

Comparing Song Audio Features to Rankings on The Billboard Hot 100

Kevin Carr 501150122

Supervisor: Ceni Babaoglu, PhD

Date: December 3, 2022

Table of Contents

[Abstract 3](#_Toc120959984)

[Introduction 4](#_Toc120959985)

[Literature Review 5](#_Toc120959986)

[Data Description 8](#_Toc120959987)

[Data Sources 8](#_Toc120959988)

[Data Pre-Processing and Organisation 8](#_Toc120959989)

[File and Calculation Locations 10](#_Toc120959990)

[Descriptive Statistics 11](#_Toc120959991)

[Audio Feature Descriptions 13](#_Toc120959992)

[Histograms 14](#_Toc120959993)

[Historical Changes in Audio Features 19](#_Toc120959994)

[Correlation Analysis 24](#_Toc120959995)

[Analysis of Genres 27](#_Toc120959996)

[Project Approach 29](#_Toc120959997)

[Outliers 29](#_Toc120959998)

[Data Mining 29](#_Toc120959999)

[Predictive Analytics 30](#_Toc120960000)

[Results 30](#_Toc120960001)

[Outliers 30](#_Toc120960002)

[Clustering 31](#_Toc120960003)

[Classification 32](#_Toc120960004)

[Comparison of Results – Effectiveness 35](#_Toc120960005)

[Visualisation of Results Using Principal Component Analysis 36](#_Toc120960006)

[Statistical Analysis 43](#_Toc120960007)

[Ranking of Models 44](#_Toc120960008)

[Comparison of Results – Efficiency 45](#_Toc120960009)

[Comparison of Results – Stability 47](#_Toc120960010)

[Discussion and Conclusions 47](#_Toc120960011)

[Discussion and Limitations 47](#_Toc120960012)

[Conclusions 48](#_Toc120960013)

[Future Work 48](#_Toc120960014)

[References 50](#_Toc120960015)

[Attachments 54](#_Toc120960016)

# Abstract

The central problem in this project is to utilise publicly available audio feature data to predict whether or not a song is likely to appear on the Billboard Hot 100 charts. Music streaming services employ data models to characterise audio features for songs. This data is provided and publicly available for multiple streaming services, notably Spotify. The Billboard Hot 100 has been a music industry standard for approximately 70 years. The Billboard Hot 100 contains weekly rankings for songs which are based on sales, plays, and surveys.

Audio feature data and Billboard Hot 100 chart data for this project have been obtained from multiple sources and combined. Billboard Hot 100 charts were obtained for the entire history of the Billboard Hot 100, from 1958 to 2021. Two additional databases were obtained consisting of approximately 10 million songs with audio feature data. Were possible, this audio feature data was merged with songs from the Billboard Hot 100 charts. In cases where audio feature data for Billboard Hot 100 songs were not in available datasets, missing audio features were obtained using the Spotify API, where available. Audio features were obtained for approximately 75% of songs in the Billboard Hot 100 charts, as well as an approximately 10 million additional songs.

For this project, two main techniques have been employed, namely data mining and predictive analytics. First, data mining and knowledge discovery was used to explore the data, cluster audio features, and determine correlations between audio features. Second, predictive analytics was used to attempt to build a predictive model using the data. By utilising knowledge discovered during the data mining phase of the project, predictive analysis has been broken into sets of clustered songs with similar audio features and/or genres.

After employing a number of predictive machine learning models on clustered and un-clustered data, classification success was mixed. Since the dataset is highly unbalanced, achieving high accuracy was trivial. However, optimising for more nuanced metrics such as precision, recall, or F1-score has proven difficult. Based on visual inspection and statistical analysis, the predictions appear to be correct, albeit not as useful anticipated. This is due to the fact that a large number of songs exist with audio features consistent with hit songs, but only a small portion of all songs become hits. Therefore, popular audio feature characteristics predicted using these methods may be necessary for a song to achieve popularity but are not sufficient for commercial success. More research is necessary to confirm this hypothesis.

# Introduction

Music streaming services employ data models to characterise audio features for songs, and use this data to recommend songs and playlist to their listeners. This data is provided and publicly available for multiple streaming services, notably Spotify (Spotify, n.d.).

The Billboard Hot 100 has been a music industry standard for approximately 70 years (Wikipedia, 2022). The Billboard Hot 100 contains weekly rankings for songs which are based on sales, plays, and surveys.

For this project, data mining and predictive analytics have been employed. Data mining and knowledge discovery were used to explore the data, cluster audio features, and determine correlations between audio features. Predictive analytics was used to attempt to build a predictive model.

This analysis has the potential to predict future trends in music or the performance of an individual song. These predictions could be useful to musicians or producers attempting to optimise success, or listeners looking for something new.

# Literature Review

The primary focus of this literature review was to gather understanding on previous research related to music clustering techniques and the prediction of popularity for songs, especially in cases where song audio features were used for clustering or prediction. Google Scholar (Google Scholar, n.d.), PaperDigest.org (Paper Digest, n.d.), and Elicit.org (Elicit, n.d.) were used to investigate and gather research materials.

Popularity can be defined in numerous ways. In this study, popularity is simply considered to be an appearance on the Billboard Hot 100 charts. The magnitude of this popularity may be further be defined using total weeks on the chart, or top rank on the chart (Lee et. al., 2018). Using appearance on the Billboard Hot 100 as a metric for popularity has been used in other similar studies (Reiman et.al., 2018). Another commonly used popularity metric is ‘popularity’ as defined in the Spotify API (Kim, 2021; Gao, 2021). Since the Spotify ‘popularity’ metric does not include historical popularity data and considers only Spotify streaming, this study focusses on the Billboard Hot 100 charts in order to get a longer-term, and wider perspective of music popularity.

Audio features used in music classification have evolved through various stages. MIDI (Musical Instrument Digital Interface) format musical notation has been used to cluster music into categories (Cilibrasi et. al., 2004; Cataltepe et. al., 2007). Low-level audio features such as mel-frequency cepstral coefficients, spectral flatness, and number of zero crossings have been used to predict steaming popularity (Yang et. al., 2017; Lee et. al., 2018; Araujo et. al., 2019), improve music recommendation systems (Li et. al., 2007; Schedl, 2013), and to classify emotion in music (Jia, 2022). High-level audio features such as danceability, instrumentalness, and speechiness are included in track information available from the Spotify API. These high-level audio features have been used to identify song attributes (Febirautami et. al., 2018), predict popularity (Reiman et.al., 2018; Martín-Gutiérrez et. al., 2020; Kim, 2021; Gao, 2021), to classify music into genres (Setiadi et. al., 2020), and to classify music into moods (Chen et. al., 2021).

Clustering music has been used for recommender systems (Li et. al., 2004; Li et. al., 2007; Huo, 2021), as well as to categorise music (Honingh et. al., 2011). In this study, overall trends, genres, and audio feature clusters have been considered to attempt to improve predictive analytics. It was hypothesised that the accuracy of predictions of song popularity using audio features could be improved by separating predictions by genre (Reiman et.al., 2018). Music genres have been used in combination with high-level audio features to predict popularity (Kim, 2021). It has been noted that audio features within the Billboard Hot 100 within genres are relatively consistent over time (O'Toole et. al., 2022).

In this study, clustering and classification have been used. Similar studies have had success with a variety of techniques and models. Neural networks have been used to predict popularity (Yang et. al., 2017; Gao, 2021), improve music recommendation systems (Li, 2021; Shi, 2021), or classify music (Jia, 2022; Li et. al., 2022). K-Means Clustering has been used in a number of studies to cluster data (Li et. al., 2007; Xu et. al., 2021; Kim et. al., 2021). A variety of classification models have been used to classify music, notably Support Vector Machines (Laurier et. al., 2009; Lee et. al., 2018; Reiman et.al., 2018; Araujo et. al., 2019; Setiadi et. al., 2020; Wilkes et. al., 2021), K-Nearest Neighbours (Cataltepe et. al., 2007; Reiman et.al., 2018; Kim, 2021), Decision Trees / Random Forests / Boosted Trees (West, 2008; Febirautami et. al., 2018; Chen et. al., 2021; Gao, 2021), and Logistic Regression (Reiman et.al., 2018; Chen et. al., 2021; Gao, 2021). In addition, Principle Component Analysis has been used to reduce dimensionality and improve predictive results (Gao, 2021).

Predicting popularity on the Billboard Hot 100 has been investigated in other studies (Lee et. al., 2018; Reiman et.al., 2018). However, no study has successfully used high-level audio features to predict popularity as defined above. Although high-level audio features were used, Reiman et.al. were not able to accurately predict song popularity. This was potentially due to an overly diverse dataset for non-hit songs. It was hypothesised that the accuracy of predictions of song popularity using audio features could be improved by separating predictions by genre (Reiman et.al., 2018). Additionally, although it has been demonstrated that popularity can be predicted using audio features alone, this was demonstrated using low-level audio features and different statistical descriptions for popularity (Lee et. al., 2018). This study aims to accurately predict song popularity, defined as appearance on the Billboard Hot 100 charts, using high-level audio features and genre data available from the Spotify API. In addition, none of the studies listed above utilised highly-unbalanced data to predict popularity, even though popularity, by definition is highly-unbalanced. This study aims to make predictions using a large and highly-unbalanced dataset.

# Data Description

## Data Sources

The data gathered for this project have been taken from multiple sources and combined. Data was found using the Google dataset search engine (Google Dataset Search, n.d.).

Three of the relevant sources were found on Kaggle.com, a popular online data science community where users can share datasets (Dhruvil Dave, 2021; Malte Grosse, 2022; Rodolfo Figueroa, 2020). Audio features from the large datasets were matched with the list of songs from the Billboard Hot 100. Missing data were obtained, where available, from the Spotify API.

## Data Pre-Processing and Organisation

Data Importing and preprocessing was completed in 3 stages:

1. Import and setup data types
2. Get missing data from Spotify API
3. Merge, clean, and save optimised datasets

Data import and cleaning is included as **Attachment 1**.

The “Billboard ‘The Hot 100’ Songs” dataset (Dhruvil Dave, 2021) was available in CSV format, and includes date, rank, song title, artist, last-week, peak-rank, and weeks-on-board. This CSV was imported into Python as a Pandas dataframe. This data did not include Spotify song ids, genres, release dates, or audio features, so the Spotify API was used to gather this data.

The SQLite dataset, “8+ M. Spotify Tracks, Genre, Audio Features” (Malte Grosse, 2022) included 9 tables totalling 44 columns. The database was queried to combine song title, artist, Spotify id, release date, and audio features. The queried data were exported to CSV and imported into Python as a Pandas dataframe.

The “Spotify 1.2M+ Songs” dataset (Rodolfo Figueroa, 2020) was available in CSV format. It included combine song title, artist, Spotify id, release date, and audio features. This CSV was imported into Python as a Pandas dataframe.

Based on findings from the literature review portion of this study, genre data was gathered for easily available songs. Since genre data was included in the SQLite database for many of the songs (Malte Grosse, 2022), this data was queried, exported as CSV, and imported into Python. Since multiple genres were often available for a given artist, all genres were populated, then the results were sorted by most common genre. This results in songs being categorised with only one genre. It should be noted that this is the most common genre, and not necessarily the most applicable genre. It should also be noted that not all songs have genre data associated with them. Approximately 69% of songs included genre data (approximately 6.6M entries).

Songs from the Billboard Hot 100 were queried by artist and song name using the Spotify API, gathering Spotify id were available. Songs with Spotify id were queried to gather audio features, genre data, and release dates. Since multiple genres were often available, all genres were obtained from the API, then the result corresponding to the most common genres from the SQLite query data were populated as the song’s genre. Similar to above, it should be noted that this is the most common genre, and not necessarily the most applicable genre, and not all songs have genre data associated with them.

Since Get Requests from the Spotify API were time consuming, undefined genres from the “Spotify 1.2M+ Songs” dataset (Rodolfo Figueroa, 2020) were left undefined due to time constraints.

The data from each dataset was combined to form 3 non-distinct working datasets:

* All songs including audio features (approximately 10M entries)
* The Billboard Hot 100 historical charts (approximately 300k entries)
* Songs from The Billboard Hot 100 that include audio features and genre (approximately 20k entries)

Since the dataframes are not distinct, in cases where mutually exclusive groups were necessary, Pandas dataframe query functions and vectorized formula may be implemented to segment data as required.

Once working datasets were compiled, they were exported as Pickle format. Pickle has a number of advantages over CSV format, most notably file size, the retention of data type formats, and the amount of time required to re-load the file into Python Pandas dataframe.

## File and Calculation Locations

Datasets and calculations used in this study can be found at the following URL:

<https://github.com/KevinCarr42/Billboard-100-Audio-Feature-Analytics>

Files too large to upload to GitHub been uploaded to a shared Google Drive folder (shared with Toronto Metropolitan University Google Drive accounts):

<https://drive.google.com/drive/folders/10wpORzZURV11VAUPKmCDDKxHFCvwjloR>

## Descriptive Statistics

Descriptive statistics for the datasets are included in the following tables. More detailed descriptive calculations are included in **Attachment 2**.

Table 1. Descriptive Statistics - All Songs With Audio Features

| **Audio Feature** | **Mean** | **Standard Deviation** | **Minimum** | **Quartiles** | | | **Maximum** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **25%** | **50%** | **75%** |
| acousticness | 0.42 | 0.37 | 0.00 | 0.03 | 0.34 | 0.82 | 1.00 |
| danceability | 0.53 | 0.19 | 0.00 | 0.40 | 0.54 | 0.68 | 1.00 |
| duration\_ms | 238,312 | 156,918 | 1,000 | 169,587 | 216,933 | 275,400 | 6,072,187 |
| energy | 0.55 | 0.28 | 0.00 | 0.31 | 0.57 | 0.79 | 1.00 |
| instrumentalness | 0.26 | 0.38 | 0.00 | 0.00 | 0.00 | 0.66 | 1.00 |
| key | 5.2 | 3.5 | 0 | 2 | 5 | 8 | 11 |
| liveness | 0.21 | 0.18 | 0.00 | 0.10 | 0.13 | 0.26 | 1.00 |
| loudness | -11.0 | 6.3 | -60.0 | -13.7 | -9.2 | -6.4 | 7.2 |
| mode | 0.66 | 0.47 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| speechiness | 0.10 | 0.14 | 0.00 | 0.04 | 0.05 | 0.08 | 0.97 |
| tempo | 118.6 | 30.9 | 0 | 95 | 119 | 137 | 250 |
| time\_signature | 3.84 | 0.57 | 0.00 | 4.00 | 4.00 | 4.00 | 5.00 |
| valence | 0.47 | 0.28 | 0.00 | 0.23 | 0.47 | 0.71 | 1.00 |

Table 2. Descriptive Statistics - Billboard Hot 100

| **Audio Feature** | **Mean** | **Standard Deviation** | **Minimum** | **Quartiles** | | | **Maximum** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **25%** | **50%** | **75%** |
| rank | 50.5 | 28.9 | 1 | 26 | 51 | 76 | 100 |
| last-week | 47.6 | 28.1 | 1 | 23 | 47 | 72 | 100 |
| peak-rank | 41.0 | 29.3 | 1 | 13 | 38 | 65 | 100 |
| weeks-on-board | 9.2 | 7.6 | 1 | 4 | 7 | 13 | 90 |
| acousticness | 0.28 | 0.28 | 0.00 | 0.04 | 0.18 | 0.47 | 1.00 |
| danceability | 0.60 | 0.15 | 0.00 | 0.51 | 0.61 | 0.71 | 0.99 |
| duration\_ms | 226,926 | 65,973 | 37,013 | 183,560 | 221,400 | 258,533 | 1,292,293 |
| energy | 0.63 | 0.20 | 0.02 | 0.48 | 0.64 | 0.79 | 1.00 |
| instrumentalness | 0.03 | 0.13 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 |
| key | 5.2 | 3.6 | 0 | 2 | 5 | 8 | 11 |
| liveness | 0.19 | 0.16 | 0.02 | 0.09 | 0.13 | 0.24 | 1.00 |
| loudness | -8.6 | 3.6 | -29.5 | -11.0 | -8.1 | -5.8 | 2.3 |
| mode | 0.73 | 0.44 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| speechiness | 0.06 | 0.07 | 0.00 | 0.03 | 0.04 | 0.06 | 0.94 |
| tempo | 120.4 | 27.9 | 0 | 100 | 119 | 136 | 241 |
| time\_signature | 3.94 | 0.29 | 0.00 | 4.00 | 4.00 | 4.00 | 5.00 |
| valence | 0.61 | 0.24 | 0.00 | 0.42 | 0.63 | 0.81 | 0.99 |

Table 3. Descriptive Statistics - All Songs From Billboard Hot 100 With Audio Features

| **Audio Feature** | **Mean** | **Standard Deviation** | **Minimum** | **Quartiles** | | | **Maximum** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **25%** | **50%** | **75%** |
| acousticness | 0.32 | 0.29 | 0.00 | 0.05 | 0.22 | 0.56 | 1.00 |
| danceability | 0.59 | 0.15 | 0.00 | 0.49 | 0.60 | 0.70 | 0.99 |
| duration\_ms | 217,614 | 67,768 | 37,013 | 169,707 | 210,533 | 251,260 | 1,292,293 |
| energy | 0.61 | 0.20 | 0.02 | 0.46 | 0.63 | 0.78 | 1.00 |
| instrumentalness | 0.04 | 0.14 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 |
| key | 5.2 | 3.6 | 0 | 2 | 5 | 8 | 11 |
| liveness | 0.19 | 0.16 | 0.02 | 0.09 | 0.13 | 0.25 | 1.00 |
| loudness | -8.9 | 3.6 | -29.5 | -11.3 | -8.5 | -6.1 | 2.3 |
| mode | 0.74 | 0.44 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| speechiness | 0.07 | 0.08 | 0.00 | 0.03 | 0.04 | 0.06 | 0.94 |
| tempo | 120.5 | 28.2 | 0 | 100 | 119 | 137 | 241 |
| time\_signature | 3.93 | 0.33 | 0.00 | 4.00 | 4.00 | 4.00 | 5.00 |
| valence | 0.61 | 0.24 | 0.00 | 0.42 | 0.64 | 0.81 | 0.99 |

Table 4. Descriptive Statistics - Songs From Billboard Hot 100 With Audio Features And Genre

| **Audio Feature** | **Mean** | **Standard Deviation** | **Minimum** | **Quartiles** | | | **Maximum** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **25%** | **50%** | **75%** |
| acousticness | 0.31 | 0.29 | 0.00 | 0.05 | 0.22 | 0.54 | 1.00 |
| danceability | 0.59 | 0.15 | 0.00 | 0.49 | 0.60 | 0.70 | 0.99 |
| duration\_ms | 219,137 | 67,456 | 37,013 | 171,881 | 212,027 | 252,237 | 1,292,293 |
| energy | 0.61 | 0.20 | 0.02 | 0.47 | 0.63 | 0.78 | 1.00 |
| instrumentalness | 0.03 | 0.14 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 |
| key | 5.2 | 3.6 | 0 | 2 | 5 | 8 | 11 |
| liveness | 0.19 | 0.16 | 0.02 | 0.09 | 0.13 | 0.25 | 1.00 |
| loudness | -8.8 | 3.6 | -29.0 | -11.3 | -8.4 | -6.0 | -0.4 |
| mode | 0.74 | 0.44 | 0.00 | 0.00 | 1.00 | 1.00 | 1.00 |
| speechiness | 0.07 | 0.08 | 0.00 | 0.03 | 0.04 | 0.06 | 0.94 |
| tempo | 120.6 | 28.2 | 0 | 100 | 119 | 137 | 231 |
| time\_signature | 3.93 | 0.33 | 0.00 | 4.00 | 4.00 | 4.00 | 5.00 |
| valence | 0.61 | 0.24 | 0.00 | 0.42 | 0.63 | 0.81 | 0.99 |

### Audio Feature Descriptions

Audio features available from the Spotify API used in this study are described in detail in the below table (Spotify, n.d.).

Table 5. Description of Audio Features From Spotify API

| **Audio Feature** | **Type** | **Description** | **Min** | **Max** |
| --- | --- | --- | --- | --- |
| acousticness | number  <float> | A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. | 0 | 1 |
| danceability | number  <float> | Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable. | 0 | 1 |
| duration\_ms | integer | The duration of the track in milliseconds. | 0 | N/A |
| energy | number  <float> | Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. | 0 | 1 |
| instrumentalness | number  <float> | Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0. | 0 | 1 |
| key | integer | The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1. | -1 | 11 |
| liveness | number  <float> | Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live. | 0 | 1 |
| loudness | number  <float> | The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db. | -60 | 0 |
| mode | integer | Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0. | 0 | 1 |
| speechiness | number  <float> | Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks. | 0 | 1 |
| tempo | number  <float> | The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. | 0 | N/A |
| time\_signature | integer | An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4". | 3 | 7 |
| valence | number  <float> | A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). | 0 | 1 |

### Histograms

Histograms for each of the audio features are shown in the below figures, comparing songs on the Billboard Hot 100 with all songs in this study.

Histogram

Description automatically generated with low confidence

Figure 1. Acousticness Histogram

Chart, bar chart

Description automatically generated

Figure 2. Danceability Histogram

Chart, bar chart, histogram

Description automatically generated

Figure 3. Duration Histogram

Chart, bar chart

Description automatically generated

Figure 4. Energy Histogram

Chart, bar chart

Description automatically generated

Figure 5. Instrumentalness Histogram

Chart, histogram

Description automatically generated

Figure 6. Key Histogram

Chart, histogram

Description automatically generated

Figure 7. Liveness Histogram

Chart, bar chart

Description automatically generated

Figure 8. Loudness Histogram

A picture containing chart

Description automatically generated

Figure 9. Mode Histogram



Figure 10. Speechiness Histogram

Chart, bar chart

Description automatically generated

Figure 11. Tempo Histogram

Chart, bar chart

Description automatically generated

Figure 12. Time Signature Histogram

Chart

Description automatically generated

Figure 13. Valence Histogram

## Historical Changes in Audio Features

Historical line plots for each of the audio features are shown in the below figures, comparing audio feature averaged by release year for songs on the Billboard Hot 100 compared with all songs in this study.

Chart, line chart

Description automatically generated

Figure 14. Acousticness History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 15. Danceability History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 16. Duration History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 17. Energy History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 18. Instrumentalness History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 19. Liveness History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 20. Loudness History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 21. Mode History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 22. Speechiness History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 23. Tempo History Comparing the Billboard Hot 100 with All Songs

Chart, line chart

Description automatically generated

Figure 24. Valence History Comparing the Billboard Hot 100 with All Songs

The following figure shows the Billboard Hot 100 chart yearly audio feature averages over time. In contrast to the above figures, this figure is grouped by appearance on the charts in a given year, and does not necessarily correspond to the release year for any given song. Also of note, all audio features were normalised to range from 0 to 1 in order to create a consistent plot.

Chart, histogram

Description automatically generated

Figure 25. Billboard Hot 100 Historical Charts

A more detailed analysis of each audio feature over time is included in **Attachment 2**.

## Correlation Analysis

Correlation analysis is outlined in the below figure. A more detailed correlation analysis in included in **Attachment 2**. The analysis investigated which audio features correlate with each other, differences between Billboard Hot 100 songs and songs not on the list, as well as which features correlate with popularity (as defined above).

Chart

Description automatically generated

Figure 26. Correlation Analysis Summary

In the above figure, the “in\_B100” feature denotes whether or not the song appeared on the Billboard Hot 100 charts.

With a few exceptions, the analysis showed a relatively weak correlation between audio features. This is unsurprising because audio features have been created to represent music simply, accurately, and without redundancies. In later stages of this study, these low correlation values imply that all audio features have the potential to be relevant for predictive analytics, and none of these features can simply be dropped as redundant.

Also of note, every audio feature correlated very weakly with popularity (the “in\_B100” feature). As noted in the literature review, this is likely due to large varieties of musical styles all being included in the calculations. Separating analysis into genres yielded stronger correlations for some of the audio features, although correlations were still relatively weak. However, as shown later in this report, clustering into genres prior to classification did not improve classification results. Genres are discussed in the next section, and detailed analysis is included in **Attachment 2**. Classification is discussed later in this report.

Correlation analysis was also conducted strictly on the Billboard Hot 100 dataset. The below figure shows the correlation between audio features and Billboard Hot 100 performance, both peak-rank and weeks-on-board.

Chart

Description automatically generated

Figure 27. Billboard Hot 100 Correlation Analysis Summary

As shown in the above figure, a number of audio features have a small correlation with performance on the Billboard Hot 100. The following figures show the sorted correlation coefficients corresponding to weeks-on-board and peak-rank, respectively. Note that the order of the charts is reversed because lower peak rank is optimal, whereas higher weeks-on-board is optimal. More detailed analysis is included in **Attachment 2**.

Chart

Description automatically generated

Figure 28. Ranked Audio Feature Correlations With Weeks On Billboard Hot 100 Charts

Chart, scatter chart

Description automatically generated

Figure 29. Ranked Audio Feature Correlations With Peak Rank On Billboard Hot 100

## Analysis of Genres

In order to improve correlations and future predictions, genre information has been investigated. Correlation analysis showed stronger correlation between audio features and popularity when restricting analysis to specific genres. However, as shown later in this report, clustering into genres prior to classification did not improve classification results. Detailed genre correlation analysis is included in **Attachment 2**. Classification is discussed later in this report.

Even after excluding all but the most popular genre for each artist, there are still over 5,000 genres in the dataset. Many genres are obscure, and most could more accurately be referred to as sub-genres. For this reason, bins of genres were grouped together to approximate more general genres. Regular Expressions were used to group songs into categories by matching patterns within genre info from the Spotify API. These genre groups were able to reduce variation in audio features to an extent, as shown in the below figure. Details of this genre analysis are included in **Attachment 2**.

Chart, bar chart

Description automatically generated

Figure 30. Comparison of Audio Feature Variations Between Genres

It is important to note that these categories are based on assumptions about which genres belong together, and some of these grouping decisions have been made arbitrarily.

# Project Approach

The central problem in this project is to utilise publicly available audio feature data to predict the popularity of a song. For this project, two main techniques were employed, namely data mining and predictive analytics.

## Outliers

Outliers were investigated using a number of common methods, as well as domain knowledge. The goal of excluding audio features is to exclude all non-musical information from the dataset to attempt to improve the veracity of the analysis. Detailed outlier analysis is included in **Attachment 3**.

## Data Mining

First, data mining and knowledge discovery has been used to explore the data and cluster songs by audio features. Genres and audio feature correlation have been explored in detail. Details are included in the **Data Description** section above, and detailed calculations are included in **Attachment 2**.

For this analysis, K-Means Clustering was employed, as it was the most common clustering algorithm noted in the Literature Review. Details of clustering analysis are included in **Attachment 4.**

## Predictive Analytics

Secondly, predictive analytics was used to build a number of predictive models. These models attempt to predict whether or not a song will be make it onto the Billboard Hot 100 charts using audio features for that song. Some models incorporated the clusters from the previous phase of the analysis as an attempt to improve predictive accuracy.

This analysis investigated a variety of supervised classification algorithms, namely Logistic Regression, Decision Trees, K-Nearest Neighbours, Random Forests, AdaBoost, and Neural Networks. Statistical significance and predictive power were compared between the utilised algorithms to evaluate the best performance. Details of predictive analytics are included in **Attachment 5.**

# Results

## Outliers

In order to exclude non-musical tracks from the dataset, a number of methods were investigated, including inter-quartile range (IQR), statistical score, percentiles, and domain knowledge.

None of the standard outlier exclusion methods worked perfectly in isolation. Since musical tracks may have audio features in nearly any range, outliers were still primarily musical in most cases, and thus not excluded. However, a few audio feature ranges were found to consist primarily of non-musical audio tracks. The audio feature ranges excluded from the dataset are as follows:

* Valence = 0
* Speechiness > 0.947 (top 0.5%)
* Temp = 0
* Loudness < -34.7 dB (bottom 0.5%)
* Danceability < 0.0644 (bottom 0.5%)

Additionally, domain knowledge was used to exclude non-musical tracks. In order to determine non-musical tracks, genres were inspected for category names and added to a list of genres to exclude. These excluded genres included categories such as “sleep,” “sound effects,” and “ringtone.” A full list of excluded genres is included in **Attachment 3**.

In addition, music which may not be considered to be “songs” has been excluded (e.g., entire performances, commercials, tv show intros, etc.). Based on domain knowledge, audio tracks less than 1 minute or greater than 10 minutes in length have been excluded.

Detailed outlier analysis is included in **Attachment 3**.

## Clustering

The clustering analysis consisted of the following steps:

1. Remove outliers
2. Drop columns not to be included in the clustering
3. Transform data so that all audio features range from 0 to 1
4. Cluster using the K-Means model, optimising K for silhouette score
5. Cluster using the K-Means model, using K equal to the same number of categories as used in the manual genre groupings as defined in **Attachment 2** and **Attachment 4**
6. Cluster data manually using the genre groupings as defined in **Attachment 2** and **Attachment 4**

Clustered data was saved for use in the classification analysis. Details of clustering analysis are included in **Attachment 4.**

## Classification

The following machine learning models have been considered in the classification portion of this assessment:

* Logistic Regression
* Decision Trees
* K-Nearest Neighbours
* Random Forests
* AdaBoost

Five-fold cross-validation was used to evaluate the effectiveness of these models. Consistent random seeds were used for the stratified folds and random undersampling in order to have consistency between tests. This enabled a direct matched comparison of out-of-fold predictions between modelling scenarios.

In addition to the above-noted models, neural networks as implemented in TensorFlow were included in the preliminary analysis, but not in the detailed finalised modelling. Details of preliminary classification using neural networks are included in the GitHub repository, but have been excluded from the results and conclusions of this report for simplicity.

Also initially considered but discarded, Support Vector Machine (SVM) models were not feasible do to the extremely large dataset. Some combination of undersampling and PCA may allow for the use of SVM models, but these methods were excluded from this report for simplicity and due to time constraints.

Since the dataset is highly unbalanced, undersampling and oversampling were considered to improve results. Both oversampling and undersampling were shown to improve precision and recall significantly versus the unbalanced calculations. However, undersampling was found to calculate significantly faster without sacrificing precision or recall. Therefore, oversampling was not used for final modelling scenarios. The same random seed was used for all undersampling to assure that all out-of-fold predictions were made on the identical partitions of the dataset, and consistency between modelling scenarios was maintained.

In order to improve results, a large range of hyperparameters was considered for each of the models. A grid search was utilised to find the optimal parameters for each of the models in order to optimise ROC AUC (receiver operating characteristic area under the curve). Due to time constraints, only Logistic Regression and Decision Trees were tuned for the entire dataset. However, limited selections of the dataset were considered for each of the remaining models. Details of this partial dataset tuning are included in the GitHub repository, but have been excluded from the results and conclusions of this report.

Predictions were made with and without clustering the data in order to attempt to improve results. Each of the clusters described above was considered in the classification analysis.

Although a large number of permutations of model, cluster, tuning parameters, and sampling techniques were possible, this assessment was limited to the following scenarios for simplicity and due to time constraints:

1. Logistic Regression – Default Hyperparameters
2. Decision Tree – Default Hyperparameters
3. K-Nearest Neighbours – Default Hyperparameters
4. Random Forest – Default Hyperparameters
5. AdaBoost – Default Hyperparameters
6. Logistic Regression – Tuned Hyperparameters
7. Decision Tree – Tuned Hyperparameters
8. Logistic Regression – Clustered By K-Means Version 1
9. Logistic Regression – Clustered By K-Means Version 2
10. Logistic Regression – Clustered By Genre

For each scenario, the classification analysis consisted of the following steps:

1. Split the data into 5 stratified folds, using a consistent random seed for consistency between tests
2. Train the model using 4 of the folds, balanced using undersampling with consistent random seed for consistency between tests
3. Predict the class for the entire, unbalanced test fold
4. Repeat for each fold, populating out-of-fold predictions into a dataframe including all predictions for cross-validation evaluation

In addition to the analysis outlined above, rough calculations were completed to investigate each of the following topics in greater detail. These extra analysis notebooks are included in the GitHub repository, entitled as follows:

* NOTEBOOK 5A. Choosing Classification Models
* NOTEBOOK 5B. Tuning Classification Models
* NOTEBOOK 5C. Neural Network
* NOTEBOOK 5D. Tuning Classification Models using Spotify popularity

The results of these analyses informed the finalised analysis included in **Attachment 5** (“NOTEBOOK 5. Classification Calculations and Results”), but the results of these rough calculations were not utilised directly in the results and conclusions of this study. They were therefore only included in the repository, and not referenced or included in detail within this report.

Details of predictive analytics and cross-validation results are included in **Attachment 5**, with additional calculations included in the GitHub repository for this project.

## Comparison of Results – Effectiveness

Out-of-fold predictions were calculated for each scenario as described above. Each scenario was compared using visualisation and statistical analysis. Since all stratified folds and undersampling utilised consistent random seeds, the results are matched for statistical consistency between tests.

Detailed calculations and visualisations of results are included in **Attachment 6.**

After employing the classification models on clustered and un-clustered data, classification success was mixed. Since the dataset is highly unbalanced, achieving high accuracy was trivial. However, optimising for more nuanced metrics such as precision, recall, or F1-score has proven difficult. Based on visual inspection and statistical analysis, the predictions appear to be correct, albeit not as useful anticipated. This is due to the fact that a large number of songs exist with audio features consistent with hit songs, but only a small portion of all songs become hits. More details are provided below.

### Visualisation of Results Using Principal Component Analysis

In order to visualise the distribution of songs within 11-dimensional audio feature space, the dimensionality of the dataset was reduced using Principal Component Analysis (PCA). In this way, a scatterplots can be used to visualise songs within audio feature space. As a result, it is possible to visually compare songs on the Billboard Hot 100, songs not on the Billboard Hot 100, and songs predicted to be popular.

The figures below show scatterplots of the first two principal components for each prediction scenario outlined above.

Chart

Description automatically generated

Figure 31. Predicted vs Actual Hits: Logistic Regression - Default Hyperparameters

Chart

Description automatically generated

Figure 32. Predicted vs Actual Hits: Decision Tree - Default Hyperparameters

Chart

Description automatically generated

Figure 33. Predicted vs Actual Hits: K-Nearest Neighbours - Default Hyperparameters

Chart

Description automatically generated

Figure 34. Predicted vs Actual Hits: Random Forest - Default Hyperparameters

Chart

Description automatically generated

Figure 35. Predicted vs Actual Hits: AdaBoost - Default Hyperparameters

Chart

Description automatically generated

Figure 36. Predicted vs Actual Hits: Logistic Regression - Tuned Hyperparameters

A picture containing chart

Description automatically generated

Figure 37. Predicted vs Actual Hits: Decision Tree - Tuned Hyperparameters

Chart

Description automatically generated

Figure 38. Predicted vs Actual Hits: Logistic Regression - Clustered By K-Means Version 1

Chart

Description automatically generated

Figure 39. Predicted vs Actual Hits: Logistic Regression - Clustered By KMeans Version 2

Chart

Description automatically generated

Figure 40. Predicted vs Actual Hits: Logistic Regression - Clustered By Genre

As shown in the figures above, most models performed fairly well at classifying songs into audio feature ranges consistent with Billboard Hot 100 hits. However, some of these modelling scenarios included audio features that deviated more than one would expect based on visual inspection of the PCA scatterplots. Notably, clustering by genre appears to have created results significantly worse than the un-clustered scenarios.

In addition to PCA scatterplots, histograms have been generated for each audio feature, as well as the first 2 principal components to visually inspect predicted results. For each of the histograms, the “Predicted Popular” columns have been subdivided into modelling scenarios ordered as outlined above. To avoid clutter, labels for these series have been excluded from the figures.

The figures below show histograms for the first two principal components comparing songs on the Billboard Hot 100, songs not on the Billboard Hot 100, and songs predicted to be popular.

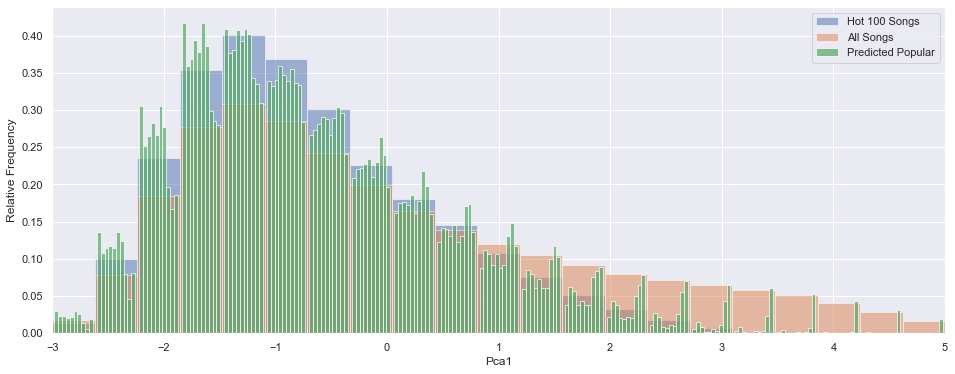


Figure 41. Comparing Hot 100 Songs with Predicted Popularity – First Principal Component

Chart, histogram

Description automatically generated

Figure 42. Comparing Hot 100 Songs with Predicted Popularity – Second Principal Component

Similar to the PCA scatterplots, these histograms show predictions relatively consistent with audio features as distributed within the Billboard Hot 100. Also similar to the scatterplots, some predictions deviate more than others, most notably the scenario clustered by genre.

In addition to the above histograms, similar figures for each audio feature are included in **Attachment 6**.

### Statistical Analysis

In addition to the visual inspection outlined above, statistical analysis was performed to evaluate the performance of the models and to assess which model performed the best.

Since results for each scenario have similar recall and precision measures, with and without hyperparameter tuning or clustering, it was unknown whether these models were statistically distinct. To assess whether all model were statistically equivalent, a Friedman Test was performed on all matched predictions. The null hypothesis, that all models are statistically equivalent, was rejected to any arbitrary degree of certainty; the p-value rounds to exactly zero, meaning it is many orders of magnitude below zero.

Since the null hypothesis for the Friedman Test was rejected, at least one of the models is statistically different from the others, and a more detailed statistical test was required. The Wilcoxon Signed Rank Test was performed to compare each model with each other model. Similar to above, the results of these tests indicate that all models are statistically distinct to any arbitrary degree of certainty. Therefore the null hypothesis was rejected for every test. Each of the modelling scenarios is statistically distinct.

Details of statistical analysis are included in **Attachment 6.**

### Ranking of Models

Based on the results of the modelling, the models were evaluated for performance. Multiple metrics were considered when evaluating the performance of the models. The most applicable metric for this study is likely the F1-Score, because it incorporates both Recall and Precision. Scoring metrics for each modelling scenario are presented in the below table.

Table 6. Model Performance By Scenario - Scores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelling Scenario** | **Scoring Metric** | | | |
| **Precision** | **Recall** | **F1 Score** | **Accuracy** |
| Logistic Regression - Default Hyperparameters | 0.004 | 0.802 | 0.009 | 0.569 |
| Decision Tree - Default Hyperparameters | 0.004 | 0.647 | 0.009 | 0.653 |
| K-Nearest Neighbours - Default Hyperparameters | 0.005 | 0.762 | 0.009 | 0.612 |
| Random Forest - Default Hyperparameters | 0.006 | 0.794 | 0.012 | 0.678 |
| AdaBoost - Default Hyperparameters | 0.005 | 0.797 | 0.01 | 0.61 |
| Logistic Regression - Tuned Hyperparameters | 0.004 | 0.802 | 0.009 | 0.57 |
| Decision Tree - Tuned Hyperparameters | 0.005 | 0.757 | 0.01 | 0.65 |
| Logistic Regression - Clustered By K-Means Version 1 | 0.004 | 0.681 | 0.009 | 0.627 |
| Logistic Regression - Clustered By K-Means Version 2 | 0.004 | 0.694 | 0.009 | 0.618 |
| Logistic Regression - Clustered By Genre | 0.002 | 0.125 | 0.005 | 0.872 |

Based on the scores in the above table, the models are ranked as shown in the below table.

Table 7. Model Performance By Scenario - Ranks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modelling Scenario** | **Rank** | | | |
| **Precision** | **Recall** | **F1 Score** | **Accuracy** |
| Logistic Regression - Default Hyperparameters | 5 | 1 | 4 | 10 |
| Decision Tree - Default Hyperparameters | 5 | 9 | 4 | 3 |
| K-Nearest Neighbours - Default Hyperparameters | 2 | 5 | 4 | 7 |
| Random Forest - Default Hyperparameters | 1 | 4 | 1 | 2 |
| AdaBoost - Default Hyperparameters | 2 | 3 | 2 | 8 |
| Logistic Regression - Tuned Hyperparameters | 5 | 1 | 4 | 9 |
| Decision Tree - Tuned Hyperparameters | 2 | 6 | 2 | 4 |
| Logistic Regression - Clustered By K-Means Version 1 | 5 | 8 | 4 | 5 |
| Logistic Regression - Clustered By K-Means Version 2 | 5 | 7 | 4 | 6 |
| Logistic Regression - Clustered By Genre | 10 | 10 | 10 | 1 |

As shown in the above tables, the Random Forest generally outperformed the other models. This is consistent with the visual inspection of the results shown previously.

Interesting to note, the clustering did not improve any of the models, and clustering by genre lead to the worst results. This is an unexpected result based on the Literature Review.

## Comparison of Results – Efficiency

Unfortunately, the Random Forest model has many tuning parameters and calculates relatively slowly, so could take on the order of days or weeks to completely tune the hyperparameters, assuming ideal conditions. Therefore, an optimised Random Forest model was not tested. It is anticipated that this model would perform better than the untuned model. This optimised Random Forest model is outside the scope of this project, but is discussed in the Future Work section below.

Another time limitation involved oversampling versus undersampling. Oversampling was significantly more time consuming due to the large size of the dataset. Incorporating multiple model runs to perform hyperparameter tuning further increased time to calculate, since the model needs to be re-tested for each set of hyperparameters. The following table outlines the approximate times required to calculate each model using the oversampled and undersampled data, with and without hyperparameter tuning.

Table 8. Efficiency By Machine Learning Model - Training

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Machine Learning Model** | **Approximate Time to Complete Training** | | | |
| **Single Scenario Undersampled** | **Single Scenario Oversampled** | **Tune Hyperparameters Undersampled** | **Tune Hyperparameters Oversampled** |
| Logistic Regression | 28 seconds | 50 minutes [1] | 61 minutes | 4 days [1] |
| Decision Trees | 27 seconds | 50 minutes [1] | 67 minutes | 5 days [1] |
| K-Nearest Neighbours | 18 minutes | 30 hours [1] | 60 hours [1] | 300 days [1] |
| Random Forest | 3.1 minutes | 5 hours [1] | 90 hours [1] | 400 days [1] |
| AdaBoost | 1.3 minutes | 2 hours [1] | > 15 hours [2] | 80 days [1] |

Notes: [1] Estimated based on known factors, likely to be significantly longer based on Note [2], which was estimated to be significantly faster. Models may vary significantly based on time complexity, these estimates should be considered to be very approximate.

[2] Run did not complete, but was aborted after approximately 15 hours of calculation.

As shown in the above table, Logistic Regression and Decision Trees are the fastest models. The remaining models were found to be too time intensive to be fully evaluated in this study.

In order to assess the efficiency of the models, prediction times for fully trained models was also assessed. The below table shows the amount of time each model took to perform

Table 9. Efficiency By Machine Learning Model - Predictions

|  |  |  |
| --- | --- | --- |
| **Machine Learning Model** | **Time to Prediction Entire Dataset** | **Approximate Predictions Per Millisecond** |
| Logistic Regression | 0.81 seconds | 11,000 |
| Decision Trees | 1.6 seconds | 5,400 |
| K-Nearest Neighbours | 18 minutes | 8 |
| Random Forest | 2.2 minutes | 68 |
| AdaBoost | 0.83 minutes | 180 |

As shown in the above table, Logistic Regression is the fastest model. However, for singular predictions or small batches of data, any of the models may be adequate.

Due to limitations discussed above, fully tuned models were not considered for this evaluation. It is possible that fully tuned models could perform more or less quickly than default hyperparameter models.

## Comparison of Results – Stability

As shown in the above comparison of results, predictions are relatively stable, even using different machine learning algorithms. Although there exists room for improvement, all models provide predictions consistent with the known class data, with the possible exception of clustered data. Statistical analysis and visual inspection confirm these observations.

Although an optimised Random Forest or Neural Network may perform best, a Logistic Regression model performs nearly as well in most cases. More testing would be required to determine an optimal model. Additionally, methods outlined in the Future Work section below should be considered before choosing an optimal model.

# Discussion and Conclusions

## Discussion and Limitations

Although results are not as precise as desired, the models appear to be functioning correctly. Based on the nature of the problem, it appears that there may be no way to avoid a large percentage of false positives. There is likely too much variance in each of the audio features to predict popularity with any precision, even after clustering or grouping by genre.

Looking at the PCA scatterplots and histograms of predictions, it appears that predictions are lined up well with actual popular songs. Although there is noticeable room for improvement, the main issue with precision involves popular songs taking up a large portion of 2-dimensional audio feature space (and presumably also the full dimensional audio feature space). Therefore, even with a perfectly optimised model, we would still expect low precision due to the highly unbalanced data, which is inherent to the nature of popularity.

## Conclusions

This study had the initial goal of predicting music popularity. Although predicting popularity has proven difficult, it is possible to quantify features that are characteristic of most popular songs. These features may not be sufficient to get a song onto the Billboard Hot 100, but in most cases, a range of audio features does appear to be necessary in order to have a chance of achieving popularity.

# Future Work

Based on the findings of this study, a number of potential future areas of investigation are possible. A few interesting options are listed below.

Due to time constraints, an optimised Random Forest model was not assessed. It would be interesting to assess the effectiveness of this model, potentially in comparison to or conjunction with the other future work outlined in this section.

In terms of prediction, rather than predicting popularity, future models may be used to predict whether or not commercial success is possible. By focussing on whether or not the song’s audio features lie within the optimal range for popular music, more accurate and useful predictions could be possible.

The use of PCA was utilised in this study for visualisation of higher dimensional data. However, some sources have noted improvements in predictions using PCA as part of the classification pipeline. It would be interesting to evaluate the performance of these models with and without PCA. PCA could also be used to reduce complexity and speed up slow models like the Random Forest and AdaBoost models, and may allow for the use of SVM models.

In addition, PCA has the potential to be used analytically by fitting a system of inequalities to the lower-dimension PCA space in order to back-calculate potential audio feature ranges for popular songs. This study could involve using the 2-dimensional PCA plots to fit an ellipse or two curves, within/between which popularity is more likely. Using the back-calculated inequalities, a system of equations could be used to describe the potential audio features which are most likely to achieve popularity. This method has the potential to be quicker and more intuitive than machine learning methods, and could potentially lead to higher precision and recall.

Once a useful and streamlined model is available for predicting a song’s popularity potential, it would be useful to develop a plugin for musical software. If this was achieved, it would not be necessary to upload music to Spotify before checking the popularity potential of a song. This would require extra steps, most notably the calculation of audio features from raw audio data. It is unknown whether this is possible, due to the proprietary nature of the Spotify API and algorithms. More refined versions of this plugin could even offer advice to help achieve audio feature ranges with improved potential for popularity.

# References

Araujo, C. V. S., Cristo, M. A. P., & Giusti, R. (2007). Predicting Music Popularity on Streaming Platforms. ANAIS DO SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO MUSICAL (SBCM 2019).

Billboard Hot 100. (2022, September 6). Wikipedia. <https://en.wikipedia.org/w/index.php?title=Billboard_Hot_100&oldid=1108834581>

Cataltepe, Z., Yaslan, Y., & Sonmez, A. (2007). Music Genre Classification Using MIDI and Audio Features. EURASIP JOURNAL ON ADVANCES IN SIGNAL PROCESSING.

Chen, Y. C., Chen, Z. C., & Hsia, C. H. (2021). Music Mood Classification System for Streaming Platform Analysis Via Deep Learning Based Feature Extraction. 2021 IEEE INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS-TAIWAN (ICCE-TW).

Cilibrasi, R. L., Vitányi, P., & Wolf, R. D. (2004). Algorithmic clustering of music. Proceedings of the Fourth International Conference onWeb Delivering of Music, 2004. EDELMUSIC 2004..

Dhruvil Dave. (2021, November 9). Billboard "The Hot 100" Songs [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DS/1211465>

Elicit. (n.d.). <https://elicit.org/>

Febirautami, L. R., Surjandari, I., & Laoh, E. (2018). Determining Characteristics of Popular Local Songs in Indonesia's Music Market. 2018 5TH INTERNATIONAL CONFERENCE ON INFORMATION SCIENCE AND CONTROL ENGINEERING (ICISCE).

Gao, A. (2021). Catching the Earworm: Understanding Streaming Music Popularity Using Machine Learning Models. E3S Web of Conferences 253, 03024

Google Dataset Search. (n.d.). <https://datasetsearch.research.google.com/>

Google Scholar. (n.d.). <https://scholar.google.com/>

Honingh, A. K., & Bod, R. (2011). Clustering and Classification of Music by Interval Categories. MCM.

Huo, Y. (2021). Music Personalized Label Clustering and Recommendation Visualization. Complex..

Jia, X. (2022). A Music Emotion Classification Model Based on The Improved Convolutional Neural Network. COMPUTATIONAL INTELLIGENCE AND NEUROSCIENCE.

Kim, J. H. (2021). Music Popularity Prediction Through Data analysis of Music’s Characteristics. International Journal of Science, Technology and Society.

Kim, S., Park, J., Seong, K., Cho, N., Min, J., & Hong, H. (2021). Music-Circles: Can Music Be Represented With Numbers?. ARXIV.

Laurier, C. , Lartillot, O. , Eerola, T. , & Toiviainen, P. (2009). Exploring relationships between audio features and emotion in music.

Lee, J. , & Lee, J. S. (2018). Music Popularity: Metrics, Characteristics, and Audio-Based Prediction. IEEE Transactions on Multimedia.

Li, L. (2021). Learning Recommendation Algorithm Based on Improved BP Neural Network in Music Marketing Strategy. COMPUTATIONAL INTELLIGENCE AND NEUROSCIENCE.

Li, Q., Kim, B. M., Guan, D. H., & Oh, D. W. (2004). A Music Recommender Based On Audio Features. SIGIR.

Li, Q., Myaeng, S. H., & Kim, B. M. (2007). A Probabilistic Music Recommender Considering User Opinions and Audio Features. INF. PROCESS. MANAG..

Li, X., & Li, J (2022). Music Classification Method Using Big Data Feature Extraction and Neural Networks. JOURNAL OF ENVIRONMENTAL AND PUBLIC HEALTH.

Malte Grosse. (2022, March 23). 8+ M. Spotify Tracks, Genre, Audio Features [Data set]. Kaggle. <https://www.kaggle.com/datasets/maltegrosse/8-m-spotify-tracks-genre-audio-features/>

Martín-Gutiérrez, D. , Peñaloza, G. H., Belmonte-Hernández, A. , & García, F Á. (2020). A Multimodal End-to-End Deep Learning Architecture for Music Popularity Prediction. IEEE ACCESS.

O'Toole, K., & Horvát, E. Á. (2022). Novelty and Cultural Evolution in Modern Popular Music. ARXIV.

Paper Digest. (n.d.). <https://www.paperdigest.org/>

Reiman, M., & Örnell, P. (2018). Predicting Hit Songs with Machine Learning. EXAMENSARBETE INOM TEKNIK, GRUNDNIVÅ, 15 HP.

Rodolfo Figueroa. (2020, December 22). Spotify 1.2M+ Songs [Data set]. Kaggle. <https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs>

Schedl, M. (2013). Ameliorating Music Recommendation: Integrating Music Content, Music Context, and User Context for Improved Music Retrieval and Recommendation.

Setiadi, D. R. I. M., Rahardwika, D. S. , Rachmawanto, E. H. , Sari, C. A. , Susanto, A., Mulyono, I. U. W. , Astuti, E. Z. , & Fahmi, A. (2020). Effect of Feature Selection on The Accuracy of Music Genre Classification Using SVM Classifier. 2020 INTERNATIONAL SEMINAR ON APPLICATION FOR TECHNOLOGY OF INFORMATION AND COMMUNICATION (ISEMANTIC).

Shi, J. (2021). Music Recommendation Algorithm Based on Multidimensional Time-Series Model Analysis. COMPLEX..

Spotify for Developers. (n.d.). <https://developer.spotify.com/documentation/web-api/>

West, K. (2008). Novel Techniques for Audio Music Classification and Search. ACM SIGMULTIMEDIA RECORDS.

Wilkes, B., Vatolkin, I., & Müller, H. (2021). Statistical and Visual Analysis of Audio, Text, and Image Features for Multi-Modal Music Genre Recognition. ENTROPY (BASEL, SWITZERLAND).

Xu, Y., & Xu, S. (2021). A Clustering Analysis Method for Massive Music Data.

Yang, L. C., Chou, S. Y., Liu, J. Y., Yang, Y. H., & Chen, Y. (2017). Revisiting the problem of audio-based hit song prediction using convolutional neural networks. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

# Attachments