Spotify Hit Detector

ABSTRACT

With the increase in access to technology to the common man, a lot of information is available to them. Artists around the world can share their work through the internet, making it available to everyone. With this magnitude of information and data moving around, music artists find it difficult to get peoples' attention without the necessary promotions of the track. However, with the increase in the cost of advertising, music artists starting their careers will want a tool that will help them in identifying tracks that have the potential to become popular.

INTRODUCTION

An average human nowadays has access to thousands of tracks in each of their favourite genres. However, there are multiple instances where a track that has real potential to become a hit song, hasn't. The main reason for this is that the artist decided to spend equal amounts of money promoting all his songs, effectively reducing the money they are putting into each song in terms of marketing and promotion.

By doing this, they are effectively killing their chances of making it big in the music world, as all it takes for an artist to gain some traction in the public's eye is one great song. As beginning artists, they may be experimenting with the tracks and style of music they want to create. Therefore, the Spotify Hit Detector tool will help them understand which songs of theirs have the potential to become a hit, and which don't. By using this, they will be able to increase the amount of money they can put into marketing and promotion of each of these 'potential hit' tracks and thus, reach more people with their best music.

Additionally, this could also help some record label companies who are trying to look for new artists. So far, the selection process is quite subjective, in that the decision is made by a single person. This, however, could result in some bias creeping into the decision-making process, which could positively/negatively affect the company, as well as the artist. Using this the tool could help them remove this subjectivity from the talent acquisition process and hence, improve the quality of music in the long run.

In terms of challenges, a major challenge that this tool faces at this point is the limitation that it only works well with tracks that have the United States as a market. This means that songs from other countries such as the UK, India, Australia, etc. will not work well with this tool now. Talking about impact, this tool will help improve the quality and standard of music if used regularly. However, this would increase the quality of music in extremely popular genres, and thus, this improved quality of the music will reach a lot of people in the world.

PROBLEM STATEMENT

Use the track features developed by Spotify to predict whether a particular track has the potential to become a 'Hit Song' or not.

PROPOSED SOLUTION

OVERVIEW

This is a service, where the user can input the link/ID of the song and can find out whether the song has the potential to become a hit song. The way this would work would be, that we use the link/ID to get the track features from the Spotify API endpoint and feed these features to our ML model to predict whether the song could be a hit or not. Our machine learning model will use song data to see which songs have been a hit based on their track features. The Machine Learning model that was used to train the data is the XGBClassifier, which makes use of Gradient Boosting Decision Trees to help classify data.

DETAILED OUTLINE OF THE PROCESS

We started with three different datasets, one for each of the decades (the 90s, 00s, and 10s). These datasets were combined, and any duplicates were removed. Additionally, the records in the master set were shuffled to not include any bias related to the time of release and the time when it was popular. Once this was done, we were left with 18,316 records of data. This data was perfectly balanced, meaning that exactly 50% of the data had a target of 0 and the other 50%

of the data had a target of 1. This will help with the training process, as there aren't any class imbalance biases that can affect the model.

The first model tried was Logistic Regression. The reason for this was the fact that we had a lot of numeric data and very few categorical variables. This proved to be adequate to try Logistic Regression to see the results and set a baseline for any other models that would be tried. The best accuracy score that we could with Logistic Regression using Randomized Search for hyperparameter tuning was around 0.82. On the testing set, it performed slightly better, getting an accuracy of around 0.832.

Next, Random Forest Classifier was used as the model on the data. This was the choice as I wanted to test out a decision tree model and Random Forest has been very effective in the past for binary classification. Once again, using Randomized Search for hyperparameter tuning, the best score I got was an accuracy of around 0.822, which isn't a significant increase compared to the Logistic Regression model. When used on the testing set, once again, it performed better, with an accuracy of around 0.8341.

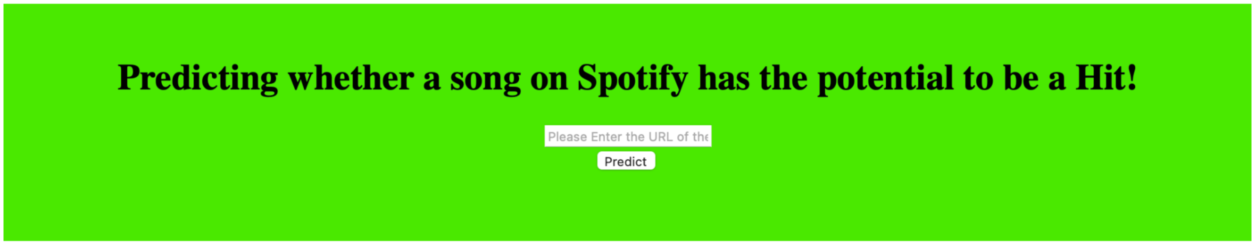
Looking at the promising results that the Random Forest model gave, I used the XGBClassifier for the final model. Once again, its record has been quite good, consistently performing well. The best score of the model after a Randomized Search was accuracy of around 0.866. This is a significant increase compared to the Logistic Regression and Random Forest models. On the testing set, it performed even better, getting an accuracy score of around 0.882. This model showed great accuracy as well as a good generalization to new data. Therefore, this is the model that was chosen to use in the tool.

Next, this model was serialized into a pickle file. Two simple and basic HTML files were created (landing page, and results-display page) to act as the interface for the user to use the tool. The tool was created using Flask to create an API endpoint for our tool, which will get the user's input (song URL). From the URL, the song ID was extracted and sent to a module that interacted with the Spotify, API to retrieve the necessary features of the audio. Simultaneously, the model's pickle file was deserialized to use the model to predict the target for new inputs.

Once all the features were obtained, a single row data frame was created to supply the input to the model. Based on the result, an output sentence was formatted to be displayed on the results-display HTML page. An important part of the process of interacting with the Spotify API was to obtain an OAuth token from their website. These tokens, however, expire in an hour. An error message will be displayed when the token has expired, making it clear that the model won't work at all and giving an output that the track is a flop song.

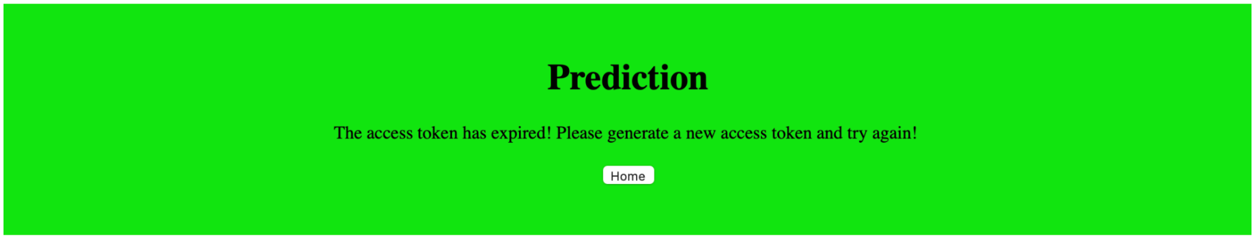
USER EXPERIENCE PROCESS

Below is a screenshot of the landing page of the tool.



Predicting whether a song on Spotify has the potential to be a Hit!

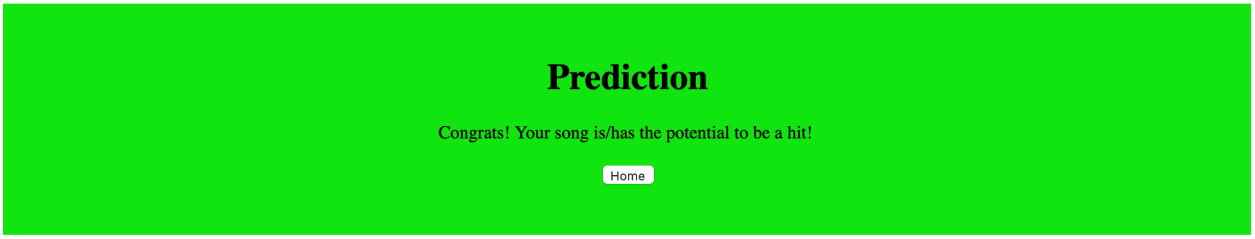
Now, we can see the error message when the OAuth token has expired.



Prediction

The access token has expired! Please generate a new access token and try again!

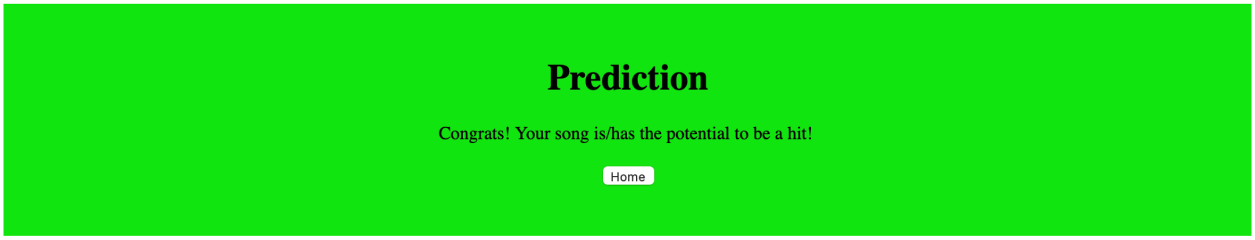
After generating a new access token, and inputting the link for the track Starboy - The Weeknd, the following is the output by the tool.



Prediction

Congrats! Your song is/has the potential to be a hit!

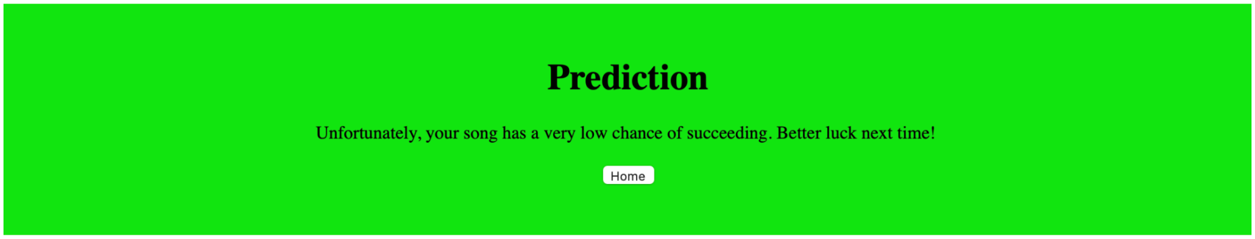
Now, let’s test it with a smaller artist. Inputting the link for the song Wonky - Point Blank (297,390 plays) , the following is the output.



Prediction

Congrats! Your song is/has the potential to be a hit!

Let us use another song from the same artist with lesser plays. Inputting the link of RITMO - Point Blank (7,759 plays), the following is the output.

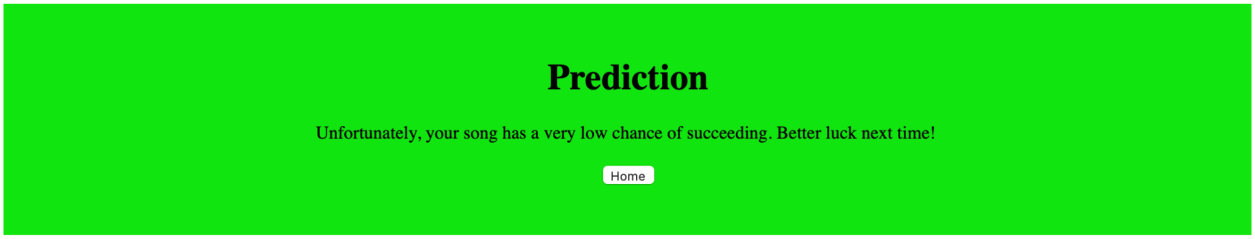


Prediction

Unfortunately, your song has a very low chance of succeeding. Better luck next time!

EXPERIMENTAL RESULTS

I wondered what the output would be when I tried to input a song that wasn't from the US. Inputting the link of a song by my friend, Kubool Hai Na - Arvind, the following is the output. This song was made in the UK, but mainly meant for audiences in India.



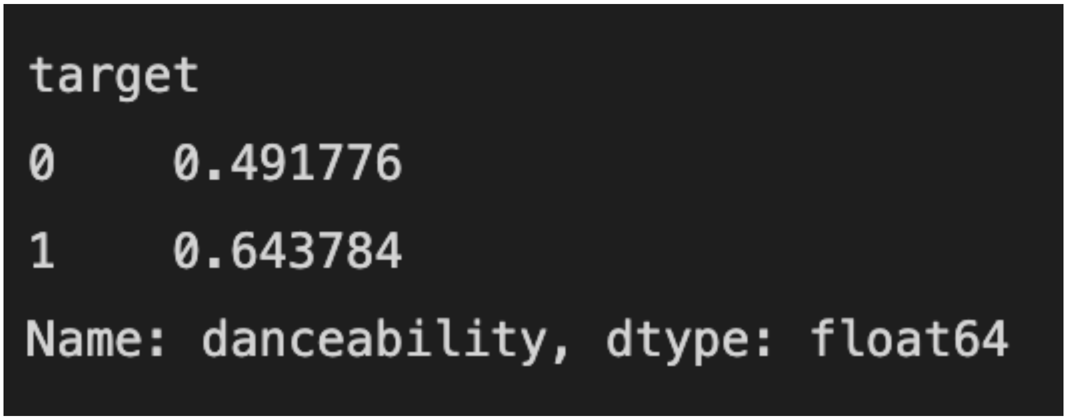
Prediction

Unfortunately, your song has a very low chance of succeeding. Better luck next time!

EXPLORATORY DATA ANALYSIS

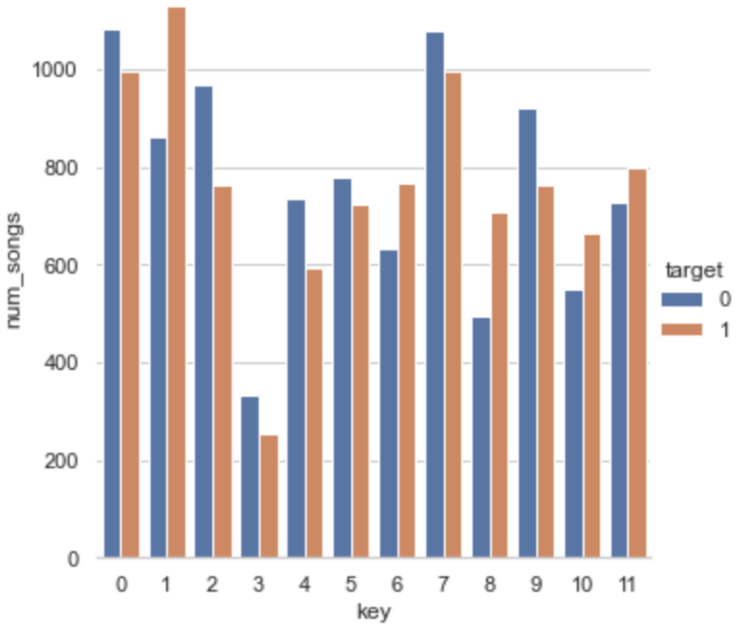
I tried to further explore the data to see if there are any patterns, I can find that can help artists make more popular songs. The following are the results of this.

We can see below that hit songs, on average, have a higher danceability value, which makes sense. However, the difference was quite surprising.

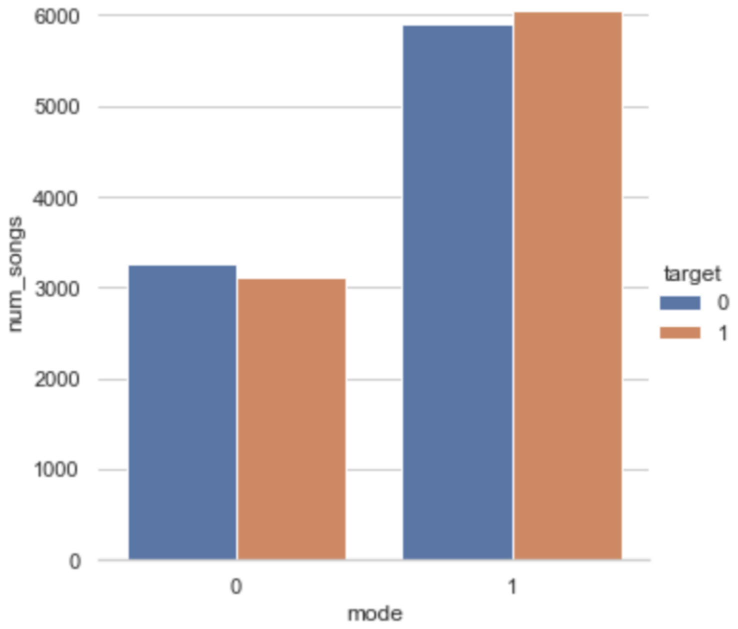


```
target
0    0.491776
1    0.643784
Name: danceability, dtype: float64
```

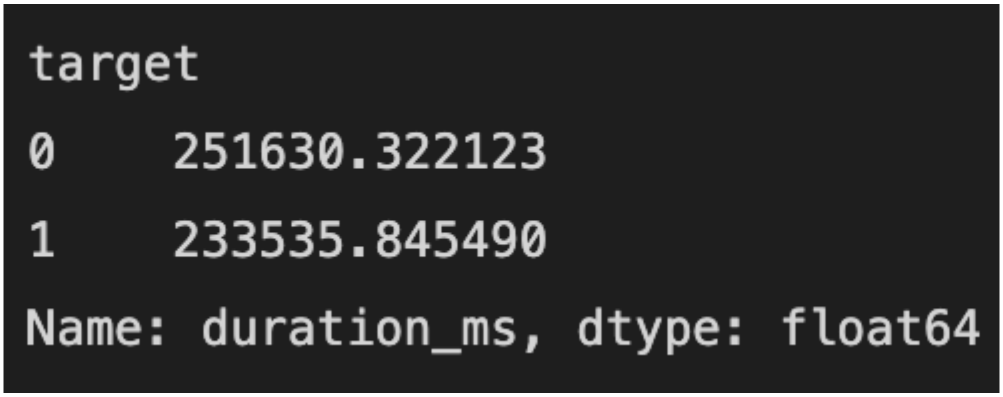
I also tried to see the pattern between keys and the target. The result is displayed below.



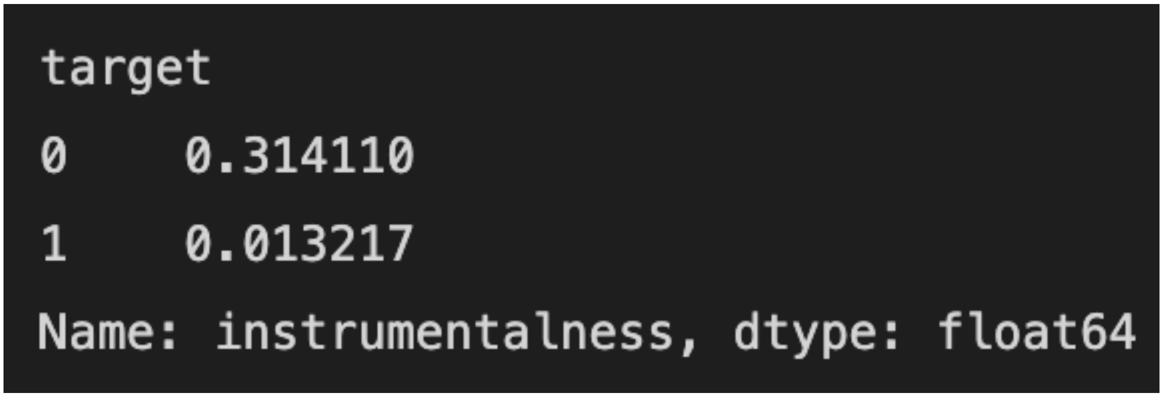
Additionally, I wanted to see the pattern between a song's modality (major/minor) and the target distribution.



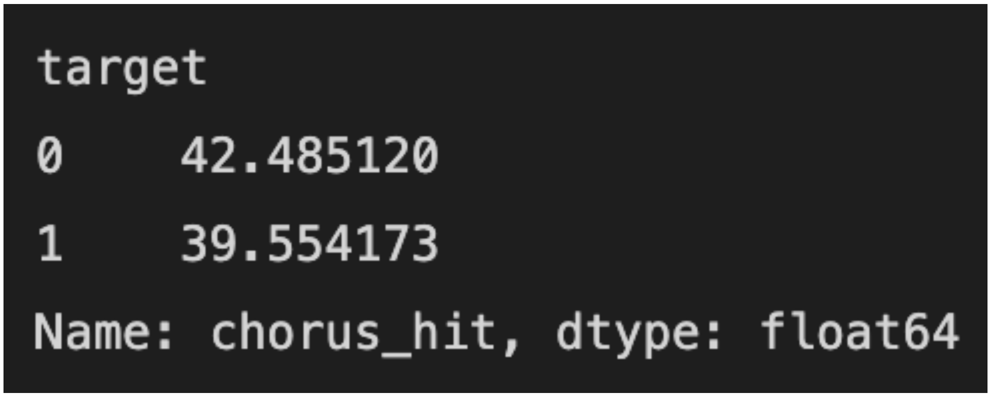
I also tried to see the correlation between the duration of a song, instrumentalness and the timestamp when the first chorus started. Below are the results.



```
target
0    251630.322123
1    233535.845490
Name: duration_ms, dtype: float64
```



```
target
0    0.314110
1    0.013217
Name: instrumentalness, dtype: float64
```



```
target
0    42.485120
1    39.554173
Name: chorus_hit, dtype: float64
```

FUTURE IMPROVEMENTS

While this tool is quite useful, its full functionality hasn't come around which is the reason it hasn't already been deployed. There are 4 main improvements for the future at this point:

1. Improving the UI of the tool. The current UI has no visual appeal and would not encourage users to try it out and see the results for themselves. Visual appeal is a very important aspect of a product and due to lack of time, there was no opportunity to improve that yet.

2. Increasing the domain of the dataset to include other markets as well. Right now, this tool works well only on songs that have the US as a primary market. This is very limiting for the possibilities that the tool can achieve, and more data has to be put into the model to train for songs from other markets as well.

3. Automating the OAuth token retrieval. Right now, I can use this as I can manually generate a new OAuth token when I want to use the tool. However, once it is deployed, this will no longer be possible and hence, we need to find a way to automate the process entirely.

4. Giving more information about the decision to the user. At this point, all the tool does is say whether the song has the potential to be a hit or not. However, future versions will also include a mini dashboard that will help the user understand which attributes of the song have resulted in the decision and some general statistics about the track.

CONCLUSION

With the growing landscape in music today, people have so many choices regarding what they listen to. This makes it incredibly hard for new artists to break into the field and create an impact. With the Spotify Hit Detector, artists know exactly where they stand in terms of their ability to make good music, as well as help them focus their attention on the songs that, statistically, have a much higher potential of becoming popular. Also, it provides record-label companies with a more objective method of choosing talent and new artists to represent. This will give talented artists the chance to create better music and overall, help the increased quality of music in the long run.