Song Prediction Algorithm

Introduction

In today's world, when people want to listen to music, they open streaming services such as Amazon Music, Apple Music, or Spotify on their phones or computers. These services also keep detailed records of all songs that people search for and listen to. This data is used to create automatically curated playlists for users to listen to, often with songs and artists that they have never heard before. I am going to analyze the data on my own liked and disliked songs in order to create an algorithm that can predict whether or not I like a song. Such an algorithm could be applied to anyone.

Problem

The main problems I found during my initial research were:

1. How can the unstructured data of a music track be fed into a regression function?
2. What correlation might there be between songs in different genres.
3. Spotify's current prediction algorithm is proprietary, so I do not know how the app currently chooses songs.
4. Can something as esoteric and personal as music preference be easily and correctly quantified.

Survey

There has been significant research on this subject before. The music industry is a multi-billion dollar market and its products are consumed by billions around the globe. Every streaming service has its own algorithms for song prediction and playlist curation. For example, Spotify uses acoustic analysis to create data for their own algorithm. While I do not believe my method will be better than existing algorithms for general use, I believe mine will work better for a me.

Proposed Method

The first step in my method for predicting whether I will like a song is to download data on my past liked and disliked songs from the Spotify API. After collating and cleaning the data by removing unnecessary info and structuring it correctly, I will create a pairwise plot and correlation heatmap to find any obvious correlations between variables. After deciding on independent and dependent variables, I will split the data into a training and testing set. Now the logistic regression model can be created and tested.

Experiment and Evaluation

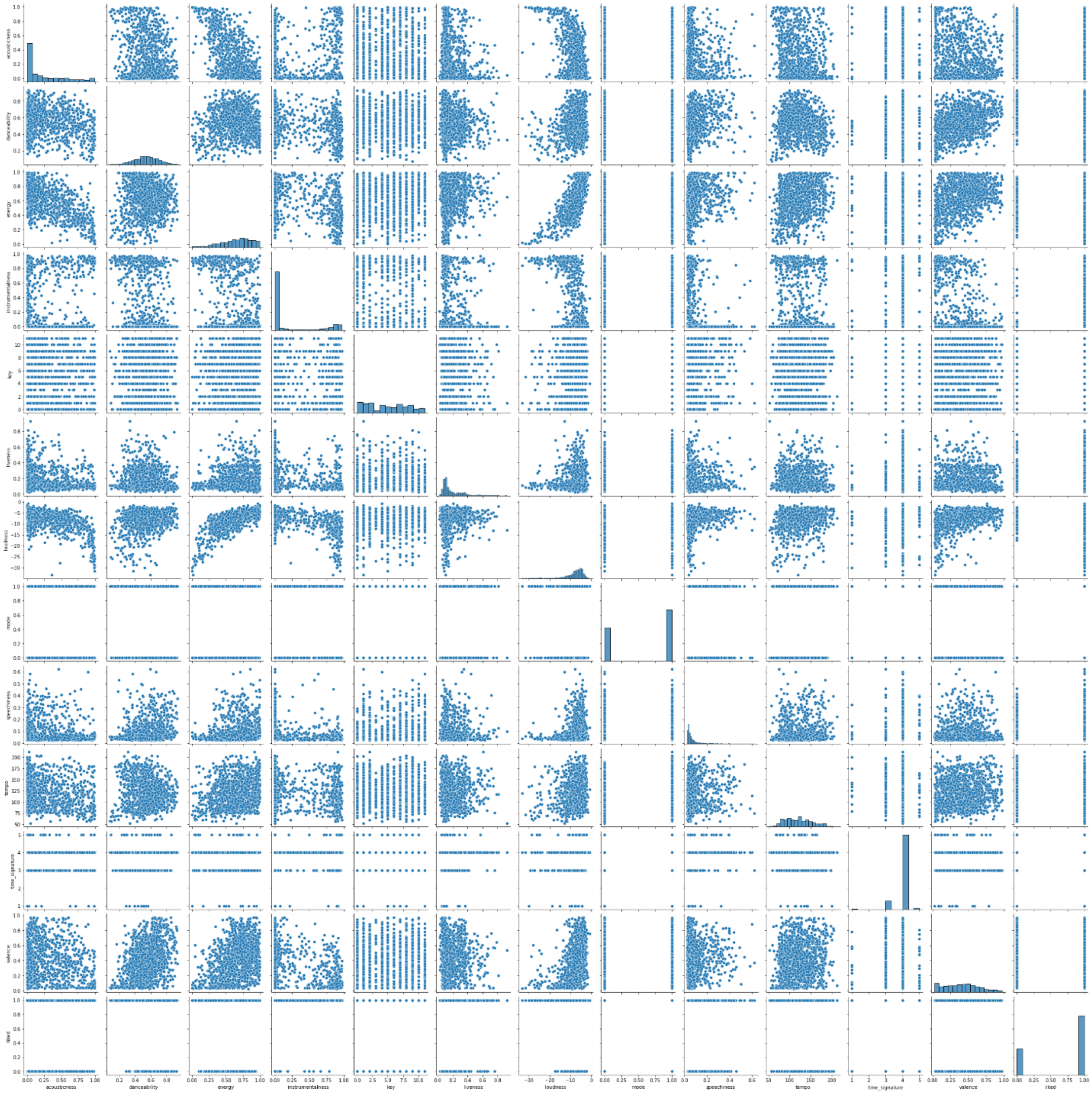
To create the data, I used an https GET request to download a list of song IDs. I then used the ID list in another GET request to pull metadata for each song. After reading the collected JSON files into a DataFrame I began my analysis.

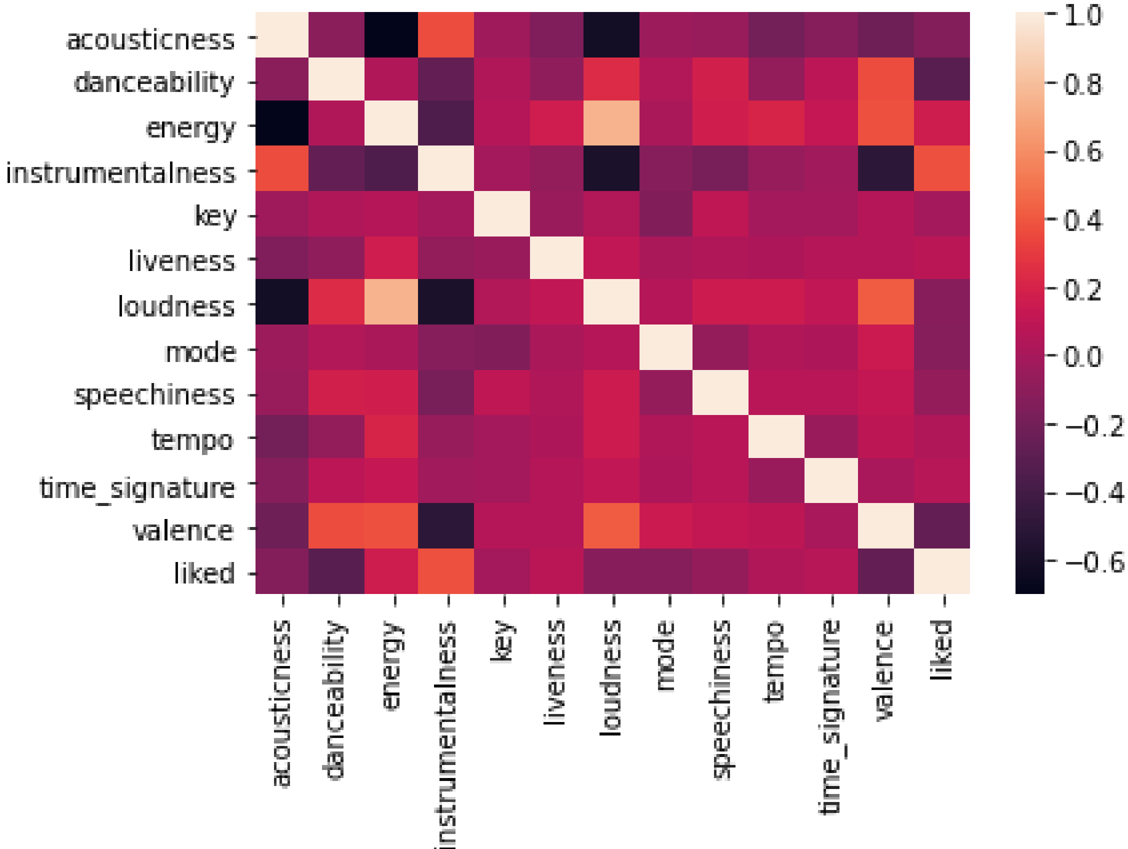
The first thing I did was remove unneeded data, leaving only the following categories:

Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
Energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Instrumentalness	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
Key	The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.
Liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
Loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
Mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
Tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
Time Signature	An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4".
Valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
Liked	Binary value of 0 or 1 representing whether or not the song is disliked or liked

(Above descriptions from https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features)

I then created a pairwise plot and correlation heatmap of all the variables:




acousticness
danceability
energy
instrumentalness
key
liveness
loudness
mode
speechiness
tempo
time_signature
valence
liked
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Through these charts I was unable to find any correlations, so I moved onto my regression model. I used logistic regression because of this is a classification problem.

```
Optimization terminated successfully.
         Current function value: 0.386798
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                  liked   No. Observations:                  838
Model:                          Logit   Df Residuals:                      825
Method:                           MLE   Df Model:                           12
Date:                Thu, 09 Dec 2021   Pseudo R-squ.:                  0.3773
Time:                        11:33:36   Log-Likelihood:                -324.14
converged:                       True   LL-Null:                       -520.57
Covariance Type:            nonrobust   LLR p-value:                 1.226e-76
====================================================================================
                       coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------
const               -2.5213      1.437     -1.755      0.079      -5.337       0.295
acousticness        -1.4908      0.494     -3.015      0.003      -2.460      -0.522
danceability        -3.2498      0.770     -4.221      0.000      -4.759      -1.741
energy               5.5264      0.902      6.126      0.000       3.758       7.294
instrumentalness     6.2857      1.003      6.267      0.000       4.320       8.251
key                 -0.0128      0.029     -0.443      0.658      -0.069       0.044
liveness             0.4746      0.795      0.597      0.550      -1.083       2.032
loudness            -0.1304      0.054     -2.423      0.015      -0.236      -0.025
mode                -0.3678      0.209     -1.763      0.078      -0.777       0.041
speechiness          0.4037      1.048      0.385      0.700      -1.650       2.457
tempo                0.0029      0.003      0.897      0.370      -0.003       0.009
time_signature       0.2875      0.232      1.240      0.215      -0.167       0.742
valence             -1.7463      0.518     -3.371      0.001      -2.762      -0.731
====================================================================================
```

I then used the model to create an array of predicted values and created a confusion matric and accuracy score:

Confusion Matrix	70	50
	39	255
Accuracy Score: 0. 785024154589372		

With an accuracy score of approximately 79%, I believe that this model would have a high chance of choosing songs that I would like.

Conclusion

After finishing my analysis and model, I believe that song preferences can indeed by quantified correctly, and that given enough data, songs can be chosen for users with a reasonable guarantee that they will be enjoyed.

However, I have also come to the conclusion that while the analysis is sound, the result is inherently flawed due to a lack of data on things such as the user's mood, ongoing activities, and time of day. Most people have some differences between the music they listen to in a car, while doing work, or working out. The data also does not take into account the differences between songs in different languages. If these problems are addressed then this method may be used to create an application for automatic music recommendations.

Distribution of Effort

This was a solo project, and I did all the work myself.