Classifying Songs Into the Top 200 Charts of Spotify

Kanin Bender
Indiana University
Luddy School of Informatics, Computing, and Engineering

Anna Kim
Indiana University
Luddy School of Informatics, Computing, and Engineering

Samantha Sharp
Indiana University
Luddy School of Informatics, Computing, and Engineering

With Spotify's dominance in the music streaming market, having a song chart in the top 200 songs on the platform can be a very lucrative event for an artist. Having a song chart on the platform begs the question of whether the song's charting position is a result of the musical features of the track, the listenership and/or reach of an artist's audience, or some combination in between. To answer this question, we evaluate five different statistical models for classification.

Keywords—Spotify, music features, rating, classification, artificial intelligence

I. Introduction

Boasting over 172 million premium subscribers, Spotify is a leading company in the music streaming industry [1]. Users go to Spotify to listen to music, follow other users and artists, and create and browse playlists. Users can create their own playlists, or listen to playlists curated by Spotify, including the playlist this project focused on: the Top 200 Weekly charts. These charts consist mostly of the latest releases from mainstream, famous artists, but there is the occasional song that is completely unexpected whether it be from an unknown artist or different genre. This project focuses on each song's characteristics and features to try and classify whether it would chart in the Top 200 Weekly Global Charts on Spotify.

All code and datasets used are on our GitHub (https://github.iu.edu/sharpsam/B365-Spotify-Project).

II. Datasets

There were two datasets that were used to create classification models. The first dataset was songs that appeared on the Top 200 Weekly charts any time between December 27, 2019 until July 23, 2021, hereinafter referred to as Top 200, and the second dataset was all songs on Spotify, hereinafter referred to as All Songs.

A. Top 200

Collected from Kaggle, this dataset contains information of the Top 200 Weekly Global charts of Spotify in 2020 and 2021. The features in this data are highest charting position, number of times charted, week of highest charting, song name, song ID, number of streams, artist, artist followers, genre, release date, weeks charted, popularity, danceability, acousticness, energy, chord, instrumentalness, liveness, loudness, speechiness, tempo, valence, and duration. Some of the features were not to be used; however, a description of the important features will be described in Section V, Part A.

B. All Songs

Also collected from Kaggle, this contains data of songs found on Spotify between 1922 and April 16, 2021. The features on this dataset are song ID, song name, popularity, duration, explicit, artists, artists' IDs, release date, danceability, energy, key, mode, speechiness, acousticness, tempo, valence, instrumentalness, liveness, and time signature. In order to start cleaning the data, only the songs released between January 1, 2020 and April 16, 2021 were used.

III. Data Cleaning

Data cleaning was performed in R. Once both datasets were combined, all duplicate columns were removed since the datasets had overlapping scopes. Then, because All Songs contained songs that were also in the Top 200 dataset, the duplicate songs in both datasets were removed. After the datasets did not contain duplicate attributes or entries, a column with binary values was added to determine the class for classification. If the song was in the Top 200, the class would be "True", anything else would be "False." This new data was then merged into a csv file for data collection.

IV. Data Collection

Since both datasets had different attributes in their original datasets, once merged, there were remaining columns that were blank for one set and other columns that were blank for the other. To solve this issue, a Python script was created that parsed through each row of the Excel document, determining whether the data was from the Top 200 dataset or the All Songs dataset, and based on the identification, would place the value needed into the blank column. Once all of the entries were parsed through, the entries were saved into a new Excel sheet.

A. Spotipy Library

For the Spotipy library to function, it required a client ID and client secret to be passed to the Spotify API for data collection.

A config file was created at the start that contained the identification information for this project. This was obtained by creating an account on the Spotify Developer page.

B. Openpyxl Library

This library was used to open, edit, and save the new and old Excel documents to obtain and insert the data.

C. Attributes Obtained

The song was determined to be in the Top 200 if the value in the Highest Charting Position column was "NA". These songs needed the attributes mode, time signature, explicit, artists' IDs.

If the song was not in the Top 200, it needed the number of followers[1], genres[2], and chord.

The program also removed brackets for columns genre and artists to unify the two datasets.

Streams on Spotify was not collected due to the Spotify API not having a method to obtain this value; however, the value would be unnecessary for the scope of this project since higher streams would directly correlate to the song being in the Top 200 dataset.

V. Feature Selection

None of the non-numeric/categorical attributes were considered due to the models being unable to consistently evaluate the data, This left behind the following attributes: number of followers of the artist, popularity, danceability, energy, loudness, speechiness, acousticness, liveness, tempo, duration, valence, explicit, mode, instrumentalness, time signature, and key. Popularity was also excluded from the final models because it is calculated at the time that it is collected, so the value can be unreliable, and directly correlates to whether a song would be in the Top 200 dataset.

A. R Code

Using the Boruta library, each attribute was rated based on the mean importance, determined by the Boruta library, listed below [2][3].

1) Artist Followers: Number of followers of the main artist

2) Explicit: Boolean of whether the track contains explicit words

3) Energy: Represents a perceptual measure of intensity and activity, measured from 0.0 to 1.0

4) Loudness: Overall loudness of the track in decibels

5) Speechiness: Presence of spoken words. The more speech-like, the closer to 1.0 the value is

6) Acousticness: Measure of whether the track is acoustic from 0.0 to 1.0

7) Duration: Length of track in milliseconds

8) Instrumentalness: Predicts whether a track contains no vocals. The track likely has no vocal content when the value is closer to 1.0

9) Danceability: How suitable the track is for dancing. This is based on tempo, rhythm stability, beat strength, and overall regularity. A value of 1.0 is the most danceable

10) Valence: The musical positiveness conveyed by a track. Measured from 0.0 to 1.0

11) Tempo: The overall beats per minute a track is

12) Liveness: The presence of an audience in the recording. The higher the value, the higher the probability that the track was performed live

13) Time Signature: The overall beats in each bar of the track

14) Mode: Modality of the track: 0 is minor and 1 is major

15) Key: Main chord of the song instrumental in key format

VI. Classification Models

The below models were implemented with Python scripts using the scikit-learn library. Any import statement mentioned is from this library.

A. Naïve Bayes Classifier

For this classifier, a Gaussian Bayes model was run on the attributes described in part V. There is no feature selection in this model. The model outputs train/test size accuracy, recall, and precision into a file, each train size increasing by 0.5 each iteration, starting with 0.05.

B. K-Nearest Neighbor (KNN)

Using the KNeighborsClassifier import, the data was normalized using the StandardScaler, at first with all the attributes in part V, but later used forward selection for training and testing. After forward selection, the only attribute determined to be the most useful in creating the best model is artist followers. To refine the model, the GridSearchCV import was used with the optimal amount of neighbors equal to one.

C. Decision Tree

For this decision tree, the DecisionTreeClassifier import was used for its creation. To prevent overfitting, the program found the best alpha, and then created the decision tree where the accuracy, precision, and recall comes from. Similarly to Naïve Bayes Classifier, the results were printed with each train size increasing by 0.5 each iteration starting with 0.05. For the selected training size of 70%, the attributes that the decision tree found the most important were artist followers, duration, acousticness, loudness, energy, instrumentalness, and explicit.

[1] Since some songs have multiple artists, there can be different amounts of followers for the artists. To make it one number to be comparable, only the greatest number of followers was kept in the dataset

[2] Spotipy can only record the genre of the artists themselves and not the genre of the song, so this value is the compilation of all genres of the featured artists.

D. Random Forest

The Random Forest model also utilized the pandas library and is first trained with the maximum depth being 5 and a penalized fit of 0.03. From here, any attribute with a less than 0.05 impact was removed and the model was retrained and reran without a maximum depth. Attributes deemed important in this model are artist followers, valence, energy, and speechiness.

E. Neural Network

The MLPClassifier import was used to create this model. It used the rectified linear unit function for the activation function, the stochastic gradient-based optimizer for the solver, and had one hidden layer with 100 nodes. This was normalized using Standard Scalar and did not do any feature selection.

VII. Results

Model	Accuracy	Precision	Recall
Naïve Bayes	0.91	0.49	0.32
KNN	0.97	0.83	0.81
Decision Tree	0.93	0.58	0.42
Random Forest	0.97	0.85	0.79
Neural Network	0.93	0.58	0.45

Results are based on 70% training and 30% testing based on the Naïve Bayes Classifier, making it our benchmark. All other models were expected to do better than the benchmark, making this training/testing split the best for evaluation.

VIII. Conclusion

After analyzing the results of the classifiers, a track's audio features were not as important as the popularity of the track and the artist. Because of how the models were created, a lot of the song's information was lost as well due to some attributes not being quantifiable numbers.

A next step would be to retest the models with new data. Due to the time frame and scope of this project, new data to further test and develop the classifiers was not able to be obtained. For another study, being able to evaluate the categorical data not used in this project might result in different models and important characteristics.

IX. Contributions

1) Kanin: Worked on cleaning the data, bayes classifier, and random forest.

2) Anna: Worked on the presentation, K-Nearest Neighbor, and Neural Network.

3) Samantha: Worked on data collection, Decision Tree, and the write-up.

[1] "Spotify Users - Subscribers in 2021." Statista. Statista Research Depeartment, November 4, 2021. https://www.statista.com/statistics/244995/number-of-paying-spotify-subscribers/.

[2] Spotify Web API. Spotify AB. Accessed November 1, 2021. https://developer.spotify.com/documentation/web-api/reference/#/.

[3] Pillai, Sashank. "Spotify Top 200 Charts (2020-2021)." Kaggle, August 16, 2021. https://www.kaggle.com/sashankpillai/spotify-top-200-charts-20202021/tasks.