

**TECHNICAL PROJECT REPORT**

**MUSIC POPULARITY PREDICTION**

GROUP 9

GROUP MEMBERS:

Arush Rao

Rohit Yadav

Moneel Kothare

Jenny Jiang

**CONTENTS**

[**1. EXECUTIVE SUMMARY** 3](#_Toc39676235)

[**2. BACKGROUND / CONTEXT** 4](#_Toc39676236)

[a. Domain 4](#_Toc39676237)

[b. Brief description of the scenario 4](#_Toc39676238)

[c. Decision(s) of interest 4](#_Toc39676239)

[d. Decision maker(s) 4](#_Toc39676240)

[**3. BUSINESS UNDERSTANDING** 5](#_Toc39676241)

[a. Business Objective 5](#_Toc39676242)

[b. Situation assessment 5](#_Toc39676243)

[c. Data Mining goals 5](#_Toc39676244)

[**4. DATA UNDERSTANDING** 7](#_Toc39676245)

[a. Data requirements 7](#_Toc39676246)

[b. Describe data 7](#_Toc39676247)

[c. Sources: 9](#_Toc39676248)

[d. Quality 9](#_Toc39676249)

[**5. DATA PREPARATION** 10](#_Toc39676250)

[a. Data Selection 10](#_Toc39676251)

[b. Data Cleaning 11](#_Toc39676252)

[c. Prepare Data 11](#_Toc39676253)

[**6. MODELLING – BUILDING DECISION SUPPORT MODELS** 13](#_Toc39676254)

[a. Describe Data in Detail 13](#_Toc39676255)

[b. What type of decision-making model(s)is appropriate for the decision‐making tasks? 17](#_Toc39676256)

[c. Provide rationale for choice of model(s) 17](#_Toc39676257)

[d. Detail model development and output 21](#_Toc39676258)

[**7. DSM EVALUATION** 24](#_Toc39676259)

[**8. DISCUSSION** 26](#_Toc39676260)

[a. DSM recommendations 26](#_Toc39676261)

[b. DSM Limitations 26](#_Toc39676262)

[c. Influences for Decisions 26](#_Toc39676263)

[d. Enhancements/Future Work 26](#_Toc39676264)

# **1. EXECUTIVE SUMMARY**

Spotify has become one of the top streaming providers in the Music industry. It has become considerably difficult for the management to decide what record labels and artists should be offered a contract and the kind of music or artists that should be promoted via paid advertisements and whether their music should be part of their free or premium service.

The goal of this project was to predict whether a song is popular or not based on the song attributes(Metadata) and genre that Spotify provides. We ran an analysis on about approximately 6000 hit and non-hit songs to help understand what makes a song popular and if we could predict whether it will be on the Top 100 Billboard or not.

After exploring Machine Learning and Data mining algorithms throughout this course, we chose to run Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Tree algorithms on our data set. We found that Logistic Regression yielded the highest accuracy of 80.54% on the validation set in predicting whether a song will be on the Top 100 billboard or not.

# **2. BACKGROUND / CONTEXT**

## a. Domain

The domain for our project is the Music and Entertainment Industry. We take a look at the insights from Spotify, an international media services provider, whose primary business is to provide an audio [streaming](https://en.wikipedia.org/wiki/Streaming_media) platform that provides [DRM](https://en.wikipedia.org/wiki/Digital_rights_management)-protected music, videos and [podcasts](https://en.wikipedia.org/wiki/Podcast) from record labels and media companies. As a [freemium](https://en.wikipedia.org/wiki/Freemium) service, basic features are free with advertisements or automatic music videos, while additional features, such as offline listening and commercial-free listening, are offered via paid subscriptions.

## b. Brief description of the scenario

It is difficult for companies in the music industry to decide what artists/labels they should sign on and what kind of music they should onboard to the platform. We aim to help in contributing to the decision-making process by finding patterns in the music data over the last decade.

## c. Decision(s) of interest

Key decisions such as ‘Investing in an artist/song/company’ and ‘Promoting producing of a certain type of music’. Based on this, what type of music should be made free and what music should be part of the premium service, would be the decisions of interest here.

## d. Decision maker(s)

There will be a couple of decision makers in this case:

1. Spotify, i.e. the business itself which would use this model to promote songs of a popular/trending genre or regulate the streaming of less popular genres. This would also help them sign deals with record labels focusing on genres which are currently in demand, based on the analysis.
2. The second decision maker would be the artists or music producers themselves who are associated with Spotify. This would be key in determining the kind of music they produce and the artists from a genre with which they collaborate.



# **3. BUSINESS UNDERSTANDING**

## a. Business Objective

The business objective is to assess music streaming patterns to improve business decisions mentioned before, by carrying out accurate predictions using insights received from the data being analyzed. Here the focus will be on two areas of interest namely, the attributes/features of a non-popular song as compared to the attributes/features of a popular song and the correlation attributes/features. The approach towards achieving the business objectives is explained as follows:

1. Does Genre matter? Do the listeners really care about the genre? We intend to analyze user behavior and streaming choices and whether a genre really determines how popular a song is on the billboard and how often a song from the same genre appears there.
2. What features(metadata) contribute to a song being popular? We aim on comparing the songs over the years, based on certain variables and factors such as common audio features which include but are not limited to loudness, speech, energy, acoustics, etc. along with the artist and the genre to which the song belongs. The orderly data gathered using these factors will help guide us in our analysis for determining/predicting the popularity of a song.
3. Could we really predict the popularity of a song based on its attributes provided by Spotify?

## b. Situation assessment

For Online streaming services like Spotify one of the major ways of generating revenue is to convert the free user base into subscribers. By having accurate data of a large number of songs over the decade, Spotify would like to have the knowledge on a song or artist to promote/advertise the most thereby pushing the most preferred songs as suggestions to its users. Also, Spotify can make contractual decisions with different record labels by analyzing the trends in the track selection and number of times a song is played/streamed.

## c. Data Mining goals

Once we explore the data in depth and clean it based on our business objective, we would move on to achieving a model for predicting popularity. We start off by creating a split on our initial data for training and validation. We intend to have a 70-30 split, where 70% of the data would be used as our training data and the remaining 30% would be used for validation. The initial and most basic form of analysis would be to classify the data based on the outcome of it being popular or not based on its presence in the Top 100.

We should then be able to form clusters where each cluster defines a category of songs either by popularity, genre or a defining metadata characteristic thereby enabling us to analyze songs category belonging to a cluster and examining individual tracks belonging to a single cluster. The analysis carried out should help determine which would be the trending category and why. Here the ranking of each category would be done, by the number of top trending songs falling into that genre/category. The information gained here would thus clarify what type of category the focus should be, when it comes to carrying out predictive analysis.

Based on the information obtained about the metadata features of popular songs and the outcome as popular(1) and not-popular(0), we would perform logistic regression to predict the probability of the popularity of a song or at least to understand what would be the important metadata features that contribute to making a song popular. It would be crucial to determine the strongest predictors amongst the variables and then regress to find a fitting model.

# **4. DATA UNDERSTANDING**

## a. Data requirements

We would require a large dataset with popular and non-popular songs since we intend to analyze trends for the entire decade. This would mean gathering data from every year’s billboard (Top 1-100) on Spotify. A comprehensive dataset from the streaming platform that highlights the tracks that have been on the billboard and gives us insights on the popularity of a song/genre, features and attributes (metadata) of the tracks, album details and other details which might help us in establishing correlations and help us view patterns in these varied tracks.

|  |  |
| --- | --- |
| **Business Objective** | **Data Requirement** |
| Could we really predict the popularity of a song? | Existing song data classified as popular or non-popular |
| Does Genre matter | Genre data for the corresponding song data |
| What features(metadata) contribute to a song being popular? | Feature Metadata for the corresponding song data |

## b. Describe data

The dataset we are using is a raw dataset of the top songs of the decade starting from 2010 to 2019 on Spotify’s billboard. It has a collection of metadata features for each song such as Track name, Track ID, popularity, Acoustic-ness, Danceability, Duration(MS) , Energy, Instrumental-ness, Key, Liveness, Loudness, Mode, Speech-ness, Tempo etc. It also gives us an overview about the track/song such as the year it was produced and released, the artist or artist(s) if at all there was a collaboration and the genre it was classified under on Spotify. We do not consider a single artist to belong to the same genre as there are artists who explore genres and we would like to respect the fact that an artist should be free to produce music in the genre they desire to.

|  |  |  |
| --- | --- | --- |
| **Key** | **Value Type** | **Value Description** |
| **acousticness** | **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. |
| **danceability** | **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. |
| **duration\_ms** | **int** | The duration of the track in milliseconds. |
| **energy** | **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. |
| **instrumentalness** | **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. |
| **key** | **int** | [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.](https://en.wikipedia.org/wiki/Pitch_class) |
| **liveness** | **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. |
| **loudness** | **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 typical range between -60 and 0 db. |
| **mode** | **int** | 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. |
| **speechiness** | **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. |
| **tempo** | **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. |
| **time\_signature** | **int** | An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). |
| **type** | **string** | The object type: “audio\_features” |
| **uri** | **string** | The Spotify URI for the track. |
| **valence** | **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). |
| **Track** | **text** | Name of the track |
| **Artist** | **text** | Artist Name of the corresponding track |
| **Top 100** | **Int** | Whether the track has ever been on the Top 100. 1- YES, 2 – NO. |

## c. Sources:

There were a lot of datasets available online on:

1. [www.kaggle.com](http://www.kaggle.com)
2. [www.github.com](http://www.github.com)
3. [www.reddit.com](http://www.reddit.com)

that give us an overview about the billboard details. They are several music platforms which offer the same music. However, for the purpose of this project, since we consider doing an analysis on the Spotify billboard, the objective was to find a dataset that incorporates all the songs that have been on the billboard in the past decade. We explored sources such as Reddit, Kaggle, GitHub and finally decided to choose a dataset from Kaggle which was the closest we could get for finding our desired dataset which contained all the attributed mentioned in the data description section. Although there are also other datasets which give us a more detailed analysis on the music data from every month in each year, our objective was to look for data that gave us popularity insights on the overall year as compared to an individual month in a year. Our final data set was made using a combination of these two datasets.

<https://www.kaggle.com/abdulmeral/spotify-deep-learning-in-10-steps>

https://www.kaggle.com/leonardopena/top-spotify-songs-from-20102019-by-year

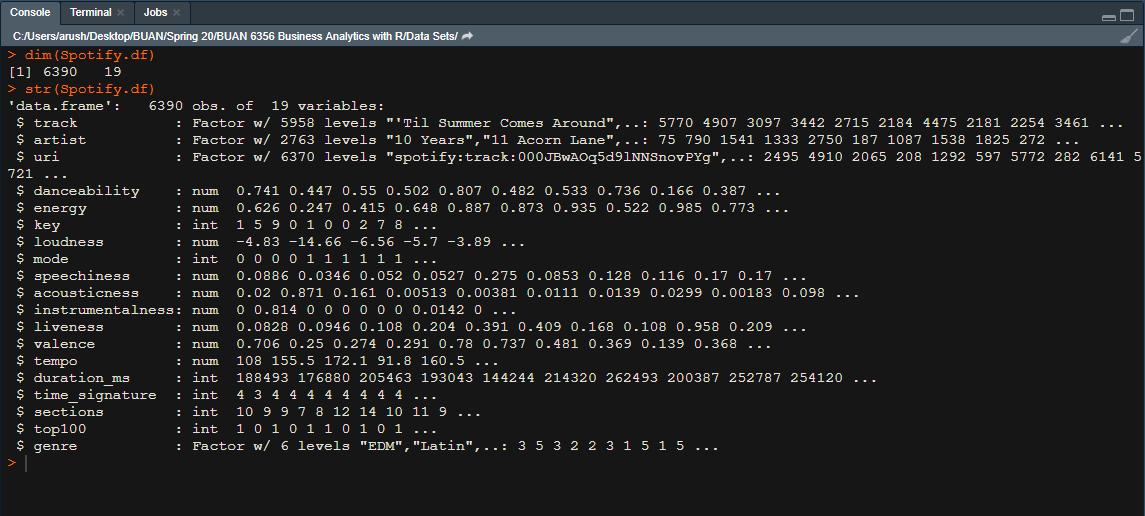
## d. Quality

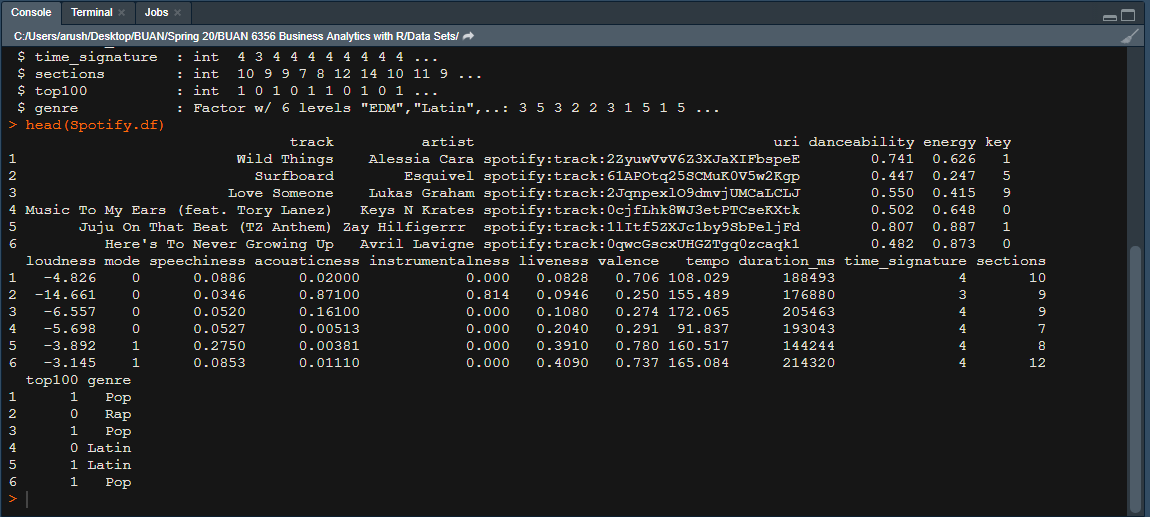
The dataset we obtained, was originally fetched from Spotify using a REST API that is available to the public. This API can be used to extract the desired data from Spotify using a valid token. Metadata features such as Tempo which can be translated to BPM(Beats per minute), Danceability, energy etc. were indeed the values that Spotify assigned to these tracks. The music data and its corresponding metadata is highly accurate as it is directly from the business source. The data overall, is also enough considering the analysis we would like to do and the results that we’d like to see. The consistency between the data we have on hand and the areas of interest that we’d like to explore have a strong correlation too. The data relates to our business requirements and also includes all the predictors we are looking for to predict the popularity.

# **5. DATA PREPARATION**

## a. Data Selection

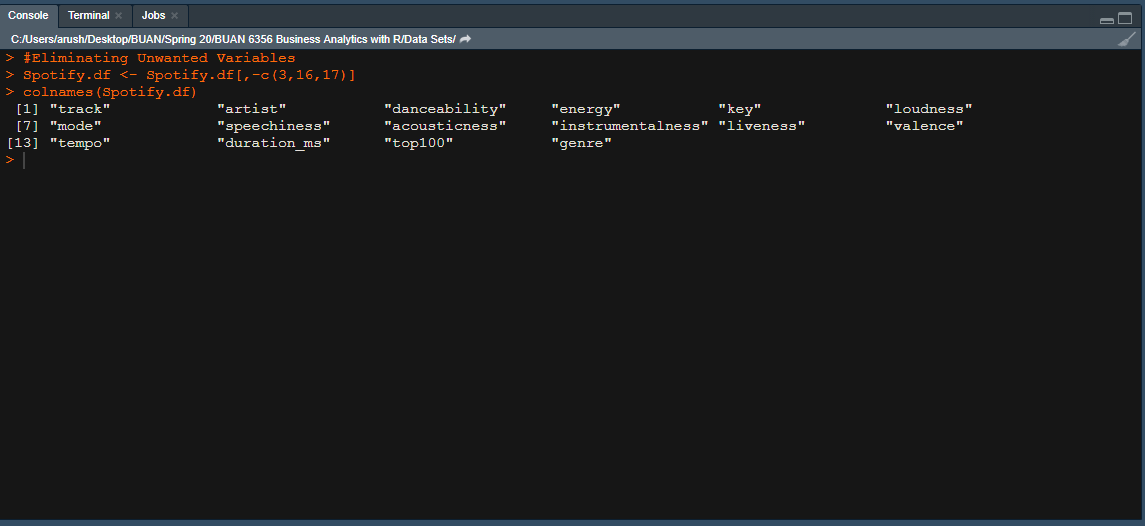
The dataset we chose was obtained from Spotify’s API. This dataset initially consisted of 6398 records spread across 19 variables. The dataset represents a mix of tracks over the last decade. They are classified as ‘Top 100’ based on whether they were, at any point of time, during the last decade at any position in the ‘Top 100’ weekly charts. This means that even if the track was at Number 100 for a week, it will be classified as a ‘Top 100’ track. However, to have an accurate model, we need to reduce the number of predictors(variables) based on what impact they have on our outcome variable. Since our goal is to predict the popularity of a song, we choose the variable ‘Top 100’ as the outcome variable.





## b. Data Cleaning

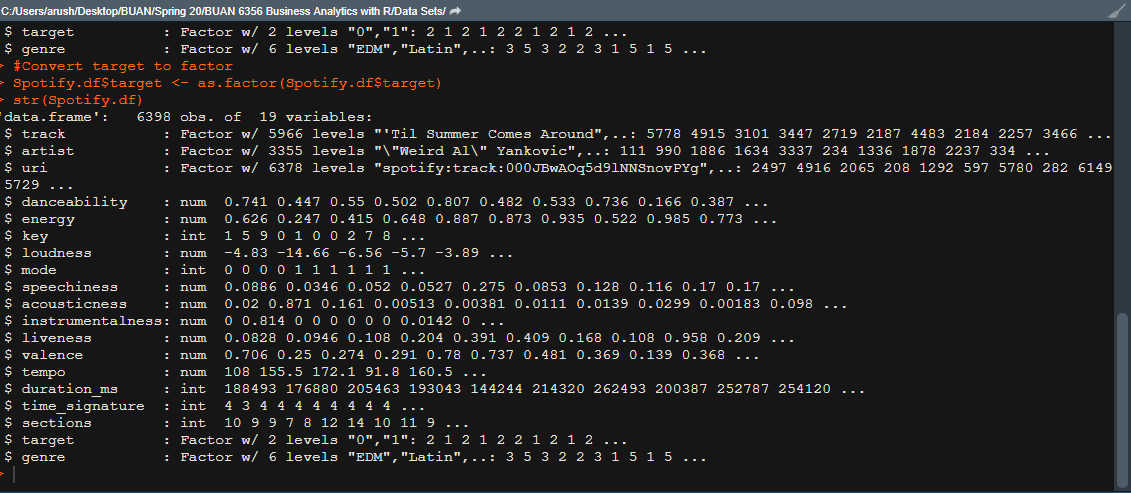
We see that the data is neat and does not consist of any N/A’s. But an in-depth analysis of the data shows that there are a lot of special characters in artist names and track names due to encoding conversions. We eliminate these special characters as they are not relevant. We begin by eliminating variables that we most certainly will not require during any stage of this project including exploratory data analysis as well as any visualizations. We eliminate the variables ‘**uri**’, and ‘**sections**’ as these are particular to Spotify and are links to access these tracks. Also, the initial few renders of our models showed that the ‘time\_signature’ does not really influence the model in any way as it is relatively constant for all the tracks. Hence, we also eliminate the **‘time\_signature’** variable. They do not in any way impact the data and will not be necessary for our analysis.



## c. Prepare Data

For ease of understanding and usability, we make a few changes to the data structure of certain variables. We convert the ‘**duration\_ms’** (duration in milliseconds) to seconds since it is easier to interpret. We also change the data structure of the **‘Top 100’** variable from integer to factor since it is the outcome variable and the classifier.

Since we intend to consider numeric variables as predictors, we also create a separate data frame consisting only of the numeric variables so that we may use this data frame during certain computations only involving numeric predictors. We also create a data frame consisting of only popular songs so that we may visualize the attributes of these songs.



# **6. MODELLING – BUILDING DECISION SUPPORT MODELS**

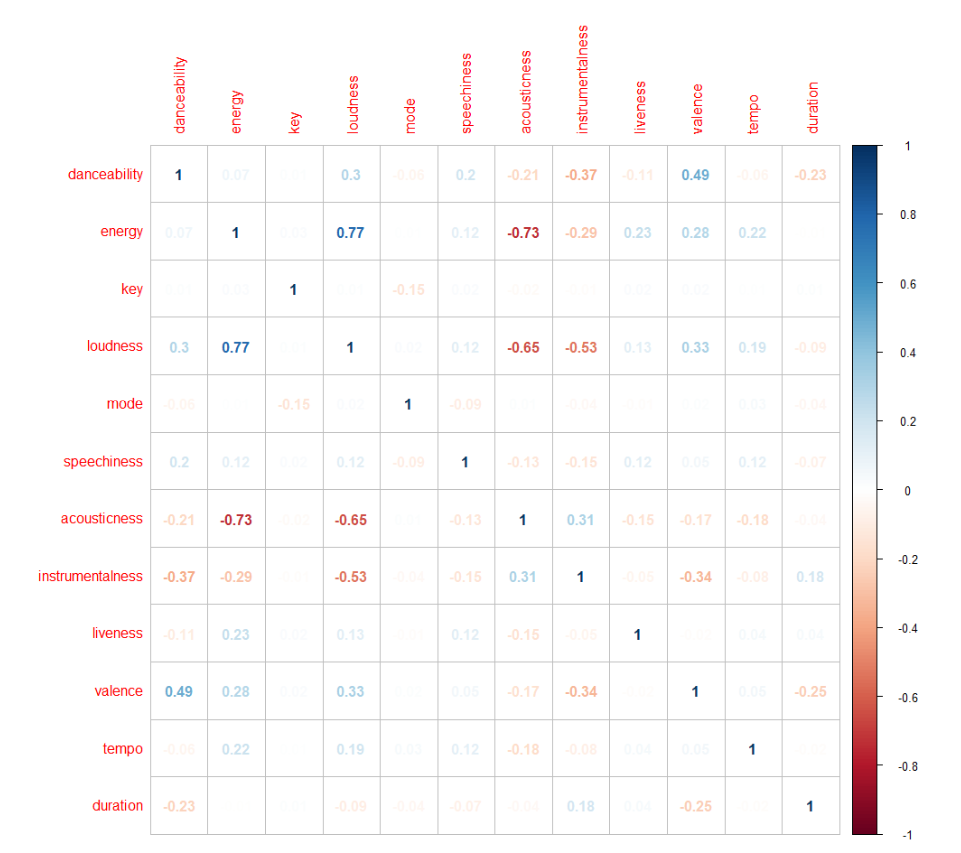
## a. Describe Data in Detail

**A close up of a map

Description automatically generated**

We plot all the numerical features of the song data to understand the distribution of the features. This helps us in understanding the distribution and variability of the dataset based on the features. A few facts about this dataset based on the plots can be listed as follows:

1. Majority of the songs are not acoustic.
2. Majority of the songs have not been recorded live.
3. Speechiness is distributed close to 0 indicating very few audiobooks and more songs with lyrics.
4. Majority of the songs have a high tempo indicating a high BPM value.
5. Majority of the songs have a significant danceability value.



**A screenshot of a cell phone

Description automatically generated**

We also plot a correlation table of the numerical variables for all songs v/s popular songs to check of there are any correlations between the predictors. However, the results are surprising, showing very few correlations indicating that most of these variables are strongly independent for both sets except for the following:

1. Energy and Acousticness have a strong negative correlation.
2. Loudness and Energy have a strong positive correlation.
3. Loudness and Acousticness have a strong negative correlation.

A picture containing screenshot, microwave

Description automatically generated

We see that there is a good ratio of hits to non-hits in the data set. This ensures that we are dealing with good examples of each category of the classified songs.

A close up of a map

Description automatically generated

Since these variables are positively correlated, based on the correlation plot, we do a density plot for these variables for the popular/hit songs and see that their distributions appear to be similar. This helps us in understanding that the hit songs have high danceability and energy.

A close up of a map

Description automatically generated

As we see, the songs in the Top 100 are mostly loud.

A picture containing drawing

Description automatically generated

We see that our dataset has songs distributed over genres with Rock being the #1 Genre and Latin and Pop coming in at #2 and #3 positions respectively.

A close up of text on a white background

Description automatically generated

The number of hit/popular songs based on genre tell us a different story indicating that Pop ranks as the most popular genre.

## b. What type of decision-making model(s)is appropriate for the decision‐making tasks?

Since our data has the outcome ‘Top 100’ which classifies ‘not popular’ as ‘0’ and ‘popular’ as 1, our goal is to be able to predict whether a song falls in the 0 or 1 category. This is a single value binary classification problem . Hence, we choose to implement the following decision-making models:

1. Logistic Regression
2. K-Nearest Neighbors
3. Support Vector Machine
4. Decision Tree

## c. Provide rationale for choice of model(s)

1. **Logistic Regression**

Logistic Regression is one of the basic and popular algorithms to solve a classification problem. It is named as ‘Logistic Regression’ because it’s underlying technique is quite the same as Linear Regression. The term “Logistic” is taken from the **Logit function** that is used in this method of classification. Logistic Regression is a classification algorithm which can be used when the output is categorical, as in our case we aim to **categorize a song as popular or not popular**.

1. **K-Nearest Neighbors**

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It is easy to implement and understand but has a major drawback of becoming significantly slows as the size of the data in use grows.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

In the case of classification and regression, we saw that choosing the right K for our data is done by trying several Ks and picking the one that works best.

**Advantages**

1. The algorithm is simple and easy to implement.
2. There is no need to build a model, tune several parameters, or make additional assumptions.
3. The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

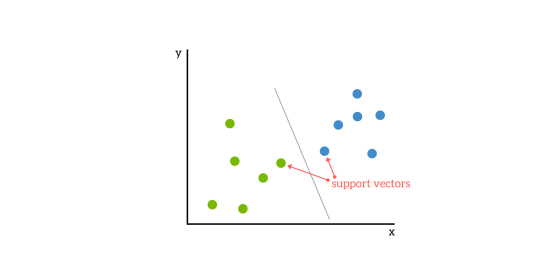
**Disadvantages**

1. The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.
2. **Support Vector Machine**

Support Vector Machines is a powerful algorithm which can be used for both regression and classification tasks.

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems and as such this why we used this algorithm in our project.

SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes, as shown in the image below.



Support Vectors

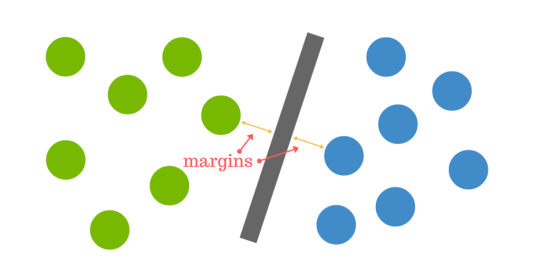
* Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

What is a hyperplane?

* As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.
* Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.
* So, when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

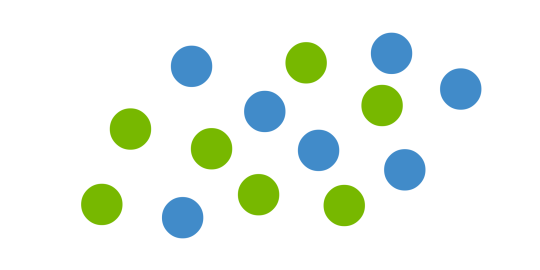
How do we find the right hyperplane? Or, in other words, how do we best segregate the two classes within the data?

* The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.

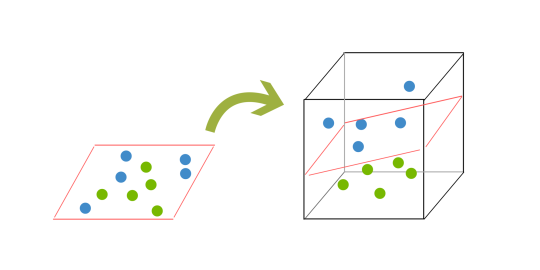


But what happens when there is no clear hyperplane?

* This is where it can get tricky. Data is rarely ever as clean as our simple example above. A dataset will often look more like the jumbled balls below which represent a linearly non separable dataset.



* In order to classify a dataset like the one above it is necessary to move away from a 2d view of the data to a 3d view. Explaining this is easiest with another simplified example. Imagine that our two sets of colored balls above are sitting on a sheet and this sheet is lifted suddenly, launching the balls into the air. While the balls are up in the air, you use the sheet to separate them. This ‘lifting’ of the balls represents the mapping of data into a higher dimension. This is known as kernelling.



* Because we are now in three dimensions, our hyperplane can no longer be a line. It must now be a plane as shown in the example above. The idea is that the data will continue to be mapped into higher and higher dimensions until a hyperplane can be formed to segregate it.

Pros & Cons of Support Vector Machines

* Pros

1. Accuracy
2. Works well on smaller cleaner datasets
3. It can be more efficient because it uses a subset of training points

* Cons

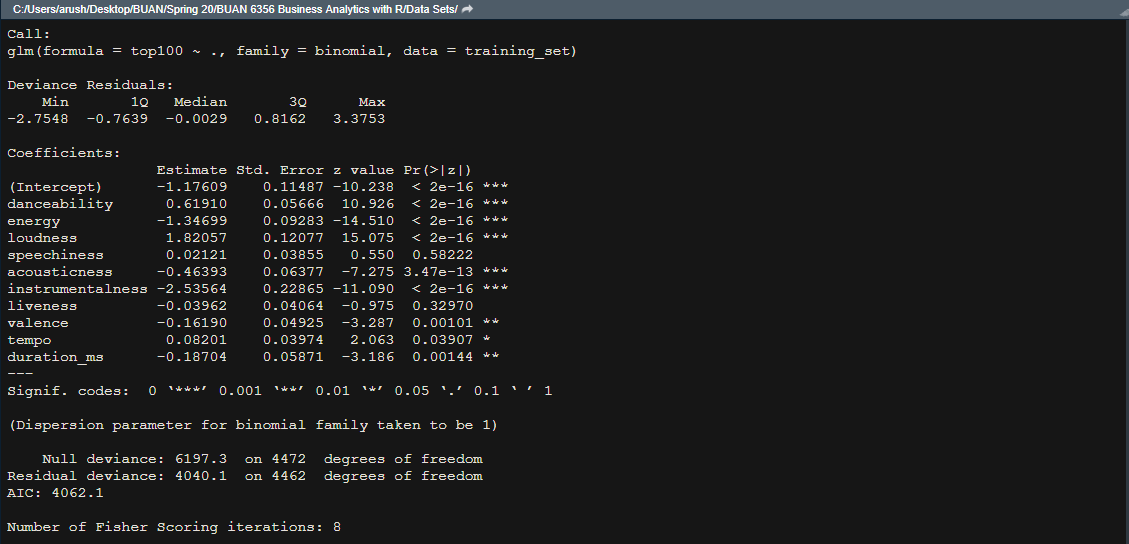
1. Is not suited to larger datasets as the training time with SVMs can be high
2. Less effective on noisier datasets with overlapping classes
3. **Decision Tree**

Similarly, Decision trees are also a popular classification method. Decision Tree algorithm belongs to the family of supervised learning algorithms. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data). The Decision Tree which has a categorical target variable is called a **Categorical variable decision tree** which we will be using.   
The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

## d. Detail model development and output

1. **Multiple Logistic Regression:**

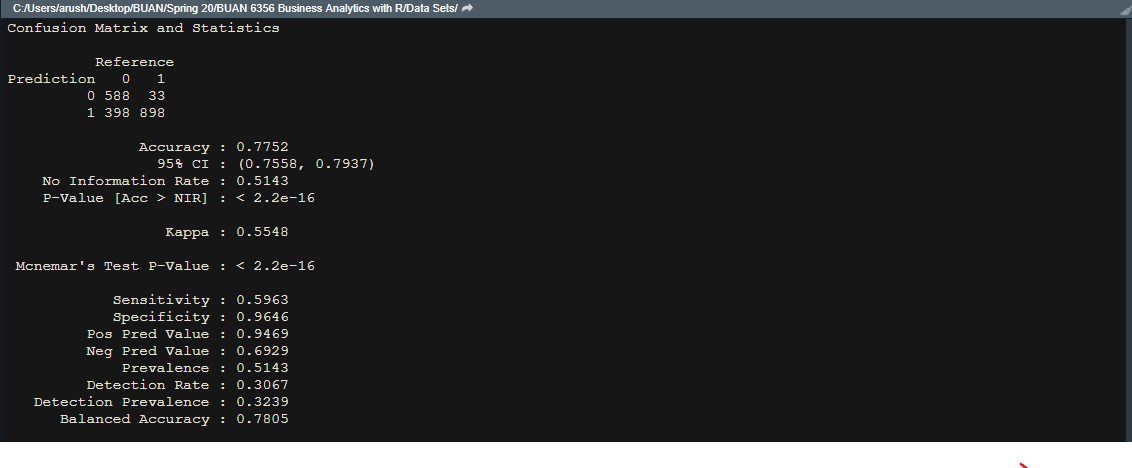
We begin by splitting our data set into training and validation sets in a 70-30 ratio. We then construct a logistic regression model using glm, with family as binomial since the outcome variable is a ‘yes/no’ classifier. This model assumes that every song can be linearly separated into 2 categories. We set the threshold/cutoff value at 0.5 indicating that, if the estimated probability of a record based on the model is greater than 0.5 then it would be classified as 1 i.e. a popular song signifying it is likely to be on the Top 100 bilboard. This model assigns weights to every feature/variable of a song and then uses these weights to predict whether a song falls in the Top 100 category or not. We use the glm() function in R to build a logistic model predicting the probability of the outcome variable. The output of the summary function that we ran on our logistic model was as follows: 

We go through this table to check the coefficients of the variables we used while creating this model. A positive coefficient indicates that the variable has a positive influence or impact on contributing to the popularity of a song and a negative coefficient indicates that the said variable has a negative impact on the popularity of a song. Since the p-value of most of these coefficients is small, we see that these predictors are strong in influencing the popularity of a song.

We understand that the model outputs probabilities for the likelihood of a song being in the top 100. We train the model using the training set and then use the validation set to make predictions.

1. **K-Nearest Neighbors:**

Next, we use the K- nearest neighbors classifier using knn and using the same split of 70-30 of training and testing for training the model and then validating it. We choose the value of K as 66 based on the rule of thumb which states k should be the square root of the number of records in the training set. As a pre-processing step, we also scale all the features so that the data is normalized, and we can have speedy calculations. We use the knn() function to train a model based on the training set. The knn() function identifies the k-nearest neighbors using Euclidian distance where k is a user-specified number, which in our case is 66. It returns a factor value of predicted labels for each of the records in the specified data set.



1. **SVM**

We construct a simple SVM using the C- Classification type since this is a classification problem. We use a linear kernel to build this model to see if the data can be distinguished with a linear separation. What it does with the data is that in case of non-linear separations, it leverages the kernel to create a few new features based on existing ones to find a decision boundary. The algorithm outputs an optimal hyperplane. We train the model using the training set and then use the validation set to make predictions. We use the svm() function in R where we define our data, the type which will be c-classification and the kernel which will be ‘linear’ in our case.

1. **Decision Tree**

We intend to use a decision tree to classify a song as popular or not in turn telling is whether is likely to end up on the billboard or not. A song classified as ‘0’ would be non-popular and a song classified as ‘1’ would be a popular one. We start of by taking the same split of 60-40 of training and testing for training the model and then validating it. We consider all the predictors we used previously and then set the parameters of Minbucket = 50, maxdepth = 5, cp= 0.0001. We use the rpart function to build an initial classification model.

We then plot the CP table to check complexity parameters of the tree, this function provides optimal prunings of the tree based on the cp value and to avoid overfitting. We then proceed to use the prp() function to plot the best pruned tree. On comparing the two trees we see there is considerable difference between the initial tree and the best pruned tree.

A picture containing text, map

Description automatically generated

**Initial Tree**

A close up of a map

Description automatically generated

**Best Pruned Tree**

Based on the best pruned tree, we see that the most important predictors of popularity of a song in our model are instrument, energy, danceability and loudness.

Some of the rules can be listed down as:

1. If instrument  >= -0.51 then class = 0
2. If instrument  < -0.51  and energy >= 1.1 then class = 0
3. If energy < 1.1 and instrument  >= -0.52  and danceability < -0.12 then class = 0
4. If energy < 1.1 and instrument  >= -0.52  and danceability >= -0.12 and valence >= 1.4 then class = 0
5. If energy < 1.1 and instrument  >= -0.52  and danceability >= -0.12 and valence < 1.4 then class = 1
6. If energy < 1.1 and instrument  < -0.52  and loudness < -0.58 then class = 0
7. If energy < 1.1 and instrument  < -0.52  and loudness >= -0.58 then class = 1

# **7. DSM EVALUATION**

1. **Logistic Regression**

We build a confusion matrix to understand the TPR and FPR of the model. ***ACCURACY – 80.54%***

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Actual Class** | |
|  |  |
|  |  | 0 | 1 |
| **Predicted Class** | 0 | 703 | 283 |
| 1 | 90 | 841 |

1. **KNN**

To evaluate the model performance, we check the results on the validation set that the model predicted as compared to their actual values. ***ACCURACY – 77.52%***

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Actual Class** | |
|  |  |
|  |  | 0 | 1 |
| **Predicted Class** | 0 | 588 | 33 |
| 1 | 398 | 898 |

1. **SVM**

We build a confusion matrix to understand the TPR and FPR of the model. ***ACCURACY – 79.81%***

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Actual Class** | |
|  |  |
|  |  | 0 | 1 |
| **Predicted Class** | 0 | 674 | 75 |
| 1 | 312 | 856 |

1. **Decision Tree**

Finally, we use decision trees to check if this technique can be used to classify the dataset with better accuracy. We build a confusion matrix to understand the TPR and FPR of the decision tree. ***ACCURACY – 80.02%***

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Actual Class** | |
|  |  |
|  |  | 0 | 1 |
| **Predicted Class** | 0 | 693 | 90 |
| 1 | 293 | 841 |

|  |  |  |
| --- | --- | --- |
| **MODEL** | **ACCURACY** | |
|  | Training Data | Validation Data |
| **Logistic Regression** | **79.3** | **80.54** |
| **KNN** | **76.95** | **77.52** |
| **SVM** | **78.74** | **79.81** |
| **Decision Tree** | **80.48** | **80.02** |

A screenshot of a map

Description automatically generated

As expected, all these models have a similar accuracy rate indicating that they have a very similar performance rate for this classifying task. However, the logistic regression model yields the best accuracy with 79.3% for training and 80.54% for validation data with the Decision Tree coming in at a close 2nd place.

# **8. DISCUSSION**

## a. DSM recommendations

We see that the DSM yields a significant accuracy which can be used to predict whether a song will reach the Top 100 or not based on the audio features provided. The analysis and model built here focuses on both, the audio industry as well as the listeners/end users. Audio labels/ talent acquisition agencies can use this data to decide whether they should be interested in signing an artist. Artists can use this as a reference while creating content to decide what attributes their tracks must include to ensure it lands in the Top 100. Listeners/End users can use this to decide what songs they should explore or listen to based on their personal favorite attributes.

## b. DSM Limitations

The limitations of the DSM are as follows:

1. It considers a single artist representing every track, ruling out the possibility of collaborations between artists influencing song popularity.
2. This model depends on the audio features provided by Spotify which is a unique feature exclusive to them. It is not an industry standard hence the analysis will be limited to data provided by Spotify.
3. It does not consider external factors such as current environment, artist popularity, rhymes, current trends, advertisements etc. which are very influential to the popularity of a song.

## c. Influences for Decisions

The attributes danceability, energy and loudness of a track and the genre it belongs too will be the top influencers for the business decisions to be made such as an upcoming artist to be signed or an album to be promoted or advertised. The decision system considers, all these factors while making a prediction except for the external factors that influence popularity.

## d. Enhancements/Future Work

We would like to have more attributes about the songs so that we can provide better results for the task of predicting popularity. User data such as the number of streams, age group etc. would help us make more accurate and specific predictions. A few attributes about external factors influencing popularity would also be helpful. For example: knowing the advertising rate of amount spent at advertising etc. We would also like to implement Neural Networks and deep learning techniques to understand if we can classify the songs better.

**REFERENCES:**

1. <https://www.kaggle.com/abdulmeral/spotify-deep-learning-in-10-steps>
2. <https://www.kaggle.com/leonardopena/top-spotify-songs-from-20102019-by-year>
3. <http://cs229.stanford.edu/proj2018/report/16.pdf>
4. www.kdnuggets.com