MSBA 212
Fall 2020

Group #9
Chengwu Weng
Mira Daya
Misha Khan
Yue Fang

Table of Contents

I. Overview

Spotify is a Swedish based company that was founded by Daniel Ek and Martin Lorentzon in April 2006 and launched in October 2008. Spotify provides a streaming service for digital music, podcast, and video. During 2006, Spotify was created in a response to the "growing piracy problem the music industry was facing… [thus costing] the music industry millions each year because now [users] are not paying for songs." [1] Founder Daniel Ek stated, "The only way to solve the problem was to create a service that was better than piracy and at the same time compensates the music industry." [2]

A. Business Implications

Spotify generates 91% of its revenue from user subscriptions and 9% from advertisements while also keeping 30% of the remaining sum of licensing and contracts from artists. [3] Spotify has various subscription tiers: free, premium, student, and family. Users can stream for free with shuffle only and ad-supported playbacks. Users can also opt for a premium subscription and pay $9.99 per month, $4.99 per month if they are a student, and $14.99 per month for the family plan of six users. Paid users can listen to music offline and ad free. For advertisements, Spotify will play fifteen second to a minute ads between every 15 minutes of streaming for free users as well as display banners or sponsored playlists on their homepage.

Some favorable features among users are the personable playlists and exploration of new artists. Because of Spotify's unique personalization algorithm, this enhances a user's experience, especially among millennials within ages 25-34, to help discover new music within their desired taste. Spotify's machinery allows small artists to gain massive exposure in a way where users may not search for organically. This platform makes it easier for rising musicians to become popular at a faster rate compared to other streaming services.

Spotify uniquely stands out from its competitors for it allows users to create playlists and share music with one another in an intimate and social aspect all while maintaining a youthful, appealing aesthetic. Spotify dominates the market share with 35% with its main rival, Apple

---

[1] How Spotify Came to Be Worth Billions." BBC News, BBC, 1 Mar. 2018, www.bbc.com/news/newsbeat-43240886
[2] Ibid
[3] Desai, Mittul. "How Spotify Makes Money - Business Model." *Medium*, Uncovering Music Tech, 29 Apr. 2019, medium.com/dissecting-music-tech/how-spotify-makes-money-business-model-ca0a71a19163

Music, following with only 19%. The company now aims to "move into a territory occupied by the likes of Billboard… Rolling Stone… and the Official Charts Company in the UK" [4] and compete with larger rankings but is limited by only their own data collection.

For our project, we created a model that predicts whether a song that was released in 2020 is a hit song by creating a threshold for the popularity score. Songs above the threshold will be considered popular while songs below are not. Using machine learning applications, we can predict if a song will be popular through observing its composition. By examining the similar compositions of a song, we can see a pattern in songs that are viral or not. Considering such attributes will help improve streaming companies, like Spotify, with reducing advertising costs and expanding an artist's platform to boost revenue income.

B. Related Work

Spotify's success is largely due to how accurate their music recommender algorithm is. This personalized experience creates a strong relationship directly with their consumers. For example, Spotify curates a uniquely selected playlist every week called "Discover Weekly" for each individual user - free or premium. Spotify's generated playlists is mostly based on a listener's recently played artists. The algorithm assigns affinity scores to artists and finds other similar songs based on how central it is to a user's music taste. With the help of machine learning, recommendations are improved over time and increases user engagement.

Spotify monitors when a user selects a song and looks for other users playing similar music. Through collaborative filtering, the algorithm monitors the user's behavioral trends. Similar to Netflix's rating system, Spotify uses implicit feedback and records the number of times a user has played a particular song, saved a song, or clicked on the artist's profile. This helps Spotify curate relevant recommendations for a user.

Spotify also utilizes Natural Language Processing (NLP) to improve their algorithm. It looks at what people are saying online about certain artists and songs to identify descriptive terms and texts associated with those artists or songs. These keywords are filtered into categories "top terms" and "cultural vectors". Each term is assigned a weight to reflect the relevance of importance. The system is also able to identify new music terms and trends not only in English language but across all cultures like Latin music. Audio models are another way to analyze data

[4] Ibid

from raw tracks and categorize songs correspondingly. For example, when a new song is released, NLP may not pick up on coverage especially if social media is slow. Audio models leverage song data while collaborative filtering models are able to recommend similar songs to a user. [5]

II. Data

A. Data Acquisition

We collected our data from two sources: Kaggle.com and Billboard's API. From Kaggle, we were able to acquire attributes and use those variables to train our models (variables listed below). We also web scraped our data using Billboard's API to get the Hot 100 Songs from 1970-2020. We needed Billboard's data to discern which songs were qualified as a hit song or not.

B. Data Summary

From the Kaggle dataset, the variables have unique descriptions that are important to address. Our two most important variables to consider are "hitsong" and "popularity". "Hitsong" is a binary variable created from our Billboard dataset which we utilized during our exploratory data analysis. We also created another binary variable called "highRate", based on "popularity", which we used the models to predict on.

Variable	Description	Summary Statistics
acousticness	Measures the acousticness of a track from 0.0 to 1.0 with 1.0 being the highest acoustic	Mean: 0.296220 Standard Deviation: 0.305025
artists	Artist of the song	
danceability	Measures how suitable a track is for dancing based on tempo, rhythm stability, beat strength, & regularity from 0.0 to 1.0 with 1.0 being the most danceable	Mean: 0.566567 Standard Deviation: 0.174132
duration_ms	The duration of the track in milliseconds.	Mean: 2.435079e+05 Standard Deviation: 1.014723e+05

[5] Ipshita Sen is a UK-based entrepreneur. Originally from Dubai, Ipshita. "How AI Helps Spotify Win in the Music Streaming World." *Outside Insight*, 26 Nov. 2018, outsideinsight.com/insights/how-ai-helps-spotify-win-in-the-music-streaming-world/.

energy	Measures intensity and activity based on dynamic range, loudness, timbre, onset rate, & general entropy from 0.0 to 1.0 with 1.0 being the most energy	Mean: 0.605473 Standard Deviation: 0.240014
explicit	Measures how much profanity, inappropriate references, & other unsuitable content for children with 0.0 being non explicit and 1.0 being explicit	Mean: 0.121331 Standard Deviation: 0.326513
hitsong	Top 100 songs from Billboard dataset by year (2014-2019)	Mean: 0.000354 Standard Deviation: 0.018815
id	The Spotify ID for the track	
instrumentalness	Measures the instruments in a track from 0.0 to 1.0 with 1.0 being purely instrumental (rap or spoken word tracks are very vocal so they would be considered more towards 0.0)	Mean: 0.095581 Standard Deviation: 0.240922
key	All keys on octave encoded as values ranging from 0 to 11, starting on C as 0, C# as 1 and so on	Mean: 5.247433 Standard Deviation: 3.549025
liveness	Measures the presences of an audience in the track from 0.0 to 1.0 with 1.0 being track was performed live	Mean: 0.201672 Standard Deviation: 0.181356
loudness	Measures the overall loudness of a track in decibels from -60 db to 0 db	Mean: -9.387156 Standard Deviation: 4.898656
mode	Measures the modality (major = 1.0, minor = 0.0) of a track	Mean: 0.701041 Standard Deviation: 0.457804
name	Name of the song	
popularity	Popularity of the artist from 0 to 100 with 100 being the most popular	Mean: 45.229509 Standard Deviation: 13.599671
release_date	Date of release in mostly YYYY-MM-DD format (precision may vary)	
speechiness	Measures the presence of spoken word from 0.0 to 1.0 with 1.0 being mostly speech like podcasts, audio books, poetry, etc	Mean: 0.080092 Standard Deviation: 0.093447
tempo	Measures the beats per minute/ pacing of the track	Mean: 120.629224 Standard Deviation: 30.045454
valence	Measures musical positivessness of the track from 0.0 to 1.0 with 1.0 being most positive (happy, cheerful, euphoric) and 0.0 being more negative (sad, angry)	Mean: 0.539649 Standard Deviation: 0.258032
year	Year of release	Mean: 1994.920015 Standard Deviation: 14.679558

C. Data Preprocessing

From our data, we dropped variables “id”, “name”, and “release date”. We then scraped data from Billboard’s Hot 100 songs since 1970 in order to generate the hit songs list. We cleaned up “trackname” and “artist” in order to make a cleaner dataset. Afterwards, we added years to the Billboard’s data to examine when hit songs were released. By assigning 0’s and 1’s, we classified if a song was a hit or not depending on if it exceeds our popularity score threshold. We also joined our genre data with the main dataset in order to predict which genres are popular as well.

Before we started our analysis, we first cleaned “artists” and “track name” in order to join our Kaggle and Billboard datasets. This gives us the best match when we join datasets.

Before ->

Kaggle

	artists	name
0	['Linkin Park']	Final Masquerade
1	['Hippie Sabotage']	Ridin Solo - Njomza Remix
2	['Bleachers']	Wild Heart
3	['together PANGEA']	Sick Shit
4	['David Guetta', 'Showtek', 'VASSY']	Bad (feat. Vassy) - Radio Edit

Billboard

	trackname
0	1. 'Happy' by Pharrell Williams
1	2. 'Dark Horse' by Katy Perry Featuring Juicy J
2	3. 'All Of Me' by John Legend
3	4. 'Fancy' by Iggy Azalea Featuring Charli XCX
4	5. 'Counting Stars' by OneRepublic

After ->

	artists	name
0	linkin park	final masquerade
1	hippie sabotage	ridin solo njomza remix
2	bleachers	wild heart
3	together pangea	sick shit
4	david guetta showtek vassy	bad feat vassy radio edit

	trackname	artist
0	happy	pharrell williams
1	dark horse	katy perry featuring juicy j
2	all of me	john legend
3	fancy	iggy azalea featuring charli xcx
4	counting stars	onerepublic

Afterwards, we joined the Kaggle and Billboard Hot 100 songs datasets together by track name. This helped us determine if a song is a hit or not by creating the binary variable “hitson” using Billboard’s data for Spotify. For example, if we look at Ariana Grande’s song “in my head”, hitsong is equal to 1 meaning the song was listed on the Billboard Hot 100 songs.

	genre	track_id	acousticness	artists	danceability	duration_ms	energy	explicit	instrumentalness	name	popularity	hitsong
817	Dance	4T652DlATVHe0jdLKaN3Bw	0.17300	Ariana Grande	0.662	222947.0	0.600	1.0	0.000137	in my head	72.0	1
818	Dance	6QfS2wq5sSC1xAJCQsTSlj	0.41600	Lady Gaga Bradley Cooper	0.575	217213.0	0.330	0.0	0.000000	Shallow - Radio Edit	77.0	0
819	Dance	27356GVuMPFWiJSZCragoM	0.08440	Ariana Grande	0.671	140693.0	0.714	1.0	0.000001	make up	68.0	0
820	Dance	4VUwkH455At9kENOfzTqmF	0.34600	Bazzi Camila Cabello	0.638	180000.0	0.717	0.0	0.000000	Beautiful (feat. Camila Cabello)	79.0	0
821	Dance	2qT1uLXPVPzGgFOx4jtEuo	0.04000	Ariana Grande	0.699	205920.0	0.713	0.0	0.000003	no tears left to cry	82.0	1

Lastly, we created a cutoff value to determine how many songs are popular or not. Our prediction variable is called “highRate” which determines if a song is popular or not. We set our own threshold for popularity using the median 62 because the median is less susceptible to outliers than the mean. We decided on this based on our boxplot. After setting the threshold value, songs that were above the popularity score 62 were considered popular while below 62 were considered not popular. From this, we considered 47.78% of songs to be popular while 52.22% of songs not popular.

Positive Instance Percentage	0.47783489
Negative Instance Percentage	0.52216510

$$highRate = \begin{matrix} 1, & if\ popularity > 62 \\ 0, & otherwise \end{matrix}$$

III. Exploratory Analysis

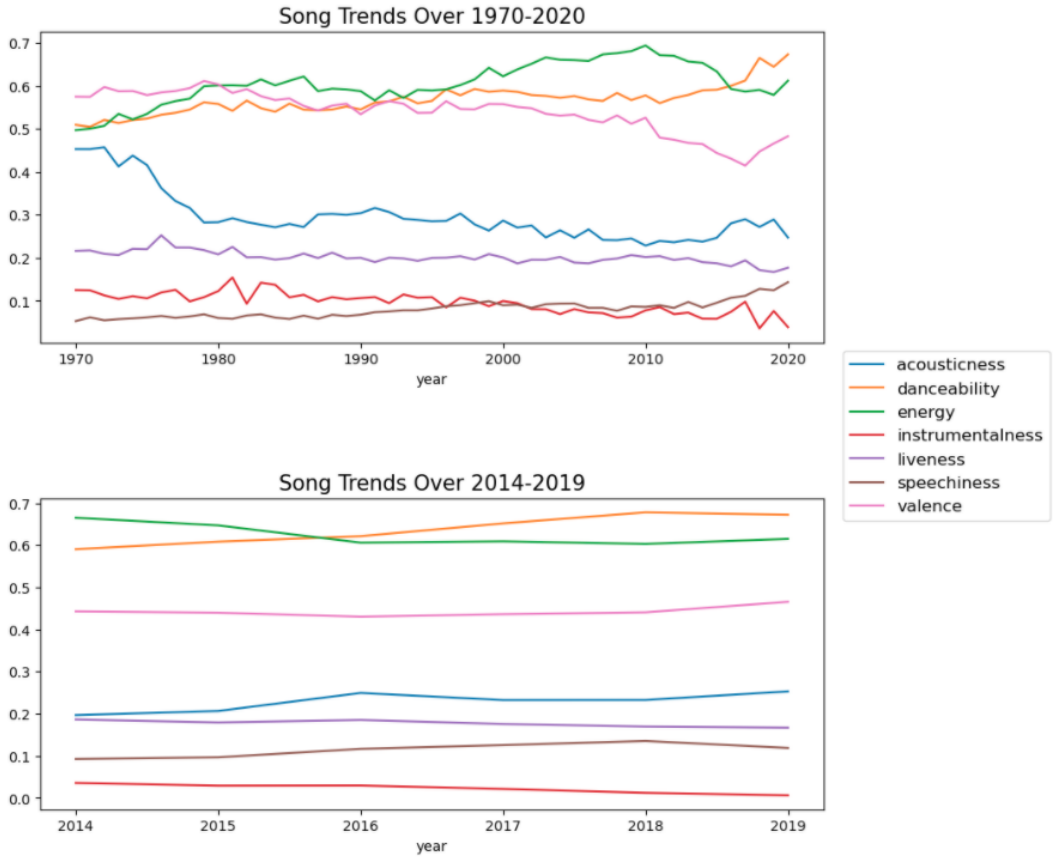


Figure 1: (a) Song Attributes Over Time (b) Song Attributes from 2014-2019

Figure 1 (a) above shows the trends overtime for song attributes. For the most part, the trends are constant overtime. Note that the two attributes with the most noticeable changes are *acoutstiness* and *energy*. Acousticness starts to decrease from the 70's till the early 80's then is constant for the rest of the timeline. This can be due to the movement in making electronically derived music which results in a loss of acoustic properties to songs. Additionally, energy starts to spike around the early 2000's to about 2015 and then dips again. Figure 1 (b) is a focused on years 2014 to 2019, which is the time frame we are basing our Machine Learning models on.

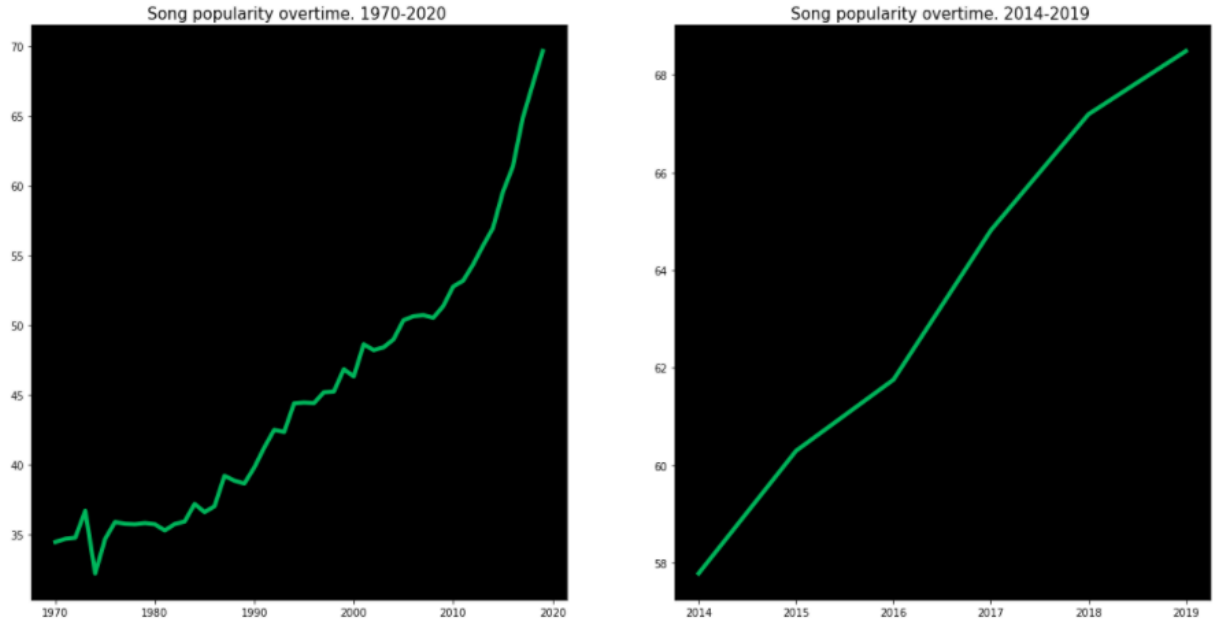


Figure 2: (a) Song popularity score 1970 - 2019,
(b) Song popularity score 2014 - 2019

Figure 2 (a) above shows the trend in popularity core from 1970 to 2019, 2020 is not included as the data was not complete for the year. Generally, popularity is increasing with song release year. Figure 2 (b) is for the timeframe 2014-2019. Spotify assigns popularity of a track based on (1) the total number of plays compared to other tracks and (2) how recent those plays are. Over time, if a song isn't listened to very much then the popularity score will go down.

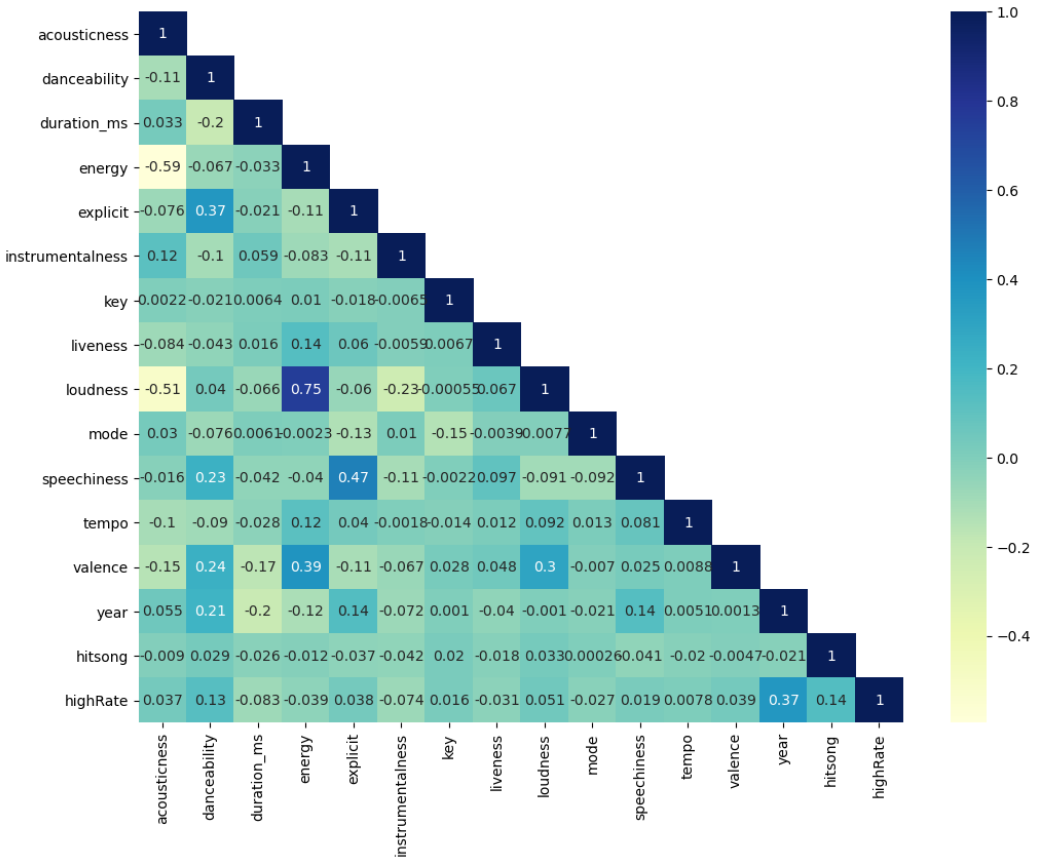


Figure 3: Correlation Matrix Between Numerical Variables

We used Figure 3 to address concerns for multicollinearity. There exists multicollinearity with variables such as Loudness and Energy, Speechiness and Explicit, Year and highRate.

```
The most linear correlated features to POPULARITY are:
popularity          -->  0.80 (abs)
year_2018           -->  0.26 (abs)
year_2014           -->  0.23 (abs)
pop                 -->  0.18 (abs)
hitsong             -->  0.14 (abs)
danceability        -->  0.13 (abs)
year_2015           -->  0.13 (abs)
year_2019           -->  0.12 (abs)
year_2017           -->  0.12 (abs)
```

Figure 4: Top linearly correlated features to *Popularity*

Figure 4 shows us which variables are most correlated with our popularity score variable. Notice that the variable *pop*, which represents if a song's genre is pop or not, is somewhat

correlated to a song's popularity score. In Figure 6a and 6b we will explore more about genres present in this dataset and how those relate to if a song is considered to be popular.

genre	Not Popular	Popular	Percentage Change %
acousticness	0.215912	0.234719	8.710782
danceability	0.614144	0.651891	6.146146
duration_ms	226500.257007	217887.326921	3.802614
energy	0.632306	0.617821	2.290733
explicit	0.405953	0.443108	9.152596
instrumentalness	0.032957	0.016335	50.435634
key	5.136553	5.252120	2.249891
liveness	0.182914	0.174426	4.640387
loudness	-6.931887	-6.651304	4.047705
mode	0.637711	0.611442	4.119385
speechiness	0.111864	0.116243	3.914114
tempo	120.794659	121.251517	0.378210
valence	0.430704	0.447575	3.916874
hitsong	0.056289	0.140615	149.808822
duration_min	3.775004	3.631455	3.802614

Figure 5

Figure 5 was used to get a baseline understanding of how different variables affect our binary variable highRate, which is used to classify whether a song is considered popular or not. We use the difference between means for popular and not popular songs to see the variables effect on popularity. Notice that the highest percent changes are for the following attributes: (1) Hit Song, (2) Instrumentalness, (3) Explicit, (4) Acousticness, (5) Danceability. These attributes contribute to what makes a song popular or not.

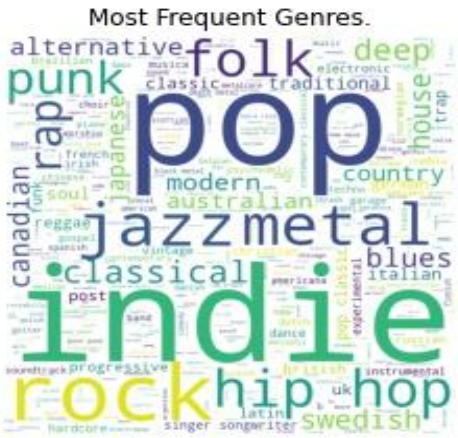


Figure 6 (a)

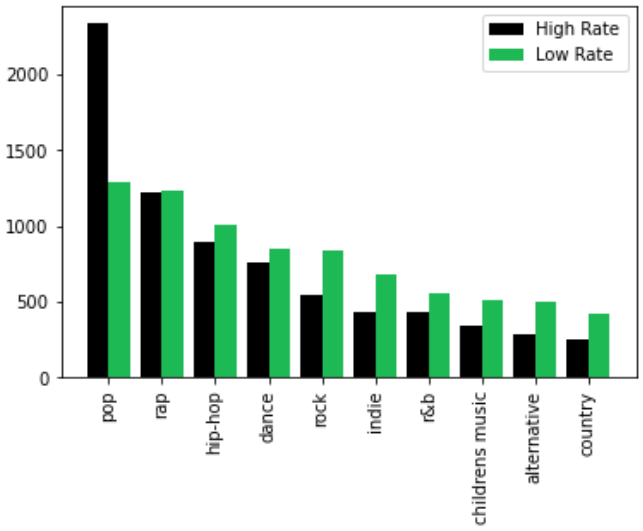


Figure 6 (b)

Finally, Figures 6 (a) and 6 (b) dive into the types of genres present in our dataset. Figure 6 (a) is a word cloud that represents the frequency of genres in the original dataset. Notice that the most frequent types of genres are Pop, Indie, Rock, Hip Hop, Jazz, Metal, etc.

Figure 6 (b) shows us the frequency of genres by whether or not a song is considered popular, highRate = 1, or not popular, highRate = 2. The patterns for popular and not popular are similar. The highest frequency of popular songs are pop, then rap, hip hop and so on. The same goes for not popular songs.

Additional word clouds along with brief descriptions are included in the Appendix.

IV. Methods

Since our goal is to classify if a song is popular or not on spotify, we used the following models:

1. Logistic Regression as benchmark
2. Polynomial Features & Logistic Regression
3. Polynomial Features & Random Forest

By using a 10-fold cross validation method, our dataset will be divided into 10 subgroups. Each subgroup will be reserved as a test set, and the remaining part of the dataset will be used to train the model. As for evaluating the models, we used accuracy rate and confusion

matrix to test the performance of our model. All the models were built using a Python package called Sklearn.

A. Logistic Regression

The first model we explored is Logistic Regression that is also our benchmark model. Since we are classifying spotify hit songs into two categories, we don't need a complex algorithm, like SVC that can be used to classify 50 categories. In this context, Logistic Regression meets our needs and is suitable for solving our identified problems.

Table 1 Evaluation Metrics of Logistic Regression

Average accuracy	0.6790
Average precision	0.6939
Average recall	0.5875

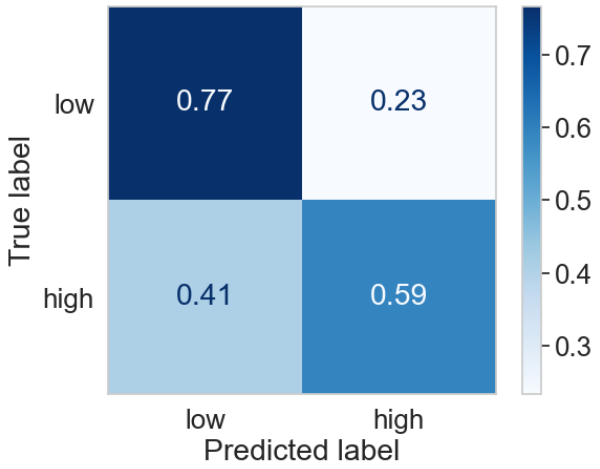


Figure 7 Confusion Matrix of Logistics Regression

The accuracy for Logistic Regression is 67.90% for our simple model as shown in Table 1. The confusion matrix in Figure 7 shows us that 77% of not-popular songs were predicted correctly, and 59% of popular songs were predicted correctly. Alternatively, 23% of not-popular songs were predicted as popular, and 41% of popular songs were predicted as unpopular.

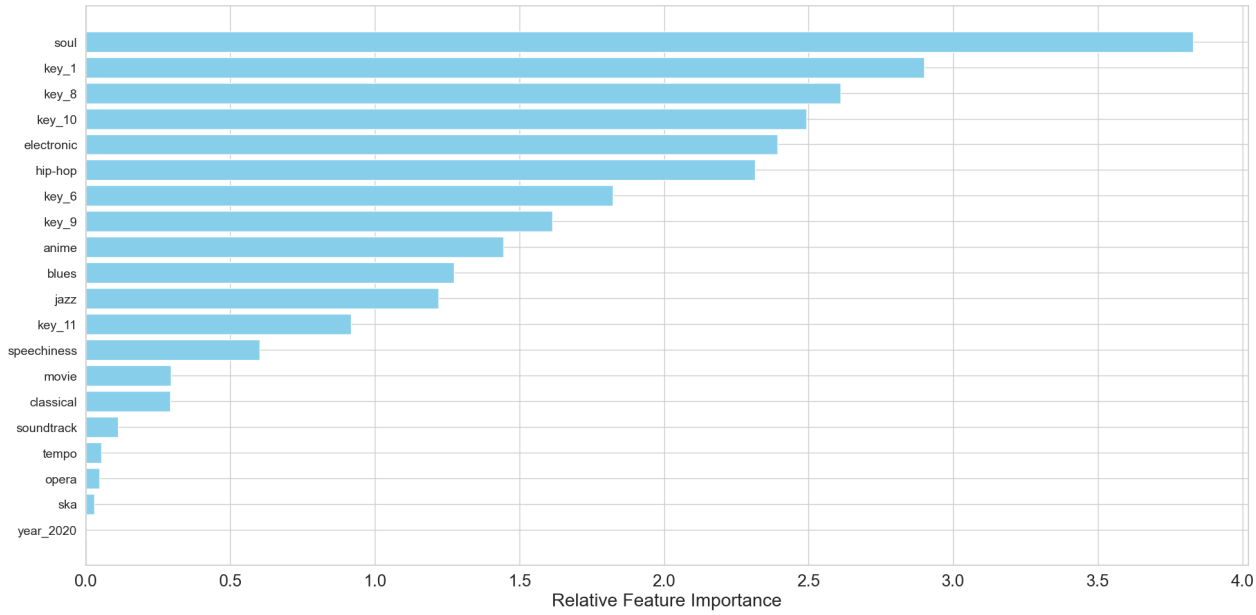


Figure 8 Relative feature importance of Logistics Regression

The relative feature importance of the Logistic Regression model in Figure 8 shows this model was built by half of genres and half of music features.

B. Polynomial Features & Logistic Regression

In the exploratory analysis, we have examined the linear relationship between each of the variables. The most linearly correlated feature with popularity is year, shown in Figure 4. The correlation coefficient between year and popularity is 0.26. Since there are no other strong linear relationships between dependent and independent variables, we enhanced our model by setting polynomial degrees of 2 and using feature selection by ranking absolute values of the magnitude of coefficients to fit a better model.

Applying a polynomial feature transformation can help expose important relationships between input variables and the target variable that may not necessarily be linear. For example, if an input sample is two dimensional and of the form [a, b], the degree-2 polynomial features are [1, a, b, a^2, ab, b^2].

Table 2 Evaluation Metrics of Polynomial Features & Logistic Regression

Average accuracy	0.6826
Average precision	0.6879
Average recall	0.6147

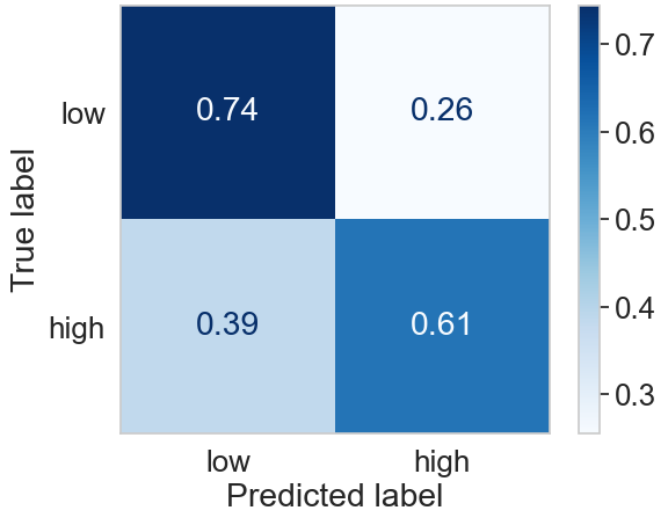


Figure 9 Confusion Matrix of Polynomial & Logistics Regression

The accuracy for Polynomial & Logistic Regression is 68.26% for our polynomial & Logistics Regression model as shown in Table 2. The confusion matrix in Figure 8 shows us that 74% of not-popular songs were predicted correctly, and 61% of popular songs were predicted correctly. Alternatively, 26% of not-popular songs were predicted as popular, and 39% of popular songs were predicted as unpopular.

Overall, this model has a slight improvement for the accuracy score. It has a better performance of predicting popular songs correctly, but a worse performance of predicting non-popular songs compared to the benchmark.

C. Polynomial Features & Random Forest

Random Forest is an improved model of Decision Tree. An ensemble of Decision Trees algorithms can deal with overfitting problems and enforce diversity. The Random Forest model can train the model on different samples of the data as well as use random subsets of features to create individual decision trees. The Random Forest model aggregates individual decision trees to limit overfitting as well as minimize error due to bias and therefore yields useful results.

i. Optimal Depth of Decision Tree

First, we wanted to figure out the optimal depth of the decision tree. According to Figure X, the trends of accuracy rate continues growing from depth of 1 to 30. Since our intention is to receive a higher accuracy rate, we left nodes expanded until all leaves are pure.

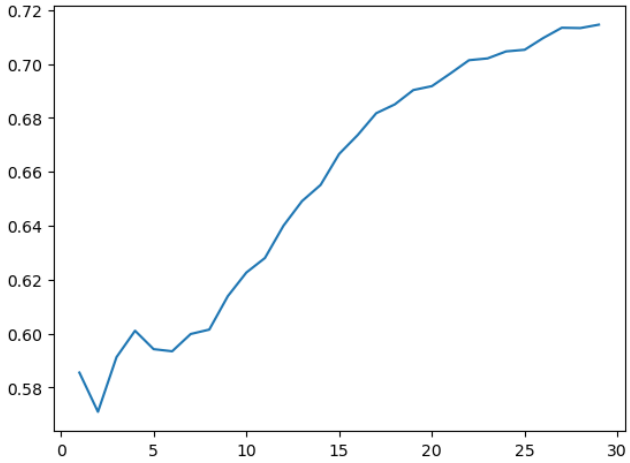


Figure 10 Accuracy Score with Different Depth

ii. Optimal Number of Trees

We created a line graph to show how the accuracy rate is affected by the number of trees. We noticed that the accuracy rate stops increasing and fluctuates starting at 20. We decided to pick 20 as the optimal number of trees to be created by the model.

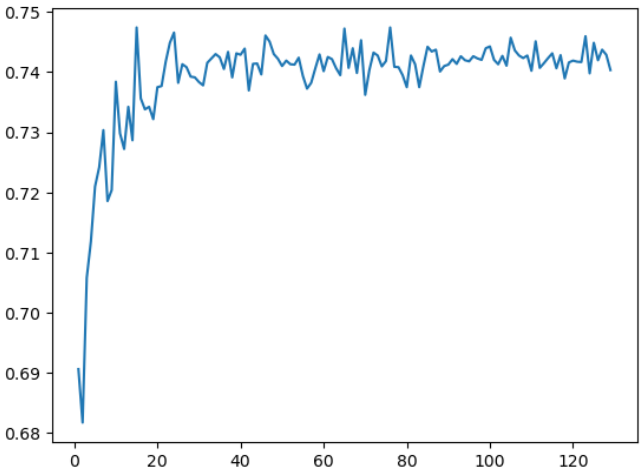


Figure 11 Accuracy Score with Different Estimators

Table 3 Evaluation metrics of Polynomial features & Random forest

Average accuracy	0.7374
Average precision	0.7633
Average recall	0.7359

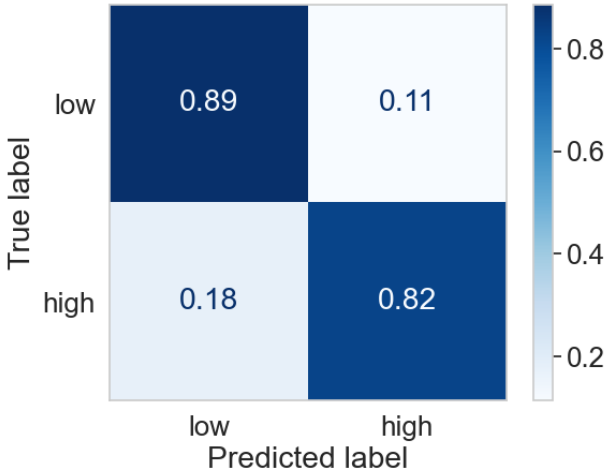


Figure 12 Confusion Matrix of Polynomial & Random Forest

The accuracy for Polynomial & Random Forest is 73.74% for our polynomial & Random Forest model as shown in Table 3. The confusion matrix in Figure 8 shows us that 89% of not-popular songs were predicted correctly, and 82% of popular songs were predicted correctly. Alternatively, 11% of not-popular songs were predicted as popular, and 18% of popular songs were predicted as unpopular.

Compared to the benchmark, the accuracy rate increased from 67.90% to 73.74%. The rates of both predicting popular songs and non-popular songs correctly are higher than the benchmark. As a result, this model has the best performance and was used to predict 2020 popular songs on .

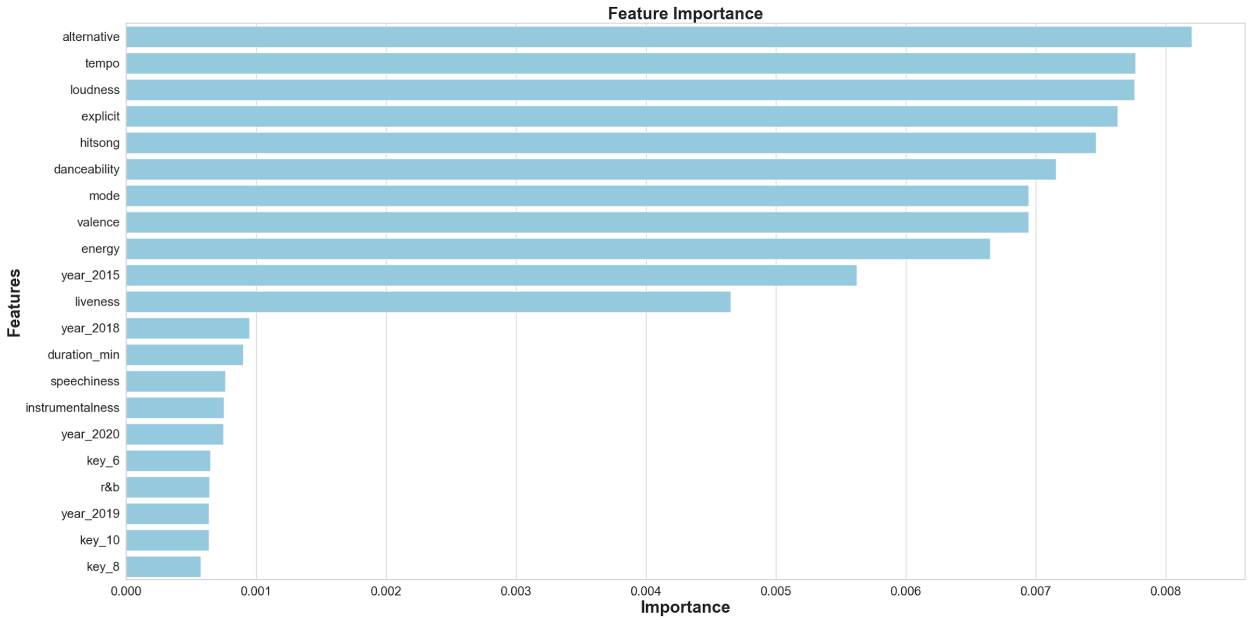


Figure 13 Relative Feature Importance of Logistics Regression

The relative feature importance of the polynomial & Random Forest model in Figure 13 shows this model was built by almost all music features and only one music genre shown in the graph. Compared to the Logistic Regression, the model built with more music features predicts popular songs more accurately.

V. Findings

Logistic Regression	Feature Cross	Random Forest
67.9044%	68.2613%	73.7374%

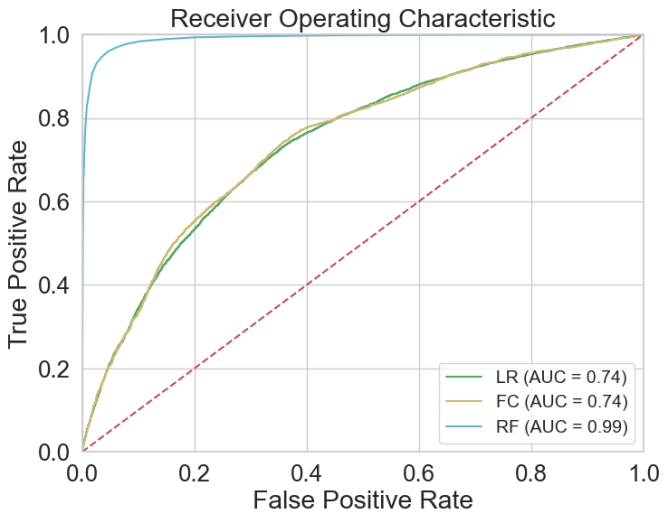


Figure X

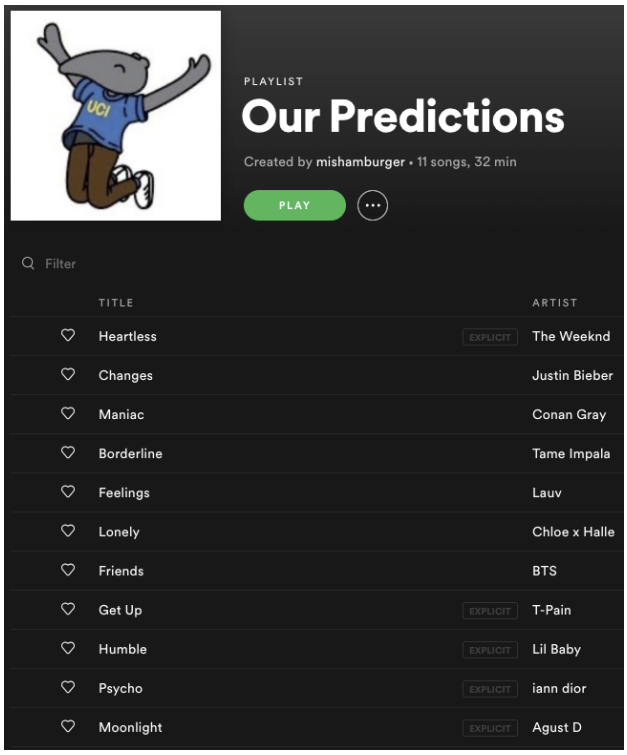
Figure X represents the ROC curve that shows the True Positive Rate against the False Positive Rate for all three of our models. We assess this curve by looking at which curve is the highest to the top left corner, indicating a better performance, as well as looking at the area under the curve, AUC, which reinforces the predictive accuracy of the models.

Random Forest has the highest accuracy at 73.73% as well as the highest AUC 0.99. This gives us confidence that Random Forest is the best model for our dataset. Both Logistic Regression with or without polynomial features come with 0.74 AUC and accuracy rates are very similar with 67.90% and 68.26% respectively.

VI. Conclusion

Total number of 2020 songs are 1104 and we predicted 501 songs as popular songs. 75 out of 121 songs that are in the billboard top 100 in 2020 were predicted as popular songs on Spotify. There are some similarities between the billboard top 100 songs and popular songs on spotify.

Billboard Hot 100 Songs
18. Kings & Queens - Ava Max
20. More Than My Hometown - Morgan Wallen
24. Levitating - Dua Lipa
39. WHATS POPPIN - Jack Harlow
50. Love You Like I Used To Russell Dickerson
73. Took Her to the O - King Von
79. Good Time - Niko Moon
89. Martin & Gina - Polo G



Since our Random Forest model had the highest accuracy, we used this model to predict 2020 songs. On the left table, these are the songs that our Random Forest model predicted accurately for 2020. These songs were matched with the Billboard Hot 100 Songs. However, on the right are songs that our model predicted to be hits but were not mentioned on the Billboard Hot 100 Songs. From this, we can see that our model still needs to improve its accuracy.

VII. Next Steps?

Some additional due diligence our group would like to work on after this project is focusing on improving the models. Our accuracies were good but we believe that the models can

be better. Once we improve our models, we can also explore which specific attributes are the most important for predicting popularity.

Appendix

Popular Artists Word Clouds

Figure A

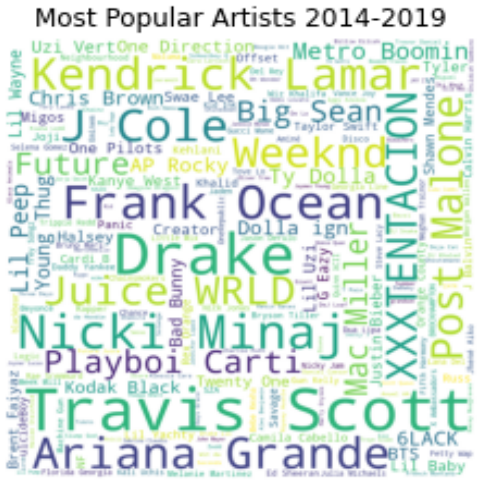


Figure B

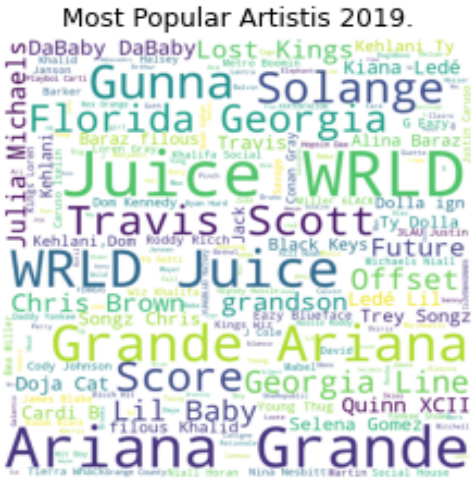


Figure C

Figure A shows us the artists with the most popular songs in the original data set. Artists like Drake, Bad Bunny, Travis Scott, and BTS held higher numbers of popular songs throughout the entire data set.

Figure B shows most popular artists from 2014-2019, the time frame our models are built on. There is a more even distribution between artists than in Figure A. Artists like Drake, Travis Scott are still in higher frequency but now we also see Niki Minaj, Ariana Grande, and Kendrick Lamar.

Figure C shows most popular artists in 2019 by counting how many popular songs artists had.