whole.
Methodology

Data Extraction

The data were collected from the Spotify Web API server using the open source spotipy library in Python [8]. Similar data can be found on Kaggle [9] for Taylor Swift specifically. However, the Kaggle dataset does not have the most recent albums released and lacks some of the features Spotify estimates. Using the Python script in the Appendix, song data was pulled from each of Taylor Swift’s albums. Figure 1 outlines the Python script used to pull data from Spotify.

Figure 1: Python Script Summary

1-5
• Download and import libraries

6-11
• Input required credentials for Spotify API

12-27
• Pull every album from a specified artist (Taylor Swift)

36-49
• Loop through each album to pull every song from each album

28-35;50-70
• Loop through every song to pull feature and audio data

71-138
• Organize data in a dataframe to export as CSV

Data

The initial data pull includes a total of 464 songs including every album (excluding singles or EP’s) released in every market. This includes both the initial and deluxe versions of albums in addition to albums that were released multiple times in different markets. To ensure the dataset includes only one record for each song, only the most recent released album in the United States was used. For example, Taylor Swift released Red, Red (Deluxe Edition), and Red (Taylor’s Version), but only Taylor’s Version (the most recent) was considered in this analysis.

Every song from the following nine albums is included except three voice memos from 1989 (Deluxe Edition) for a total of 174 songs:

- Taylor Swift
- Speak Now (Deluxe Edition)
- 1989 (Deluxe Edition)
- reputation
- Lover
- folklore (deluxe version)
- evermore (deluxe version)
- Fearless (Taylor’s Version)
- Red (Taylor’s Version)

Variables
Every feature included was directly pulled from Spotify’s developer application programming interface or API. These features are a mix of song information, audio analysis, and audio features [10, 11].

- Track number: position of the track on the album
- Total tracks on album: Number of tracks on the album. From this dataset, lengths of albums range from 15 songs to 30. On average, a Taylor Swift album has about 21 songs (Figure 3)
- Release year: Year the album was released. When the data was initially pulled from Spotify, the entire release date was given. This was shortened to just the year to include in this analysis. The albums in this analysis were released between 2006 and 2021.
- Duration: length of track in milliseconds. The average length of a Taylor Swift song is about 4 minutes (Figure 3)
- Explicit: a binary variable that denotes if a track includes explicit lyrics (1) or no explicit lyrics (0). Of the 174 songs in this analysis, 13 contained explicit language.
- Popularity: an integer between 0-100 calculated by Spotify that denotes how often a track has been listened to and how recently. The higher the popularity value, the more popular a song is. The value used in this analysis is the popularity of each song when the data was pulled on February 2, 2022. These values appear normality distributed between 50 and 92 (Figure 4) with an average popularity score of 67.6 (Figure 3)

Audio analysis variables:
- Loudness: overall loudness of a song in decibels. The average loudness of Taylor Swift songs is -7.124 decibels (Figure 3).
- Mode: a binary variable that indicates the modality of a track where major = 1 and minor = 0. 159 of the 174 songs in this dataset have a major mode.
- Tempo: overall tempo of a song in beats per minute. The average tempo for these Taylor Swift songs is 126.51 beats per minutes but ranges from 68.53 to 208.92 beats per minutes (Figure 3).
- Time signature: an estimate of the time signature with a value from 4 to 7 with 4 being 4/4 time and 7 being 7/4 time. The most common time signature in this dataset is 4/4 time (Figure 5).
- Key: key of the song in pitch class notation ranging from 0 = C to 11 = B. Since key is a categorical variable, these values were converted into 11 dummy binary variables for each key. In linear regression, all but one of the categories need to be represented by a binary variable. For this analysis, key = 5 or a key of F was chosen as a baseline and to be excluded because it was seen to be the most different key when looking at the p-values across different baselines. The most common keys among Taylor Swift songs are C (0) with 32 songs and G (7) with 30 songs (Figure 6).
• Acousticness: a confidence measure from 0 to 1 whether the track is acoustic. 1 represents high confidence the track is acoustic. The acoustic values in this dataset have a wide range from 0.0002 to 0.97 with an average value is 0.31 (Figure 3).
• Danceability: a value between 0 to 1 describing how suitable a track is for dancing based on a combination of tempo, rhythm stability, beat strength, and overall regularity. A value of 0 is least danceable. Danceability values range from 0.29 to 0.90 with an average of 0.58 (Figure 3).
• Energy: a measure between 0 and 1 representing the perceptual measure of intensity and activity. Perceptual features include dynamic range, perceived loudness, timbre, onset rate, and general entropy. Energy ranges from 0.13 to 0.95 with an average value of 0.59 (Figure 3).
• Instrumentalness: a value between 0 and 1 predicting whether a track contains no vocals. A value close to 1 indicates a high likelihood a track has no vocals. Instrumentalness values have a relatively small range from 0 to 0.18 (Figure 3).
• Liveness: a value between 0 and 1 that detects the presence of an audience in the recording. Higher values represent an increased probability that the track was performed live. Liveness ranges from 0.038 to 0.38 with an average of 0.14 (Figure 3).
• Speechiness: a value between 0 and 1 that detects the presence of spoken words in a track. The more speech-like a recording is, the closer the value is to 1. Speechiness ranges from 0.02 to 0.52 with an average of 0.05 (Figure 3).
• Valence: a measure from 0 to 1 describing the positiveness of a track. A value close to 1 sound happy while tracks with a low valence sounds sad. Valence values range from 0.05 to 0.94 with an average of 0.41 (Figure 3).

**Predictive Modeling**

Multiple linear regression is a statistical technique that attempts to explain the linear relationship that various independent variables have on a dependent variable and is written in the form \(y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p\). The statistical software Minitab was used to estimate all linear regression models.

One model was created taking interaction into consideration. Interaction explains the role a variable has in explaining the dependent variable at different levels of another variable. Interaction can be viewed as products in full second order regression models. Variables that were not significant in the main effects model nor interacted significantly in the second order model were removed from the model, and the dataset was split based on interacting variables to create a suite of models rather than one model describing all the data. These methods are detailed elsewhere [12].

A second model was created using the backward elimination method. Backward elimination is a stepwise regression method that starts with every available variable in the model and iteratively drops one variable at a time until all remaining variables have a p-value less than some predetermined alpha level.

**Model Performance**

For each model, Ryan-Joiner, Levene’s, and Lack of Fit tests were performed to address linear regression model assumptions like normality and constant error variance in residuals. Models were compared using Bayesian Information Criteria (BIC). BIC is a model selection criterion where a model with a low BIC value is preferred. It balances model performance on the training
dataset with the complexity of the model to pick one that is likely to have the highest performance on a completely new dataset. BIC has a relatively high penalty on the number of variables included in the model, so it generally prefers simpler models. BIC was calculated according to the following equations [13]:

\[
BIC = -2\ln(\text{likelihood}) + (p + 1)\ln(n) \\
-2\ln(\text{likelihood}) = n\ln\left(\frac{RSS}{n}\right) + n + n\ln(2\pi)
\]

Where

- \( n \) = sample size
- \( p \) = number of coefficients including the constant
- RSS = residual sum of squared errors
Results

Suite of Models
The analysis began with a first order regression model using all 27 independent variables and popularity as the dependent variable in Minitab. Any variable that was not significant based on alpha of 0.05 nor interacted significantly with other variables as seen in the full second order model was removed. Figure 2 summarizes the division of the data to create four different models that will be further elaborated on in this section.

Prior to the first split in data, the following variables were insignificant in predicting a song’s Spotify popularity and were removed from models.

- Explicit
- Time Signature
- Track Number
- Total Tracks
- Duration
- Danceability
- Energy
• Instrumentalness
• Mode
• Tempo
• Speechiness
• Binary key variables 1, 2, 3, 4, 6, 7, 8, 10, 11

The data was split on release year because it significantly interacted with liveness. K-means clustering procedure was used to split the data into k = 2 groups. The data was split to have one group of songs that were released between 2006 – 2014 and the other between 2017 – 2021. K-means clustering procedure is an algorithm that will group the data into k number of clusters to minimize the distance of each datapoint to the cluster’s center value. On the dataset that includes songs released prior to 2015, all variables were removed except release year to get Model 1.

\[ Model 1: popularity = -900 + 0.477 \times release_{year} \]

On the remaining data that includes songs released after 2015, the binary key variable 9 was removed before splitting the data again on binary key variable 0 based on the significant interaction this variable had with release year and valence. Since it is a binary variable, the data was split on Key0 = 0 and Key0 = 1. When Key0 = 1, all variables except acousticness were removed to get Model 2.

\[ Model 2: popularity = 76.43 - 8.61 \times \text{acousticness} \]

From the remaining 95 datapoints, valence was removed before splitting on liveness. The data was split on liveness due to a significant interaction with acousticness. Using a k-means clustering procedure with k = 2, the data was split on the liveness value of 0.2. For liveness < 0.2, loudness was removed from the model to get Model 3.

\[ Model 3: popularity = 2328 - 1.113 \times release_{year} - 8.88 \times \text{acousticness} - 49.3 \times \text{liveness} \]

When liveness > 0.2, all but acousticness was removed to get Model 4.

\[ Model 4: popularity = 67.26 + 16.09 \times \text{acousticness} \]

All four models are summarized in Table 1.
Table 1: Summary of Models

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-900</td>
<td>76.43</td>
<td>2328</td>
<td>67.26</td>
</tr>
<tr>
<td>Acousticness</td>
<td>-8.61</td>
<td>-8.88</td>
<td>16.09</td>
<td></td>
</tr>
<tr>
<td>Liveness</td>
<td></td>
<td>-49.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Year</td>
<td>0.477</td>
<td>0.113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Size</td>
<td>51</td>
<td>28</td>
<td>83</td>
<td>12</td>
</tr>
</tbody>
</table>

**Backward Elimination Model**

Minitab has the functionality to automatically generate a regression model using stepwise regression. Using a backward elimination model, all 27 independent variables were initially considered. Iteratively, the software runs a linear regression model and removes the variable with the highest p-value until all remaining variables have a p-value lower than the predetermined alpha level of 0.05. Table 2 shows the features and their coefficients that were retained from the backward elimination model.

Table 2: Significant Variable Coefficients in Backward Elimination Model

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Backward Elimination Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-2031</td>
</tr>
<tr>
<td>Acousticness</td>
<td>-7.37</td>
</tr>
<tr>
<td>Duration MS</td>
<td>0.000018</td>
</tr>
<tr>
<td>Key = 0</td>
<td>4.06</td>
</tr>
<tr>
<td>Key = 9</td>
<td>4.33</td>
</tr>
<tr>
<td>Release Year</td>
<td>1.0403</td>
</tr>
<tr>
<td>Speechiness</td>
<td>19.47</td>
</tr>
<tr>
<td>Tempo</td>
<td>-0.0327</td>
</tr>
</tbody>
</table>
**Discussion**

Across all four models, there is not one common significant variable. Acousticness is significant in three of the four models, and release year is significant in two of the four models. For songs released before 2015, Model 1 shows the only significant predictor is release year. For songs released after 2015 and not in a key of C, there is interaction between acousticness and popularity at varying liveness levels. When the song has a high liveness value, higher acousticness values lead to a higher value for popularity. When the song has low liveness values, a higher value of acousticness leads to a lower popularity value.

Compared to the suite of models, the backward elimination model is less complex since it estimates fewer parameters. It finds that higher values of song duration, key of C and A, release year, and speechiness lead to a higher popularity value while higher values of acousticness and tempo lead to lower values of popularity.

The backward elimination models identify more significant predictors than the suite of models. The backward elimination model finds duration, key of 9, speechiness, and tempo significant while the suite of models does not. The suite of models find liveness significant where the backward elimination model does not.

Based on BIC, the suite of four models is preferred over the backward elimination model. The four models’ BIC values sum to 1073.23 which is slightly lower than the backward elimination model’s BIC value of 1117.19. The suite of models collectively estimates ten parameters and is more complex than the backward elimination model that estimates eight parameters. However, the suite of models has a significantly lower error that makes up for the increase in model complexity (Table 3). It is reasonable to theorize that the suite outperforms in part because it acknowledges and utilizes the existence of interaction.

Table 3 show that the division of the data into four smaller sets decreased the variation and spread of the data. The sum of total sum of squares across the four models is approximately half of the sum of squares of the entire dataset. The data was naturally split in groups with similar popularity values. By splitting the data and decreasing the variation, it becomes easier to identify and model trends in the data which, in this case, lead to more accurate models.

When considering the tests for model adequacy, the backward elimination model performed better and may be more accurate when using these models to predict future performance of songs. Since there was no repeat observation in the backwards elimination model, only the Ryan-Joiner test could be done to test for normality. At an alpha level of 0.05, the backwards elimination model supports the assumption of normality in residuals.

Model 1 was the only model with repeat observations. At an alpha level of 0.05, it passes the Ryan-Joiner and Lack of Fit test but fails the Levene’s test indicating an unequal variance in the residuals. However, since this model only includes data from the past and therefore would not likely be used to predict trends going forward, there is less importance on passing model adequacy tests. Models 2-4 had no repeat observations, so only the Ryan-Joiner test was done. Of the three models, Model 2 fails the Ryan-Joiner test at an alpha level of 0.05 indicating the residuals are not normally distributed.

---

**Table 3: Comparison of Models**
Conclusion

In this paper, the popularity of Taylor Swift songs was predicted with linear regression. One set of models considered the interaction of variables to divide the data into four subsets. Another model used backward elimination to find one model that describes all the data. These models identified acousticness and release year as key predictors influencing song popularity. Based on BIC, the set of models that considered interaction was preferred. However, when looking at model adequacy, the backward elimination model may be preferred. Model 1 does not pass Levene’s test for equal variance and Model 2 does not pass Ryan-Joiner test for normality. Therefore, there should be some hesitancy in using these models to predict the future performance of songs. Another consideration before choosing a model to predict popularity would be to look at model predicted rankings of song popularity and compare them to observed ranking of popularity in the dataset.
Appendix

```python
!pip install spotipy
import spotipy
import time
import numpy as np
import pandas as pd
from spotipy.oauth2 import SpotifyClientCredentials
client_id = 'fa7bf51d5c0242058e1737db56270f6f'
client_secret = 'f43f7dcade33458bb737b2f641685d86'
client_credentials_manager = SpotifyClientCredentials(client_id=client_id, client_secret=client_secret)
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
name = "Taylor Swift"
result = sp.search(name)
artists_uris = result['tracks']['items'][0]['artists'][0]['uri']
artist_albums = sp.artist_albums(artists_uris, album_type='album')
artist_album_names = []
artist_album_uris = []
albums = artist_albums['items']
while artist_albums['next']:
    artist_albums = sp.next(artist_albums)
    albums.extend(artist_albums['items'])
for album in albums:
    artist_album_names.append(album['name'])
    artist_album_uris.append(album['uri'])

def album_songs(uri):
    album = uri
    spotify_albums[album] = {}
    spotify_albums[album]['album'] = []
    spotify_albums[album]['track_number'] = []
    spotify_albums[album]['id'] = []
    spotify_albums[album]['name'] = []
    spotify_albums[album]['uri'] = []
    tracks = sp.album_tracks(album)
    for n in range(len(tracks['items'])):
        spotify_albums[album]['album'].append(artist_album_names[album_count])
```

spotify_albums[album]['track_number'].append(tracks['items'][n]['track_number'])
spotify_albums[album]['id'].append(tracks['items'][n]['id'])
spotify_albums[album]['name'].append(tracks['items'][n]['name'])
spotify_albums[album]['uri'].append(tracks['items'][n]['uri'])

spotify_albums = {}
album_count = 0
for i in artist_album_uris:
    album_songs(i)
    print(str(artist_album_names[album_count]) + 
    album songs has been added to spotify_albums dictionary)
    album_count+=1

def audio_features(album):
    spotify_albums[album]['acousticness'] = []
    spotify_albums[album]['danceability'] = []
    spotify_albums[album]['energy'] = []
    spotify_albums[album]['instrumentalness'] = []
    spotify_albums[album]['key'] = []
    spotify_albums[album]['liveness'] = []
    spotify_albums[album]['loudness'] = []
    spotify_albums[album]['mode'] = []
    spotify_albums[album]['speechiness'] = []
    spotify_albums[album]['tempo'] = []
    spotify_albums[album]['time_signature'] = []
    spotify_albums[album]['valence'] = []
    spotify_albums[album]['popularity'] = []
    spotify_albums[album]['album_type'] = []
    spotify_albums[album]['total_tracks'] = []
    spotify_albums[album]['available_markets'] = []
    spotify_albums[album]['release_date'] = []
    spotify_albums[album]['duration_ms'] = []
    spotify_albums[album]['explicit'] = []

track_count = 0
for track in spotify_albums[album]['uri']:
    features = sp.audio_features(track)
    spotify_albums[album]['acousticness'].append(features[0]['acousticness'])
    spotify_albums[album]['danceability'].append(features[0]['danceability'])
    spotify_albums[album]['energy'].append(features[0]['energy'])
    spotify_albums[album]['instrumentalness'].append(features[0]['instrumentalness'])
    spotify_albums[album]['key'].append(features[0]['key'])
    spotify_albums[album]['liveness'].append(features[0]['liveness'])
spotify_albums[album]['loudness'].append(features[0]['loudness'])
spotify_albums[album]['mode'].append(features[0]['mode'])
spotify_albums[album]['speechiness'].append(features[0]['speechiness'])
spotify_albums[album]['tempo'].append(features[0]['tempo'])
spotify_albums[album]['time_signature'].append(features[0]['time_signature'])
spotify_albums[album]['valence'].append(features[0]['valence'])
info = sp.track(track)
spotify_albums[album]['popularity'].append(info['popularity'])
spotify_albums[album]['album_type'].append(info['album']['album_type'])
spotify_albums[album]['total_tracks'].append(info['album']['total_tracks'])
spotify_albums[album]['available_markets'].append(info['album']['available_markets'])
spotify_albums[album]['release_date'].append(info['album']['release_date'])
spotify_albums[album]['duration_ms'].append(info['duration_ms'])

track_count += 1

sleep_min = 2
sleep_max = 5
start_time = time.time()
request_count = 0
for i in spotify_albums:
    audio_features(i)
    request_count += 1
    if request_count % 5 == 0:
        print(str(request_count) + " playlists completed")
        time.sleep(np.random.uniform(sleep_min, sleep_max))
        print('Loop #: {}'.format(request_count))
        print('Elapsed Time: {} seconds'.format(time.time() - start_time))

dic_df = {
    'album': [],
    'track_number': [],
    'id': [],
    'name': [],
    'uri': [],
    'album_type': [],
    'total_tracks': [],
    'available_markets': [],
    'release_date': [],
    'duration_ms': [],
    'explicit': [],
    'acousticness': []}
for album in spotify_albums:
    for feature in spotify_albums[album]:
        dic_df[feature].extend(spotify_albums[album][feature])

len(dic_df['album'])
dataframe = pd.DataFrame.from_dict(dic_df)
final_df = dataframe.sort_values('popularity', ascending=False).drop_duplicates('name').sort_index()
final_df.to_csv("taylor_swift_album.csv")

General information on Spotify features:

- Track number: position of the track on the album
- Total tracks on album: Number of tracks on the album. From this dataset, lengths of albums range from 15 songs to 30. On average, a Taylor Swift album has about 21 songs (Figure 3)
- Release year: Year the album was released. When the data was initially pulled from Spotify, the entire release date was given. This was shortened to just the year to include in this analysis. The albums in this analysis were released between 2006 and 2021.
- Duration: length of track in milliseconds. The average length of a Taylor Swift song is about 4 minutes (Figure 3)
- Explicit: a binary variable that denotes if a track includes explicit lyrics (1) or no explicit lyrics (0). Of the 174 songs in this analysis, 13 contained explicit language.
- Popularity: an integer between 0-100 calculated by Spotify that denotes how often a track has been listened to and how recently. The higher the popularity value, the more popular a song is. The value used in this analysis is the popularity of each song when the data was pulled on February 2, 2022. These values appear normality distributed between 50 and 92 (Figure 4) with an average popularity score of 67.6 (Figure 3)

Audio analysis variables:

- Loudness: overall loudness of a song in decibels. The average loudness of Taylor Swift songs is -7.124 decibels (Figure 3).
- Mode: a binary variable that indicates the modality of a track where major = 1 and minor = 0. 159 of the 174 songs in this dataset have a major mode.
- Tempo: overall tempo of a song in beats per minute. The average tempo for these Taylor Swift songs is 126.51 beats per minutes but ranges from 68.53 to 208.92 beats per minutes (Figure 3).
- Time signature: an estimate of the time signature with a value from 4 to 7 with 4 being 4/4 time and 7 being 7/4 time. The most common time signature in this dataset is 4/4 time (Figure 5).
- Key: key of the song in pitch class notation ranging from 0 = C to 11 = B. Since key is a categorical variable, these values were converted into 11 dummy binary variables for each key. In linear regression, all but one of the categories need to be represented by a binary variable. For this analysis, key = 5 or a key of F was chosen as a baseline and to be excluded because it was seen to be the most different key when looking at the p-values across different baselines. The most common keys among Taylor Swift songs are C (0) with 32 songs and G (7) with 30 songs (Figure 6).

Audio features:
- Acousticness: a confidence measure from 0 to 1 whether the track is acoustic. 1 represents high confidence the track is acoustic. The acoustic values in this dataset have a wide range from 0.0002 to 0.97 with an average value is 0.31 (Figure 3).
- Danceability: a value between 0 to 1 describing how suitable a track is for dancing based on a combination of tempo, rhythm stability, beat strength, and overall regularity. A value of 0 is least danceable. Danceability values range from 0.29 to 0.90 with an average of 0.58 (Figure 3).
- Energy: a measure between 0 and 1 representing the perceptual measure of intensity and activity. Perceptual features include dynamic range, perceived loudness, timbre, onset rate, and general entropy. Energy ranges from 0.13 to 0.95 with an average value of 0.59 (Figure 3).
- Instrumentalness: a value between 0 and 1 predicting whether a track contains no vocals. A value close to 1 indicates a high likelihood a track has no vocals. Instrumentalness values have a relatively small range from 0 to 0.18 (Figure 3).
- Liveness: a value between 0 and 1 that detects the presence of an audience in the recording. Higher values represent an increased probability that the track was performed live. Liveness ranges from 0.038 to 0.38 with an average of 0.14 (Figure 3).
- Speechiness: a value between 0 and 1 that detects the presence of spoken words in a track. The more speech-like a recording is, the closer the value is to 1. Speechiness ranges from 0.02 to 0.52 with an average of 0.05 (Figure 3).
- Valence: a measure from 0 to 1 describing the positiveness of a track. A value close to 1 sound happy while tracks with a low valence sounds sad. Valence values range from 0.05 to 0.94 with an average of 0.41 (Figure 3).
Citations


