Exploratory Data Analysis within the Music Industry

Chaithanya Mohan

Submitted for the Degree of Master of Science in
Applied Data Science

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK
December 1, 2024

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 16473

Student Name: Chaithanya Mohan

Date of Submission: 1 Dec 2024

Signature: Chaithanya Mohan

Abstract

As Spotify and YouTube are major players in music streaming industry, it is important to study more about the scores and metrics provided by them. Using a Spotify and YouTube dataset from Kaggle with around 20,000 values, the study delves into one classification and one regression problem. The study aims to try to classify the songs into danceable and not danceable songs based on Danceability score provided by Spotify. The machine learning algorithms used for this classification are Support Vector Machine, neural network and logistic regression. While the data was divided into danceable and not danceable classes based on a threshold, it created two imbalanced classes which were balanced using synthetic oversampling. The SVM, neural network and logistic regression had an accuracy score of 0.75, 0.78 and 0.69 respectively on unseen test set. The models had chances of predicting few not danceable songs as danceable due to low precision score in danceable class. The study highlighted a lack of values in danceable class. As for the regression problem, this study's objective was to predict the likes to views ratio of audio on YouTube. Random forest, XGBoost and linear regression algorithms were used for this regression task. The random forest, XGBoost and linear regression algorithms had a R^2 value of around 0.09, 0.07 and 0.08 respectively on test set but with small MAE values ranging from 0.005 to around 0.006 for each model showing that the models are able to predict values close to the actual values but are unable to explain the variance in the target variable. The models were tried to be hyperparameter tuned but the model's R^2 value did not improve. This study concludes that the poor R^2 value of the models could be attributed to the lack of linear or monotonic correlation between the target variable and the independent variable.

Contents

1. Introduction

Music streaming markets are predicted to grow from 36 Billion USD to 125.7 Billion USD by 2032 (GlobeNewswire, 2024). As music recommendation is becoming a significant part of these streaming platforms (Subbiah and Aggarwal, 2024), the relevance of studying about the song features is increasing. One of the features is the Danceability score provided by Spotify. It is a score between zero to one and higher the score is, the more danceable a song is.
Analysing the danceability score could be helpful in recommending danceable music. It can also help users in choosing their own danceable music. In this analysis, three classification algorithms are used to predict if songs are danceable or not. Due to lack of previous literature on danceability and due to the subjective nature of the score, the threshold is chosen based on the 50th and 75th percentile of the danceability scores. The threshold creates an imbalanced danceable class. The study aims to see if the three classification models, neural network, SVM and logistic regression classifiers, are able to classify the songs into danceable and non-danceable songs.
Likes, views, shares and comments are four main engagement metrics of YouTube. YouTube is present in over 100 countries (Zhu, 2020) and is the second largest search engine after Google (Davies, 2024), which makes it a relevant streaming platform for artists to try to market their songs. Understanding the positive engagement rating can be beneficial to understand what type of songs tend to get popular. In the dataset used for this study, the likes and views counts of each song in YouTube is provided. Likes to views ratio on YouTube is used to find the positive engagement ratio.
Analysing the likes to views ratio will provide us with an understanding about how many users interacted with the song positively out of all the users that viewed the song. The comments are not considered in this study as the positive and negative comments counts were not provided separately in the dataset used. Additionally, the number of shares is also not taken into account because it is not provided in the dataset either. Likes to views ratio will help in easily comparing different artists with different amounts of views and likes. In this study, three regression models, random forest, XGBoost and linear regressors are used to predict the likes to views ratio.
Subsequently in this report, the dataset used for this study will be explained in detail followed by details about both the research questions, data methodologies, data science technologies. Additionally the report will also have a section which discusses the development methods used for both the questions and the results.

2. Background Research

2.1 Dataset description

The Spotify and YouTube dataset contains top 10 songs of various artists on Spotify and YouTube platforms. It was posted in Kaggle by Salvatore Rastelli. Marco Guarisco and Marco Sallustio are collaborators (Rastelli, 2023).
The dataset contains 20,718 data entries with 27 attributes. The attributes are Artist, Url_spotify, Track, Album, Album_type, Uri, Danceability, Energy, Key, Loudness, Speechiness, Acousticness, Instrumentalness, liveness, Valence, Tempo, Duration_ms, Url_youtube, Title, Channel, Views, Likes, Comments, Description, Licensed, Official_video, Stream. The codebook below shows all the variables, values and their labels.

Variable Name	Variable value and label
Artist	Names of the artists
Url_spotify	URL links to Spotify
Track	Names of the tracks
Album	Names of the albums
Album_type	Indicates if song is part of an album, compilation or a single. Album: Part of an album Single: A single song Compilation: part of a compiled list
Uri	Uri links of the songs
Danceability	Indicates if the song is suitable for dancing. Continuous values from 0 to 1. 0 is least danceable and 1 is most danceable.
Energy	Measures the intensity of the songs. Continuous value from 0 to 1. 1 is highest energy and 0 shows least energy.
Key	Refers to the key the song is in. Discrete values of 0-11 based on the keys.
Loudness	Average of loudness of a song. Continuous values between -60db to 0db. -60db is the least loud and 0 is the loudest.
Speechiness	Identifies if spoken words are present in the audio. >0.66: mostly spoken words like podcast >0.33 and <0.66: mix of spoken words and music like raps <0.33: mostly music
Acousticness	Continuous values from 0-1. 1 shows high confidence a song is acoustic and 0 shows least confidence that a song is acoustic.
Instrumentalness	Indicates if a song contains any vocals. Values > 0.5 indicate instrumental music without vocals. The highest value is 1.
Liveness	Indicates if the audio is of a live performance. Value > 0.8 indicates that the song was performed live.
Valence	Valance indicates the musical positiveness. Continuous value between 0 and 1.

	Higher values indicate that song is more cheerful, euphoric etc. Lower values indicate song has a more negative tone showing emotions like anger or sadness.
Tempo	Indicates the tempo of the song.
Duration_ms	Duration of the song in milliseconds.
Url_youtube	Links to YouTube videos
Title	Titles of the YouTube videos
Channel	Channel Names
Views	Number of views
Likes	Number of likes
Comments	Number of comments for each song
Description	Descriptions in YouTube videos
Licensed	Boolean values indicates whether the video is licensed or not. True: Video is licensed False: Video is not licensed
Official_video	Boolean values indicates whether the audio has official video or not. True: It has an official video False: It does not have an official video
Stream	Continuous value with number of streams in Spotify.

2.1.1 Real world applications

The dataset was collected on the 7th of February 2023 using APIs (Rastelli, 2023), the data collected is not quite old and it gives us insight into the current trends and characteristics of popular songs.
Spotify exclusively provides song audios while YouTube provides both audio and video. The dataset has metrics of both the platforms which helps in comparing performance metrics of songs in both platforms. Artists can use this feature to compare which type of songs work in each platform and the listeners will have a tailored music experience according to the platform they are using. Using features like Danceability, Energy, Loudness and valence, music streaming platforms can recommend songs based on mood or occasion or place. This will provide listeners option to choose music according to their mood or location.

2.1.2 Research questions

We can formulate questions based on general characteristics of songs and performance metrics on YouTube and Spotify. Some of the questions that can be answered with this dataset are:

- Can an audio in Spotify be classified as a podcast or a song?
  The variables we can use are Artist, Danceability, Energy, Key, Acousticness, Instrumentalness, Liveness, Valence, Tempo. Using these variables, we can predict if the song is a podcast or a song using a classification model.

- Can we predict if a song is danceable or not danceable?
  Using Energy, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence and Tempo we can predict if the song is danceable or not. Classification algorithms can be used for this prediction.

- Will songs which are popular in Spotify be popular in YouTube too?

Using a regression model, the Likes attribute can be predicted. The variables that can be used for this prediction are Artist, all characteristic attributes and Stream variable. The popularity of each song can then be decided by setting a threshold for Likes.

- Will songs with music videos on YouTube be more popular in Spotify?
  Using Artist, all song characteristic attributes, Likes, Comments and the Stream attribute can be predicted using a regression algorithm. The Likes and Streams can be compared by setting a threshold for both separately.

- Can we predict if a song will have positive engagement or not?
  Using Artist, song characteristics and Stream attributes the ratio of Likes and Views variables can be predicted. We can use a regression algorithm to predict this value.

2.1.3 Questions chosen for dissertation

The questions which will be further explored in the dissertation are:

- Can we predict if a song is danceable or not danceable?
  If the model can predict if a song is danceable or not danceable, people will be able to build customized danceable playlist for workouts and personalized activities. Self-selected music has an impact on our energy efficiency while working out (Karageorghis and Priest 2012). It will also help the music labels or artists in marketing the song to the right audience. Dance music has become UK's second most popular genre and music (News, 2022) which makes it a key area of interest.

- Can we predict if a song will have positive engagement or not in YouTube?
  Number of views and comments affects the popularity of a song, but it does not show if the engagement is positive or not. One of the most disliked videos on YouTube is "YouTube Rewind of 2018". It has a 232 million views but has only 3.1 million likes as of 2024 September (YouTube, 2018). The comments provided in the dataset can be positive or negative. However, likes to view ratio can help us understand if there is positive engagement or not. Since every artist have different number of views and likes based on their popularity, this will also help us in comparing the positive engagement of different artists.

2.2 Questions and data exploration

In this section of the report, the two questions chosen for analysis will be discussed in detail. The rationale for choosing the questions, methodology and limitation is discussed for each question.

2.2.1 Can we predict if a song is danceable or not danceable?

Rationale for this question
Danceability score provides insight into how danceable a song is. Spotify does not provide complete details on how this score is calculated but it is calculated based on music elements like tempo, rhythm stability etc. (Spotify, 2024). Danceability can be very subjective (Hu, 2024). There are few studies which delves into reducing the subjectivity of the danceability characteristic. Two of the methods suggested to tackle the subjectivity issue are adding intelligent system that takes user preference into consideration (Geng, 2021) and capturing motions made by listener to make dance music suggestions based on the motions (Gong and Yu, 2021).

Danceability score can help companies or artists in understanding which songs can be marketed to which audience. Since the pandemic time the dance music industry has been steadily growing. In 2023, dance music's revenue had a growth of 17% (Andre, 2024). The danceability score will allow companies to fully understand the trends and improve their revenue.
There are several dance challenges made by artists in social media networks like TikTok and generally within few hours of the original video getting uploaded, multiple videos are created on the same music (Wang et al., 2022). This could help the song gain popularity in streaming applications. Based on the danceability scores, the algorithm in social media can provide suggestions of danceable songs to users. As suggested in the dataset description, giving recommendations of danceable music on streaming platforms will also provide listeners option to build their own playlist which can help in improving the efficiency of workouts (Karageorghis and Priest, 2012). In a study by Howlin and Rooney (2020), the authors suggests that if given a choice, most of the patients chose danceable and upbeat music for managing their pain. Additionally, music recommendations for danceable music improves customer experience and helps in getting economic benefits for companies (Geng, 2021).

Analysis of this question
The Danceability attribute will be used to create a new column by setting a threshold for a song to be danceable. Due to the subjective nature of the variable, there is no fixed threshold and hence the threshold for this analysis is decided based on the average of 50th percentile and 75th of the Danceability score in this dataset. This will ensure that the threshold is not too high or low and all the songs are above average danceable based on the scores. Since this dataset has around 20,000 songs from various artists, it is a good representation of general tendency for similar music datasets.

$$\text{Threshold} = \frac{0.637+0.740}{2} \approx 0.68$$

The histogram shows the distribution of the danceability scores.

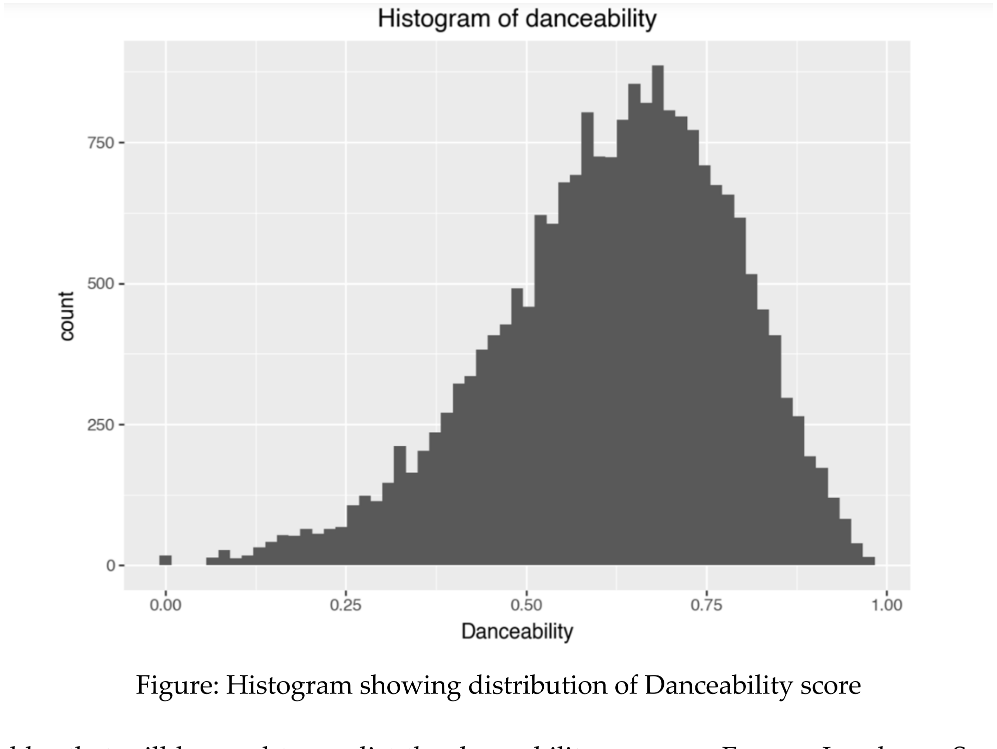


Figure: Histogram showing distribution of Danceability score

The variables that will be used to predict the danceability score are Energy, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo. Three classification algorithms will be used to predict if each song is danceable or not.

Limitations of this question
Even though the music is a universal language, it is perceived in different ways based on the culture we are part of (Hu, 2024). This also makes danceability very subjective. What is considered danceable for one might not be danceable to other. Since the dataset is collected based on songs available on Spotify whose user base is majorly from Europe, Latin America and North America (Backlinko, 2023), there is a possibility that the danceability scores majorly reflect the western perspective of danceability. Over the years, there are few characteristics like high energy levels that hasn't changed for a song to be considered danceable but there are also characteristics like Acousticness which has changed. Until 1960s, more acoustic music was considered more danceable, but it has changed over the years (Gao, 2014). The model might not be able to capture the changing concept of danceability. Even though the dataset is collected in 2023, which is quite recent, but due to the changing trends in social media platforms like TikTok the way people perceive music changes (Vizcaíno-Verdú and Abidin, 2022) this can affect the danceability score too. The model will need training data with updated danceability scores every time there is a change in trends.

2.2.2 Can we predict if a song will have positive engagement or not on YouTube?

Rationale for this question
YouTube has 10% viewership on TVs in US which is more than 7.6% of Netflix viewership (Sherman, 2024). YouTube streams are also counted towards widely recognised weekly music charts, Billboard Hot 100 charts (Ochoa, 2020). Since YouTube has a significant viewership and impact, it is essential that music labels and artists maintain positive engagement on the platform. Predicting the ratio of likes to views ratio can help in understanding what attributes generally produce a positive engagement.
Predictive analysis of the social media engagement helps in evaluating the marketing methods and setting a reference point to which we can compare the actual result (Kennedy, Kunkel and Funk, 2021). Based on research conducted on young Latin American customers, social media engagement shows their sentiments (Bailey, Bonifield and Arias, 2018). The likes on videos marketing videos could also be a good indicator of customer sentiments, it could help the company to analyse and understand customer behavior.
The Likes and Views attributes can vary widely based on the popularity of the video or artist. When we take the ratio of likes to views, it makes comparing the values between different artists easier.

Analysis of this question
The ratio of Likes to Views attributes is created as a new column. Three regressions algorithms will be used to predict the likes to views ratio.
It is not explicitly mentioned in the dataset description on Kaggle that the Likes and Views attributes are YouTube performance metrics but, in this analysis, it is assumed it is likes and view of YouTube videos because the performance metrics of Spotify is only monthly listeners, followers and total streams (vault, 2024).
Comments were not considered because comments in YouTube consists of both negative and positive sentiments (Siersdorfer et al., 2010) and the comments were not separated based on the base sentiment in the dataset. YouTube rules mentions that members should not be spamming, this suggests that there can also be multiple comments from the same account under a video (google, 2019). Unlike comments, likes on a video are unique for each user which gives us a better understanding of the unique positive engagement on a video.
If a video has around 4% or better likes to views ratio, then the video is said to have good amount of positive engagement compared to their views. This means that for every 100 viewers, there are around 4 likes. A very low likes to views ratio shows that while the views are high, there is very less or no positive engagement (Viralyft, 2024).
The reason why 4% is generally considered good is not explicitly mentioned in any sources checked. Many of the very popular songs which were checked had a ratio between 2-5%. For example: Ed Sheeran's shape of you has a likes to view ratio of around 5% (Sheeran, 2017). This might be why 4% is

a general industry standard set based on videos which are considered popular. The below graph shows the distribution of data and the maximum likes to view ratio in the dataset is around 2.5%.

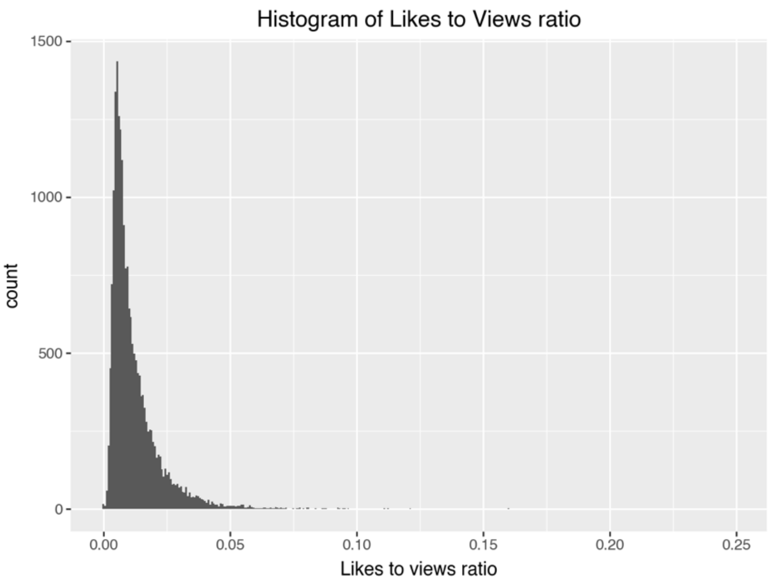


Figure : Histogram showing Likes to Views ratio

Limitations of this question
YouTube allows user to like, comment and share the videos posted on YouTube. In the study conducted by Khan (2017), the author concluded that likes, shares and comments were influenced by different factors. Khan suggested that likes were mainly used if the video meets casual entertainment purpose, comments were used mainly for interaction and shares were used when viewers wanted to share the information or general concept. While likes are considered for this study, comments and shares are not counted in the likes to views ratio. The nuances are overlooked when we only select likes to views ratio. Since the dataset does not separate positive and negative comments, it is not feasible to use comments for finding positive engagement. The number of shares is also not provided in the dataset due to which it was not used either.
There are videos which receive views and likes using bots and click farms, these are called “counterfeit attention” methods (Drott, 2020). Due to wide usage of such methods, there is a possibility that some of the views and likes are manipulated, this might result in model giving misleading predictions.
Additionally, the positive engagement does not show level of popularity. For example: If a song has 15 likes and 300 views, the positive engagement is 5%, this does not mean the song is popular. Despacito, a song with of the greatest number of likes (Kworb, 2014), has around 8.5 billion views and 53 million likes as of September 2024 (LuisFonsiVEVO, 2017). It has positive engagement of 0.6%. This is less compared to the 4% industry standard, even though the song is one of the most popular videos of all time (Greenburg, 2024).
Even though it is likes to views ratio, if the likes are zero or close to zero, the positive engagement rating becomes zero even if the views are a good amount. When this happens, the metrics tells us nothing about the views. Another limitation is even though we assume the likes are unique, there is a possibility that some people have multiple accounts, and they might be using several accounts to like videos of artists they like.

2.3 Data Methodologies

In this section, we will be looking into data collection, data cleaning, data modelling and evaluation.

2.3.1 Data Collection

The performance metrics of YouTube and Spotify like comments, likes and views, streams etc. are available for the public to view on each platform. It can be manually collected based on the research topic. In a study by Yin et al. (2022), the authors conducted a study comparing automated and manual data collection of covid patients, they noticed that the manual process is more time consuming and has possibilities of human errors. This could be true for manual data collection from YouTube and Spotify too.

Using the existing datasets can help us in reducing the manual labour. The data present in an existing dataset can be transformed or data tuned as per requirements (Gandhi et al., 2024). Dataset can be used from data repositories like UCI machine learning repository, Kaggle, Google dataset search etc. (Khosla, 2019).

APIs can be used to collect data needed for analysis. In a study by Malik and Tian (2017), they suggest the YouTube data API as a method to collect metadata from YouTube. The authors mentioned the major limitations are dependency on the API and the only data which are provided by the API can be collected. Spotify web API also provides a good way of retrieving data related to songs on the Spotify platform (Sciandra and Irene Carola Spera, 2020).

Web scraping is also known as web harvesting or web extraction. Using this technique data from any website can be automatically extracted. While web scraping can be a useful tool specifically for websites without API, it might not be legal to get data from all the websites. The web scraper will also need a server to function. Lastly, each web scraper might function differently and it would require a good amount of programming knowledge to use it (Singrodia, Mitra and Paul, 2019)

2.3.2 Data Cleaning

The data cleaning consists of handling single values, duplicate values, missing values, imbalanced dataset, categorical variables and outliers and data transformation.

Handling missing values

Missing values can either be missing completely at random or missing not at random or missing at random (Emmanuel et al., 2021). Regardless of its type, it needs to be handled. There are two types of issues which arise due to missing values, they are bias and error. Bias is introduced when we estimate a value which is very different from the actual value. Error occurs during hypothesis testing (Newman, 2014). There are several ways of handling missing values, they are discussed subsequently in this report.

List wise deletion is one of the most traditional ways and works well when the missing data is not a lot and it is missing completely at random. If the missing data is less than 10 or 15% of the dataset, then generally these missing values can be deleted (Lin and Tsai, 2019). The disadvantage of this method is that it can lead to bias if the values are not missing completely at random (Acock, 2024). Pairwise deletion makes use of two variables with non-missing entries even though there are other missing values in the same row. This method preserves more data compared to list wise deletion (Acock, 2024).

Single imputation methods like imputation with mean, median and mode are very easy to use but when we are working with a high dimensional dataset, it may introduce bias (Emmanuel et al., 2021). Imputation using mean value does not consider non-responsive bias. There may be high and low income individuals who do not fill their income value in surveys, filling them with mean would cause us to come to wrong statistical conclusions (Patrician, 2002). Imputation by median and mode value can be considered for data which is not normally distributed. This also can lead to bias in the dataset (Jadhav, Pramod and Ramanathan, 2019). Furthermore, mode can be used with categorical data.

Regression imputation is also known as conditional mean imputation. A regression model is made first and then the missing values are filled using the regression model. This model can work well if we know the target variable is a linear combination of the other variables. The downside of this model is that it only forms one regression curve and all the missing values are set based on the same curve. If there are more than one missing variables, a multivariate regression model can be used (Emmanuel et al., 2021).

In K Nearest Neighbour imputation method, the missing values are filled using the nearest similar neighbour. The distance is generally calculated using the Euclidean distance (Emmanuel et al., 2021). It is not ideal to use this when the dataset is large as it will consume a lot of time and the missing values will change based on the value of k (Jadhav, Pramod and Ramanathan, 2019).

In multiple imputation methods, several plausible values are imputed instead of imputing a single value. This produces multiple dataset and the analysis needs to be done on multiple dataset and the results can be studied and consolidated (Jadhav, Pramod and Ramanathan, 2019). Unlike the single imputation methods mentioned above, when using multiple imputation, it does not underestimate the errors due to using a specific value (Acock, 2024). The disadvantages of this method are that it takes more effort and time to use different imputations and analysing them. There is a possibility of different types of errors occurring in each dataset (Patrician, 2002). This method might also require more domain knowledge to analyse multiple models and consolidate it.

In iterative imputation, firstly, an initial value is set for all missing values, generally the mean is taken. The model parameters are set based on these parameters and iterated. Using these parameters, the conditional expectation of missing values are then found out. These values are then replaced as the missing values. The iteration is done until the imputed values are within a manually specified tolerance level(Liu and Brown, 2013).

MICE(Multiple Imputation by Chained Equations) and a python package available in scikit-learn, called IterativeImputer are two examples of iterative imputation methods. Initial missing values in the IterativeImputer are set based on mean, median, mode or a constant. The regression iteration is done using RidgeCV or Bayesian Ridge. While MICE generates multiple imputations, each variable can have a separate imputation model (Platias and Petasis, 2020). Based on a study by Klomp (2022), IterativeImputer showed varying bias across multiple predictors while mice hardly showed any bias. According to this study mice performed better than IterativeImputer.

Data transformations

When a dataset has multiple units or scales or the data in different columns is in different ranges, normalisation can be used to bring the values between a certain range. If the data is already in a normal distribution, the z score normalisation can be used. This changes the mean to zero and the standard deviation to one. This type of normalisation is called standardisation (Jamal et al., 2014).
In logarithm transformation each value changes based on the logarithm (Lee, 2020). As standardisation and normalisation does not use log, they are non-logarithmic transformations. There are also transformations which are done based on power functions like Box-Cox transformation. This transformation helps in steadying the variance of errors (Lee, 2020).

Handling duplicate values

Duplicate entries in a dataset can lead to overfitting of data. It might make the model put more weightage to the values with duplicates. It can also lead to miscalculated accuracy (Anishnama, 2023). The duplicated values can be in the test set and training set, which can cause the model to perform well with the test set but perform badly with any other generalised data.

Deleting duplicate values is a straightforward process and can be done by using drop_duplicates() function from the pandas library (Ma, Li and Li, 2024). Deleting partial duplicates is a lengthier and time taking process. In a study by Chevallier et al. (2022), they suggested using a classifier to detect the nearly duplicated data.

Handling Outliers

Outliers can arise due to several reasons including mechanical faults, human error, technical error or changes in general population (Hodge and Austin, 2004). Detecting and removing outliers can help in training the model better (Smiti, 2020). The meaning of what data is taken might vary depending on the method which is used for finding the outlier. In the studies by Hodge and Austin, they analysed how a data point which can be considered as an outlier in the box plot, cannot be considered an outlier in the scatter plot when compared to the regression line.

Statistically if a point deviates from its standard distribution it can be considered as an outlier. The parametric methods can be used when the distribution of the data is already known (Smiti, 2020). They are suitable for large datasets too as it helps in fast detection of outliers (Hodge and Austin, 2004). The Gaussian based method and the regression methods come under the parametric methods. Checking outliers using box plots is one of the most commonly used gaussian based methods (Smiti, 2020). In the regression based method, the variables are plotted in a scatter plot to see the outliers. The outliers are decided based on the regression line (Smiti, 2020).

Histograms and kernel based methods are generally used in non-parametric methods. In parametric methods, there is a need for the user to have prior knowledge about the distribution but in non-parametric methods, no prior knowledge is required. In kernel based methods any data points which lie outside the distribution space are considered as an outlier(Hodge and Austin, 2004). Histogram and Kernel can be used for numerical data but cannot be used for categorical data, the outliers change based on the bin width and bandwidth (Smiti, 2020).

Outliers can also be removed using proximity based methods and the distance is calculated based on distance from the nearest points. The points which have a high distance from the points can then be considered as outliers. In K Nearest neighbour is an example of distance based outlier detection methods. Local outlier factor is calculated by checking the density difference which is calculated based on the distance between neighbours. If a point is in a less dense area, there is a high possibility that it is an outlier based on local outlier factor (Boukerche, Zheng and Alfandi, 2021).

Handling imbalanced dataset

Even though a balanced dataset is quite useful for training a model, most of the real life problems have an imbalanced dataset. When we have an imbalanced dataset, the machine learning model tends to be biased towards the majority class. This will lead to machine learning models having great precision on the majority class but bad efficiency on the minority class. If the imbalance is not very prominent, for example, 60% in one class and 40% in another, the imbalance needs not be addressed. If the imbalance is notable with around 90% for one class and 10% for another class, it needs to be addressed (Islam, Arifuzzaman and Islam, 2019). When the data is imbalanced, predictive accuracy is not a good way of checking how well a model is performing because the model might only be performing well with the majority class (Chawla et al., 2002).

One of the most common ways to handle an imbalanced dataset is by using oversampling techniques. SMOTE (Synthetic Minority Oversampling Technique) is one of the eminent oversampling methods. In this technique, the synthetic examples are used for oversampling rather than using substitutes for oversampling (Chawla et al., 2002).

Handling single value column

When we have a single value column, it has no variance and it creates no change to the dependent variable but there are algorithms whose performance are impacted by these columns. One example of such an algorithm is Principle Component Analysis (PCA) (Hargreaves, 2023).

Handling categorical variables
Most of the machine learning algorithms only can work on numerical variables and hence before a model is applied, the categorical variables need to be encoded into numerical values. Ordinal encoding can be used for encoding when there is an ordinal relationship between the variables. It does not add any new columns to the dataset (Potdar, S. and D., 2017). It might assume an ordinal relationship is present even if there is none. Unlike ordinal encoder, One hot encoder does not assume any ordinal relationship between the variables but it leads to higher dimensionality in the dataset (Samuels, 2024). This can often lead to higher memory consumption and more complex datasets (Seger, 2018). Target encoding can be used on higher dimensionality dataset. Comparatively, models perform better with target encoding than one hot encoding. It uses mean value to compute the encoded value, while this is quite useful, it also misses other information about the dataset distribution while encoding (Larionov, 2020).

2.3.3 Data Modelling

Supervised and Unsupervised Learning
In supervised learning the training set with labels is provided and the model learns from it and predicts the values of the test set (Learned-Miller, 2014). The training data is given in the form (xi,yi) where each value contains a n dimensional vector and an output which is scalar. For example in a credit card risk prediction dataset, the details of the person and their transaction details would be in a vector form and the credit card profit or loss which needs to be predicted will be in scalar form (Barto and Dietterich, 2004). Linear regression, Logistic regression, Support vector machines are examples of supervised learning algorithms (IBM, 2021). There are several data use cases, the data is not completely labelled (Engelen and Hoos, 2019). Supervised learning can help in detecting fraud, classifying images, assessing the risk, making recommendations etc (Google Cloud, 2024). The downside is that supervised algorithms are time consuming and they cannot cluster or classify data on its own (IBM, 2021).
In semi supervised learning, both labelled and unlabelled data is used. This is particularly useful when the unlabelled data is easily available and the labelled data is more expensive to obtain. The majority of the semi-supervised learning studies are concentrated on classification problems. The predictions are made using the underlying structure and patterns of unlabelled data along with the labelled data. The downside of these models are that they are only useful when there is underlying information which helps in better prediction. There is no easy way of knowing when there is an underlying pattern or not. There is a possibility of the performance going down because of the presence of unlabelled data. Some of the examples for semi supervised learning algorithms includes, semi supervised support vector algorithms, semi- supervised neural networks: mean teacher model (Engelen and Hoos, 2019).

Unlike supervised learning, unsupervised learning does not have any labels for the target variable. It would only have a set of attributes. The machine is not provided any feedback from the environment too. Using the unlabelled data, the underlying tasks are found out (Ghahramani, 2004). Clustering, Association and Dimensionality reductions are three of the main uses of unsupervised models (Delua, 2024). Principal component Analysis and K-means clustering are examples of the unsupervised learning algorithms. These algorithms can be used in anomaly detection, customer grouping, Natural language processing etc (Google Cloud, 2024). The disadvantages include computational cost and possibility of errors (IBM, 2021).

Machine learning tasks
Within this section, the machine learning tasks, classification, regression and clustering will be discussed.

Classification is used to label or classify data based on the training data. It is one of the most widely used machine learning techniques. K-Nearest Neighbour, Support Vector Machine are two examples of classification algorithms (Soofi and Awan, 2017). Binary-class classification and Multi-class classification are the two important classes of classification. In binary class classification, there are only two classes and in multi class classification, there are multiple classes (Jha, Dave and Madan, 2019). Classifying the emails as spam or not spam is an example of binary class task while labelling the data into four types of flower species is a multi-class classification problem.

Regression is done by supervised learning algorithms, the model trains based on the training data provided and predicts a continuous value. The regression problem can also be solved using classification algorithms by converting these continuous values into discrete values. These discrete values can be taken as classes and classified based on it (Luís Torgo and Gama, 1999). There are specific regression algorithms which can be used for solving regression based problems. Some of the examples of regression algorithms are: Linear regression, Ridge and Lasso regression(Kurama, 2024).
Clustering is an unsupervised learning task which groups unlabelled data based on their similarities (Google for Developers, 2024). Clustering algorithms can be divided into hierarchical and partitional clustering. In hierarchical clustering, the clusters are divided based on hierarchy. Top to down or bottom to up method is followed. In partitional clustering, the clusters are formed based on sequence and not hierarchy. Clustering can be used for tasks like image processing and segmenting, text mining, dimensionality reduction etc. DBSCAN and K-Means are two examples of clustering algorithms (Ezugwu et al., 2022).

2.3.4 Data evaluation

Regression evaluation metrics
Mean Squared Error (MSE) is calculated by summing the squared values of both predicted and actual values. It penalises the larger errors more because it is the squared difference (Hugo Matalonga, 2023). If there are large outliers, the MSE value could be very high. They are more sensitive to large outliers compared to other metrics (Chicco, Warrens and Jurman, 2021).

$$\text{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y_i - p_i)^2$$

Where n = total number of datapoints,
y_i *= observed value ,*
p_i *= predicted value*

While MSE has a different unit than the target variable, Root Mean Squared Error (RMSE) has the same unit as the target variable and is easier to interpret. It is the root of Mean Squared Error. This metric is also sensitive to large errors. Mean Absolute Error (MAE) metric is straightforward and is not sensitive to large errors like MSE and RMSE. The disadvantage of this metric is that it does not penalise large errors (Hugo Matalonga, 2023). MSE, RMSE and MAE score lies between 0 to infinity. The best value is 0.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - p_i)^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=0}^{n}|y_i - p_i|$$

Where n = total number of datapoints,
y_i *= observed value ,*
p_i *= predicted value*

Mean Absolute Percentage Error (MAPE) is useful when the relative variations are more important than the absolute variations. It does not work well when there are large errors. R-squared value is also known as the coefficient of determination. It shows the level of variation between the predicted value and the actual value. The score lies between 0 and 1, where 1 is the best possible scenario. R-squared assumes there is a linear relationship between the target variables and other variables. It is also highly sensitive to outliers (Fitzpatrick, 2023).

$$\text{MAPE} = \frac{1}{n}\sum_{i=0}^{n}\frac{|y_i - p_i|}{y_i}$$

Where n = total number of datapoints,

y_i *= observed value ,*

p_i *= predicted value*

$$R^2 = 1 - \frac{Sum\ of\ Squares\ of\ Residuals}{Total\ Sum\ of\ Squares}$$

Classification evaluation metrics

Accuracy is calculated by summing up the number of correct predictions and dividing it by the total number of predictions. It is used often because it is easy to compute and understand. If there are imbalanced classes, accuracy scores can be misleading (Amer, 2022). If the majority class has mostly true positives and true negatives, the accuracy score could be high even if the minority class has less or no true positive and true negative values.

$$\text{Accuracy} = \frac{Correct\ predictions}{All\ predictions}$$

Precision measures the true positives out of all the positives outcomes while Recall measures the true positives divided by actual positives. The optimisation of recall or precision negatively impacts the other value. The decision of which of the two metrics to choose vary between different tasks. For example, while classifying spam email, it's better to have high precision so that the number of non-spam emails classified as spam is less even though some emails which are spam are not classified as spam. F1 score provides the balance between both precision and recall (Amer, 2022). Since precision and recall both do not include true negatives, F1 score does not take true negatives into consideration. Both recall and precision scores are also affected by class imbalance (Juba and Le, 2019).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$\text{F1 score} = \frac{Precision\ x\ Recall}{Precision + Recall}$$

Confusion matrix shows the true positive, negative and false positive and negative of each class. It makes it easy to see how the model is performing with each class (Amer, 2022). When it is binary classification, the confusion matrix only has two rows and two columns. Confusion matrices can be used to represent multi class classification too but they fail to represent multi-label classes where one object can belong to multiple classes (Krstinić et al., 2020).

Receiver Operating Characteristic (ROC) curve is plotted by plotting sensitivity against false positive rate which is 1-specificity. If the ROC curve is the same as the y=x intercept, the true positives are equal to false positives (Hoo, Candlish and Teare, 2017). Area Under the Curve (AUC) shows the overall performance of a classification algorithm. Even though AUC is a good metric to measure performance, the computational costs can be quite high especially when it is a multiclass classification problem (Hossin and Sulaiman, 2015) .

Log-Loss is also known as cross entropy score. It is calculated based on prediction probability. The lower values suggest the model is performing well. Unlike accuracy, it considers the probability distribution rather than just matching the predicted values to actual values (Aggarwal et al., 2021). The authors also points out that the loss function can be used to find the true label and this could breach privacy when the machine learning models are trained on sensitive data.

$$\text{Log loss } = -\frac{1}{n}\sum_{i=0}^{n}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Where n = total number of datapoints,
y_i *= observed value ,*
p_i *= predicted value*

2.4 Data Science Technologies

In this section, broad view of the technologies or tools that can be used in the data visualisation and data pre-processing will be discussed.

2.4.1 Data science Programming languages

Python is an open source programming language which is ranked as one of the top programming languages in the world by TIOBE index and PYPL index (Canales Luna, 2020). Python is a programming language that can be used by both professionals and beginners as it is easier to understand compared to other programming languages (Colliau et al., 2017). While python has several statistical packages and can be used for statistical analysis, python was not made with the intention of statistical analysis hence it lacks the analytical power of R programming (Hill et al., 2024).

R Is a programming language which is made specifically for statistical analysis which makes It a powerful analytical software. R is also an open source and free software and the community driven discussions can be used for debugging programming errors. Compared to SAS, another powerful statistical analysis tool, it only requires one programming language, while SAS requires three programming languages to do the same analytical tasks (Colliau et al., 2017). Major disadvantage of this language is the time it takes to master it. It also is computationally slower compared to python (Canales Luna, 2020). In a comparative experiment on Python and R, the author tried finding 10,001 prime numbers, and concluded that R was 17 fold slower than Python for this specific problem when a byte code compiler was not used. Even when a byte code compiler was used, it was slower that Python's computational speed for this problem (Simmering , 2013).

SQL (Structured Query Language) is a language used to manipulate and interact with relational databases. It is set according to ANSI standards (Melton, 1996). SQL is known for the speed at which the data can be manipulated and retrieved. The database can easily be transferred from one technology to another. SQL also permits users to view data in multiple ways (Singhal, 2022). While SQL is easy to learn, it also has its own disadvantages. Compared to python its usage is limited. It has fewer third party libraries which can aid in data analysis (Buuck, 2024). For example, while python has libraries for getting data from API, reading data from PDF etc, SQL does not have any libraries for these tasks. Python can work with several types of data including images, but SQL cannot analyse these data (Start Data Engineering, 2024).

2.4.2 Data visualisation technologies

Data visualisation technologies can be used for various tasks, including but not limited to analysing emerging trends, social media conversation, understanding underlying patterns in a dataset etc. (Lavanya et al., 2023). Subsequently, the visualisation tools will be discussed.

Tableau and Power BI are both Business Intelligence(BI) tools. Tableau was the first popular software that allowed users to analyse different datasets without using any code. It produces interactive graphs and also allows users to collaborate over the cloud. Tableau is known for its customisation abilities and flexibility. While learning Tableau requires a significant amount of effort, power BI is easier to learn and implement. Power BI comes with a powerful data editor and an open API and also allows users to post customised plugins (Carlisle, 2018). Power BI cannot handle a lot of datasets, the performance is impacted when there are several datasets (Jackson-Barnes, 2024).

Matplotlib is one of python's most popular visualisation libraries and is built on NumPy libraries. It provides several customisable 2D and 3D plotting options. It is known for its ease of integration with other python libraries (Lavanya et al., 2023). While it is flexible and can be used to make quick visualisations, it still requires customisations to be manually added compared to Seaborn, which comes with built-in visualisations with more details (Oberoi and Chauhan, 2019).

Seaborn is also one of the libraries which is available in python. Seaborn is flexible with the input data and it has the ability to transform data during visualisation. Unlike Matplotlib, it can handle wide-format data. It also comes with built in themes which makes it more aesthetically pleasing. Even though Seaborn has several advantages, it cannot completely replace Matplotlib. More complicated graphs would require Matplotlib and Matplotlib functions are still used to customise Seaborn plots. In spite of Seaborn being quite similar to ggplot2, it does not use "Grammar of Graphics" and we cannot fully control how the graph is displayed. It is more useful for quick exploratory analysis of data (Waskom, 2021).

Ggplot2 is a package in R which uses "Grammar of Graphics". In "Grammar of Graphics", the graph that needs to be drawn is first described and then built layer by layer, which makes it possible to create custom graphs as needed for each situation. The graphs made in ggplot2 are also reproducible (Wickham, 2011). Plotnine is the python equivalent of ggplot2 and it provides more features on top of what is already provided in ggplot2 (Lavanya et al., 2023). Since ggplot2 follows "Grammar of Graphics" (Wickham, 2011), it might be harder for beginners to create graphs in ggplot2 or Plotnine.

Tableau and Power BI are both Business Intelligence tools which can be used when a user wants to draw visualisations without using any codes (Edmond and Crabtree, 2022). Matplotlib can be used when the user wants to customise their graphs and built it from scratch. Ggplot2 also permits users to customise the graphs, but Matplotlib provide more polished graphs. Ggplot2 is a great option if the user wants to use visualisation package which has more data handling capabilities (Patro, 2021). Users who prefer to use less code with more aesthetic visualisation features can use Seaborn package (Karl, 2024).

2.4.3 Integrated Development Environment

Integrated Development Environment (IDE) is a software which works as a code editor, compiler and debugger. Some of the useful IDEs are discussed below:
Jupyter, which is a follow-up version of iPython, is free and open-sourced and quite easy to use. Jupyter can be used to write code and run it simultaneously. It also provides features like converting the code into PDF, html file etc. The interactive coding environment also allows users to add comments

along with code. It also allows you to switch between various code languages (Saabith, Thangarajah and Fareez, 2021). Even though it supports more than 40 programming languages, it works with only .ipynb format and it has no auto code completing capabilities like PyCharm and Visual Studio (Garkusha, 2024).

Google Colab or colaboratory is a free cloud service based on Jupyter, where datasets can be analysed. It provides large computational space compared to Jupyter. While other IDEs are installed offline and the computational power depends on the computer or device, Google Colab provides high computational power on all devices. Most of the modules which are needed for analysis come pre-installed in Google Colab (Naik, 2023). Even though we can write interpreted language directly on Google Colab, we need to write compiled languages offline then add it to Google Colab. Since Google Colab is a free service with no guarantees, Google may decide to recall it or change its available computational power anytime which may result in loss of project or the file (Carneiro et al., 2018).

Orange is an free, easy to use tool with visual programming abilities, that facilitates tasks including data preprocessing, data modelling and data evaluation. These visual programming widgets allow even beginners to use this IDE with ease and it provides flexibility and shows the interaction between each widget. Orange also allows users to use programming scripts along with visual programming facilities (Demšar et al., 2004). Even though orange allows users to use different methods and do the tasks, it does not store the data for each step anywhere. While this allows users to experiment (Dobesova, 2024), it might also lead to users losing data unless they have saved or exported it.

PyCharm is an IDE which allows the user to work on multiple scripts which interact with each other. It has a free version and users can choose to upgrade to a paid version (Saabith, Thangarajah and Fareez, 2021). Even though it provides autocomplete and auto suggestion features and comprehensive support for programming languages, it is compatible with fewer programming languages compared to Jupyter notebook (Garkusha, 2024). Another disadvantage is that the basic version has lesser features compared to premium features. For example, PyCharm basic version does not support languages like SQL, CSS etc. Furthermore, It does not have support for frameworks like Flask and FastAPI (JetBrains, 2024).

Visual Studio is an IDE developed by Microsoft. It provides support for building web applications as well as data analysis. It comes with PyLint, which checks for errors and also makes good python programming suggestions, which can be very useful for beginners in programming. Similar to PyCharm, it also provides auto code completion. It comes with a lot of features for advanced users (Saabith, Thangarajah and Fareez, 2021), so it could prove to be difficult for beginners to understand and use it.

2.5 Machine Learning algorithms

The following section will explore the six machine learning algorithms that will be used for analysing the two questions in hand.

2.5.1 Can we predict if a song is danceable or not danceable?

To classify the songs into danceable and not danceable songs, Three algorithms are selected. They are Support Vector Machine algorithm, neural network and logistic regression.

Support Vector Machine algorithm

Support Vector Machine is a Kernel based method which can be used for both classification and regression tasks (Cervantes et al., 2020). In a classification task, the classes are divided using a hyperplane. The points that fall on the boundary are known as the support vectors (Boswell, 2002). If the data points in two classes are not linearly separable, the optimal hyperplane would not have high generalising capabilities due to the inability to create an optimal hyperplane, hence the original input space is transformed into a feature space using kernel functions and then the support vector is used to find the optimal hyperplane. (Cervantes et al., 2020). When the classes cannot be linearly separable, soft margin can also be used. It allows some training data to be misclassified but there is a penalty which the model tries to reduce (Kammoun and AlouiniFellow, 2021).

$$w \cdot x + b = 0$$

Where w is the weight of the vector,
x is the input feature and b is the bias value

Pseudocode for Support Vector Algorithm by Saigal and Khanna(2020).

```
Input: X vector of features and Y vector of labels
Output: Class labels for the test data
Steps:
    1. The regularisation and Kernel values are selected.
    2. The SVM model is trained with the training data.
    3. The support vectors and hyperplane is chosen.
    4. The samples are classified into the class label that has the largest decision function value
```

SVM algorithm works really well for binary classification. Its ability to generalise well and distinguish between classes has made it a very popular classification algorithm (Cervantes et al., 2020). Running SVMs can be computationally very expensive when used on large datasets. The kernel matrix that maps to a higher dimensionality space grows quadratically as the dataset gets bigger so the training process can be time consuming. If the dataset has n data points, the kernel matrix will be of size nxn. This might not be too bad on the dataset which is used for this analysis as the dataset is a medium sized dataset. Another disadvantage is that SVM does not have an optimised solution for classifying multi-class classification. The multi-class classification will have to be solved as multiple binary class classification (Cervantes et al., 2020). However, the classification question on which the SVM is used in this study is a binary classification problem. SVMs do not perform well on an imbalanced dataset. It makes it harder for SVM algorithm to draw an optimal hyperplane when the dataset is imbalanced (Cervantes et al., 2020). Even though the classes are not balanced in the dataset used for this study, the imbalance is not very severe so this disadvantage might not affect the classification.

Neural network

Artificial neural networks are computing models built based on information processing by the human brain. It has several nodes where each node has an activation function and a weight is set between each pair of nodes. The output is decided based on these networks between nodes, weight and the incentive function. It typically consists of three types of layers, input layer, output layer and hidden layer (Wu and Feng, 2017). The nodes function independently to each other, they only use the local information to provide an output. Each layer can have a random number of nodes called bias nodes (Isaac Abiodun et al., 2018). The below function shows how the node calculates the output.

$$y(k) = F\left(\sum_{i=0}^{m} w_i(k).\, x_i\,(k) + b\right)$$

Where x_i = input value, w_i = weight associated with the line,
b = bias value, F = activation function,
y = output and k = step or the index.

Theoretically, a Neural network can approximate any target function due to its activation function. The most commonly used activation function is ReLU (Rectified Linear Unit) function (DeVore, Hanin and

Petrova, 2021). The activation function generally is adjusted based on the type of output we require (Krenker, Bester and Kos, 2011).

Pseudocode for Neural Network by Pedro (2021) and Chen et al. (2015)

```
Input: X vector of features, Y vector of labels and w weight for all nodes and i number of nodes (Guo et al., 2019).
Output: A neural network model
Steps:
    1. The input is passed through layer by layer in a feed forward method.
    2. The loss is calculated.
    3. The input is passed from last layer to first layer using back propagation.
    4. The weight and bias values in each layer are adjusted using gradient descent
    5. The classification is done after the parameters are adjusted
```

Most of the data in real life is non-linear and artificial neural networks function well with non-linear relationships. It also has self-learning capabilities which helps with forecasting. The Neural networks also help us in finding optimal solutions at high speed (Wu and Feng, 2017). It performs really well when there is sufficient data even if the patterns are not clearly understood. It also has good generalising capabilities without overfitting on the training data. Overfitting happens when a model memorises the training data without learning the underlying pattern. This results in the model performing badly with unseen data (Dongare, Kharde and Kachare, 2008).
Neural networks might not perform well if the data is not sufficient or if there is noise in the data (Dongare, Kharde and Kachare, 2008) but considering the above mentioned advantages, it is chosen as one of the algorithms to classify danceability scores.

Logistic Regression
The logistic regression model is used to find the odds of two labels of a binary classification. It predicts the probability of an outcome which is then used to classify into a label. Unlike the linear regression models, it does not have the formula for the best estimates. The estimates are improved until they reach stability. It takes the natural logarithm of the odds as the function of predictors (LaValley, 2008).

$$ln[odds(Y = 1)] = \beta_0 + \beta_1 X$$

Where ln=natural logarithm,

β_0 = *intercept term*

$\beta 1$ = *regression coefficient*

Pseudocode for Logistic Regression by Singh et al. (2022)

```
Input: X vector of features and Y vector of labels
Output: Class labels for the test data
Steps:
    1. For loop where i=1 up to k
    2. Each training data instance is taken as d
    3. A target value is set for regression $z_i = \frac{y_i - P(1|d_j)}{[P(1|d_j)(1-P(1|d_j))]}$
    4. Initialise the weight of $d_j$
    5. Assign first class label if $p_{id} > 0.5$, otherwise other class label
```

Logistic regression models allows us to use categorical or continuous variables and also allows us to use multiple predictor variables (LaValley, 2008). These models are also not complex compared to other models, it simplifies the process of finding the impact of all variables and reduce the confounding factors (Yasar, Lawton and Burns, 2024). These models are simple models and can overfit over the training data and are also sensitive to outliers (Yasar, Lawton and Burns, 2024). In this analysis, the model will be tested on a unseen data to ensure that the model is not overfitting. They are generally best for binary classification tasks and not multiple class classification (Yasar, Lawton and Burns, 2024) but since this analysis is on binary classification, it would not create any issues for this study.

2.5.2 Can we predict if a song will have positive engagement or not on YouTube?

For this task, three regression algorithms, random Forest, XG Boost and linear regression will be used.

Random Forest algorithm

Random Forest algorithm is a supervised learning algorithm which can be used for both classification and Regression problems. In regression based tasks, random forest uses the bagging method. (Biau and Scornet, 2016). The error rate of random forest can be calculated by aggregating the error rate of individual trees using Out-of-Bag cases. For regression models, the concordance error rate is calculated. For example, if the weight has to be predicted then we use a pair of cases to see if the model has predicted the heavier of the two cases (Rigatti, 2017).

Pseudocode for Random Forest by Jo (2022). and Hambali et al., (2022).

```
Input: X vector of features and Y vector of labels
Output: A random forest model
Steps:
    6.  Bootstrap samples are created from the dataset.
    7.  m features are randomly selected.
    8.  The node is split by selecting the best feature among the m features until one leaf is attained and
        the tree is completed.
    9.  The model is trained on each bootstrap samples independently.
    10. Each tree comes up with a prediction and the average of all the predictions are taken.
```

Random forests are good at handling large datasets where it is hard to find a good estimate with just one step. It also handles large datasets without compromising on statistical effectiveness (Biau and Scornet, 2016). Compared to the single decision tree, it helps in reducing the variance (Probst and Boulesteix, 2018). It also has the ability to see the non-linear patterns of the predictors (Rigatti, 2017).
Even though the random forest algorithm is used in multiple fields like bioinformatics, face recognition (Mayumi Oshiro, Santoro Perez and Augusto Baranauskas, 2012) and has been an accredited algorithm in big data science competitions, it does have its disadvantages. It requires a significant computational power to function. Another disadvantage is that although it provides variable importance information (Ishwaran and Lu, 2018), it functions like a "black-box" where complete interpretation of the functioning of the model is not possible (Rigatti, 2017). Since the dataset is of medium size, the computation might not be very heavy and since the aim of the dissertation is to find the values and not dive deep into the effect of the predictors, the disadvantages might not have a huge impact on this study.

XGBoost

XGBoost is a supervised learning algorithm, and XGBoost is a short form of Extreme Gradient Boosting. It is a successor of AdaBoost and it uses boosted decision trees to make predictions (Gohiya, Lohiya and Patidar, 2018). Unlike bagging algorithms, which can produce trees parallelly, boosting algorithms produce trees sequentially (Bauer and Kohavi, 1999). In this algorithm, D regression trees, also known as CART (Classification and Regression Trees) are sequentially made. The $₣(Xi)$ is the total function space of regression trees and fd is the function space for individual regression trees (Pesantez-Narvaez, Guillen and Alcañiz, 2019).

$$\hat{Y}\, i = \mathrm{F}(Xi) = \sum Dd = 1fd(Xi), fd \in \mathrm{F}, i = 1, \dots, n$$

Pseudocode for XGBoost by Ben Jabeur, Stef and Carmona (2022).

```
Input: X vector of features, Y vector of labels, loss function, base learner and number of subtrees
Output: XGBoost prediction model
Steps:
    1. The model iterates over t rounds
    2. Each base learner tree is built in one of the rounds and a prediction is made.
    3. The loss function is calculated based on the actual and predicted value.
    4. Another base learner is run, and the best possible gradient descent stage size is calculated.
    5. The prediction is provided as output.
```

XGBoost is an open source package which is not only scalable but also performs really well with a wide range of problems. It is also an algorithm which has got wide appreciation in several competitions. In a Kaggle competition conducted in 2015, out of the 29 winning solutions, 17 of them used XGBoost (Chen and Guestrin, 2016). Another advantage of this model is that by manually setting the components in the model, we can reduce the bias in the model (Gohiya, Lohiya and Patidar, 2018). XGBoost uses loss function and regularisation to optimise model performance. If the model is predicting the age and the correct age is 25 and the decision tree predicts it to be 15, there is a difference of 10 years. The next tree considers this error and if the age is 6, it could predict the age to be 4, thereby reducing the loss age sequentially. The regularisation function helps the model in reducing the model from overfitting and helps the model in generalising better (Dong et al., 2022).

Even though XGBoost provides coefficient estimates when a linear booster is used, it does not provide any coefficient estimates when a non-linear booster is used. In a manner similar to random forest, the interpretability becomes difficult in these cases (Pesantez-Narvaez, Guillen and Alcañiz, 2019). Despite the algorithm having good performance, it takes high computational time. The hyper parameter tuning of the model requires several iterations and also takes a lot of time (Mitchell and Frank, 2017). As mentioned in the random forest algorithm section, these disadvantages might not have a huge impact on this study considering the scope of the study.

Linear Regression

In a linear regression model, the main aim is to create a linear relation equation between dependent and independent variables. Multivariate regression analysis is when we have multiple independent variables and one dependent variable (Uyanık and Güler, 2013).
The model commonly used method to find β is least squares method (Su, Yan and Tsai, 2019). In Scikit Linear regression, Ordinary least square method is used (Biswal, 2022) The model is formulated using the below equation.

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \varepsilon$$

Where y = dependent variable,
x = independent variable,
β = parameter,
ε = error

Pseudocode for Linear Regression by Leandro (2024).

```
Input: Matrix X of shape nxm, Y vector of target values
Output: Linear regression predictive model
```

Steps:

```
1. For loop where i=1 up to m
        Initialise the weight w for all features and the bias b
2. Predicted value is calculated using $P[i] = w^* x[i] + b$
3. The best fitted line is calculated using least squares method by minimizing $(P[i] - y[i])^2$
```

Linear regression is very easy to implement and interpret. The computational complexity and time complexity is also very less which makes it a very good algorithm if there is linear relationship between the variables (Anandhi and Nathiya, 2023). Since the previous two models work with non-linear data too, a simple linear model will be used to see if there is any difference in performance.
In a linear regression, the model assumes that there is no relationship between each data point. It is also sensitive to outliers. Since most of the real-life datasets are non-linear, there is a possibility that the model would not function well with this dataset. Additionally, as the model is simple, it might not be able to capture the underlying patterns (Iqbal, 2021).

3. Development Methods

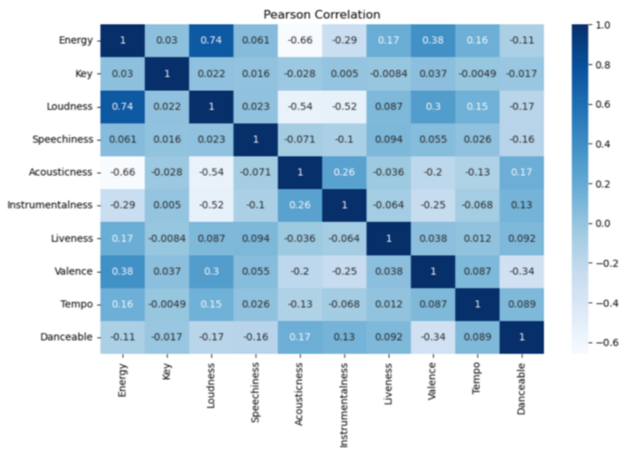
In this section, data cleaning, data modelling, data evaluation and data visualisation methods used in analysing both the questions will be discussed.

3.1 Can we predict if a song is danceable or not danceable?

3.1.1 Data cleaning

Before dealing with missing values, duplicates and outlier, columns needed for the analysis are created and selected. A new column called danceable from the Danceability column is created for the model to predict if each song is danceable or not. The threshold used was 0.68 and the column contains d (danceable) and nd (not danceable) values.

The only columns selected for analysis are the song's characteristics as other columns are not relevant to this analysis. The key column was also dropped as it was a near zero variance column. Additionally, the column has very low linear and monotonic correlation when checked with statistical tests (spearman correlation and Pearson correlation test).



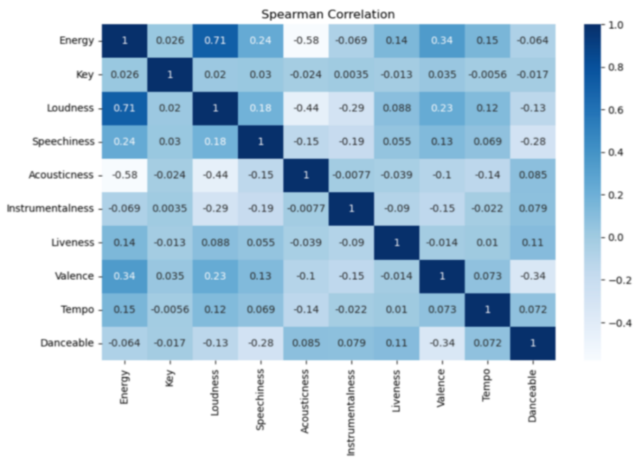
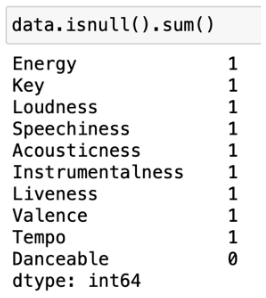


Figure: Pearson and Spearman correlation matrix

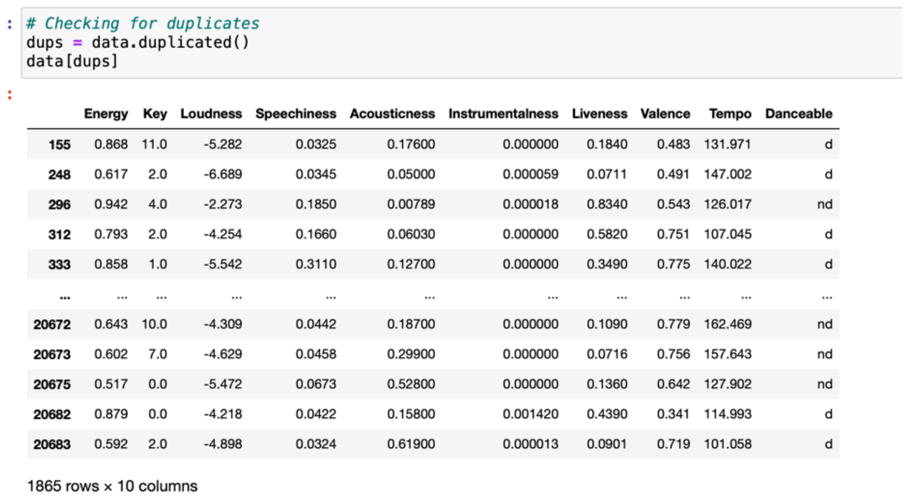
The columns selected for analysis are Energy, Key, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo and Danceable. In the 10 columns used for this analysis, there was only 1 row of missing value and since the number of missing rows were very minute, it was removed using data.dropna(axis=0). The below image shows the missing values in the 10 columns which was checked using data.isnull().sum().



```
data.isnull().sum()

Energy              1
Key                 1
Loudness            1
Speechiness         1
Acousticness        1
Instrumentalness    1
Liveness            1
Valence             1
Tempo               1
Danceable           0
dtype: int64
```

Figure: Count of missing values

The number of duplicated values were checked using data.duplicated() function. The dataset had 1865 duplicated values, and they were all dropped so that it does not provide an inflated accuracy score (Anishnama, 2023).



```
# Checking for duplicates
dups = data.duplicated()
data[dups]
```

	Energy	Key	Loudness	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo	Danceable
155	0.868	11.0	-5.282	0.0325	0.17600	0.000000	0.1840	0.483	131.971	d
248	0.617	2.0	-6.689	0.0345	0.05000	0.000059	0.0711	0.491	147.002	d
296	0.942	4.0	-2.273	0.1850	0.00789	0.000018	0.8340	0.543	126.017	nd
312	0.793	2.0	-4.254	0.1660	0.06030	0.000000	0.5820	0.751	107.045	d
333	0.858	1.0	-5.542	0.3110	0.12700	0.000000	0.3490	0.775	140.022	d
...	...	...	...	...	...	...	...	...	...	...
20672	0.643	10.0	-4.309	0.0442	0.18700	0.000000	0.1090	0.779	162.469	nd
20673	0.602	7.0	-4.629	0.0458	0.29900	0.000000	0.0716	0.756	157.643	nd
20675	0.517	0.0	-5.472	0.0673	0.52800	0.000000	0.1360	0.642	127.902	nd
20682	0.879	0.0	-4.218	0.0422	0.15800	0.001420	0.4390	0.341	114.993	d
20683	0.592	2.0	-4.898	0.0324	0.61900	0.000013	0.0901	0.719	101.058	d

1865 rows × 10 columns

Figure: Rows showing some of duplicated values

Spotify does not provide details of how the scores of each characteristics were given hence manually deciding which score is an outlier would not have been possible without a detailed study on the values and due to this reason the LocalOutlierFactor was used to find the outliers. 66 values were found to be outliers and were removed from the dataset.

The target column Danceable was the only categorical column in the dataset and it was encoded into 0 (d) and 1(nd) values using the label encoder as there were only two values. Checking the statistical distribution of the dataset using data.describe(), there are two columns, Loudness and Tempo which are not in range of 0-1. To make them into 0-1 range, these columns were normalised using MinMaxScaler(). The distribution of each column was checked and only three columns had gaussian distribution and hence the columns were not standardised.

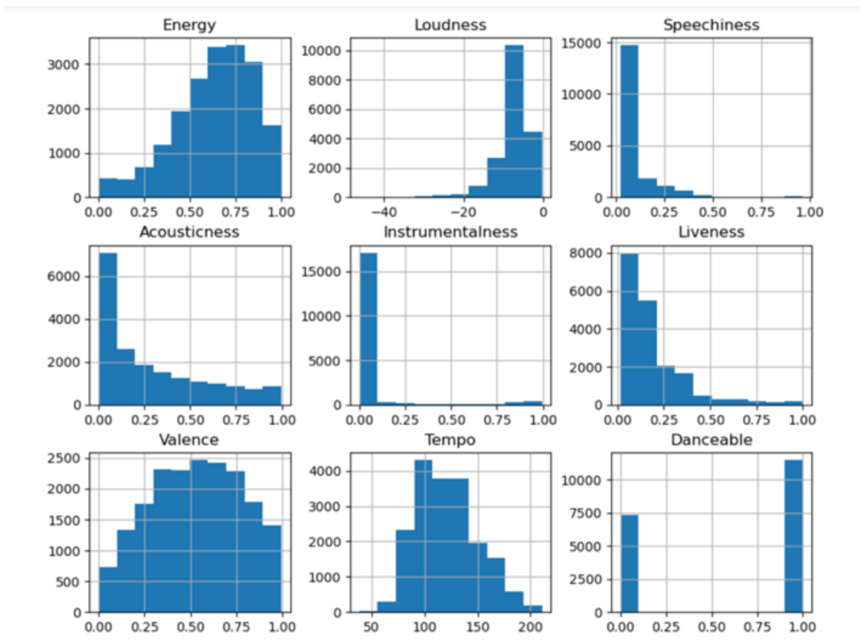
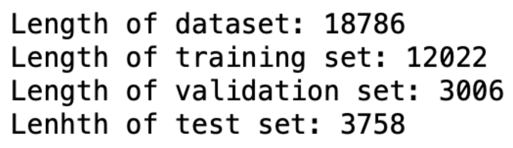


Figure: Data distribution of all the columns used in the analysis

3.1.2 Data modelling

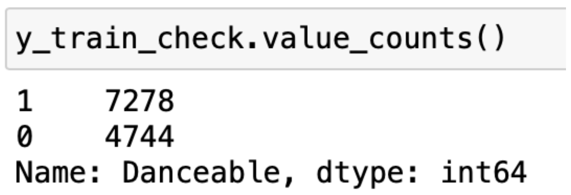
The dataset of 18786 was divided into 80% training set and 20% test set. The 80% of the training set was again divided into 80% train and 20% validation set. The training set will be used to train the models, validation set will be used for hyperparameter tuning and test set will be used for testing how well the model performs with unseen data.



```
Length of dataset: 18786
Length of training set: 12022
Length of validation set: 3006
Lenhth of test set: 3758
```

Figure: Dataset split into validation, test and training set

Support Vector Machine
The Support Vector machine was trained on the training set using regularisation value, c as 0.01 and linear kernel function. A smaller regularisation variable was chosen to allow a larger margin so as to allow the model to be flexible. The target variable has imbalanced classes and the model performed badly with class 0 which has lesser values.



```
y_train_check.value_counts()

1    7278
0    4744
Name: Danceable, dtype: int64
```

Figure: Count of values in each class in target variable

Additionally, Gini impurity value was checked and it was found to be around 0.478. A column with balanced class will have a Gini impurity value of 0.5 (Appaji, 2024). Since the value is less than 0.5, it shows there is an imbalance. An oversampling technique (SMOTE) was used to create samples in class 0 and balance both the classes. The model's performance increased.

The model was also hyperparameter tuned on the validation set using GridSearchCV. The regularisation parameter C, Kernel function and gamma which controls decision boundary was hyperparameter tuned. The model was then fitted onto the training set. The validation was used for GridSearchCV because the training set is quite big and due to the high computational power it was taking several days to complete the complete GridSearchCV. The model was optimised for best precision value as we needed less false positive or non-danceable songs to be predicted as danceable. The best parameters were used to test the model using the test set. The performance further increased even on unseen test set.

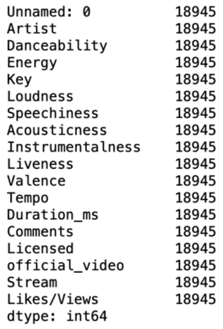
Neural Network
The neural network was trained on the training set with two hidden layers with ReLu activation function and sigmoid output layer. There is also a dropout layer which is a regularising technique and it helps in reducing overfitting (Yadav, 2022). The model was compiled with binary crossentrophy loss and adam optimizer and was fitted onto the training data using 20 epochs. A bigger epoch value was not taken so as to avoid the model from overfitting onto the training set (Marshall, 2024). The model performed better than the SVM model on both training set and test set. To check whether the model is overfitting, the model was also tested on the test set. The model was not underperforming even with test data. It had a slightly lesser but competent score.

Logistic Regression
The logistic regression model was trained with the training set without using any hyperparameters. The model's performance on the training set was average. The model was then hyperparameter tuned on penalty and C value or regularisation value using validation set. The penalty will help in deciding what regularisation will help in introducing bias and reducing variance while generalising (Akalin, 2020). Based on the best estimators, the model was tested on unseen test data and the performance went slightly down compared to neural network and SVM model's performance on the training data.

3.2 Can we predict if a song will have positive engagement or not on YouTube?

3.2.1 Data cleaning

A column named Likes/Views was created based on Likes and views columns for the model to study and predict. The original two columns were then removed from the dataset. The duplicate values were checked using data.duplicated and there were no duplicated values, hence no rows were removed. Using data.isnull.sum() the number of missing values were checked. Due to the lack of knowledge on how the scores are calculated by Spotify, the outliers were not removed manually. Using the LocalOutlierFactor, 604 outliers were removed from the dataset.



```
Unnamed: 0          18945
Artist              18945
Danceability        18945
Energy              18945
Key                 18945
Loudness            18945
Speechiness         18945
Acousticness        18945
Instrumentalness    18945
Liveness            18945
Valence             18945
Tempo               18945
Duration_ms         18945
Comments            18945
Licensed            18945
official_video      18945
Stream              18945
Likes/Views         18945
dtype: int64
```

Figure: The values remaining after using LocalOutlierFactor

The maximum number of missing values in a column was 576 and was in the Stream column. There were 400+ missing values in each of 4 other columns. If the missing values are in multiple rows in each column, deleting the values would result in a large portion of data getting deleted, hence to understand the missing values and their distribution, a missing value matrix was plotted. The below matrix shows that the maximum of the missing values are in the same rows hence all of the missing values were deleted. The total number of rows remaining for analysis were 19549 rows.

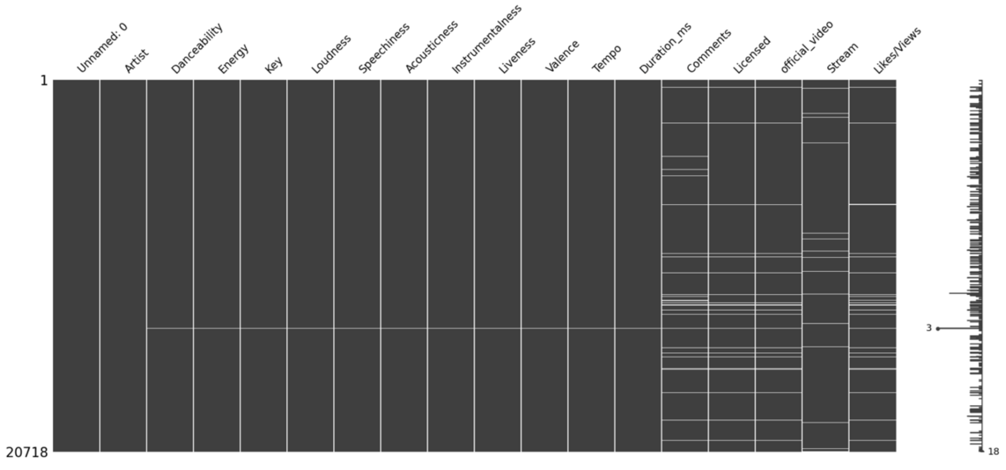


Figure: Missing values matrix

Using the spearman and Pearson correlation matrix, the monotonic and linear correlations were checked between independent and dependent variables. The top 10 labels were chosen based on the correlation values with the Likes/Views column along with the Artist column. The columns selected for analysis were, Artist, Energy, Loudness, Speechiness, Instrumentalness, Valence, Duration_ms, Comments, Licensed, Stream and the target column, Likes/Views column.

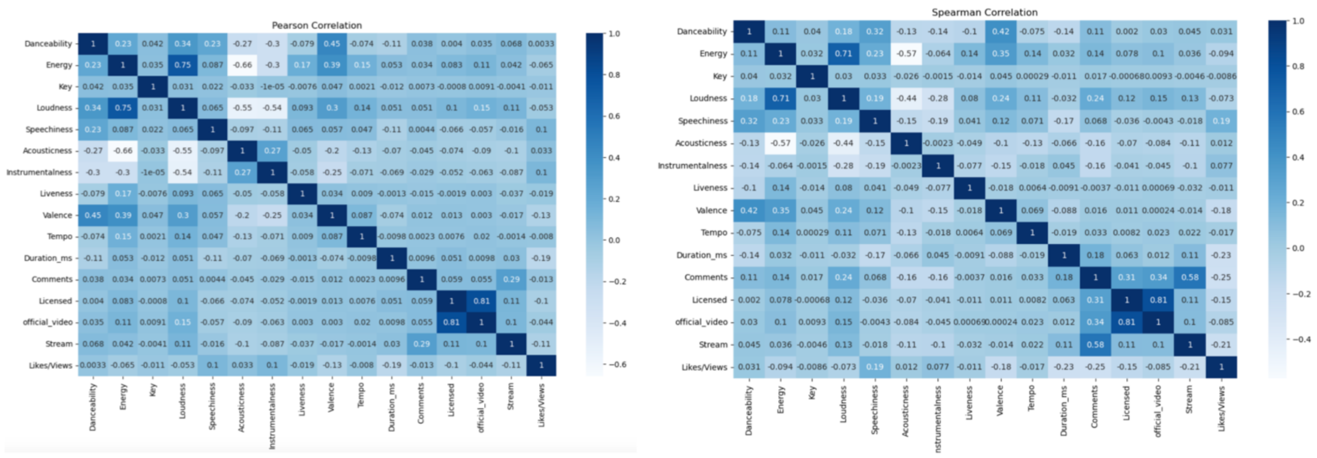


Figure: Pearson and Spearman correlation matrix

Two of the categorical variables, Licensed and Official data were encoded using Label encoder because both columns only had Boolean values. The artist column was encoded using target encoding after the data was split into training, validation and test set to avoid data leak. Target encoding was used to handle the high cardinality of the Artist column. Checking the distribution of all the columns, only two of the columns have Gaussian distribution, hence the dataset was not Standardised.

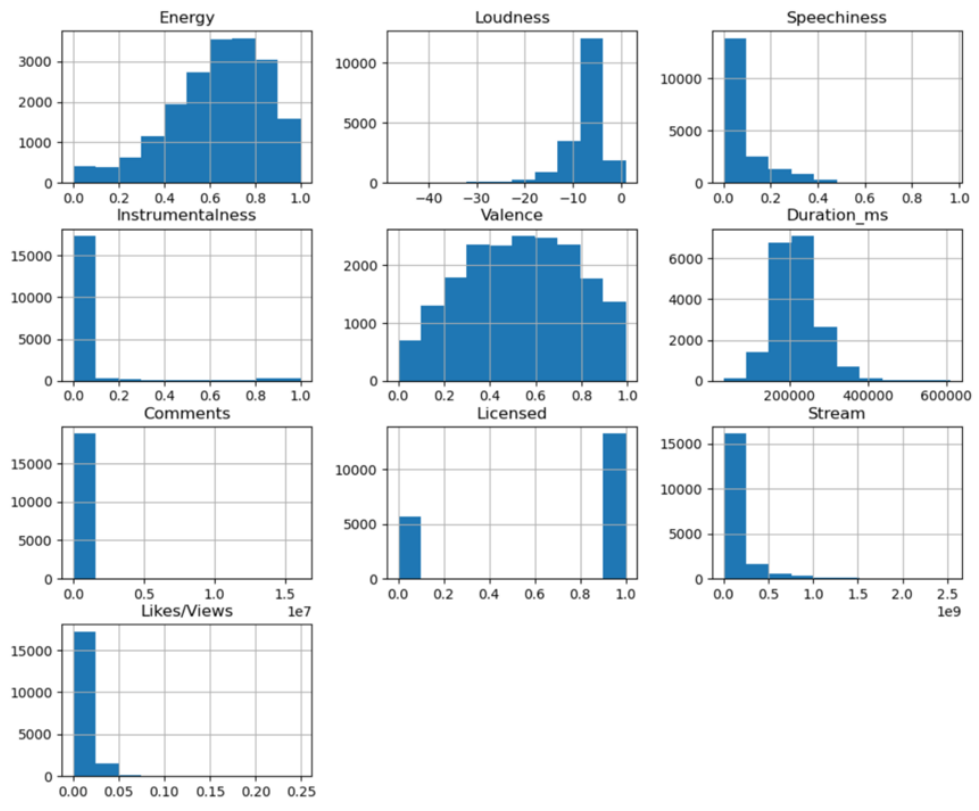
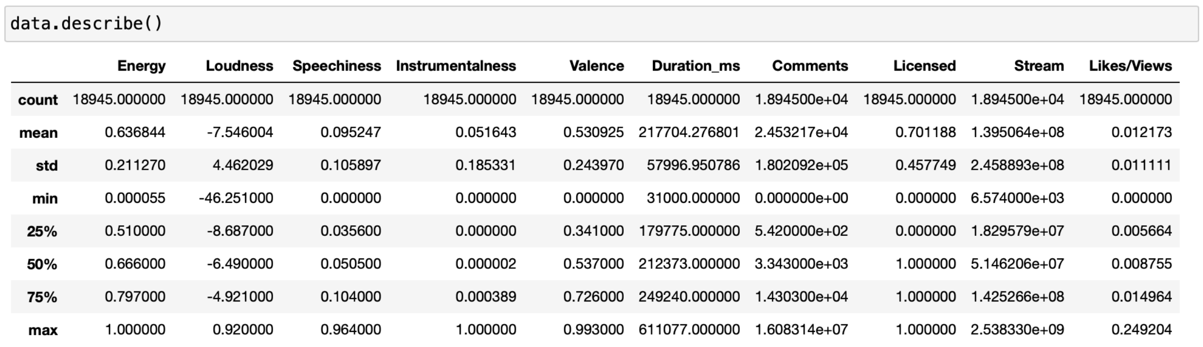


Figure: Histogram showing distribution of each column

All the columns apart from Loudness, Duration_ms, Comments and Stream, the columns are in the range 0-1 hence these were the columns which were Normalised to bring the range in between 0-1. The below image shows the statistical distribution of data.



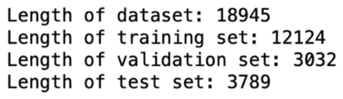
```
data.describe()
```

	Energy	Loudness	Speechiness	Instrumentalness	Valence	Duration_ms	Comments	Licensed	Stream	Likes/Views
count	18945.000000	18945.000000	18945.000000	18945.000000	18945.000000	18945.000000	1.894500e+04	18945.000000	1.894500e+04	18945.000000
mean	0.636844	-7.546004	0.095247	0.051643	0.530925	217704.276801	2.453217e+04	0.701188	1.395064e+08	0.012173
std	0.211270	4.462029	0.105897	0.185331	0.243970	57996.950786	1.802092e+05	0.457749	2.458893e+08	0.011111
min	0.000055	-46.251000	0.000000	0.000000	0.000000	31000.000000	0.000000e+00	0.000000	6.574000e+03	0.000000
25%	0.510000	-8.687000	0.035600	0.000000	0.341000	179775.000000	5.420000e+02	0.000000	1.829579e+07	0.005664
50%	0.666000	-6.490000	0.050500	0.000002	0.537000	212373.000000	3.343000e+03	1.000000	5.146206e+07	0.008755
75%	0.797000	-4.921000	0.104000	0.000389	0.726000	249240.000000	1.430300e+04	1.000000	1.425266e+08	0.014964
max	1.000000	0.920000	0.964000	1.000000	0.993000	611077.000000	1.608314e+07	1.000000	2.538330e+09	0.249204

Figure: Statistical distribution of each column

3.2.2 Data modelling

The dataset was split into 80% training set and 20% test set. The training set was again split into 80% training set and 20% validation set. The training set will be used for training the model, validation set will be used for hyperparameter tuning and the test set will be used to see the model's ability to generalise.



```
Length of dataset: 18945
Length of training set: 12124
Length of validation set: 3032
Length of test set: 3789
```

Figure: Length of training, test and validation set

Random Forest

The random forest model was trained on training data using 10 trees or n_estimators in the forest. The model performed extremely well on the training data. Since the Artist column was encoded with target encoding, there is a high possibility that there was data leak and to check it, the model needs to be checked to see if it can generalise well too. Before the model is tested with test data, it is hyperparameter tuned with RandomSearchCV using the parameter n-estimator, max_depth and max_features using validation set. Even after hyperparameter tuning the model, the R^2 value did not improve.

XGBoost

The model was trained on the training data with n_estimators taken as 10, max_depth taken as 10 and learning rate(eta) as 0.1. The maximum depth and n_estimators are chosen to be smaller values to avoid overfitting (RITHP, 2023). The model was also hyperparameter tuned with RandomSearchCV with the same parameters used in the training model. The model performed well with a high R^2 value on training data but like random Forest, the model failed to generalise well with test data.

Linear Regression

The model performance was drastically lower than random forest and XGBoost model's performance on training set. The model was not hyperparameter tuned as linear regression lack of hyperparameters. The model's performance on test set sharply reduced implying that the model was unable to learn any of the complex patterns in the data.

The performance of all the six models will be discussed in detail in the next section.

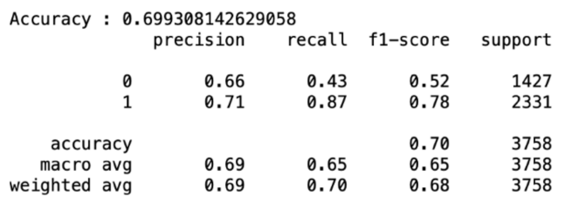
4. Experimental Results, Analysis and Conclusions

In this section the results and conclusions of all 6 models will be discussed along with the data visualisations.

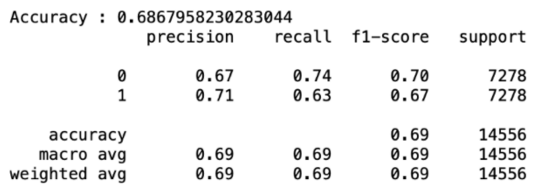
4.1 Can we predict if a song is danceable or not danceable?

4.1.1 Data evaluation

Support Vector Algorithm
The model had an accuracy score of around 0.69 before the oversampling was done. The recall and F1 score were really less. After the classes were balanced, the overall accuracy reduced from around 0.69 to 0.68 but the model had better recall and f1 score on the minority class which was underperforming previously.



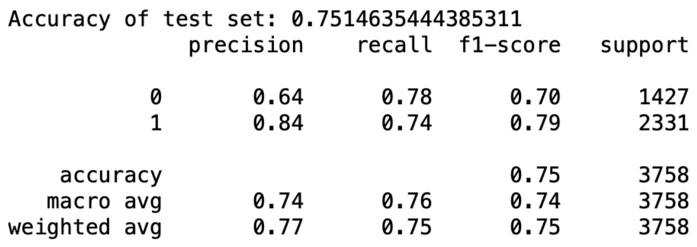
```
Accuracy : 0.699308142629058
              precision    recall  f1-score   support

           0       0.66      0.43      0.52      1427
           1       0.71      0.87      0.78      2331

    accuracy                           0.70      3758
   macro avg       0.69      0.65      0.65      3758
weighted avg       0.69      0.70      0.68      3758
```



```
Accuracy : 0.6867958230283044
              precision    recall  f1-score   support

           0       0.67      0.74      0.70      7278
           1       0.71      0.63      0.67      7278

    accuracy                           0.69     14556
   macro avg       0.69      0.69      0.69     14556
weighted avg       0.69      0.69      0.69     14556
```

Figure: Accuracy and classification report on before and after oversampling (SVM)

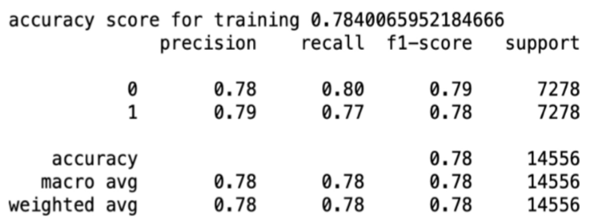
Once the model was hyperparameter tuned, the accuracy score improved significantly on the test set. It shows that the model generalises quite well to unseen data and the hyperparameter tuning helped in selecting the best estimators.



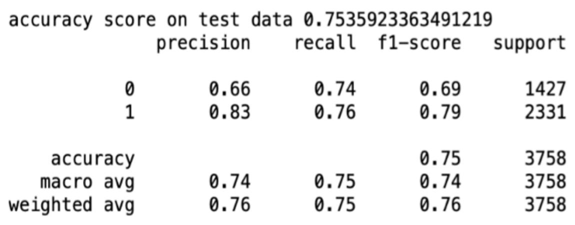
```
Accuracy of test set: 0.7514635444385311
              precision    recall  f1-score   support

           0       0.64      0.78      0.70      1427
           1       0.84      0.74      0.79      2331

    accuracy                           0.75      3758
   macro avg       0.74      0.76      0.74      3758
weighted avg       0.77      0.75      0.75      3758
```

Figure: Accuracy and classification report on test data after hyperparameter tuning (SVM)

Neural Network algorithm
The model had an accuracy score of around 0.78 on the training set. The model had a better recall value compared to precision value for the danceable class showing that the model might have more false positives. When the model was tested on the test set, the accuracy score went down, not drastically, but to around 0.75.



```
accuracy score for training 0.7840065952184666
              precision    recall  f1-score   support

           0       0.78      0.80      0.79      7278
           1       0.79      0.77      0.78      7278

    accuracy                           0.78     14556
   macro avg       0.78      0.78      0.78     14556
weighted avg       0.78      0.78      0.78     14556
```



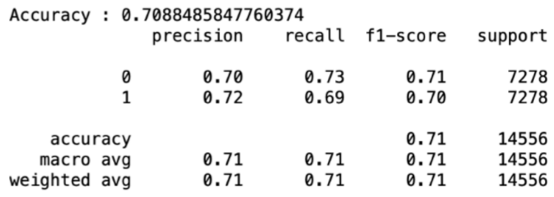
```
accuracy score on test data 0.7535923363491219
              precision    recall  f1-score   support

           0       0.66      0.74      0.69      1427
           1       0.83      0.76      0.79      2331

    accuracy                           0.75      3758
   macro avg       0.74      0.75      0.74      3758
weighted avg       0.76      0.75      0.76      3758
```

Figure: Accuracy and classification report on train and test set (Neural Network)

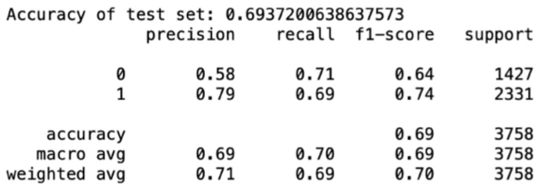
Overall, the neural network model performed better than the SVM model but the difference between the scores of SVM after hyperparameter tuning and Neural network was very slight. The models are performing better than anticipated considering the Pearson and Spearman correlation scores between independent and dependent variables were low.

Logistic Regression

The model had a accuracy score of around 0.69 on test data while it had 0.70 on training data. This shows that the model was not overfitting on training set. However, the model's precision score for class 0 went further down to around 0.58 on test data compared to other two models.



```
Accuracy : 0.7088485847760374
              precision    recall  f1-score   support

           0       0.70      0.73      0.71      7278
           1       0.72      0.69      0.70      7278

    accuracy                           0.71     14556
   macro avg       0.71      0.71      0.71     14556
weighted avg       0.71      0.71      0.71     14556
```



```
Accuracy of test set: 0.6937200638637573
              precision    recall  f1-score   support

           0       0.58      0.71      0.64      1427
           1       0.79      0.69      0.74      2331

    accuracy                           0.69      3758
   macro avg       0.69      0.70      0.69      3758
weighted avg       0.71      0.69      0.70      3758
```

Figure: Accuracy and classification report on train and test set (Logistic Regression)

Even though the overall score was between around 0.69 to 0.75 for all models, the precision score is still only around 0.64, 0.66 and 0.58 for SVM, neural network and logistic regression respectively for danceable class. It shows that the model is not very accurate in making true positive danceable classifications. This can be an issue when this algorithm is used in a music recommendation system. For example, if 100 songs were labelled as danceable by SVM, neural network and logistic regression only 64 or 66 or 58 songs will be actually danceable. The SMOTE technique helped improve the scores of minority class, but it still is not helping in getting the scores high enough to be implemented in real life scenarios. The model might perform better if more danceable songs are taken for analysis rather than using oversampling techniques to balance the classes. The issue could also be the higher danceability threshold.

4.1.2 Data visualisation

Danceability distribution

To understand the distribution of the danceability a graph was plotted with Danceability on the x axis and Tempo on the y axis. Tempo was chosen and not index of the dataset because the index vs Danceability does not show the complete distribution of Danceability as the index is around 20,000. Tempo is also related to danceability so it shows a better distribution and the Tempo only varies from 0 to 250. The visualisation shows the margin between danceable and not danceable classes. The danceable class is denoted by green and not danceable is marked with red. The class imbalance between both the classes can also be seen from the below graph.

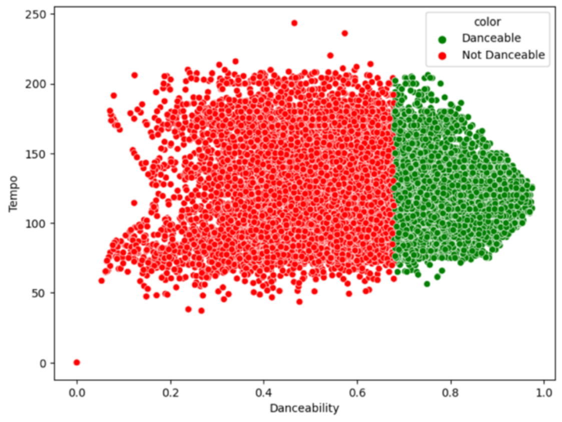


Figure: Danceability distribution vs Tempo

Learning Curve

The learning curve can be used to understand how the model performs over time. It can also help in understanding if the model is overfitting or not (Brownlee, 2019). If the model performs well with train set but performs badly with test set, the model could be overfitting (Brownlee, 2020).

In the SVM the train set performs well from the beginning but the test set learns over the time. The difference or variability between test and train set was a lot in the beginning for the SVM model but it learns over the period of time and achieves the same accuracy showing that the model is not overfitting on the train set. In neural network, the model has an oscillatory graph on both the test and the train set. The difference between both is not a lot but the difference remains till the end. The model performs well on both train and test set indicating that the model is not overfitting. In linear regression model, the model's performance declined as more data was provided showing the model was underfitting and the model is not complex enough to learn the underlying patterns. The performance on test data improved with time implying that the model could perform better if more data is provided.

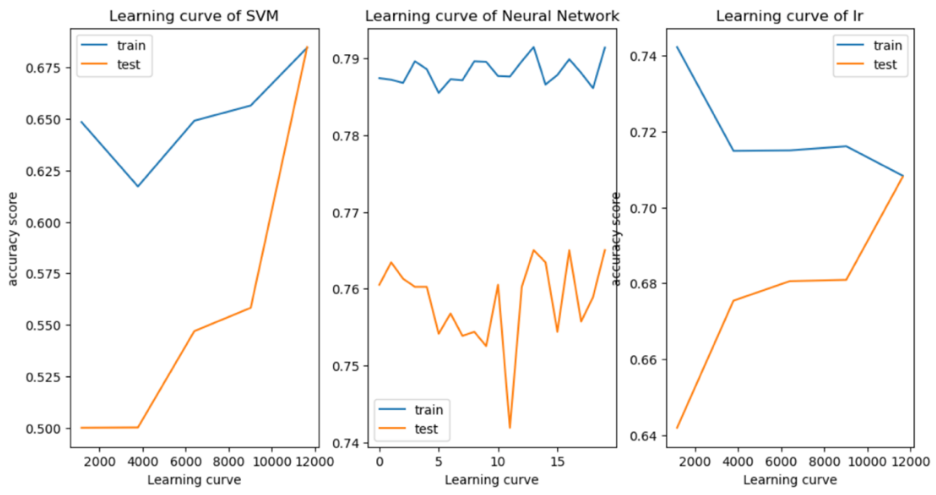


Figure: Learning curve of SVM, Neural Network and Linear Regression

ROC Curve

An Receiver Operator Characteristic(ROC) curve shows the ability of a binary classifier and is plotted between true positive rate (sensitivity) and false positive rate(1- specificity) (Chan, 2018). In the below figure, it can be seen that the ROC curve of neural network, support vector machine and the linear regression algorithm are bowing towards the top left indicating that the models are performing better than a random classifier.

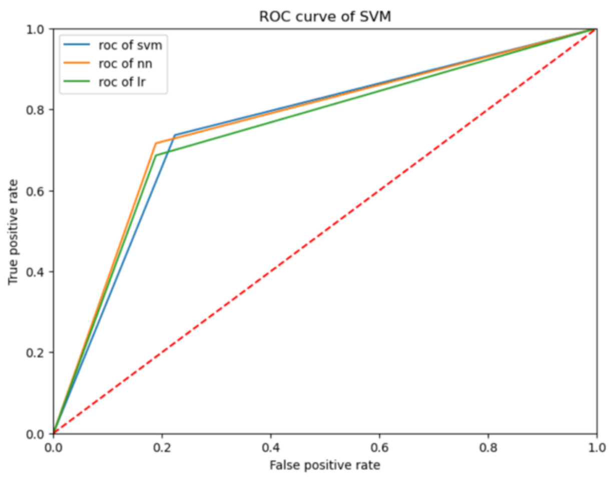


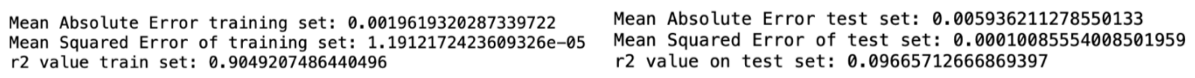
Figure: ROC curve of Neural network and SVM

4.2 Can we predict if a song will have positive engagement or not on YouTube?

4.2.1 Data evaluation

Random Forest

The random forest model had around 0.90 R^2 score on the train set. The MAE score and MSE score were also low showing that the model was very performing well on training set. After hyperparameter tuning, the model was tested on test set and the R^2 score reduced to 0.098 even though the MAE and MSE score were not high compared to the mean value of the target variable of around 0.012.



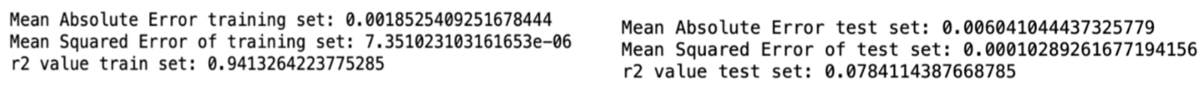
```
Mean Absolute Error training set: 0.0019619320287339722
Mean Squared Error of training set: 1.1912172423609326e-05
r2 value train set: 0.9049207486440496
```

```
Mean Absolute Error test set: 0.005936211278550133
Mean Squared Error of test set: 0.00010085554008501959
r2 value on test set: 0.09665712666869397
```

Figure: MAE, MSE and R^2 value of train and test set (random forest)

XGBoost

Like the random Forest model, the model had performed well with the training set. It scored 0.94 R^2 score and MAE score of around 0.001. When the model was hyperparameter tuned and tested with the test data, the model had scored only around 0.07.



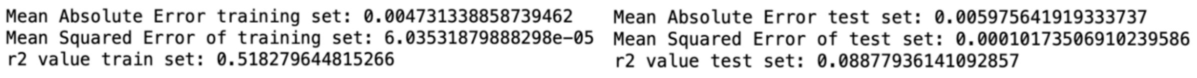
```
Mean Absolute Error training set: 0.0018525409251678444
Mean Squared Error of training set: 7.351023103161653e-06
r2 value train set: 0.9413264223775285
```

```
Mean Absolute Error test set: 0.006041044437325779
Mean Squared Error of test set: 0.00010289261677194156
r2 value test set: 0.0784114387668785
```

Figure: MAE, MSE and R^2 value of train and test set (XGBoost)

Linear regression

The linear regression model's R^2 value was only around 0.51 on training set, compared to both the other model's 0.90+. The model's R^2 value considerably dropped on the test set to around 0.088.

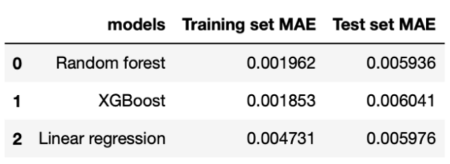


```
Mean Absolute Error training set: 0.004731338858739462
Mean Squared Error of training set: 6.03531879888298e-05
r2 value train set: 0.518279644815266
```

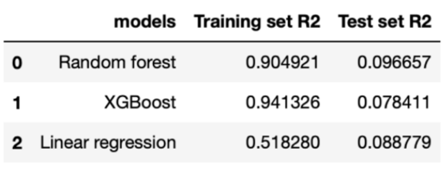
```
Mean Absolute Error test set: 0.005975641919333737
Mean Squared Error of test set: 0.00010173506910239586
r2 value test set: 0.08877936141092857
```

Figure: MAE, MSE and R^2 value of train and test set (linear regression)

Both the random forest and XGBoost models performed extremely well with the training set, but performed badly with test set, implying that the models were unable to generalise well. The high training set score of R^2 also shows that the models were overfitting on to the training data. The target encoder can often cause data leak, which could have resulted in overfitting of data on training set. The performance decreased on the training set using linear regression model implying the model could have been too simple to learn the underlying patterns. However, with test set, the models did not have high MAE or MSE score compared to the mean of the target variable (around 0.012), indicating that the predicted values were close to the actual values.



	models	Training set MAE	Test set MAE
0	Random forest	0.001962	0.005936
1	XGBoost	0.001853	0.006041
2	Linear regression	0.004731	0.005976



	models	Training set R2	Test set R2
0	Random forest	0.904921	0.096657
1	XGBoost	0.941326	0.078411
2	Linear regression	0.518280	0.088779

Figure: Comparison of MAE and R^2 values of random forest, XGBoost and linear regression model on train and test set

While checking both the linear and monotonic correlation using Spearman and Pearson correlation matrix in data cleaning section, none of the independent features had good correlation with the target variable, the highest correlation was -0.23 which was between Duration_ms showing the features used might not be suitable for predicting the target variable. The models especially the random forest and XGBoost might perform better with features which have more correlation with the target variable so as to improve the model's ability to understand the underlying pattern.

4.2.2 Data visualisation

Learning Curve

The learning curve shows the difference in performance of the models on train and test set. In the random forest model, the model performed a bit better but the performance fell as more training data was provided. The XGBoost model's performance on test data slightly improved over time without reducing, suggesting the possibility that the model might perform well with more amount of data. In the linear regression, the learning curve dramatically dropped with more data implying that the model was too simple and a more complex model is needed.

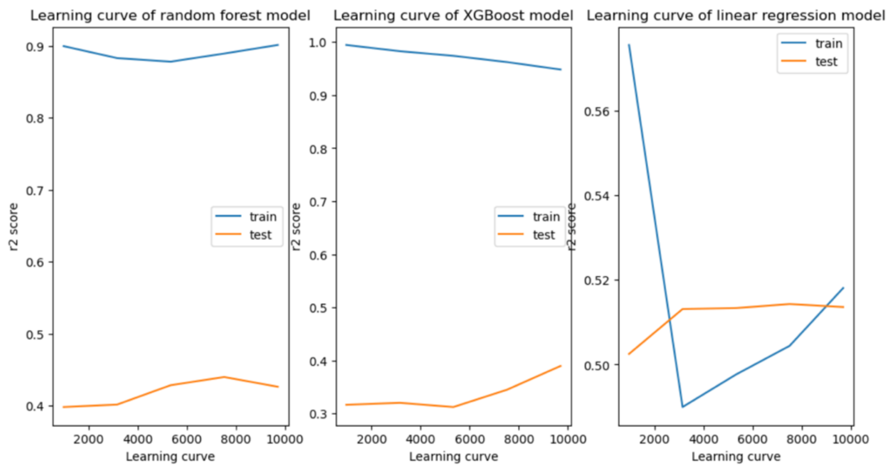


Figure: Learning curve of random forest, XGBoost and linear regression

Actual Vs Predicted Value graph

From the actual vs predicted value graph, even though difference between the actual and predicted values are not a lot in all three models, the models were unable to predict the bigger values. The actual values in the test set lies within 0 to 0.183, however most of the predicted values are between 0 and 0.050. The highest predicted value is between 0.05 and 0.10.

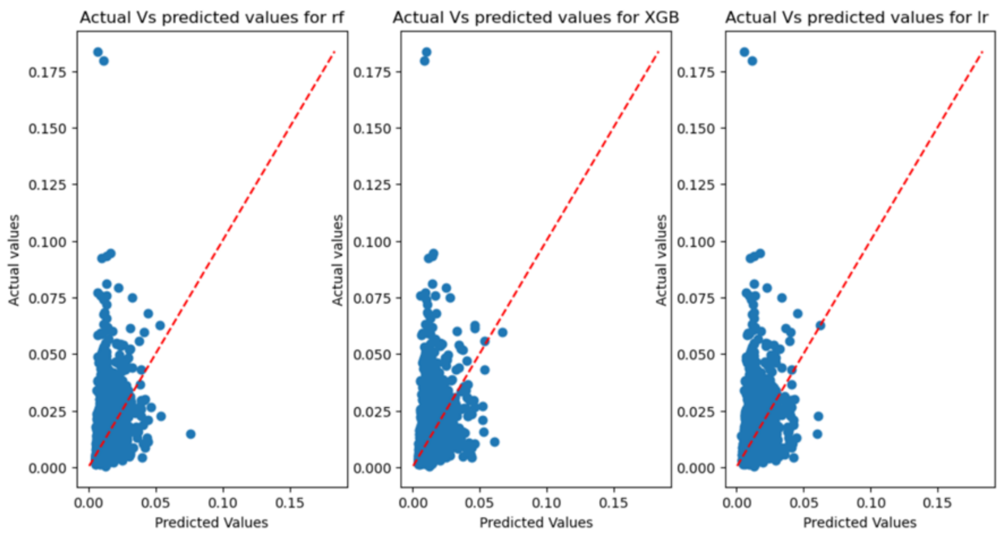


Figure: Actual Vs Predicted graph of random forest, XGBoost and linear regression

Residual Vs Fitted graph

In the residual Vs fitted graph, it can be seen that the points are fanning out indicating that there is heteroscedasticity in graphs for all three models. Heteroscedasticity means that the variance between errors is not constant (Frost, 2017). There are also two points which were predicted closer to 0.01 and has the highest residual value which is close to 0.15. These points are above the horizontal line indicating that the models have under-predicted these values.

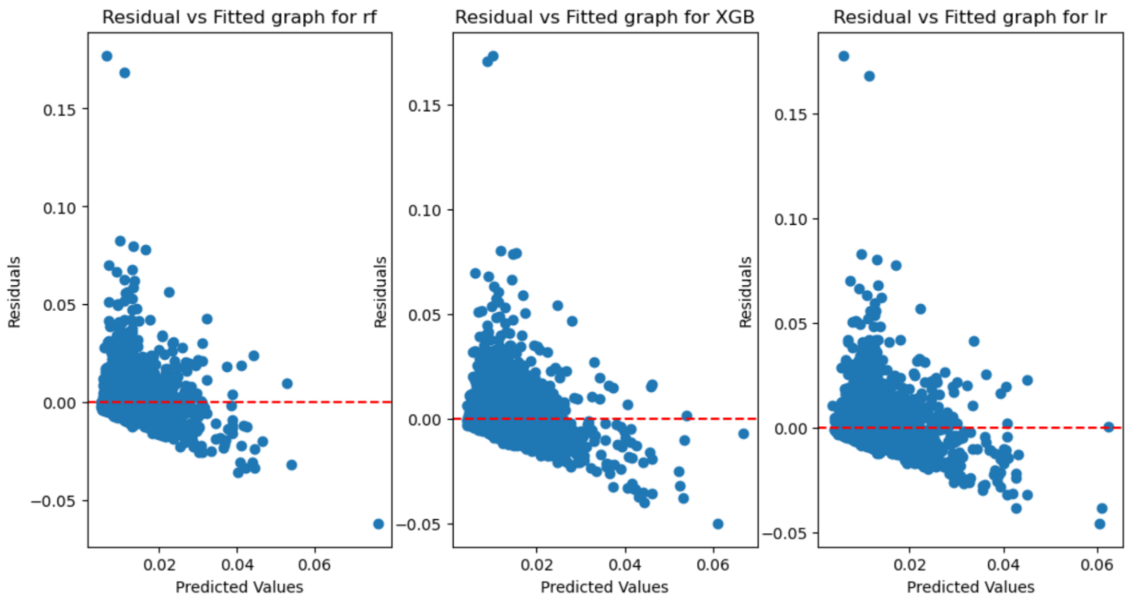


Figure: Residual Vs Fitted graph of random forest, XGBoost and linear regression

5. Reflection

In this project, I worked on two questions, one was classification problem related to Danceability score which was provided by Spotify and the other question was a regression problem, related to Likes to Views ratio provided by YouTube. While working on these questions, I faced a lot of challenges which helped me in building my technical skill as well as gaining perspective into cleaning, transforming, modelling and visualisation of data.

Overall the project went well, both of the objectives set for the project were met. Meeting the timeline set by my supervisor helped me in completing my code and report on time. It helped me in appreciating the importance of planning before starting a project. The classification of the song based on danceability was done by three models. The SVM and neural network models had an accuracy of 75% and above while logistic regression had a slightly lesser score. The likes to view ratio was predicted by random forest, XGBoost and linear regression. The MAE score was only around 0.005 to 0.006 for all the three models with mean of the actual values being 0.01, indicating that the predictions were close to the actual values. However, there are still areas of improvement in both of the questions which will be discussed subsequently in the next section along with further details about each question.

5.1 Can we predict if a song is danceable or not danceable?

While working on the data cleaning process of the danceability classification, I learnt how to find outliers when the values are not interpretable directly without examining each audio. Additionally, It gave me a chance to work with Gini impurity value and understand how it can be used to find imbalance in a dataset.

Since there was no previous research on the danceability threshold, it was hard to decide on a threshold to divide the values into danceable and not danceable. I had previously decided to use the mean of the dataset as the threshold but later changed it to an average of 50th and 75th percentile score so that the threshold is valid for songs outside this dataset too. While the generalising ability of the model increased, it resulted in two imbalanced classes. After checking with the SVM algorithm, I concluded that even though the model's overall accuracy was not bad, the model performed badly with the danceable class which was the minority class. The classes were balanced using a synthetic oversampling technique called SMOTE. The algorithms had better precision score after oversampling on the minority class. The model still did not have a competitive precision score on danceable class indicating that there will be some false positives or songs which are not danceable being predicted as danceable. I tried hyperparameter tuning using GridSearchCV for the entire training set, however due to hardware and time constraints, the GridSearchCV could not be completed and due to this I hyperparameter tuned the model using a smaller validation set. While working on SVM, the model was hyperparameter tuned on C, Kernel and gamma values. This project helped me in understanding how C and gamma values impact the model's performance. A larger value of C would have caused the margin to be smaller allowing lesser misclassification and larger value of gamma would lead to boundaries based on each point closer to the value which could both result in overfitting of the model on training data. The values of C and gamma I used for tuning were between 0.1 and 10 and 0.01 and 1 respectively. After hyperparameter tuning, the model's performance increased slightly when tested on an unseen test set. Furthermore, I used a neural network with two hidden layers and one dropout layer to avoid overfitting. The model generalised better than the SVM model. The performance of logistic regression model underperformed slightly compared to the other two models, this could be indication that the model was too simple and was unable to learn the complex patterns in the data. Even after I balanced the classes and hyperparameter tuned the models, the precision scores of the minority danceable class were less compared to the score of not danceable class in all three models, implying that even though the overall accuracy was not bad, the model might still predict few not danceable songs as danceable, which could be an issue if the model is used for music recommendation.

Based on the experience working on this project, I believe that the model could perform better if the model could be trained with larger danceability data values as the model performs better on the not danceable class. The precision score of danceability class might improve with this. The models could also be benefited by hyperparameter tuning on training set rather than the validation set. However, with more data, the model might need more computational power for hyperparameter tuning, a cloud service like EC2 could be used for better data analysis. Additionally, since the threshold was decided based on taking mean of 50th and 75th percentile of Danceability, this could actually be resulting in threshold being on higher side. A study could be conducted to understand the perception of danceability of songs to get a more user based threshold. A clustering algorithm like K-means might be helpful to group the songs into danceable and not danceable, which could divide the songs more efficiently. Furthermore, if we have a better understanding of the danceability score, it could be used for sub-classification of danceable songs into various genres.

5.2 Can we predict if a song will have positive engagement or not on YouTube?

I chose the likes/views as the target variable so as to find the positive engagement of each video on YouTube. Since the dataset had a high cardinality column, I got the chance to use target encoding while I used label encoding for the column with Boolean values. Additionally, I got to use a missing value matrix. Even though there were only around 500 missing values, the matrix was plotted to see if the missing values in different columns were concentrated in same rows. This was done to avoid deletion of multiple values if the missing values were distributed in multiple columns.

Working on this question made me realise the importance of finding variables with good correlation to the dependent variable. Even though linear and monotonic correlation were inspected using Pearson and Spearman correlation, it was seen that none of the independent variables had good correlation with the target variable. This reflected on the R^2 value of all three regression models. Even after the models were optimised using hyperparameter tuning, R^2 value did not improve. The random forest model was first hyperparameter tuned using GridSearchCV but due to the hardware constraints, I switched to RandomizedSearchCV. The hyperparameters used were total number of the trees, maximum depth of each tree and maximum number of features used for building the trees. The total number of trees checked in hyperparameter tuning was between 10 to 300, the maximum depth chosen was between 5 to 15 and the maximum number of features selected were between 0 and 100%. The XGBoost model was hyper parameter tuned on maximum number of trees, maximum depth of each tree and the learning rate. The n_estimators chosen for hyperparameter tuning was between 10 to 300, max_depth between 2 to 10 and eta or learning rate between 0.01 and 0.3. The n_estimators and max_depth for both models were not chosen to be a large number to avoid overfitting of the model on the noise. The higher the number of the n_estimators and max_depth, the computational power also increases. As the learning rate become bigger, the impact of each tree increases as the model would try to learn faster as the trees are built, to avoid this, the model was hyperparameter tuned on lower values of learning rate. The R^2 values were less on test set but checking MAE values, I saw that they were also less for all three models. The MAE values ranged between 0.005 and 0.006 showing that the predicted values were close to the actual values. The low R^2 value could be because of the small range of values of target variable. This made me understand the importance of using multiple metrics to check the performance of the model. While the model was unable to explain the variability in the values, the low MAE score shows that it still can predict values quite close to the actual values. In my opinion, if the dataset has more features with better correlation to the target variable, the model could be able to explain the variability.

Likes and Views separately have good linear correlation with Streams but when Likes/Views ratio is taken, the correlation diminishes significantly. I thought this could be due to range of Views being substantially bigger than Likes values and when Likes values were divided with Views, the nuances are lost. I tried multiplying Likes values with 1000 and then dividing with Views, however there was no significant increase in the R^2 value of the models. I also tried creating the Likes/Views column after normalising both the columns separately, however some of the values changed to Nan or Inf values and because of this reason, this method was not further explored. In future, two models could be used to predict Likes and Views separately and the likes to views ratio can be calculated based on it. This could help in having better chances at predicting the likes to views ratio. The below graph shows the correlation between Stream, Likes, Views and Likes/views.

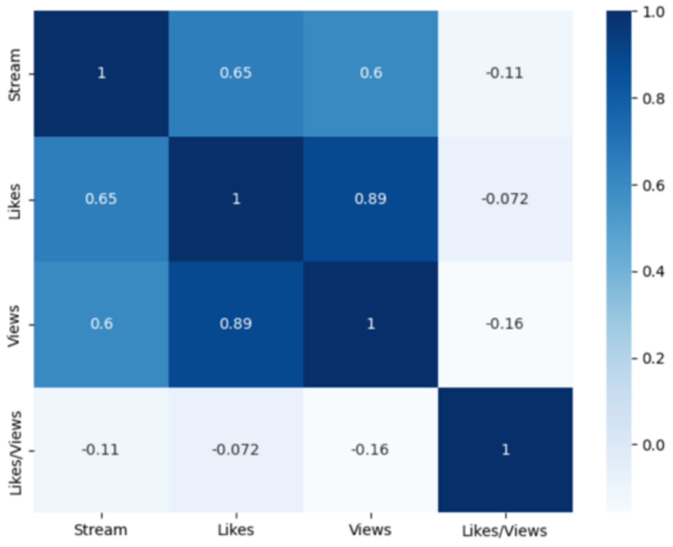


Figure : Linear correlation (Pearson's) between Stream, Likes, Views, Likes/views

Considering positive comments from YouTube along with Likes could also help in improving the ratio have more correlation with the target variable. As the views on YouTube could also be impacted by other social media posts, it is worth considering the views on social networking sites like Twitter and TikTok to predict the Likes to Views ratio for future studies.

The project gave me an opportunity to work on my skills and improve them along the way. In the beginning, every time I hit a road block, I was disheartened. Over time, I became more patient and resilient. It made me focus on solving the problem and organising my timeline. I believe that these skills would benefit me while working on future projects. Furthermore, I got the opportunity of working with a bigger dataset compared to the smaller ones I had experience with previously and the report provided me with exposure with explaining the concepts in non-technical terms.

References

Acock, A.C. (2024). Working With Missing Values. *Wiley.com*. [online] doi:https://doi.org/10.1111/j.1741-3737.2005.00191.x.

Aggarwal, A., Shiva Prasad, K., Xu, Z., Feyisetan, O. and Teissier, N. (2021). *Label Inference Attacks from Log-loss Scores*. [online] arXiv.org. Available at: https://arxiv.org/abs/2105.08266 [Accessed 24 Sep. 2024].

Akalin, A. (2020). *5.13 Logistic regression and regularization | Computational Genomics with R*. [online] Github.io. Available at: https://compgenomr.github.io/book/logistic-regression-and-regularization.html [Accessed 8 Nov. 2024].

Amer, M. (2022). *Classification Evaluation Metrics: Accuracy, Precision, Recall, and F1 Visually Explained*. [online] Cohere. Available at: https://cohere.com/blog/classification-eval-metrics [Accessed 24 Sep. 2024].

Andre , P. (2024). *IMS Business Report 2024: Global dance music industry valued at $11.8 billion*. [online] Musicweek.com. Available at: https://www.musicweek.com/live/read/ims-business-report-2024-global-dance-music-industry-valued-at-11-8-billion/089673#:~:text=According%20to%20the%20report%2C%20the,%2Fclubs%2C%20recordings%20and%20publishing. [Accessed 14 Sep. 2024].

Anishnama (2023). *How Duplicate entries in data set leads to ovetfitting?* [online] Medium. Available at: https://medium.com/@anishnama20/how-duplicate-entries-in-data-set-leads-to-ovetfitting-2e3376e309c5 [Accessed 20 Sep. 2024].

Appaji, N. (2024). *A Tale of Two Metrics: Gini Impurity and Entropy in Decision Tree Algorithms*. [online] Medium. Available at: https://medium.com/@niranjan.appaji/a-tale-of-two-metrics-gini-impurity-and-entropy-in-decision-tree-algorithms-7f61d46baacc#:~:text=The%20resulting%20Gini%20impurity%20ranges,evenly%20distributed%20among%20all%20classes). [Accessed 1 Nov. 2024].

Backlinko. (2023). *Spotify User Stats (Updated June 2024)*. [online] Available at: https://backlinko.com/spotify-users [Accessed 13 Sep. 2024].

Bailey, A.A., Bonifield, C.M. and Arias, A. (2018). Social media use by young Latin American consumers: An exploration. *Journal of Retailing and Consumer Services*, 43, pp.10–19. doi:https://doi.org/10.1016/j.jretconser.2018.02.003.

Barto, A. and Dietterich, T. (2004). *Reinforcement Learning and its Relationship to Supervised Learning*. [online] Available at: https://web.engr.oregonstate.edu/~tgd/publications/Barto-Dietterich-03.pdf.
Bauer, E. and Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, [online] 36(1/2), pp.105–139. doi:https://doi.org/10.1023/a:1007515423169.

Ben Jabeur, S., Stef, N. and Carmona, P. (2022). Bankruptcy Prediction using the XGBoost Algorithm and Variable Importance Feature Engineering. *Computational Economics*, [online] 61(2), pp.715–741. doi:https://doi.org/10.1007/s10614-021-10227-1.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, [online] 25(2), pp.197–227. doi:https://doi.org/10.1007/s11749-016-0481-7.

Biswal, A. (2022). *Sklearn Linear Regression*. [online] Simplilearn.com. Available at: https://www.simplilearn.com/tutorials/scikit-learn-tutorial/sklearn-linear-regression-with-examples [Accessed 8 Nov. 2024].

Boswell, D. (2002). *Introduction to Support Vector Machines*. [online] Available at: http://pzs.dstu.dp.ua/DataMining/svm/bibl/IntroToSVM.pdf.

Boukerche, A., Zheng, L. and Alfandi, O. (2021). Outlier Detection. *ACM Computing Surveys*, 53(3), pp.1–37. doi:https://doi.org/10.1145/3381028.

Brownlee, J. (2019). *How to use Learning Curves to Diagnose Machine Learning Model Performance - MachineLearningMastery.com*. [online] MachineLearningMastery.com. Available at: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/ [Accessed 1 Nov. 2024].

Brownlee, J. (2020). *How to Identify Overfitting Machine Learning Models in Scikit-Learn - MachineLearningMastery.com*. [online] MachineLearningMastery.com. Available at: https://machinelearningmastery.com/overfitting-machine-learning-models/ [Accessed 1 Nov. 2024].

Buuck, B. (2024). *Python vs. SQL: A Deep Dive Comparison*. [online] Software AG. Available at: https://www.softwareag.com/en_corporate/blog/streamsets/python-vs-sql.html [Accessed 27 Sep. 2024].

Canales Luna, J. (2020). *Python vs R for Data Science: Which Should You Learn?* [online] Datacamp.com. Available at: https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference [Accessed 27 Sep. 2024].

Carlisle, S. (2018). Software: Tableau and Microsoft Power BI. *Technology | Architecture + Design*. [online] doi:https://doi.org/10.1080//24751448.2018.1497381.

Carneiro, T., Medeiros Da Nobrega, R.V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V.H.C. and Filho, P.P.R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, 6, pp.61677–61685. doi:https://doi.org/10.1109/access.2018.2874767.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, [online] 408, pp.189–215. doi:https://doi.org/10.1016/j.neucom.2019.10.118.

Chan, C. (2018). *What is a ROC Curve and How to Interpret It*. [online] Displayr. Available at: https://www.displayr.com/what-is-a-roc-curve-how-to-interpret-it/ [Accessed 3 Nov. 2024].

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16(16), pp.321–357. doi:https://doi.org/10.1613/jair.953.

Chen, T. and Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp.785–794.

Chen, Z., Li, C., Journal and Vinicio Sánchez, R. (2015). Multi-layer neural network with deep belief network for gearbox fault diagnosis . 17(5).

Chevallier, M., Nicoleta Rogovschi, Faouzi Boufarès, Nistor Grozavu and Clairmont, C. (2022). Detecting Near Duplicate Dataset with Machine Learning. *International Journal of Computer Information Systems and Industrial Management Applications*, [online] 14, pp.374–385. doi:https://hal.science/hal-03722301.

Chicco, D., Warrens, M.J. and Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, [online] 7, p.e623. doi:https://doi.org/10.7717/peerj-cs.623.

Colliau, T., Rogers, G., Hughes, Z., Ozgur, C., Hughes, Z., Bennie, E. and Myer-Tyson, quot; (2017). MatLab vs. Python vs. R MatLab vs. Python vs. R. *Journal of Data Science*, [online] 15, pp.355–372. Available at: https://scholar.valpo.edu/cgi/viewcontent.cgi?article=1049&context=cba_fac_pub.

Davies, D. (2024). *Meet The 7 Most Popular Search Engines In The World*. [online] Search Engine Journal. Available at: https://www.searchenginejournal.com/seo/meet-search-engines/ [Accessed 28 Oct. 2024].

Delua, J. (2024). *Supervised vs Unsupervised Learning*. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning [Accessed 24 Sep. 2024].

Demšar, J., Zupan, B., Leban, G. and Curk, T. (2004). Orange: From Experimental Machine Learning to Interactive Data Mining. *Lecture Notes in Computer Science*, pp.537–539. doi:https://doi.org/10.1007/978-3-540-30116-5_58.

DeVore, R., Hanin, B. and Petrova, G. (2021). Neural network approximation. *Acta Numerica*, 30, pp.327–444. doi:https://doi.org/10.1017/s0962492921000052.

Dobesova, Z. (2024). Evaluation of Orange data mining software and examples for lecturing machine learning tasks in geoinformatics. *Computer Applications in Engineering Education*, [online] 32(4). doi:https://doi.org/10.1002/cae.22735.

Dong, J., Chen, Y., Yao, B., Zhang, X. and Zeng, N. (2022). A neural network boosting regression model based on XGBoost. *Applied Soft Computing*, [online] 125, pp.109067–109067. doi:https://doi.org/10.1016/j.asoc.2022.109067.

Dongare, A., Kharde, R. and Kachare, A. (2008). Introduction to Artificial Neural Network. *Certified International Journal of Engineering and Innovative Technology (IJEIT)*, [online] 9001(1), pp.2277–3754. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=04d0b6952a4f0c7203577afc9476c2fcab2cba06.

Drott, E. (2020). Fake Streams, Listening Bots, and Click Farms: Counterfeiting Attention in the Streaming Music Economy. *American Music*, 38(2), p.153. doi:https://doi.org/10.5406/americanmusic.38.2.0153.

Edmond, S. and Crabtree, M. (2022). *Power BI vs Tableau: Which is The Better Business Intelligence Tool in 2024?* [online] Datacamp.com. Available at: https://www.datacamp.com/blog/power-bi-vs-tableau-which-one-should-you-choose [Accessed 2 Oct. 2024].

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B. and Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, [online] 8(1), pp.1–37. doi:https://doi.org/10.1186/s40537-021-00516-9.

Engelen, van and Hoos, H.H. (2019). A survey on semi-supervised learning. *Machine Learning*, [online] 109(2), pp.373–440. doi:https://doi.org/10.1007/s10994-019-05855-6.

Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Laith Abualigah, Agushaka, J.O., Eke, C.I. and Akinyelu, A.A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, [online] 110, pp.104743–104743. doi:https://doi.org/10.1016/j.engappai.2022.104743.

Fitzpatrick, L. (2023). *The Limitations of R^2 in Correlation Studies*. [online] George Lee Sye. Available at: https://georgeleesye.com/the-limitations-of-r%C2%B2-in-correlation-studies/ [Accessed 24 Sep. 2024].

Frost, J. (2017). *Heteroscedasticity in Regression Analysis*. [online] Statistics By Jim. Available at: https://statisticsbyjim.com/regression/heteroscedasticity-regression/ [Accessed 4 Nov. 2024].

Gandhi, S., Gala, R., Viswanathan, V., Wu, T. and Neubig, G. (2024). *Better Synthetic Data by Retrieving and Transforming Existing Datasets*. [online] arXiv.org. Available at: https://arxiv.org/abs/2404.14361 [Accessed 25 Sep. 2024].

Gao, L. (2014). *Laura Gao*. [online] Laura Gao. Available at: https://www.lauragao.com/the-evolution-of-dance-music#:~:text=Interestingly%2C%20acousticness%20affects%20danceability%20differently,made%20a%20song%20less%20danceable. [Accessed 10 Sep. 2024].

Garkusha, S. (2024). *PyCharm vs. Jupyter Notebook | The PyCharm Blog*. [online] The JetBrains Blog. Available at: https://blog.jetbrains.com/pycharm/2024/09/pycharm-vs-jupyter-notebook/ [Accessed 27 Sep. 2024].

Geng, J. (2021). Personalized Analysis and Recommendation of Aesthetic Evaluation Index of Dance Music Based on Intelligent Algorithm. *Complexity*, 2021, pp.1–15. doi:https://doi.org/10.1155/2021/1026341.

Ghahramani, Z. (2004). Unsupervised Learning. *Lecture notes in computer science*, [online] pp.72–112. doi:https://doi.org/10.1007/978-3-540-28650-9_5.

GlobeNewswire (2024). *Music Streaming Market Projected to Grow at 15.1% CAGR, Reaching USD 125.7 Billion by 2032 | Market.us*. [online] GlobeNewswire News Room. Available at: https://www.globenewswire.com/news-release/2024/02/05/2823256/0/en/Music-Streaming-Market-Projected-to-Grow-at-15-1-CAGR-Reaching-USD-125-7-Billion-by-2032-Market-us.html [Accessed 28 Oct. 2024].

Gohiya, H., Lohiya, H. and Patidar, K. (2018). *A SURVEY OF XGBOOST SYSTEM*. [online] *International Journal of Advanced Technology & Engineering Research*. Available at: http://www.ijater.com/Files/aa09b180-add4-4a6d-b234-bc122eb305d4_IJATER_39_07.pdf.

Gong, W. and Yu, Q. (2021). A Deep Music Recommendation Method Based on Human Motion Analysis. *IEEE Access*, 9, pp.26290–26300. doi:https://doi.org/10.1109/access.2021.3057486.
google (2019). *Spam, deceptive practices, & scams policies - YouTube Help*. [online] Google.com. Available at: https://support.google.com/youtube/answer/2801973?hl=en&ref_topic=9282365 [Accessed 10 Sep. 2024].

Google Cloud. (2024a). *What is Supervised Learning? | Google Cloud*. [online] Available at: https://cloud.google.com/discover/what-is-supervised-learning# [Accessed 24 Sep. 2024].

Google Cloud. (2024b). *What is unsupervised learning? | Google Cloud*. [online] Available at: https://cloud.google.com/discover/what-is-unsupervised-learning [Accessed 24 Sep. 2024].

Google for Developers (2024). *What is clustering?* [online] Google for Developers. Available at: https://developers.google.com/machine-learning/clustering/overview/ [Accessed 25 Sep. 2024].

Greenburg, Z.O. (2024). How 'Despacito' Became The Most Popular YouTube Video Of All Time. *Forbes*. [online] 3 Jun. Available at: https://www.forbes.com/sites/zackomalleygreenburg/2017/08/07/how-despacito-became-the-most-popular-youtube-video-of-all-time/ [Accessed 16 Sep. 2024].

Guo, H., Nguyen, H., Vu, D.-A. and Bui, X.-N. (2019). Forecasting mining capital cost for open-pit mining projects based on artificial neural network approach. *Resources Policy*, [online] 74, pp.101474–101474. doi:https://doi.org/10.1016/j.resourpol.2019.101474.

Hambali, M.A., Oladele, T.O., Adewole, K.S., Arun Kumar Sangaiah and Gao, W. (2022). Feature selection and computational optimization in high-dimensional microarray cancer datasets via InfoGain-modified bat algorithm. *Multimedia Tools and Applications*, [online] 81(25), pp.36505–36549. doi:https://doi.org/10.1007/s11042-022-13532-5.

Hargreaves, T. (2023). *Efficiently Removing Zero Variance Columns (An Introduction to Benchmarking)*. [online] T-Tested | Blogging about all things data. Available at: https://www.ttested.com/removing-zero-variance-columns/ [Accessed 22 Sep. 2024].

Hill, C., Du, L., Johnson, M. and McCullough, B.D. (2024). Comparing programming languages for data analytics: Accuracy of estimation in Python and R. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, [online] 14(3). doi:https://doi.org/10.1002/widm.1531.

Hodge, V. and Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, [online] 22(2), pp.85–126. doi:https://doi.org/10.1023/b:aire.0000045502.10941.a9.

Hoo, Z.H., Candlish, J. and Teare, D. (2017). What is an ROC curve? *Emergency Medicine Journal*, [online] 34(6), pp.357–359. doi:https://doi.org/10.1136/emermed-2017-206735.

Hossin, M. and Sulaiman, M.N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, [online] 5(2), pp.01-11. doi:https://doi.org/10.5121/ijdkp.2015.5201.

Howlin, C. and Rooney, B. (2020). Patients choose music with high energy, danceability, and lyrics in analgesic music listening interventions. *Psychology of Music*, [online] 49(4), pp.931–944. doi:https://doi.org/10.1177/0305735620907155.

Hu, Y. (2024). Cross-Cultural Perspectives in Music: Analyzing the Impact of Cultural Differences on Music Preferences and Practices. *International Journal of Education and Humanities*, 15(1), pp.170–173. doi:https://doi.org/10.54097/as616s92.
IBM (2021a). *Supervised Learning*. [online] Ibm.com. Available at: https://www.ibm.com/topics/supervised-learning [Accessed 24 Sep. 2024].

IBM (2021b). *Unsupervised Learning*. [online] Ibm.com. Available at: https://www.ibm.com/topics/unsupervised-learning [Accessed 24 Sep. 2024].

Iqbal, M. (2021). *Application of Regression Techniques with their Advantages and Disadvantages*. [online] Available at: https://www.researchgate.net/profile/Muhammad-Ahmad-Iqbal/publication/354921553_Application_of_Regression_Techniques_with_their_Advantages_and_Disadvantages/links/61543c3c14d6fd7c0fb91053/Application-of-Regression-Techniques-with-their-Advantages-and-Disadvantages.pdf [Accessed 8 Nov. 2024].

Isaac Abiodun, O., Jantan, A., Esther Omolara, A., Victoria Dada, K., AbdElatif Mohamed, N. and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, [online] 4(11), pp.e00938–e00938. doi:https://doi.org/10.1016/j.heliyon.2018.e00938.

Ishwaran, H. and Lu, M. (2018). Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Statistics in Medicine*, [online] 38(4), pp.558–582. doi:https://doi.org/10.1002/sim.7803.

Islam, S., Arifuzzaman, M. and Islam, S. (2019). SMOTE Approach for Predicting the Success of Bank Telemarketing. doi:https://doi.org/10.1109/times-icon47539.2019.9024630.

Jackson-Barnes, S. (2024). *Tableau vs Power BI: How to Choose the Right BI Software*. [online] Orientsoftware.com. Available at: https://www.orientsoftware.com/blog/tableau-vs-power-bi/ [Accessed 27 Sep. 2024].

Jadhav, A., Pramod, D. and Ramanathan, K. (2019). Comparison of Performance of Data Imputation Methods for Numeric Dataset. *Applied Artificial Intelligence*, 33(10), pp.913–933. doi:https://doi.org/10.1080/08839514.2019.1637138.

Jamal, P., Ali, M., Faraj, R. and Ali, P. (2014). 1-6 Data Normalization and Standardization: A Technical Report. *Machine Learning Technical Reports*, 1(1), pp.1–6.

JetBrains. (2024). *JetBrains Products Comparison*. [online] Available at: https://www.jetbrains.com/products/compare/?product=pycharm&product=pycharm-ce [Accessed 2 Oct. 2024].

Jha, A., Dave, M. and Madan, S. (2019). Comparison of Binary Class and Multi-Class Classifier Using Different Data Mining Classification Techniques. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.3464211.

Jo, J. (2022). *Overview Of Random Forest - Jay Jo - Medium*. [online] Medium. Available at: https://medium.com/@wjj1019/overview-of-random-forest-36a9d51f4b57 [Accessed 4 Oct. 2024].

Juba, B. and Le, H.S. (2019). Precision-Recall versus Accuracy and the Role of Large Data Sets. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 33(01), pp.4039–4048. doi:https://doi.org/10.1609/aaai.v33i01.33014039.

Kammoun, A. and AlouiniFellow, M.-S. (2021). On the Precise Error Analysis of Support Vector Machines. *IEEE Open Journal of Signal Processing*, 2, pp.99–118. doi:https://doi.org/10.1109/ojsp.2021.3051849.

Karageorghis, C.I. and Priest, D.-L. (2012). Music in the exercise domain: a review and synthesis (Part I). *International Review of Sport and Exercise Psychology*, [online] 5(1), pp.44–66. doi:https://doi.org/10.1080/1750984x.2011.631026.

Karl, T. (2024). *How to choose between Seaborn vs. Matplotlib*. [online] New Horizons. Available at: https://www.newhorizons.com/resources/blog/how-to-choose-between-seaborn-vs-matplotlib#:~:text=In%20summary%2C%20both%20Seaborn%20and,every%20aspect%20of%20a%20plot. [Accessed 2 Oct. 2024].

Kennedy, H., Kunkel, T. and Funk, D.C. (2021). Using Predictive Analytics to Measure Effectiveness of Social Media Engagement: A Digital Measurement Perspective. *Sport Marketing Quarterly*, [online] 30(4), pp.265–277. Available at: https://muse.jhu.edu/article/927258.

Khan, M.L. (2017). Social Media engagement: What Motivates User Participation and Consumption on YouTube? *Computers in Human Behavior*, [online] 66(1), pp.236–247. doi:https://doi.org/10.1016/j.chb.2016.09.024.

Khosla, C. (2019). *Top 10 Dataset Repositories - Dextutor -*. [online] Dextutor. Available at: https://dextutor.com/top-10-dataset-repositories/ [Accessed 25 Sep. 2024].

Klomp, T. (2022). *Iterative Imputation in Python*. [online] Available at: https://studenttheses.uu.nl/bitstream/handle/20.500.12932/42511/ThesisTinkeKlomp.pdf?sequence=1&isAllowed=y [Accessed 17 Sep. 2024].

Krenker, A., Bester, J. and Kos, A. (2011). Introduction to the Artificial Neural Networks. *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. doi:https://doi.org/10.5772/15751.

Krstinić, D., Braović, M., Šerić, L. and Božić-Štulić, D. (2020). Multi-label Classifier Performance Evaluation with Confusion Matrix. *Computer Science & Information Technology*. [online] doi:https://doi.org/10.5121/csit.2020.100801.

Kurama, V. (2024). *Regression in Machine Learning: Definition and Examples | Built In*. [online] Built In. Available at: https://builtin.com/data-science/regression-machine-learning [Accessed 24 Sep. 2024].

Kworb (2014). *YouTube - Most Liked Music Videos*. [online] Kworb.net. Available at: https://kworb.net/youtube/topvideos_likes.html [Accessed 16 Sep. 2024].

Larionov, M. (2020). *Sampling Techniques in Bayesian Target Encoding*. [online] Available at: https://arxiv.org/pdf/2006.01317 [Accessed 22 Sep. 2024].

LaValley, M.P. (2008). Logistic Regression. *Circulation*, [online] 117(18), pp.2395–2399. doi:https://doi.org/10.1161/circulationaha.106.682658.

Lavanya, A., Gaurav, L., Sindhuja, S., Seam, H., Joydeep, M., Uppalapati, V., Ali, W. and Sagar, V. (2023). Assessing the Performance of Python Data Visualization Libraries: A Review. *International Journal of Computer Engineering In Research Trends*, [online] 10(1), pp.29–39. doi:https://doi.org/10.22362/ijcert/2023/v10/i01/v10i0104.

Leandro (2024). *Mastering Linear Regression: From Scratch to Scikit-Learn in Python*. [online] Medium. Available at: https://medium.com/@tech-adventurer/cracking-the-code-of-linear-regression-diy-vs-scikit-learn-04ea9a84312c [Accessed 8 Nov. 2024].

Learned-Miller, E.G. (2014). *Introduction to Supervised Learning*. [online] Available at: https://people.cs.umass.edu/~elm/Teaching/Docs/supervised2014a.pdf.

Lee, D. (2020). Introduction Data transformation: a focus on the interpretation. doi:https://doi.org/10.4097/kja.20137.

Lin, W.-C. and Tsai, C.-F. (2019). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, [online] 53(2), pp.1487–1509. doi:https://doi.org/10.1007/s10462-019-09709-4.

Liu, Y. and Brown, S.D. (2013). Comparison of five iterative imputation methods for multivariate classification. *Chemometrics and Intelligent Laboratory Systems*, 120, pp.106–115. doi:https://doi.org/10.1016/j.chemolab.2012.11.010.

Luís Torgo and Gama, J. (1999). *Regression by Classification*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/2377673_Regression_by_Classification.

LuisFonsiVEVO (2017). *Luis Fonsi - Despacito ft. Daddy Yankee*. *YouTube*. Available at: https://www.youtube.com/watch?v=kJQP7kiw5Fk&ab_channel=LuisFonsiVEVO [Accessed 16 Sep. 2024].

Ma, J., Li, Z. and Li, L. (2024). Research on Public Opinion Analysis Methods Based on Knowledge Mapping and Deep Learning. doi:https://doi.org/10.4108/eai.15-12-2023.2345358.

Malik, H. and Tian, Z. (2017). A Framework for Collecting YouTube Meta-Data. *Procedia Computer Science*, [online] 113, pp.194–201. doi:https://doi.org/10.1016/j.procs.2017.08.347.

Marshall, K. (2024). *How does epoch affect accuracy?* [online] Deepchecks. Available at: https://www.deepchecks.com/question/how-does-epoch-affect-accuracy/ [Accessed 1 Nov. 2024].

Matalonga, H. (2023). *🧮Choosing between MAE, MSE and RMSE*. [online] Hugo Matalonga. Available at: https://hmatalonga.com/blog/choosing-between-mae-mse-and-rmse/ [Accessed 24 Sep. 2024].

Mayumi Oshiro, T., Santoro Perez, P. and Augusto Baranauskas, J. (2012). How Many Trees in a Random Forest? *Lecture notes in computer science*, [online] pp.154–168. doi:https://doi.org/10.1007/978-3-642-31537-4_13.

Melton, J. (1996). SQL language summary. *ACM Computing Surveys (CSUR)*, 28(1), pp.141–143. doi:https://doi.org/10.1145/234313.234374.

Mitchell, R. and Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, [online] 3, p.e127. doi:https://doi.org/10.7717/peerj-cs.127.

Naik, G. (2023). *Conceptualizing Python in Google COLAB*. Shashwat Publication.
Newman, D.A. (2014). Missing Data. *Organizational Research Methods*, [online] 17(4), pp.372–411. doi:https://doi.org/10.1177/1094428114548590.

News, B. (2022). *Dance music is second most popular UK genre - BPI*. [online] BBC News. Available at: https://www.bbc.co.uk/news/newsbeat-63462786 [Accessed 8 Sep. 2024].

Oberoi, A. and Chauhan, R. (2019). Visualizing data using Matplotlib and Seaborn libraries in Python for data science. *International Journal of Scientific and Research Publications (IJSRP)*, 9(3), p.p8733. doi:https://doi.org/10.29322/ijsrp.9.03.2019.p8733.

Ochi, V., Estrada, R., Gaji, T., Gadea, W. and Duong, E. (2021). *Spotify Danceability and Popularity Analysis using SAP*. [online] Available at: https://arxiv.org/pdf/2108.02370.

Ochoa, J. (2020). *Billboard 200 Album Chart To Count Video Plays From Streaming Services Starting In 2020 | GRAMMY.com*. [online] Grammy.com. Available at: https://www.grammy.com/news/billboard-200-album-chart-count-video-plays-streaming-services-starting-2020 [Accessed 10 Sep. 2024].

Anandhi, P. and Nathiya, E. (2023). Application of linear regression with their advantages, disadvantages, assumption and limitations. *International Journal of Statistics and Applied Mathematics*, [online] 8(6), pp.133–137. doi:https://doi.org/10.22271/maths.2023.v8.i6b.1463.

Patrician, P.A. (2002). Multiple imputation for missing data†‡. *Research in Nursing & Health*, 25(1), pp.76–84. doi:https://doi.org/10.1002/nur.10015.

Patro, R. (2021). *Matplotlib VS Ggplot2 - Towards Data Science*. [online] Medium. Available at: https://towardsdatascience.com/matplotlib-vs-ggplot2-c86dd35a9378 [Accessed 2 Oct. 2024].

Pedro, I. (2021). *From Linear Regression to Neural Networks: Why and How*. [online] Medium. Available at: https://medium.com/deep-learning-sessions-lisboa/neural-netwoks-419732d6afc0 [Accessed 4 Oct. 2024].

Pesantez-Narvaez, J., Guillen, M. and Alcañiz, M. (2019). Predicting Motor Insurance Claims Using Telematics Data—XGBoost versus Logistic Regression. *Risks*, [online] 7(2), p.70. doi:https://doi.org/10.3390/risks7020070.

Platias, C. and Petasis, G. (2020). A Comparison of Machine Learning Methods for Data Imputation. doi:https://doi.org/10.1145/3411408.3411465.

Potdar, K., S., T. and D., C. (2017). A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *International Journal of Computer Applications*, 175(4), pp.7–9. doi:https://doi.org/10.5120/ijca2017915495.

Probst, P. and Boulesteix, A.-L. (2018). To Tune or Not to Tune the Number of Trees in Random Forest. *Journal of Machine Learning Research*, [online] 18, pp.1–18. Available at: https://www.jmlr.org/papers/volume18/17-269/17-269.pdf.

Rastelli, S. (2023). *Spotify and Youtube*. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube/data [Accessed 6 Sep. 2024].
Rigatti, S.J. (2017). Random Forest. *Journal of Insurance Medicine*, [online] 47(1), pp.31–39. doi:https://doi.org/10.17849/insm-47-01-31-39.1.

RITHP (2023). *The main parameters in XGBoost and their effects on model performance*. [online] Medium. Available at: https://medium.com/@rithpansanga/the-main-parameters-in-xgboost-and-their-effects-on-model-performance-4f9833cac7c [Accessed 3 Nov. 2024].
Saabith, S., Thangarajah, V. and Fareez, M. (2021). *A Review on Python Libraries and IDEs for Data Science*. [online] Available at:

https://www.researchgate.net/publication/357898994_A_Review_on_Python_Libraries_and_IDEs_for_Data_Science [Accessed 27 Sep. 2024].

Saigal, P. and Khanna, V. (2020). Multi-category news classification using Support Vector Machine based classifiers. *SN Applied Sciences*, [online] 2(3). doi:https://doi.org/10.1007/s42452-020-2266-6.

Samuels, J. (2024). One-Hot Encoding and Two-Hot Encoding: An Introduction One-Hot Encoding and Two-Hot Encoding: An Introduction. doi:https://doi.org/10.13140/RG.2.2.21459.76327.

Sciandra, M. and Irene Carola Spera (2020). A model-based approach to Spotify data analysis: a Beta GLMM. *Journal of Applied Statistics*, [online] 49(1), pp.214–229. doi:https://doi.org/10.1080/02664763.2020.1803810.

Seger, C. (2018). *An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing*. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1259073/FULLTEXT01.pdf.

Sheeran, E. (2017). *Ed Sheeran - Shape of You (Official Music Video)*. *YouTube*. Available at: https://www.youtube.com/watch?v=JGwWNGJdvx8&ab_channel=EdSheeran [Accessed 13 Sep. 2024].

Sherman, A. (2024). *YouTube dominates streaming, forcing media companies to decide whether it's friend or foe*. [online] CNBC. Available at: https://www.cnbc.com/2024/06/26/youtube-streaming-dominance-media-strategy.html [Accessed 10 Sep. 2024].

Siersdorfer, S., Chelaru, S., Nejdl, W. and San Pedro, J. (2010). How useful are your comments? *Proceedings of the 19th international conference on World wide web - WWW '10*. [online] doi:https://doi.org/10.1145/1772690.1772781.

Simmering , J. (2013). *How slow is R really? | R-bloggers*. [online] R-bloggers. Available at: https://www.r-bloggers.com/2013/01/how-slow-is-r-really/ [Accessed 2 Oct. 2024].

Singh, D., Nigam, R., Mittal, R. and Nunia, M. (2022). Information retrieval using machine learning from breast cancer diagnosis. *Multimedia Tools and Applications*. doi:https://doi.org/10.1007/s11042-022-13550-3.

Singhal, P. (2022). *Advantages and Disadvantages of SQL - Scaler Topics*. [online] Scaler Topics. Available at: https://www.scaler.com/topics/advantages-of-sql/ [Accessed 27 Sep. 2024].

Singrodia, V., Mitra, A. and Paul, S. (2019). A Review on Web Scrapping and its Applications. [online] doi:https://doi.org/10.1109/iccci.2019.8821809.

Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, 38, p.100306. doi:https://doi.org/10.1016/j.cosrev.2020.100306.

Soofi, A. and Awan, A. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences*, 13, pp.459–465. doi:https://doi.org/10.6000/1927-5129.2017.13.76.

Spotify (2024). *Web API Reference | Spotify for Developers*. [online] Spotify.com. Available at: https://developer.spotify.com/documentation/web-api/reference/get-audio-features [Accessed 13 Sep. 2024].

Start Data Engineering (2024). *SQL or Python for Data Transformations?* [online] Startdataengineering.com. Available at: https://www.startdataengineering.com/post/sql-v-python/ [Accessed 2 Oct. 2024].

Su, X., Yan, X. and Tsai, C.-L. (2019). Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3), pp.275–294. doi:https://doi.org/10.1002/wics.1198.

Subbiah , A. and Aggarwal, V. (2024). *Transformers in music recommendation*. [online] Research.google. Available at: https://research.google/blog/transformers-in-music-recommendation/ [Accessed 28 Oct. 2024].

Uyanık, G.K. and Güler, N. (2013). A Study on Multiple Linear Regression Analysis. *Procedia - Social and Behavioral Sciences*, [online] 106, pp.234–240. doi:https://doi.org/10.1016/j.sbspro.2013.12.027.

vault, music metrics (2024). *Music Metrics Vault - Spotify metrics made easy*. [online] Musicmetricsvault.com. Available at: https://www.musicmetricsvault.com/ [Accessed 10 Sep. 2024].

Viralyft. (2024). *What Is A Good Like To View Ratio On YouTube? - Viralyft*. [online] Available at: https://viralyft.com/blog/youtube-like-to-view-ratio#:~:text=Marketing%20specialists%20often%20suggest%20aiming,likes%20for%20every%201%2C000%20views [Accessed 10 Sep. 2024].

Vizcaíno-Verdú, A. and Abidin, C. (2022). Music Challenge Memes on TikTok: Understanding In-Group Storytelling Videos. *International Journal of Communication*, [online] 16(0), p.26. Available at: https://ijoc.org/index.php/ijoc/article/view/18141/3680.

Wang, J., Wang, Y., Weng, N., Chai, T., Li, A., Zhang, F. and Yu, S. (2022). Will You Ever Become Popular? Learning to Predict Virality of Dance Clips. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(2), pp.1–24. doi:https://doi.org/10.1145/3477533.

Waskom, M. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, [online] 6(60), p.3021. Available at: https://joss.theoj.org/papers/10.21105/joss.03021.

Wickham, H. (2011). ggplot2. *Wiley Interdisciplinary Reviews Computational Statistics*, [online] 3(2), pp.180–185. doi:https://doi.org/10.1002/wics.147.

Wu, Y. and Feng, J. (2017). Development and Application of Artificial Neural Network. *Wireless Personal Communications*, [online] 102(2), pp.1645–1656. doi:https://doi.org/10.1007/s11277-017-5224-x.

Yadav, H. (2022). *Dropout in Neural Networks - Towards Data Science*. [online] Medium. Available at: https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9 [Accessed 1 Nov. 2024].

Yasar, K., Lawton, G. and Burns, E. (2024). *logistic regression*. [online] Business Analytics. Available at: https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression [Accessed 5 Nov. 2024].

Yin, A.L., Guo, W.L., Sholle, E.T., Rajan, M., Alshak, M.N., Choi, J.J., Goyal, P., Jabri, A., Li, H.A., Pinheiro, L.C., Wehmeyer, G.T., Weiner, M., Safford, M.M., Campion, T.R. and Cole, C.L. (2022). Comparing automated vs. manual data collection for COVID-specific medications from electronic health records. *International Journal of Medical Informatics*, [online] 157, p.104622. doi:https://doi.org/10.1016/j.ijmedinf.2021.104622.

YouTube (2018). *YouTube Rewind 2018: Everyone Controls Rewind | #YouTubeRewind. YouTube*. Available at: https://www.youtube.com/watch?v=YbJOTdZBX1g&ab_channel=YouTube [Accessed 8 Sep. 2024].

Zhu, L. (2020). *YouTube is more important for artists than ever before — here's why.* [online] Medium. Available at: https://lab.songstats.com/youtube-is-more-important-for-artists-than-ever-before-heres-why-3401c13c12b1 [Accessed 28 Oct. 2024].

Appendix

Details of the folder
The project folder contains 2 Jupyter notebook files, report in PDF format, dataset used and one folder with results and visualisation.

The Jupyter file of the first question about Danceability is named as danceability.ipynb and the file for the second question is named as likesviews.ipynb. The dataset is also provided in the same folder. The results and graphs of each analysis are saved in a folder named results and graphs.

How to run the program
1. Open Jupyter notebook in the browser.
2. Save the ipynb files given in this folder to a folder in the computer.
3. Navigate to the folder on the Jupyter notebook interface and open the ipynb file.
4. Each cell is already executed, however can be run again by pressing the run button on the interface. or by clicking Control and Enter keys on keyboard.