ASHESI UNIVERSITY

USING MACHINE LEARNING TO PREDICT THE POPULARITY OF GHANAIAN HIPLIFE SONGS

APPLIED PROJECT

B. Sc. Management Information Systems

Sebastian Dakey

2020

ASHESI UNIVERSITY

USING MACHINE LEARNING TO PREDICT THE POPULARITY OF GHANAIAN HIPLIFE SONGS

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Management Information Systems.

Sebastian Dakey

2020

DECLARATION

I hereby declare that this [capstone type] is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this [capstone type] were supervised in accordance with the guidelines on supervision of [capstone type] laid down by Ashesi University.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgements

I will like to thank the Almighty God for his grace and mercies and seeing me through this capstone journey. I will like to also extend my heartful gratitude to my supervisor Stephane Nwolley whose guidance and advice have helped me successfully complete this capstone. Finally, I will like to thank Gordon Silvera who is a Data Scientist at Spotify for pointing me to various resources that helped me in pulling my capstone together.

Abstract

Music streaming is a popular way of consuming music today. From Spotify, Apple Music, Deezer, among others to AudioMack, these streaming platforms are an excellent way for artistes to market and monetize their music. Moreover, these platforms present opportunities for artistes, Record Labels, Artist Managers, and other stakeholders in the music industry to gain insights into how listeners consume their music. Also, the data generated from these streaming platforms are used for various research in the field of Music Information Retrieval (MIR). The examples include Music Recommendation, Music Generation, and Hit Song Prediction. This study investigates how audio features can help in predicting the popularity of a song. I scrapped 2000 Ghanaian hiplife songs using the Spotify Web API, which offers developer access to its Music Catalogue and extracted relevant audio features for each of the songs. Using Logistic Regression, K Nearest Neighbors, Random Forest, and Naïve Bayes Classification algorithms, I built a model to predict the popularity of a song. The result showed that although audio features offer a sufficiently useful way to predict the popularity of a song, there might be other features other than audio features that can explain the popularity of the song.

Table of Contents

Chapter 1: Introduction

Hiplife is a common musical genre in the Ghanaian Music Industry today. It is a genre of music most preferred, especially among the youth. The term hiplife is a blend of American hip hop music and Ghanaian highlife (a musical genre that blends African rhythms with that of Euro-American and African diaspora). Hiplife, therefore, simultaneously blends elements of both genres to produce a distinct genre [8]. Until the consumption of music went online, local artists sold their music on cassettes and subsequently compact discs. They leveraged on many cassettes/compact discs distribution centers to market their music. How far an artist's music went depended on how good their distribution strategy was (and its resulting sales) and if they got much airplay on radio and television.

The emergence of the internet and streaming platforms made music more accessible to everyone. Anyone with internet access can download and consume music. Music Streaming platforms made it even possible for an artist's music to reach a wider listening audience through the music recommendation capabilities of these platforms. On Spotify, the larger your audience, and the frequency of streams, the higher the popularity score of the song. Hence, the higher the popularity score, the more likely the song is a hit. Sarkodie is one of the finest musicians on the African continent. With over ten years of experience, his level of craftsmanship and superior understanding of the 'game' and his market is unparalleled. He is the Ghanaian artist with the highest following on Spotify. On July 25, 2019, he released a single off his 'Black Love' Album titled 'Lucky,' which is a big hit in Ghana. It has a popularity score of 38/100 on Spotify. Following this, he released two more off the 'Black Love' album titled 'DoYou' and 'Saara,' which were hits as well and has a popularity score of 36/100 and 35/100, respectively. King Promise is also another prominent musician in the country. He is by far the most popular Ghanaian artist on Spotify. With

his songs “Commando” and “Call waiting” scoring 59/100 and 55/100 as popularity scores respectively, King Promise is one of the biggest hitmakers in Ghana. It is worth noting that on Spotify, popularity scores are measure across markets (countries). Ghana unfortunately is not a market yet on Spotify. Hence, the popularity of Ghanaian songs is in relation to how they are doing well in other markets. The assumption is that if Ghanaian songs score a popularity score of 10 and above, then they are already popular in Ghana. Like Sarkodie and King Promise among a few other hitmakers in Ghana, what makes their songs unique? What exactly is there about these songs that predisposes them to such greatness? Are there specific features of a song that promises a hit?

Music Information Retrieval is a research field that seeks to extract useful information from music. Some of the many applications of this research field include hit song science, music recommendation systems, music generation, among others. Hit song science is a branch of music information retrieval that seeks to predict the success of a song based on some features. Access to the internet and social media has made many artists leverage their power to promote their music. However, what decisions go into promoting one song over the other? What do record labels look out for when producing new songs? These are some of the questions that hit song science seek to address. Determining hit songs has excellent commercial potential. First, it will help record labels and artistes to identify which songs to promote based on the likelihood of it becoming a hit. It will also help the underground artist who does not have enough capital to promote songs to focus on the essentials features that make a song a hit so that they focus on minimal marketing efforts to make their songs commercial. In this paper, I will use machine learning algorithms to build models based on the intrinsic audio features of Ghanaian popular music.

Chapter 2: Related Works

In a research conducted by Yekyung Kim, Bongwon Suh and Kyogu Lee [5], the researchers mined the listeners' behaviors on twitter to predict hit songs. Microblogs, of which Twitter is an example, are great platforms for people to share their thoughts and opinions on various subject matters with their audience. On Twitter, users use hashtags to mark specific topics and group related tweets together for various analyses. In this study, the researchers collected music listening behaviors on Twitter using music-related hashtags (#nowplaying) and built a predictive model to forecast hit songs and Billboard rankings. The researchers crawled tweets related to the music listening behavior using the Twitter Streaming search API from November 21, 2013, to January 23, 2014, which generated about 31.6 million tweets. These tweets were those that contained the #nowplaying, #itunes, and #np hashtags. Furthermore, they collected data from the Billboard Hot 100 chart for ten weeks. The Billboard Hot 100 chart is a chart by the Billboard Magazine, which shows the 100 most popular songs in the USA. The chart is issued every Thursday. The data from Billboard were collected to evaluate the prediction performance of their model. For their preprocessing, they calculated the popularity of song by counting plays and the number of tweets associated with a song. They also counted artiste popularity as the number of tweets associated with the artistes.

To obtain a quantitative evaluation, they used correlation management, regression model, and classification methods. Using Pearson's correlation coefficient under correlation management, they calculated the correlation between the song's play-count, artist popularity, and the number of weeks on the Billboard chart. In ranking the music prediction, they built three regression models to evaluate the performance of the predictive model, namely linear regression, quadratic linear regression, and support vector regression. In conclusion, the support vector regression showed the

highest squared correlation coefficient (0.75). The classification model generated 83.9% accuracy, 83% precision, and 85.3% recall. These results show that social media is a great way to understand the music consumption of users and subsequently use that information to predict the popularity of a song.

This research presented another perspective to analyzing the popularity of a song as compared with using audio features. It offers great insights into how people use social media with respect to music consumption. Through this research, one can understand how such social media usage can offer insights into investigating the popularity of a song. The only downside is that it used only twitter as a platform to analyze music consumption. Whiles this offers great insight, it would be interesting to know how the results will look like on Facebook, Instagram and other social media platforms bigger than twitter.

In a research conducted by Minna Reiman and Philippa Ornell [10], the authors analyzed the audio features of songs and built a predictive model with various machine learning algorithms to predict the popularity of a song. They extracted hit songs from Billboard Hot 100 between the years 2016 and 2018. They generated 609 non-hit songs from 13 different genres. They used Spotify's web API to generate audio features of all the songs in their datasets. Some of the audio features included valence, danceability, energy, tempo, acousticness, among others. Using Logistic Regression, K-Nearest Neighbors, Gaussian Naïve Bayes and Support Vector Machines, they built a predictive model to classify hit songs from non-hit songs. The authors used 10-fold cross-validation to estimate the accuracy of all the models. To measure the performance of the predictive models, they used the confusion matrix. The confusion matrix is a matrix that summarizes the performance of a model on test data [13]. For their results, they had the Gaussian Naïve Bayes performing best with 60.17% accuracy. From this study, the authors concluded that using their

dataset, it is not possible to completely forecast whether or not a song is a hit or not using audio features. This methodology presented another perspective to analyzing the popularity of a song. Whiles their source of data is a more objective way of getting hit songs and non-hit songs, such data source is not present in the Ghanaian market. Hence a researcher may have to device their own approaches to sourcing hit and non-hit songs.

In a research conducted by Leo Breiman, Michael Last and John Rice [3], the authors predicted hit songs based on early adopter data in addition to audio features. In [11], Rogers describes early adopters as those with the highest level of opinion leadership among other adopter categories. These are young people who are more 'socially forward' than their counterparts. They are usually described as people with the sixth sense for hit songs; meaning they can listen to hit songs even before it climbs the record charts. Furthermore, they collected a list of hit songs from the Belgian website "The Ultratop 50". The top 20 songs were identified as hits. For non-hit songs, they used the "The Bubbling Under Chart," which also comes from the "The Ultratop 50" and contains the list of 20 upcoming songs as identified by music experts. Any song that eventually made the charts (top 20) was removed. The data was gathered from July 2, 2011, to November 16, 2013. Using the Last.FM API, listening behaviors of three Last.FM groups were collected. After a series of data wrangling, cleaning, and feature engineering, a total of 140 audio features were extracted using the EchoNest API. The researchers then used classification algorithms (Logistic Regression, Support Vector Machines, Naïve Bayes to build audio feature models to predict hit songs. This generated an accuracy of 0.64 for Logistic Regression, which was the highest among the rest of the algorithms. The model built using the listening behaviors of early adopters (listeners in Last.FM groups) generated an accuracy of 0.79 for Logistic Regression, which was also the highest among the rest. The authors concluded that listening behaviors on Last.FM can help predict

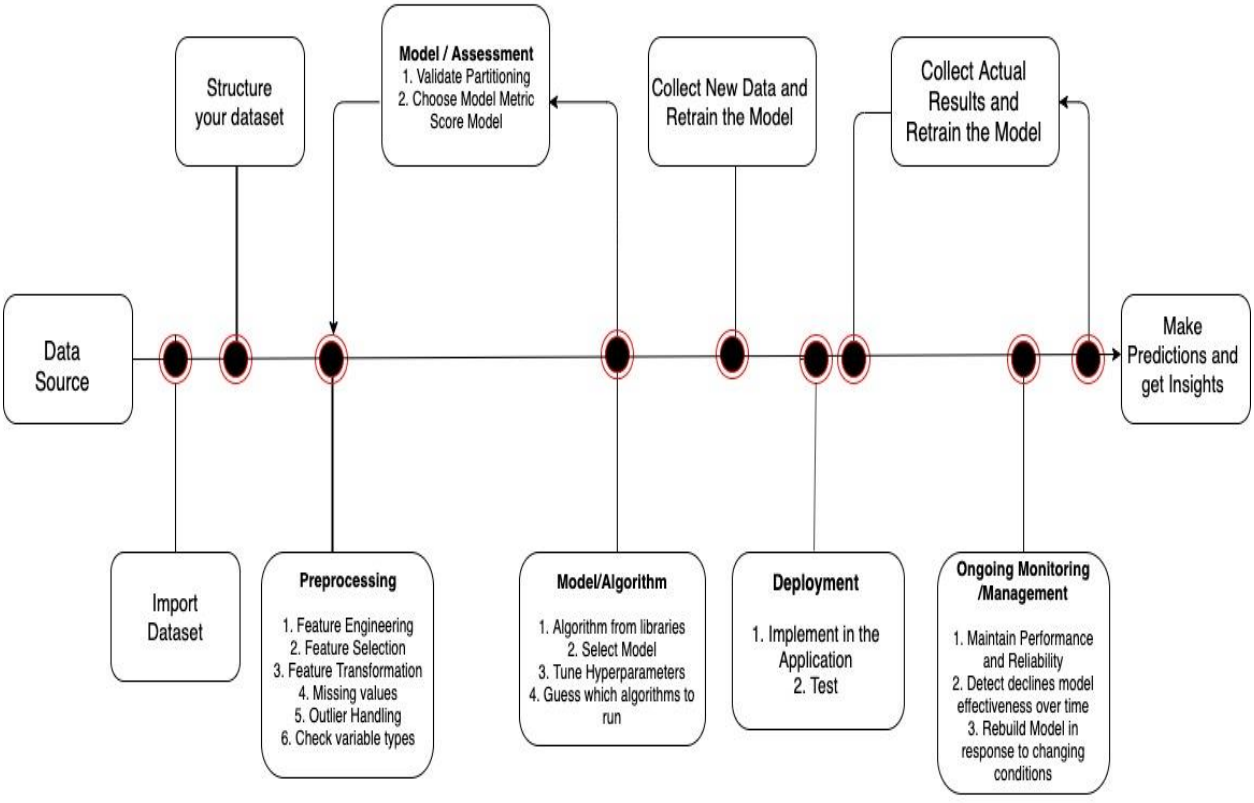
the top 20 hits. These researchers also presented another way of analyzing the popularity of a song. A great advantage of this research is the addition of early adopter data. This addition can fairly increase the accuracy of the data. The only downside to this research is that early adopters data may not be present in all markets thereby making the results of this research specific to the market within which this research was conducted.

Chapter 3: Architecture and Implementation

Overview

This chapter explains the architecture and implementation strategies employed in this project. It gives an explanation and justification for various technologies and libraries used. Lastly, it outlines the thought processes from data gathering to modeling.

Architecture



The above diagram represents the machine learning architecture of this project. It defines the major layers in the machine learning cycle and all the steps involved in transforming raw data into training data set for modeling and decision making.

Implementation tools and technologies

This section describes the various languages, technologies, and libraries used in implementing the project.

Programming Language

- **Python:** It is an interpreted, general-purpose, and high-level programming language. It is very concise, clear, and easy to read and write. The core reason for using python is the availability of third-party libraries written in python that allows one to perform more specialized and advanced tasks with python [12]. For example, third party libraries like sklean, Keras, TensorFlow, among others, enables one to perform beginner to advanced machine learning tasks to solve a wide array of problems. In this project, third party libraries written in python for data wrangling, visualizations, and machine learning modeling will be used.

Data Analytics, Machine Learning, and other Libraries

- **Pandas:** It is a python library that contains data structures and tools for working with structured data across fields like finance, and statistics, among others. The library provides methods for performing various data wrangling techniques and analyses. The use of pandas serves as the foundation for other tasks such as modeling [6].
- **Matplotlib:** It is a 2D graphics library used to create publication-quality graphs and plots in interactive environments across platforms. It can be used in Jupyter notebooks, IPython shells, and python scripts, among others. With just a few lines of code, you can generate histograms, bar charts, and scatterplots, among others [4].

- **Seaborn:** It is a python data visualization library based on matplotlib. It is used to generate high-level, more beautiful, and informative statistical graphs such as time series plot, scatter plots, and facet plots.
- **Sklearn:** It is an open-source python library for machine learning tasks. It integrates various state of the art machine learning algorithms for medium scale supervised and unsupervised machine learning problems [9]. It is the choice of machine learning library because it is easy to use, performs very well, and has extensive documentation. It contains built-in functionalities such as preprocessing, metric, train_test_split for various machine learning tasks.
- **Numpy:** It is a python library that serves as the foundational package for scientific computing. The library provides an efficient way of implementing numerical computations in a high-level language [14].
- **Beautiful Soup:** It is a python library that is used to scrape information from websites. It is used for navigating, searching and modifying a parse tree [7]. It was used to scrape music data from ghanamotion.com
- **Requests:** It is an Apache2 licensed HTTP library written in python. It is used for sending HTTP requests [2]. It is usually used with Beautiful Soup during web scrapping tasks.
- **Re:** This is a python module that provides regular expression matching operations. It was used with Requests and Beautiful Soup for web scrapping.
- **Spotipy:** This is a lightweight Python library built purposely for the Spotify Web API. It is used to get access to all the music data provided by the Spotify platform.
- **Other tools**

- **Spotify Web API:** This is an API built by Spotify that is based on the REST principles. It returns JSON metadata about music artists, albums, and tracks directly from the Spotify Data Catalogue. This tool was used to scrape artist, album, and track data of Ghanaian hiplife artists and subsequently generated audio features of Ghanaian hiplife tracks.
- **Jupyter Notebook:** It is an open-source web application that allows you to write and share notebooks (documents) that contains code, visualizations etc. Its many uses include data cleaning and transformation, data visualizations, and machine learning, among others.

Scrapping Ghanaian Music Websites

Before using the Spotify Web API to generate audio features of Ghanaian hiplife songs, I began by scrapping music data of Ghanaian hiplife songs on some of the popular music websites. With the use of Beautiful Soup, Re and request python libraries, I was able to scrape music data of 1800 songs from Ghana Motion. I was unsuccessful in scrapping music data from other music websites such as ghanamusic.com, hitxgh, ghxclusives etc because of many inconsistencies (with respect to 'class' and 'id' namings in tags) across the websites. Ghana Motion has a reasonably consistent website structure making it easy to get those music data at scale. Most of the songs I downloaded were those of upcoming artists. This is so because, like every other musician, underground artists in particular want to be heard in order to come into the mainstream; hence, they put almost all their songs on music websites for free download. Established artists usually have their songs on major streaming platforms like Spotify, Apple Music, and others.

Visualizations and Insights

Afterwards, by way of googling, I gathered music data of hit songs in Ghana from 2010 to 2019 and stored them in excel. At this stage, I wanted to explore the data I had gathered to spot any useful insights through visualizations. Using google sheets and Tableau, I generated a number of useful visualizations. The following figures below shows some of the visualizations.

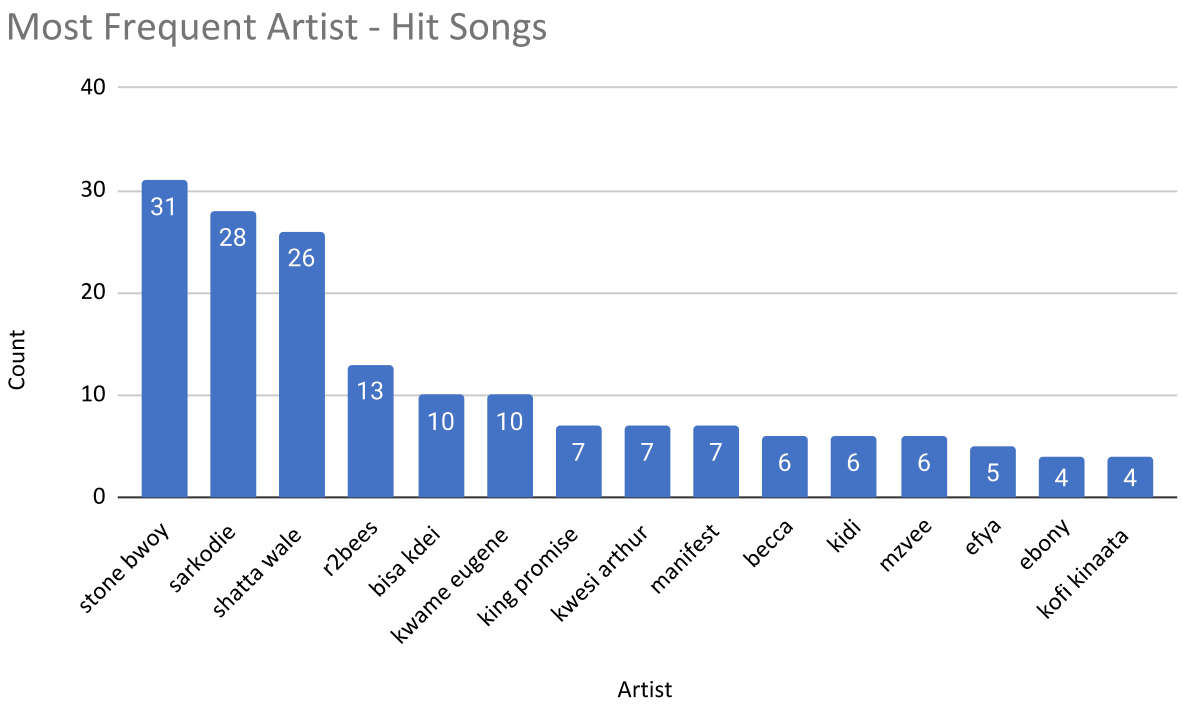


Figure 1

From the dataset I generated, figure 1 shows the top 15 artists in Ghana. These artists represent the top mainstream ones who are widely known for making hit songs. They have been

in the industry for at least six years, built their skills and understood the dynamics of making good songs within the music industry.

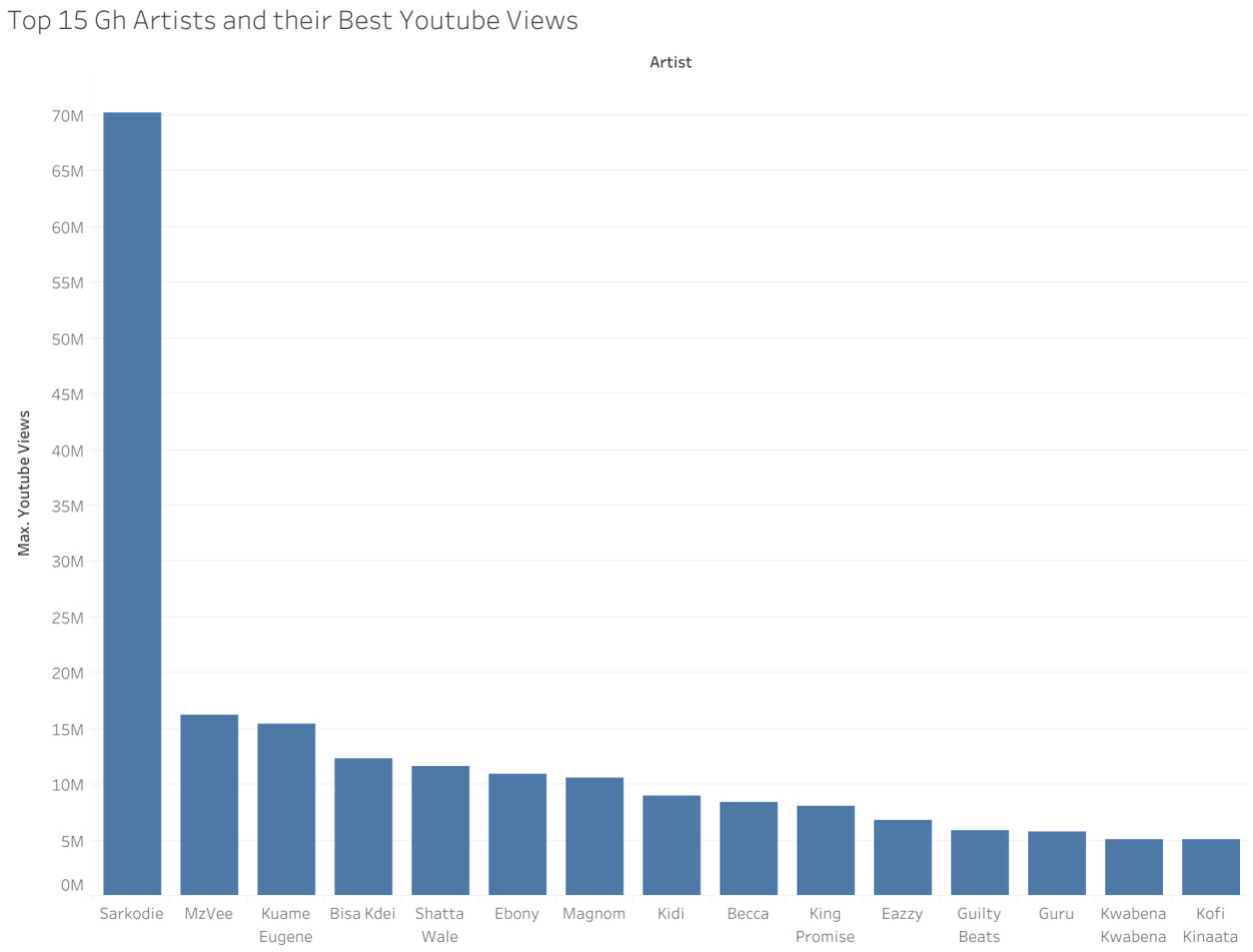


Figure 2

Figure 2 represents artists and their best YouTube views. These artists also double as some of Ghana's finest hit makers and mainstream artists. From the graph, Sarkodie is the Ghanaian artist with the highest YouTube views. He made 71 million views with his "Adonai" track featuring Castro. He released the song in 2014.

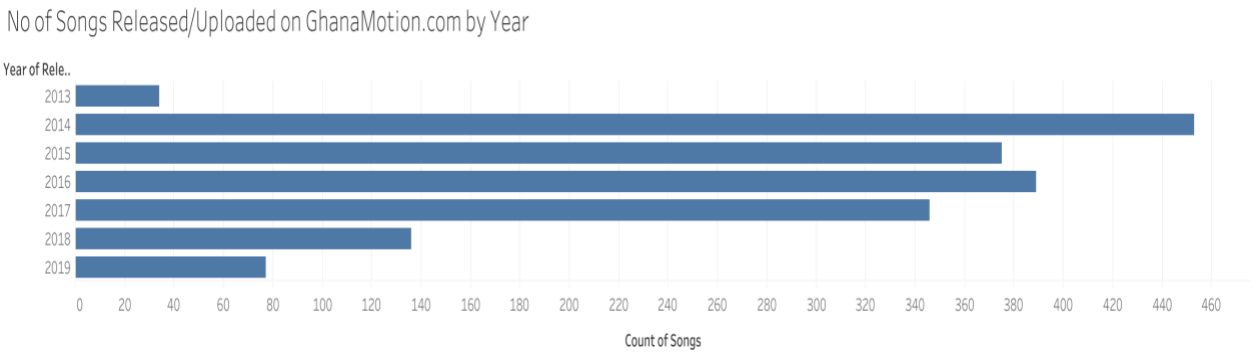


Figure 3

Figure 3 shows the number of songs released by artists/uploaded on ghanamotion.com. Although there are other Ghanaian music websites which upload songs which may not necessarily be on ghanamotion.com, the above figure gives an insight into the average number of songs uploaded on Ghana's music websites per year.

Scrapping Spotify Music Catalogue, Feature Extraction and Insights

To scrape music data from Spotify's Music Catalogue, I used Spotipy. It is a lightweight python library for the Spotify Web API. It helps developers get access to information on artists, albums, tracks and playlists. I registered as a developer and got a unique client ID and client secret which enabled me to scrape music data on the music catalogue. Following that I used the various methods available to scrape music data on Ghanaian hiplife artists.

The Spotify platform is divided into markets (countries). Ghana unfortunately is not yet a market on the music streaming platform. This means that Ghanaians (listeners) who have Spotify accounts live in other markets (countries) or are using the accounts of those who live abroad where Spotify is available. The popularity of a track or an artist is measured across markets (countries). The popularity of a track scores between 0 and 100. A song or an artist can be more popular in one

market than the other. Hence the popularity score is an aggregate across markets. An interesting observation made was that a song may have a popularity score of 10/100 across all markets on the streaming platform and still be a big hit in the Ghanaian music industry. For example, in 2016 Shatta Wale released "Dancehall King" which has a popularity score of 10/100. This was a big hit in Ghana up till now. Another example was Becca's "Daa ke da" which she released in 2013 and has a popularity score of 10/100. This song again was a big hit in Ghana. This means that to be able to get a good grasp of which songs were hits in Ghana, one needs a good contextual knowledge of the industry in addition to the popularity score on Spotify. The popularity score which changes over time depending on the streams gives a good indication of how Ghanaian artists and their songs are performing relative to global markets.

Visualizations and Insights

After conducting series of data wrangling, I generated the following visualizations:

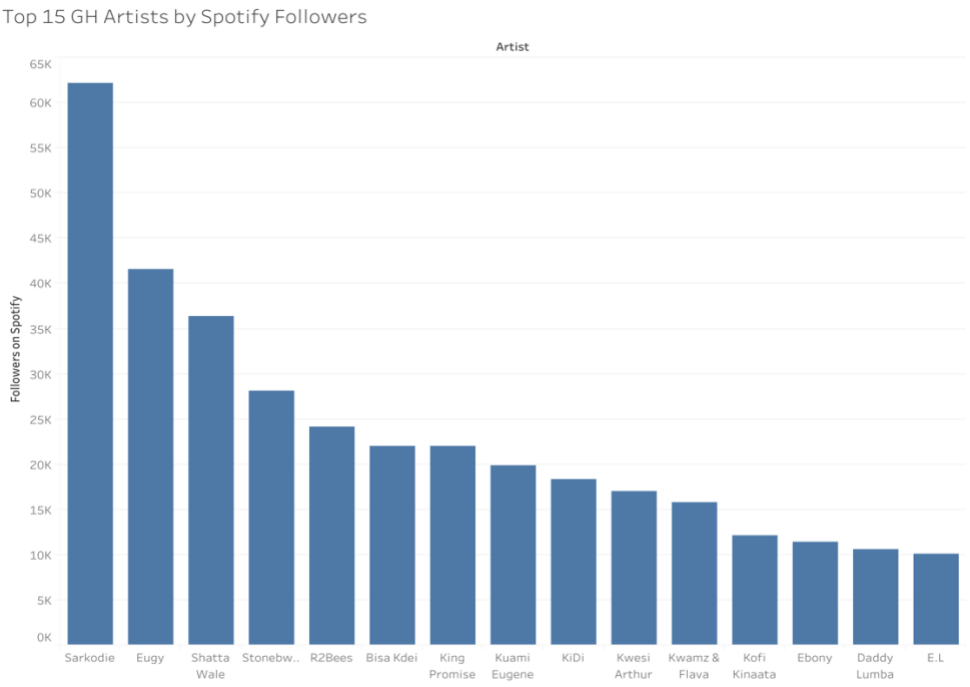


Figure 4

Figure 4 shows the top 15 Ghanaian artists by followers across all markets on Spotify. Sarkodie is the Ghanaian artist with the largest following on Spotify. From this figure, we see almost the same artists from the first task we completed (scrapping music data from GhanaMotion.com), which further confirms that these are the hitmakers.

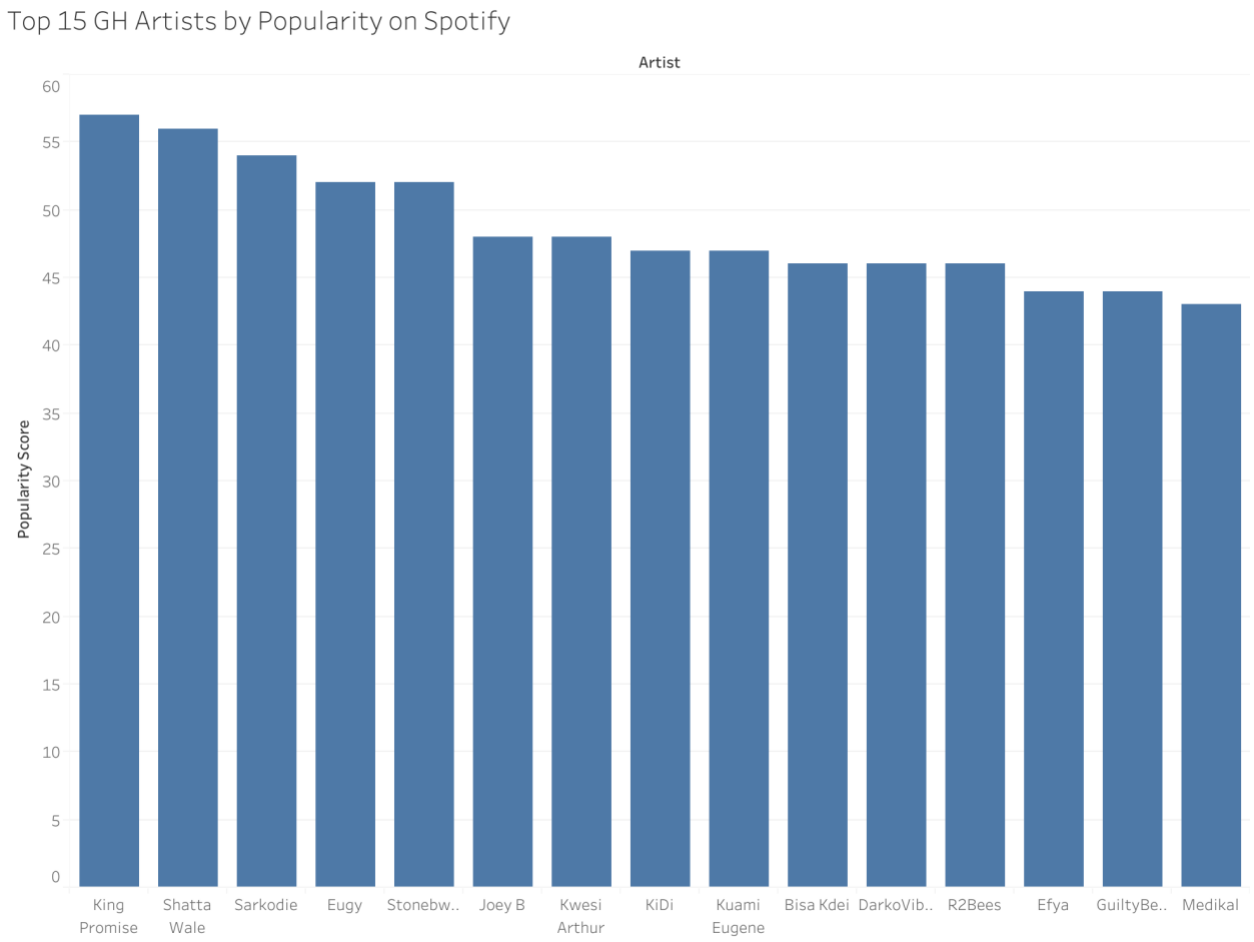


Figure 5

Figure 5 shows the top 15 Ghanaian artists and their popularity score across all markets on Spotify of which King Promise tops. The above two visualizations are useful information for a variety of scenarios. Firstly, it is excellent information for corporate firms who are considering using a Ghanaian artist as brand ambassadors for their firms. Secondly, international music companies

who want to work with local artists on any kind of project can use this visualization to gain insights into the top artists in the country.

Visualizations of Audio Features

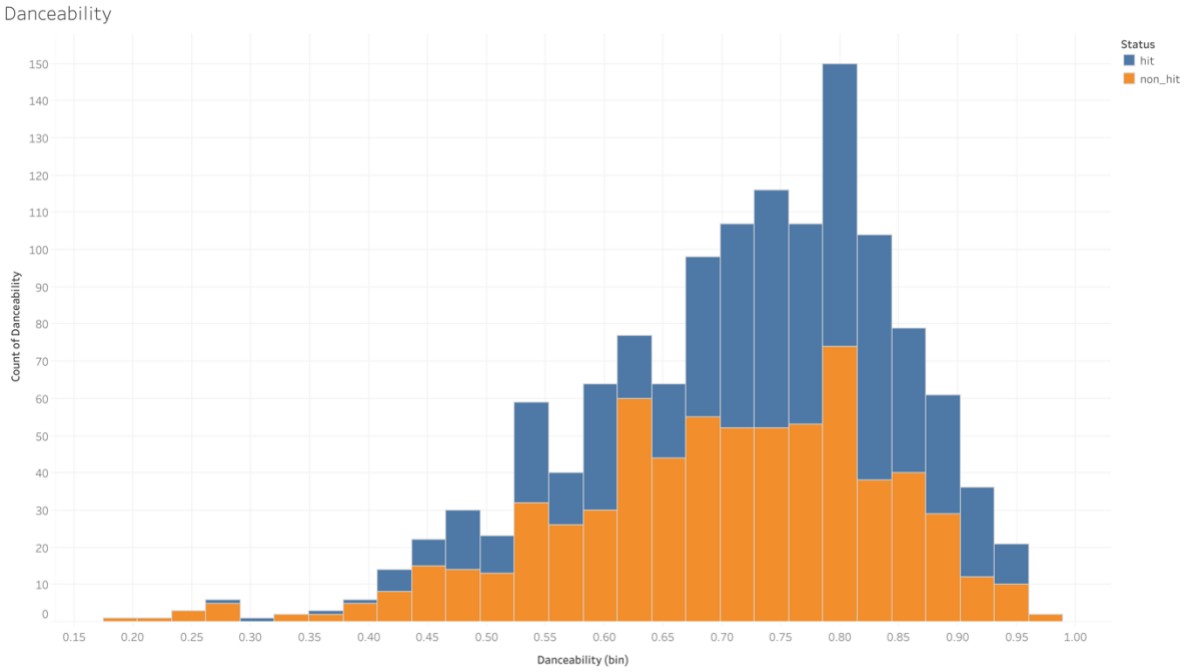


Figure 6

Figure 6 shows the distribution of the danceability of both hit and non-hit songs. Danceability refers to the suitability of a track for dancing based on musical elements such as beat strength, rhythm stability, and tempo. The average danceability for hit songs is 0.7324 while the average danceability for non-hit songs is 0.697. This means that while both hit and non-hit songs are danceable, hit songs are more danceable than non-hit songs.

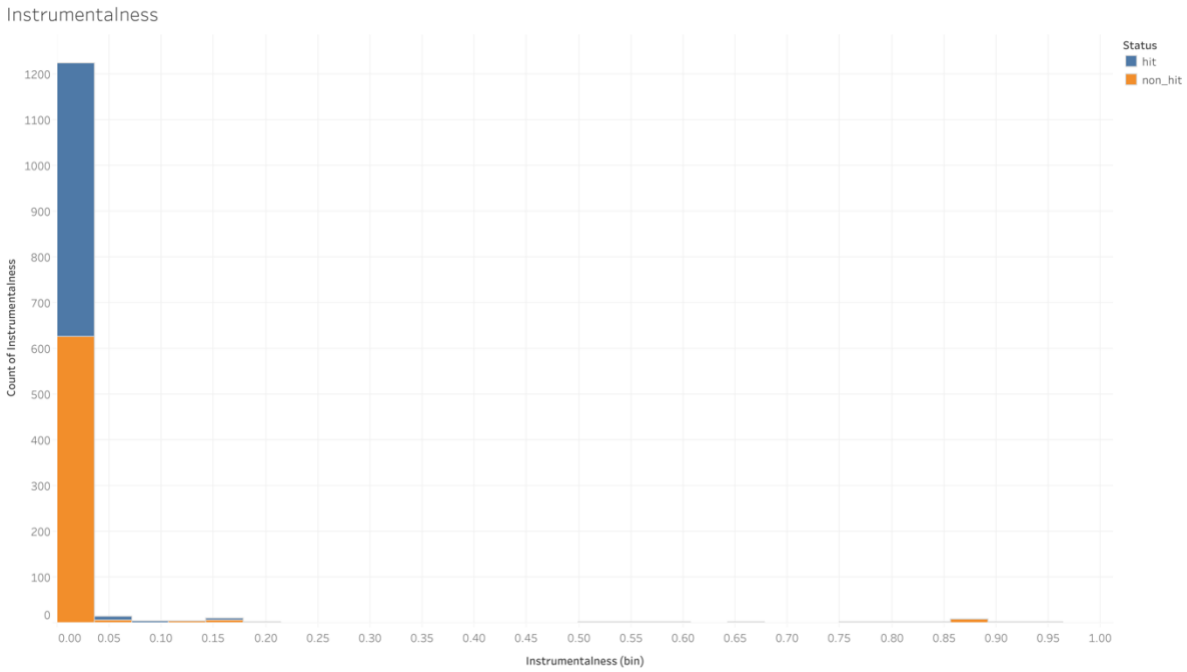


Figure 7

Figure 7 shows the distribution of Instrumentalness for both hit and non-hit songs. Instrumentalness measures the probability of a song containing no vocals. "ooh" and "aah" sounds are treated as instrumentals whilst spoken words and rap are vocals. The closer the score is to 0 means it contains vocals. The average instrumentalness for hit songs was 0.00570, and the average instrumentalness for non-hit songs was 0.0398. This means that while both hit and non-hit songs are highly vocal, hit songs are less vocal than non-hit songs.

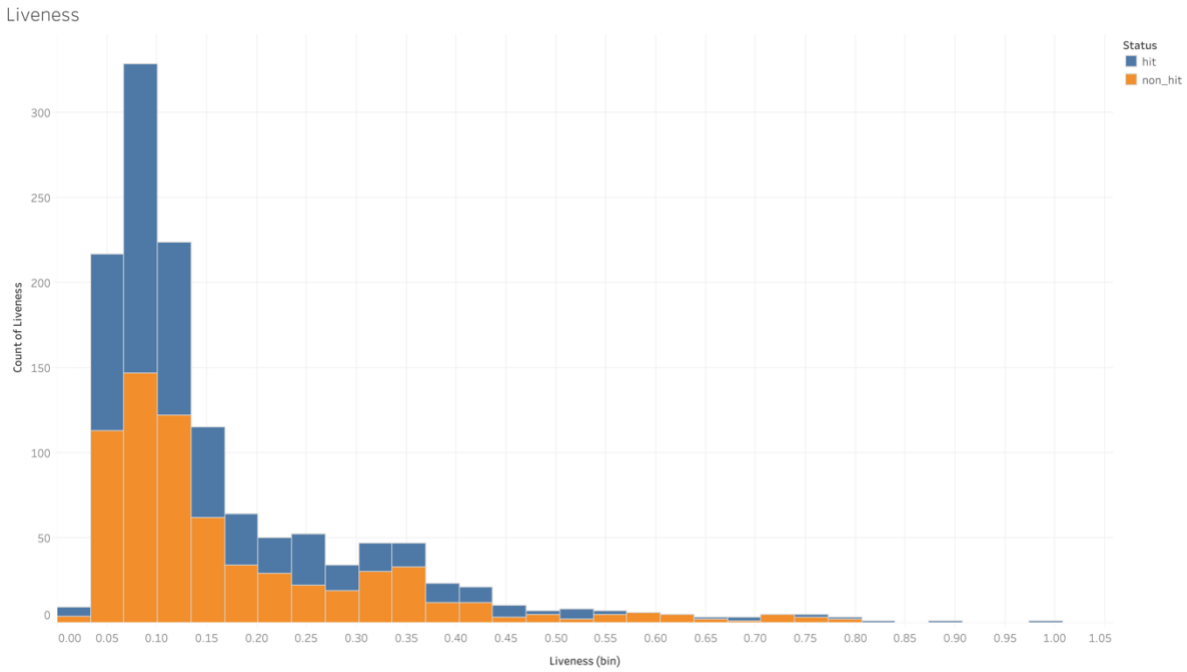


Figure 8

Figure 8 shows the distribution of liveness of hit and non-hit songs. Liveness measures the presence of an audience in the song. The higher the score, the higher the probability that the song was performed live. The average liveness for hit songs is 0.1558, and the average liveness for non-hit songs is 0.1768.

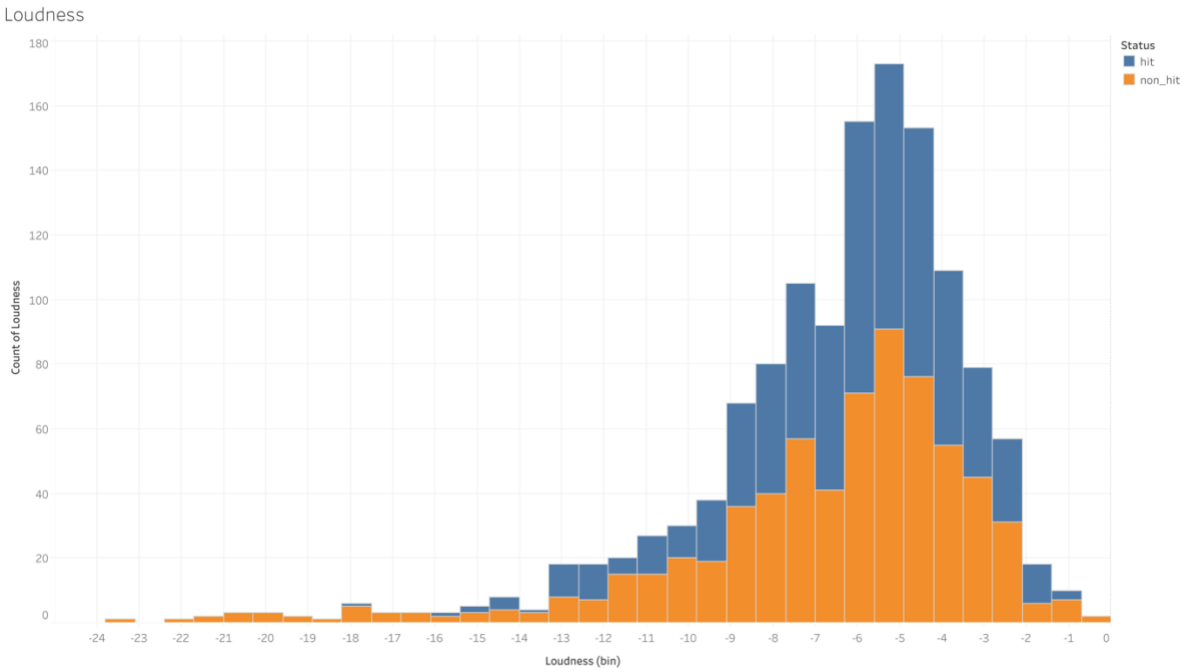


Figure 9

Figure 9 shows the distribution of loudness. Loudness measures the overall loudness measured in decibels. The average loudness for hit songs is -6.2245 decibels, and the average loudness for non-hit songs is -6.742. This means that non-hit songs are generally louder than hit songs.

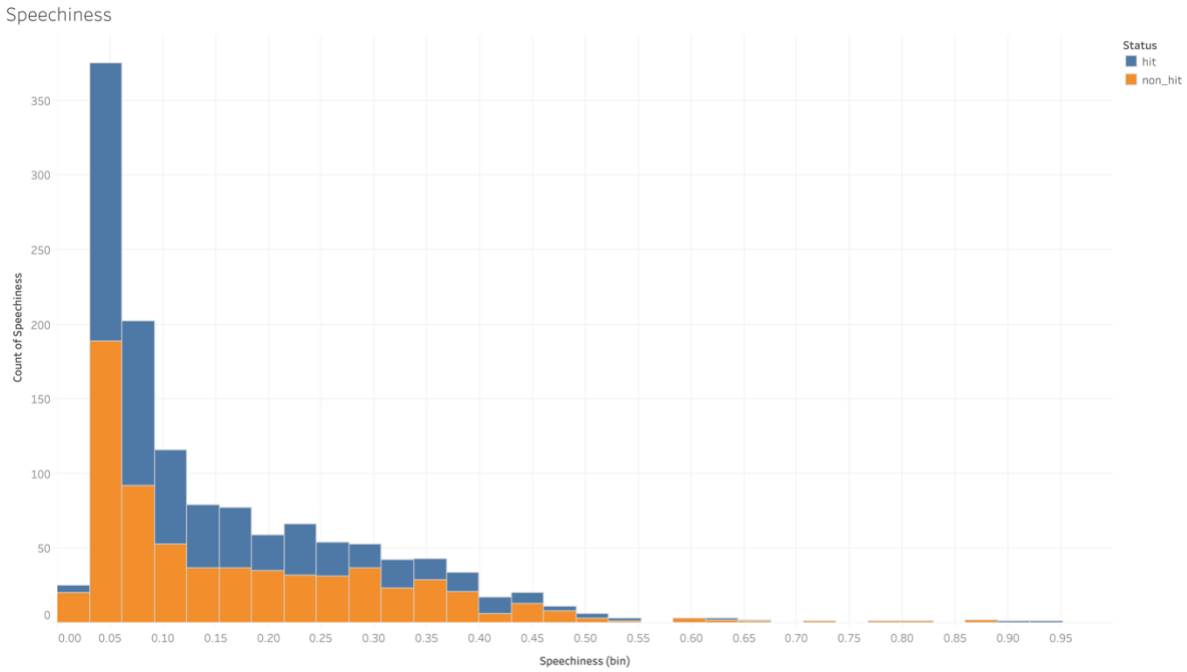


Figure 10

Figure 10 shows the distribution of speechiness in a song. Speechiness detects the presence of a spoken word in a song. The closer the value is to 1.0, the likelihood that it is poetry, audiobook, or other similar audio. If it is closer to 0.0, it means that the song contains both music and speech. The average score for hit songs is 0.14559, and the average score for non-hit songs is 0.1675.

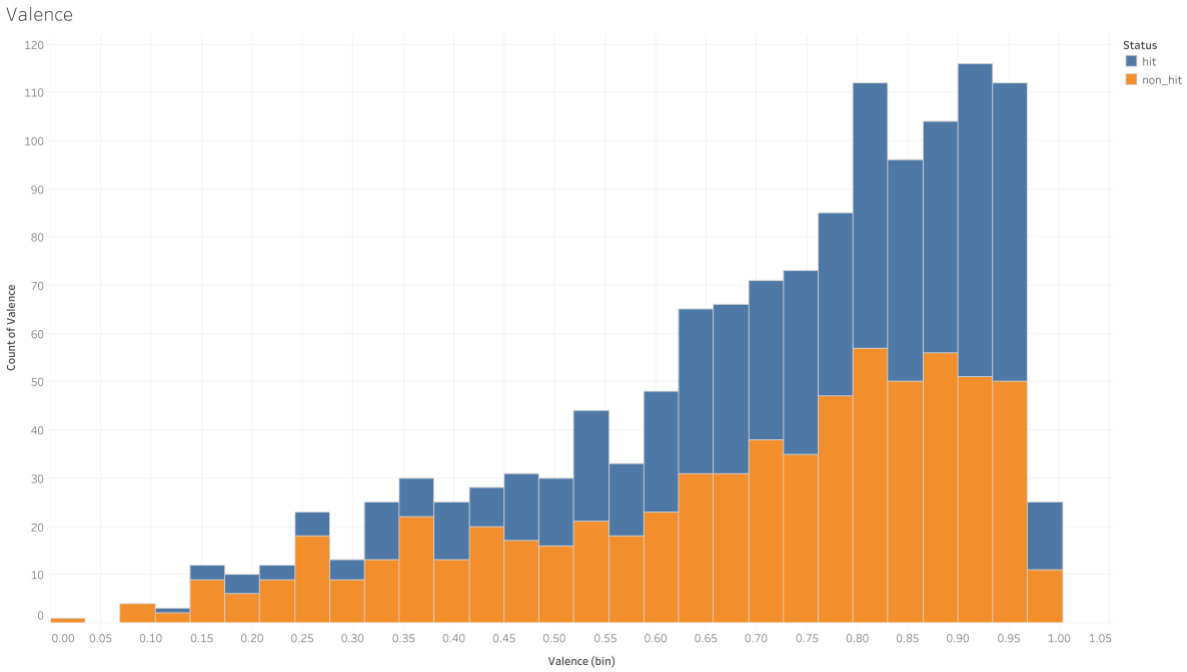


Figure 11

Figure 11 shows the distribution of valence. Valence measures how positive a song is. Songs with high valence are positive (happy, cheerful) whiles songs with low valence are negative (sad, depressed, angry). The average valence for hit songs is 0.7344, and the average valence for non-hit songs is 0.684. Although both hit and non-hit songs have high valence scores, hit songs are generally more positive than non-hit songs.

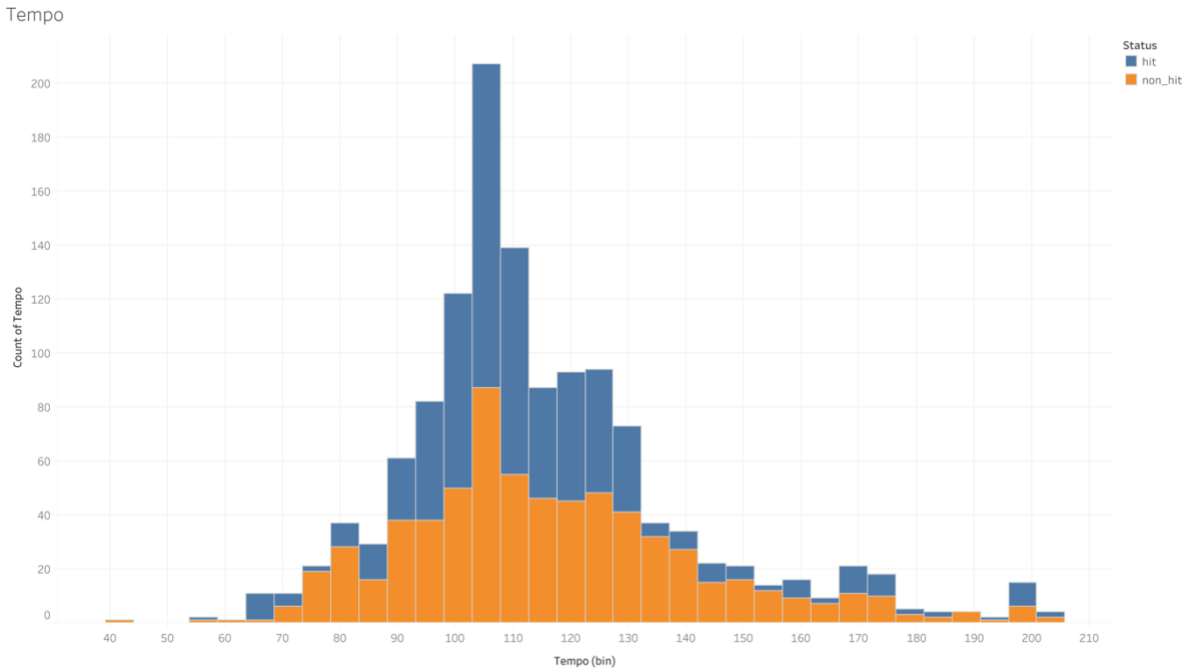


Figure 12

Figure 12 shows the distribution of tempo. Tempo refers to the speed or pace of a song. The average tempo for hit songs is 113.327, and the average tempo for a non-hit song is 116.808. Non-hit songs are generally faster than hit songs.

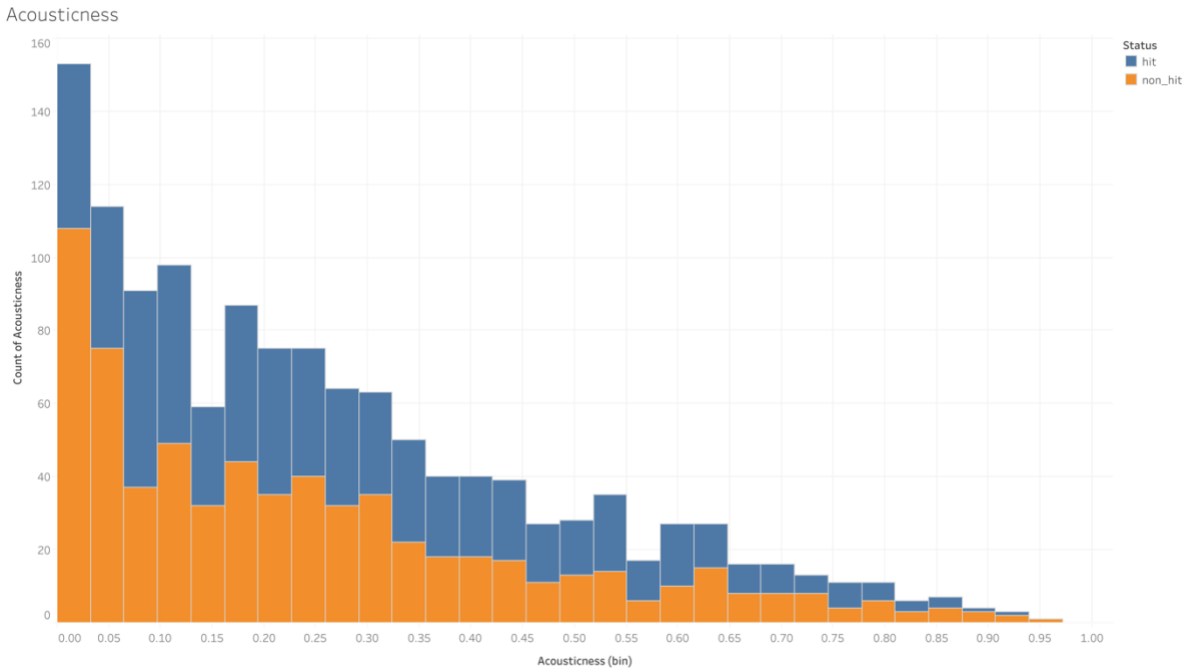


Figure 13

Figure 13 shows the distribution of Acousticness. Acousticness score measures the probability that a song is acoustic. The average acousticness for hit songs is 0.2838, and the average acousticness for a non-hit song is 0.245. Notwithstanding the observation that both hit and non-hit songs are generally less acoustic, non-hit songs are less acoustic than hit songs.

Preprocessing and Feature Selection

The final size of the dataset to be used for machine learning was 1297 rows with 21 columns. I divided the dataset into hit and non-hit songs. Songs with popularity score of 10 and above were labeled as hits. Songs with popularity score of 1 and 0 were labeled as non-hits. I created a column called ‘status’ and encoded hits as 1 and non-hits as 0. Afterwards, I checked the ratio of hits to non-hits to ensure that they were fairly balanced before proceeding. The ratio of hits to non-hits was 47.7:52.3 which is fairly balanced.

I ran a Logit function in order to assess the significance of all the features I had in my dataset. Below is the output of the Logit function.

```
Optimization terminated successfully.
         Current function value: 0.654357
         Iterations 7
                         Results: Logit
==================================================================
Model:                Logit              Pseudo R-squared:  0.055
Dependent Variable:   status             AIC:               1725.4026
Date:                 2020-03-20 13:08   BIC:               1797.7519
No. Observations:     1297               Log-Likelihood:    -848.70
Df Model:             13                 LL-Null:           -897.67
Df Residuals:         1283               LLR p-value:       4.1613e-15
Converged:            1.0000             Scale:             1.0000
No. Iterations:       7.0000
------------------------------------------------------------------
                     Coef.  Std.Err.     z     P>|z|    [0.025  0.975]
------------------------------------------------------------------
duration_min       -0.0349   0.0531 -0.6575 0.5109 -0.1390  0.0692
explicit            0.2761   0.1786  1.5458 0.1222 -0.0740  0.6262
key                 0.0046   0.0161  0.2839 0.7765 -0.0269  0.0360
mode                0.1357   0.1187  1.1433 0.2529 -0.0970  0.3685
time_signature     -0.0452   0.1103 -0.4100 0.6818 -0.2615  0.1710
acousticness        1.0743   0.2767  3.8827 0.0001  0.5320  1.6165
danceability        1.3123   0.4544  2.8880 0.0039  0.4217  2.2029
energy             -0.2663   0.5063 -0.5260 0.5989 -1.2586  0.7260
instrumentalness   -3.7605   1.0355 -3.6316 0.0003 -5.7901 -1.7310
liveness           -0.9538   0.4381 -2.1773 0.0295 -1.8125 -0.0952
loudness            0.0450   0.0230  1.9564 0.0504 -0.0001  0.0900
speechiness        -1.5524   0.4630 -3.3531 0.0008 -2.4597 -0.6450
valence             0.6260   0.3187  1.9641 0.0495  0.0013  1.2508
tempo              -0.0055   0.0023 -2.3671 0.0179 -0.0100 -0.0009
==================================================================
```

Figure 14

Based on the output, all features with a p value greater than 0.05 are insignificant and hence are to be withdrawn. The features that were taken down were *duration_min*, *explicit*, *key*, *time signature*, *mode* and *energy*. After taking them down, I generated another Logit function on the remaining features. Find below the output.

```
Optimization terminated successfully.
         Current function value: 0.656415
         Iterations 7
                         Results: Logit
================================================================
Model:              Logit            Pseudo R-squared: 0.052
Dependent Variable: status           AIC:              1718.7406
Date:               2020-03-20 13:08 BIC:              1760.0830
No. Observations:   1297             Log-Likelihood:   -851.37
Df Model:           7                LL-Null:          -897.67
Df Residuals:       1289             LLR p-value:      3.6177e-17
Converged:          1.0000           Scale:            1.0000
No. Iterations:     7.0000
----------------------------------------------------------------
                   Coef.  Std.Err.    z    P>|z|   [0.025  0.975]
----------------------------------------------------------------
acousticness       1.0912   0.2716  4.0174 0.0001  0.5588  1.6236
danceability       1.0714   0.3673  2.9167 0.0035  0.3514  1.7913
instrumentalness  -3.8117   1.0288 -3.7049 0.0002 -5.8282 -1.7952
liveness          -1.0282   0.4189 -2.4546 0.0141 -1.8493 -0.2072
loudness           0.0424   0.0182  2.3356 0.0195  0.0068  0.0780
speechiness       -1.4684   0.4387 -3.3474 0.0008 -2.3281 -0.6086
valence            0.4730   0.2951  1.6029 0.1090 -0.1054  1.0514
tempo             -0.0064   0.0019 -3.2917 0.0010 -0.0102 -0.0026
================================================================
```

Figure 15

All the p values of the remaining features were less than 0.05 meaning were significant enough to be included in building the model.

Logistic Regression

Logistic regression is a linear model for classifying outcomes into two or more categories. The model outputs the probability that an item belongs to one category or the other [9]. For our case, a song was be classified as 1(hit) or 0(non-hit). After tuning the regularization parameter (C) several times, the value of C that produced the smallest misclassification error was 1 which produced an accuracy of 0.64. The confusion matrix for this accuracy was 88 true negatives, 79 true positives, 33 false negatives, and 60 false positives. This means that 79 songs were hits, and the model predicted them as a hit. Eighty-eight of the songs were non-hits, and the model predicted them as non-hits. Sixty of the songs were non-hits, and the model predicted them as hits. Finally, 33 of the songs were hits, and the model predicted them as non-hits.

Naïve Bayes

This is a type of classification algorithm that assumes that all features to be used in the modeling are all independent. This means that the presence of one feature is unrelated to the presence of another one. After fitting the Gaussian Naïve Bayes model on the data, it generated an accuracy of 0.56. The figure below provides an additional information on the model accuracy.

```
[[80 53]
 [63 64]]
              precision    recall  f1-score   support

         0.0       0.56      0.60      0.58       133
         1.0       0.55      0.50      0.52       127

    accuracy                           0.55       260
   macro avg       0.55      0.55      0.55       260
weighted avg       0.55      0.55      0.55       260
```

Figure 16

From the figure we see that the model generated 80 true negatives which means 80 songs were non-hits and the model predicted them as non-hits. 64 (true positives) of the songs were hits, and the model predicted them as hits. 63 (false negatives) of the songs were hits, and the model predicted them as non-hits. 53 (false positives) of the songs were non-hits and the model predicted them as hits.

Random Forest

It is a supervised learning algorithm that is used for classification. A random forest generates decision trees on data samples and acquires the results from each of the trees and employing voting, selects the best results. It reduces the probability of overfitting by averaging results from each of

the decision trees [1]. After fitting the Random Forest model on the data, it generated an accuracy of 0.61. The figure below gives additional information about the accuracy of the model.

```
[[83 41]
 [60 76]]
              precision    recall  f1-score   support

         0.0       0.58      0.67      0.62       124
         1.0       0.65      0.56      0.60       136

    accuracy                           0.61       260
   macro avg       0.61      0.61      0.61       260
weighted avg       0.62      0.61      0.61       260
```

Figure 17

From the figure, we see that the model generated 83 true negatives, which means 83 songs were non-hits, and the model predicted them as non-hits. 76 (true positives) of the songs were hits, and the model predicted them as hits. 60 (false negatives) of the songs were hits, and the model predicted them as non-hits. 41 (false positives) of the songs were non-hits, and the model predicted them as hits.

K Nearest Neighbor

The idea behind K nearest neighbor is that for a new test data point to be classified; it looks for *k* training data points that are closest to the test data point. The test data point then takes on the labels of the *k* training data points closest to it. The number of neighbors, *K* can be specified with an integer value as a parameter in the *KNeighborsClassifier* class.

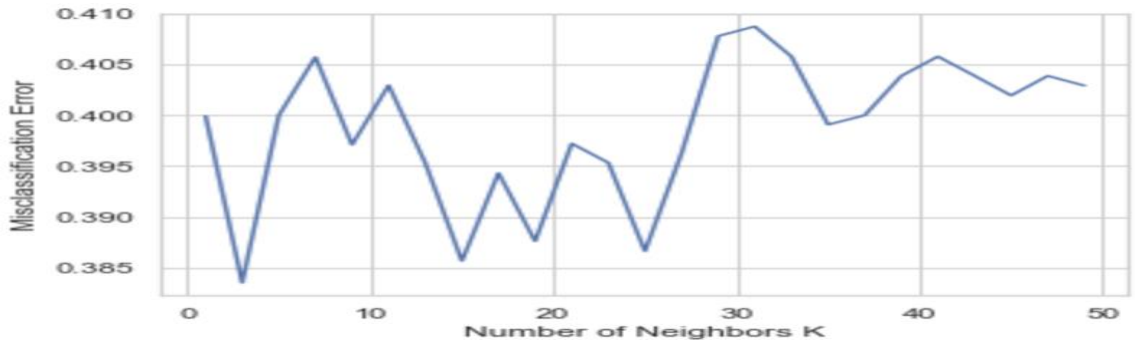


Figure 18

After performing the cross-validation training for k = [1, 50], the value of K that minimizes the misclassification error is 3. At a value of k=3, the accuracy of the model was 0.60. The figure below gives additional information about the accuracy of the model.

```
[[82 42]
 [64 72]]
              precision    recall  f1-score   support

           0       0.56      0.66      0.61       124
           1       0.63      0.53      0.58       136

    accuracy                           0.59       260
   macro avg       0.60      0.60      0.59       260
weighted avg       0.60      0.59      0.59       260
```

Figure 19

From the figure, we see that the model generated 82 true negatives, which means 83 songs were non-hits, and the model predicted them as non-hits. 72 (true positives) of the songs were hits, and the model predicted them as hits. 64 (false negatives) of the songs were hits, and the model predicted them as non-hits. 42 (false positives) of the songs were non-hits, and the model predicted them as hits.

Chapter 4: Testing and Results

After fitting the algorithms to the datasets, they were evaluated to see how well they perform on new data they have not seen before. For each algorithm, optimum parameters were chosen to ensure minimum misclassification errors. The table below shows a summary of the evaluation results of each of the algorithms. Each table shows the overall accuracy score, confusion matrix, precision, and recall.

Logistic Regression (C=1.0)

Table 1

Non-Hit Songs	88 – True Negatives	60 – False Positives
Hit Songs	33 – False Negatives	70 – True Positives

Overall Accuracy	**Precision**	**Recall**
0.64	0.65	0.64

Naïve Bayes

Table 2

Non-Hit Songs	80 – True Negatives	53 – False Positives
Hit Songs	63 – False Negatives	64 – True Positives

Overall Accuracy	**Precision**	**Recall**
0.55	0.55	0.55

Random Forest

Table 3

Non-Hit Songs	83 – True Negatives	41 – False Positives
Hit Songs	60 – False Negatives	76 – True Positives

Overall Accuracy	**Precision**	**Recall**
0.61	0.62	0.61

K-Nearest Neighbor

Table 4

Non-Hit Songs	82 – True Negatives	42 – False Positives
Hit Songs	64 – False Negatives	72 – True Positives

Overall Accuracy	**Precision**	**Recall**
0.59	0.60	0.59

From the results above, Logistic Regression with a C=1 is the best performing model overall with an accuracy of 0.64 with a precision of 0.65 and a recall of 0.64. This means this model is more likely to predict a new data point as belonging to its actual class than the rest of the algorithms, which have precision and recall less than 0.65 and 0.64, respectively.

Chapter 5: Conclusion and Future Work

In this paper, we set out to investigate whether audio features could help to predict the popularity of a song. Using the Spotify Web API, we scrapped over a 1000 Ghanaian hiplife songs and extracted audio features from them. Using four supervised machine learning classification algorithms (Logistic Regression, Naïve Bayes, K Nearest Neighbors, and Random Forest), we identified that Logistic Regression performed best, with an accuracy of 0.64, in predicting the popularity of the song after using optimum parameters for each of the algorithms.

Limitation

Given that the accuracy for the best performing model, Logistic Regression, is very low, it suggests that audio features are not enough to predict the popularity of a song. Thus, other factors might be at play. Given this, the model cannot be deployed for use in the industry. Additionally, by using Spotify's Web API to scrape songs from their music catalogue, it suggests that to predict the popularity of a song, it would have to be uploaded on Spotify's platform. While this is not ideal, it is the only available solution to predict the popularity of a song.

Future Work

Given the low accuracy of the best performing model, there is the need to explore other factors that can contribute to predicting the popularity of a song. One way to do that is to explore how the use of social media contributes to the popularity of a song. Given that modern artists rely on social media to promote their songs, it will be interesting to explore how their social media marketing strategies contribute to the popularity of their songs.

Furthermore, it will be great to build a software that generate scores for the relevant audio features. The software will prevent the instances where one will have to upload his or her songs on Spotify before getting the scores for the audio features. The generated scores for the audio features can then be used for analyses. In this situation, songs produced straight out of the studio can be run through the model to generate the probability that it is going to be a hit. In the event of low probability, the song can be reworked to improve on the probability.

References:

[1] Leo Breiman, Michael Last, and John Rice. 2003. Random Forests: Finding Quasars. In *Statistical Challenges in Astronomy*, Springer, New York, NY, 243–254. DOI:https://doi.org/10.1007/0-387-21529-8_16

[2] Rakesh Vidya Chandra and Bala Subrahmanyam Varanasi. 2015. *Python Requests Essentials*. Packt Publishing.

[3] Dorien Herremans and Tom Bergmans. Hit Song Prediction Based on Early Adopter Data and Audio Features. *The 18th International Society for Music Information Retrieval Conference (ISMIR) - Late Breaking Demo*. Retrieved March 28, 2020 from https://www.academia.edu/37332881/Hit_Song_Prediction_Based_on_Early_Adopter_Data_and_Audio_Features

[4] John D. Hunter. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering* 9, 3 (May 2007), 90–95. DOI:https://doi.org/10.1109/MCSE.2007.55

[5] Yekyung Kim, Bongwon Suh, and Kyogu Lee. 2014. #nowplaying the Future Billboard: Mining Music Listening Behaviors of Twitter Users for Hit Song Prediction. In *SoMeRA@SIGIR*. DOI:https://doi.org/10.1145/2632188.2632206

[6] Wes McKinney. 2011. pandas: a Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing* 14, 9 (2011). Retrieved March 21, 2020 from https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics

[7] Vineeth G. Nair. 2014. *Getting Started with Beautiful Soup*. Packt Publishing.

[8] Joseph Oduro-Frimpong. 2009. Glocalization Trends: The Case of Hiplife Music in Contemporary Ghana. (December 2009). Retrieved March 27, 2020 from https://www.academia.edu/414858/Glocalization_Trends_The_Case_of_Hiplife_Music_in_Contemporary_Ghana

[9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, (October 2011), 2825−2830.

[10] Minna Reiman and Philippa Örnell. 2018. Predicting Hit Songs with Machine Learning. Retrieved March 27, 2020 from https://www.semanticscholar.org/paper/Predicting-Hit-Songs-with-Machine-Learning-Reiman-%C3%96rnell/bdf8309db0e1379519760daad3fa4dfbb46ebee4

[11] Everett M. Rogers. 2010. *Diffusion of Innovations, 4th Edition*. Simon and Schuster.

[12] Mark Summerfield. 2010. *Programming in Python 3: A Complete Introduction to the Python Language*. Addison-Wesley Professional.

[13] Kai Ming Ting. 2017. Confusion Matrix. In *Encyclopedia of Machine Learning and Data Mining*, Claude Sammut and Geoffrey I. Webb (eds.). Springer US, Boston, MA, 260–260. DOI:https://doi.org/10.1007/978-1-4899-7687-1_50

[14] Stefan van der Walt, S. Chris Colbert, and Gael Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering* 13, 2 (March 2011), 22–30. DOI:https://doi.org/10.1109/MCSE.2011.37