Data Analysis Mathematics, Algorithms, and Modelling (PROG8430)

Group Project Introduction

Group-4

Aswani Pottathuparambil Unnikrishnan (8858117)

Dwarakanath Chandra (88556840)

Guru Charan Bogireddy (8902043)

Ramya Chandran Udayachandran (8895789)

Sofiya Sofiya (8902034)

Data Analysis Mathematics, Algorithms, and Modelling (PROG8430)

Group-4 Project Introduction

Spotify Songs Recommender Analysis Using Clustering

Introduction

User engagement is the prime goal of content streaming services. Especially, the music streaming services need to capture the user preferences dynamically to suggest correct recommendations for engaging the listeners. However, the user preferences are not constant all the time. Users listen to different kinds of songs and music based on their mood, environment, and how their day was going. Hence it is difficult to predict what user would listen to and what the recommendation system should suggest. This is the business problem for music streaming recommendation systems. In this project, we want to analyse and resolve this problem not from the perspective of users but from the perspective of content characteristics.

Problem Statement

Spotify is a leading music streaming service. The key to the success of Spotify among different market competitors is its ability to provide music for individual preferences. The customization plays an important role in improving the user experience. However, as mentioned earlier, customer preferences change dynamically based on a lot of factors. Identifying the customer music listening preferences is a difficult task to the streaming services.

However, rather than predicting the customer preferences based on customer related factors, we can identify and categorize different songs on their characteristics and attributes. So that, When Spotify observes any user listening from a particular category of songs, we can configure the recommendation algorithm to suggest more songs from the same cluster to the user. This helps us in retaining the user engagement on the app.

The Problem statement is “How can we categorize the Spotify songs based on the analysis of their attributes and characteristics into distinctive clusters to help the Spotify recommendation algorithm to suggest the listeners with the most relevant and engaging playlists?”

Solving this problem statement would:

1. Enhance the user experience.
2. Provide better customization.
3. Reduces the user bouncing rate.
4. Results in lower skip count.

Data Analysis Mathematics, Algorithms, and Modelling (PROG8430)

Group-4 Project Introduction

Spotify Songs Recommender Analysis Using Clustering

Abstract – Approach

To solve the above-mentioned business problem, the songs will be clustered based on their characteristics using unsupervised learning algorithms. The clustering algorithms chosen for this project are K-means clustering and Hierarchical clustering methods. As part of this project, we want to build these clustering models on Spotify data to form different clusters that capture the similarities between the songs' attributes. Once the models are ready, we want to evaluate models based on performance metrics such as similarity and dissimilarity. After the evaluation of the models, we choose the best model for clustering of songs for helping the Spotify recommendation system.

Data Gathering

To create the clustering models, we need a list of songs streaming on Spotify along with their characteristics and attributes. After a thorough search and evaluation of various alternative datasets, we found a Spotify songs dataset on Kaggle.com (Marylou, 2023). This dataset contains 2017 songs each with 17 characteristics. This is a suitable dataset to conduct the analysis and to build the clustering models.

The 2017 songs form rows of this dataset with 2017 records.

The 17 characteristic features form columns of this dataset.

The list of features available in the dataset are:

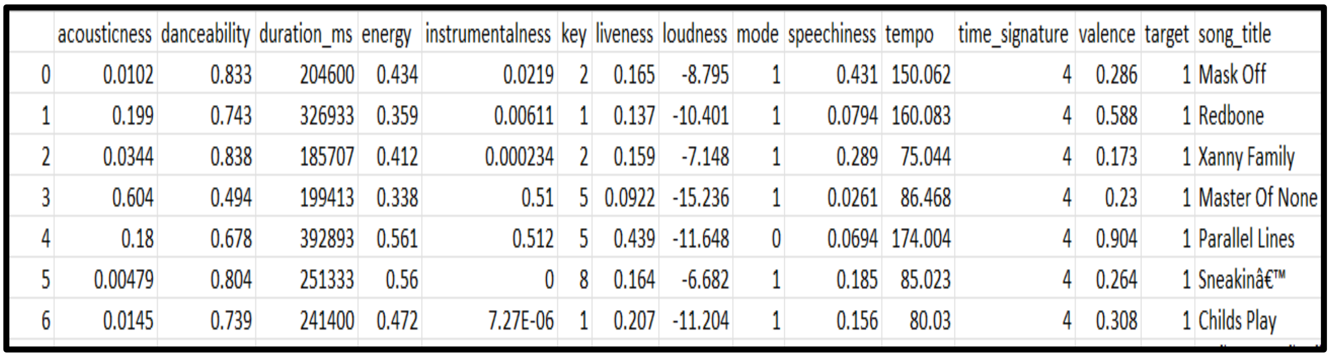
1. **acousticness:** Describes how acoustic a song is. A score of 1.0 means the song is most likely to be an acoustic one.
2. **danceability:** Describes how suitable a track is for dancing based on tempo, rhythm stability, beat strength, and regularity.
3. **duration_ms:** Song duration in milliseconds.
4. **Energy:** Represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
5. **instrumentalness:** The number of vocals in the song. The closer it is to 1.0, the more instrumental the song is.
6. **liveness:** Describes the probability that the song was recorded with a live audience.
7. **speechiness:** Speechiness detects the presence of spoken words in a track.
8. **valance:** Describes the musical positiveness conveyed by a track.
9. **song_tiltle:** Name of the song
10. **artist:** Composer & Singer of the song
11. **tempo:** Tempo on its own refers to the speed you play the song at.

12. **key:** key is the main group of pitches, or notes, that form the harmonic foundation of a piece of music.
13. **mode:** musical modes are rotations of the major or natural minor scales.
14. **loudness:** A measure of how load is the song in decibels.
15. **time_signature:** Time signatures tell musicians how to group musical notes.
16. **target:** A binary variable to indicate whether song is popular or not popular.

The following is the data snippet for further illustration of the dataset.

Fig:1

Data Snippet of Spotify Songs Data



	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	target	song_title
0	0.0102	0.833	204600	0.434	0.0219	2	0.165	-8.795	1	0.431	150.062	4	0.286	1	Mask Off
1	0.199	0.743	326933	0.359	0.00611	1	0.137	-10.401	1	0.0794	160.083	4	0.588	1	Redbone
2	0.0344	0.838	185707	0.412	0.000234	2	0.159	-7.148	1	0.289	75.044	4	0.173	1	Xanny Family
3	0.604	0.494	199413	0.338	0.51	5	0.0922	-15.236	1	0.0261	86.468	4	0.23	1	Master Of None
4	0.18	0.678	392893	0.561	0.512	5	0.439	-11.648	0	0.0694	174.004	4	0.904	1	Parallel Lines
5	0.00479	0.804	251333	0.56	0	8	0.164	-6.682	1	0.185	85.023	4	0.264	1	Sneakinâ€™
6	0.0145	0.739	241400	0.472	7.27E-06	1	0.207	-11.204	1	0.156	80.03	4	0.308	1	Childs Play

Data Cleaning

The collected Spotify songs data from the data source is raw data with a lot of unnecessary information for the analysis. Hence, data cleaning procedure conducted on the dataset to clean the data for further exploratory data analysis. This data cleaning happened in step-by-step process as mentioned below.

a. **Unnecessary Columns:** Some of the columns in the data are not useful for analytical purposes. However, they may have the numerical values which machine learning models may interpret as useful information. Hence, we need to remove such unnecessary algorithms to avoid confusion for machine learning model building. When we inspect data, there is an unlabelled column at the index = 0, which contains serial numbers for the songs. We have dropped that index column successfully.

b. **Missing Value Handling:** Missing values in data causes problems during the statistical analysis. Hence, we need to ensure the presence of any missing values in the data and handle them based on the logical context of the features. However, it is observed that the Spotify dataset doesn't contain any missing values upon investigation. As a precautionary measure, we have provided a missing value filter to prevent any future incoming data has missing values.

c. **Duplicate Values:** Duplicate values in the data causes redundancy of observations. Hence, we need to identify and remove the duplicate values as they will cause distortion in the analysis. We have provided a duplicate value filter using distinct () function based on song_title and artist features. We observed that, there are 35 duplicated songs with same name and artist in the data and removed them successfully.

d. **Type Casting Problems:** To do the analysis the features should be of correct datatypes. The discrepancy in type of features would hinder the analytical process. Hence, we have checked for the data types of all variables and ensured all numerical variables are either float or integer data types and remaining or character data types.

Hence, by performing the above basic data cleaning steps, we have ensured the data is not corrupted and is ready for further exploratory analysis.

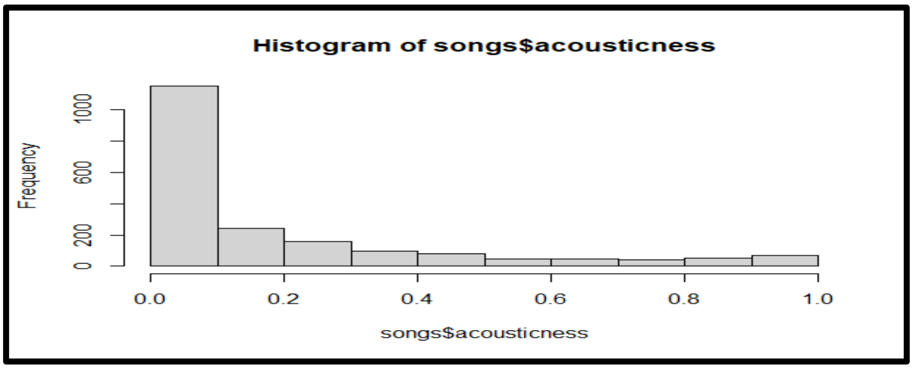
Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) is done for analysis and investigation on data sets and summarize their main characteristics, often employing data visualization methods. This helps determine how best to manipulate data sources to get the answers you need, making it easier to discover patterns, spot anomalies, test a hypothesis, or check assumptions. EDA is primarily used to see what data can reveal and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

To conduct the exploratory data analysis on Spotify songs data, basic cleaning of data was done as mentioned in the earlier section. Now, we want to explore the data distribution of various features to understand the behaviour of each feature and underlying insights. We have plotted data distribution of different features with histograms as shown below.

Figure:2

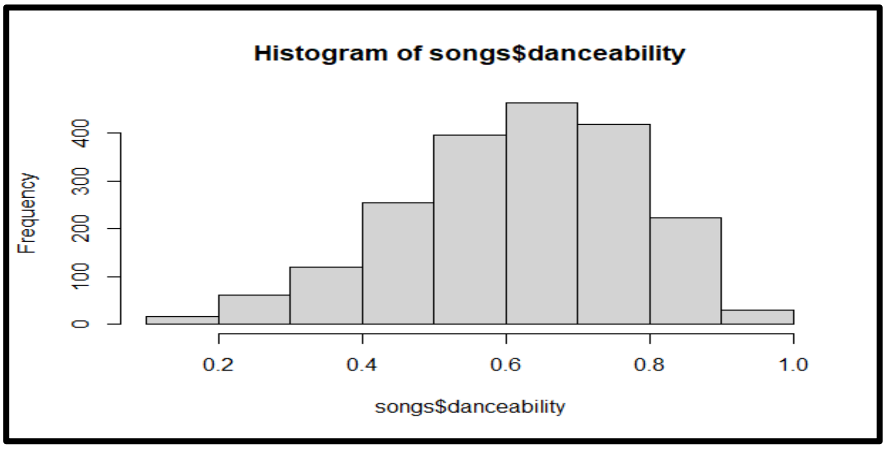
Histogram of acousticness variable



The histogram of acousticness variable illustrates that majority of songs in the data used electric and electronic music rather than acoustic one. The distribution of acousticness is right-skewed.

Figure:3

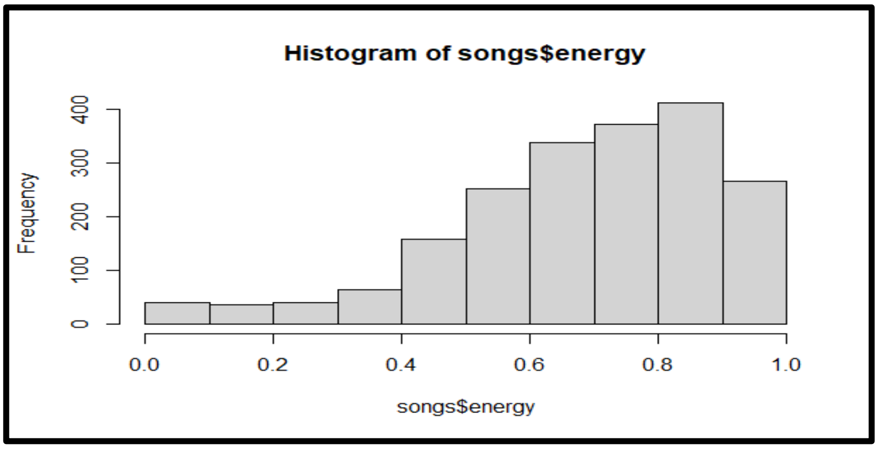
Histogram of Danceability



From the above histogram of danceability, we can see that the data is almost normally distributed. This indicates the presence of danceable songs is slightly higher in the data.

Figure-4

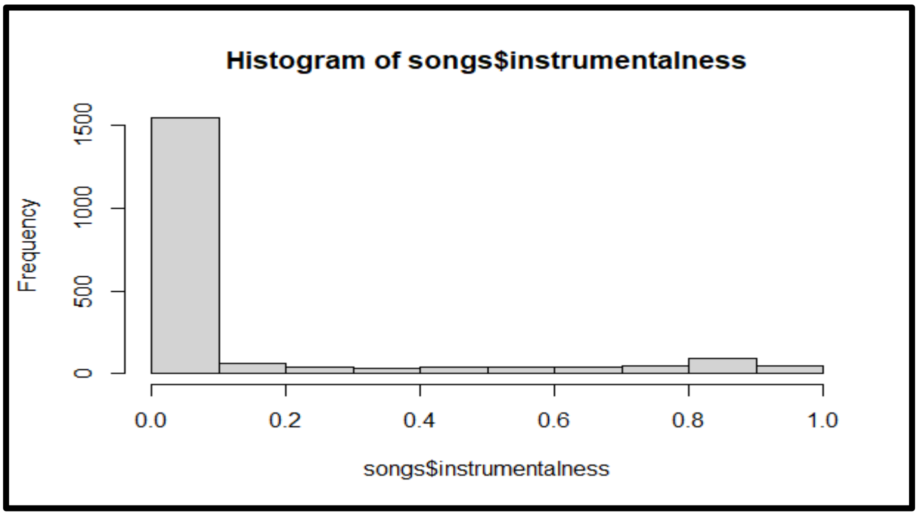
Histogram of Energy



The above histogram of energy variable indicates that there are higher number of energetic songs in the data. The energy data distribution is left-skewed.

Figure-5

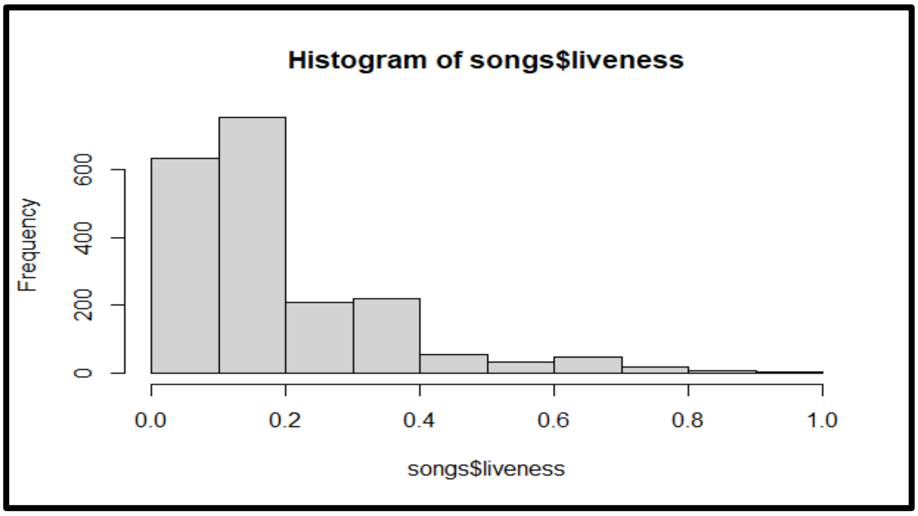
Histogram of instrumentalness



The instrumentalness shows how instrumental the song is. From the data distribution, we can see that, majority of the songs are vocal songs. However, there are small number of songs that are instrumental in the dataset. The data distribution is right-skewed.

Figure-6

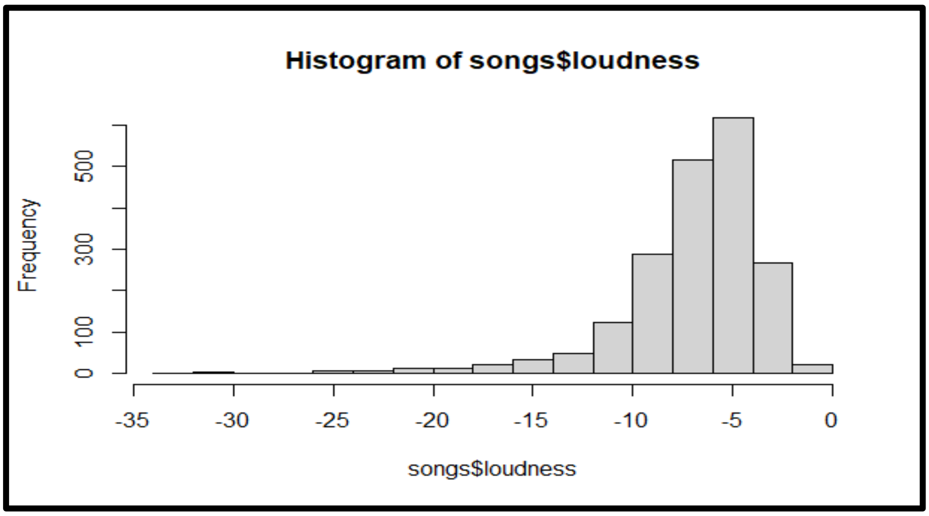
Histogram of liveness



Liveness is a measure of whether the song is recorded with live audience or not. Most of the songs are recorded without live audience. This can be observed from the data distribution. The histogram of liveness is right-skewed.

Figure-7

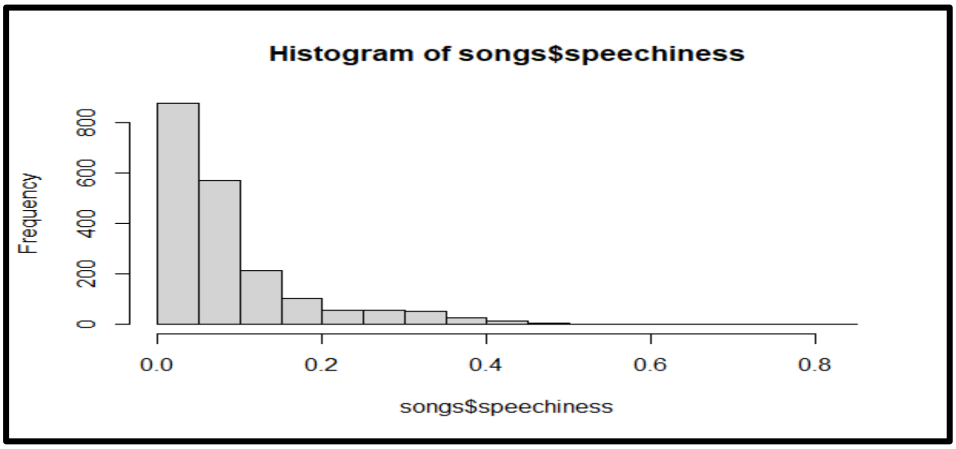
Histogram of loudness



The histogram of loudness shows that most of the songs on the Spotify data are loud in nature. The data distribution is left-skewed.

Figure-8

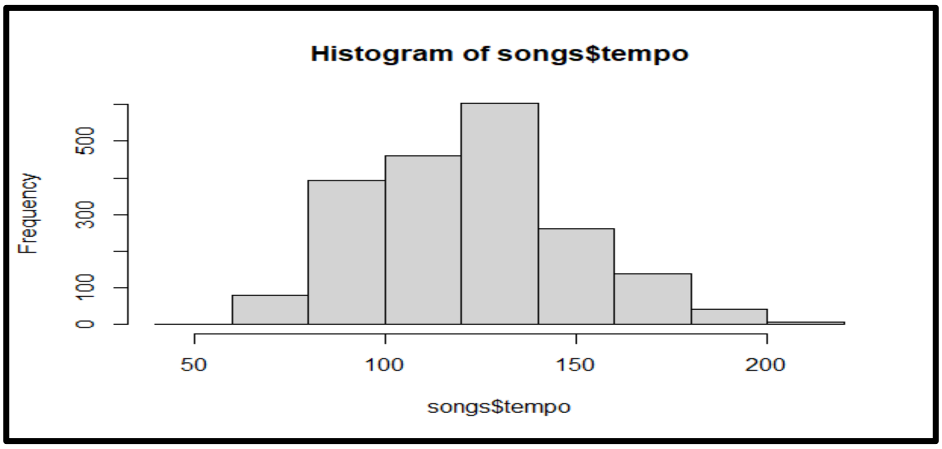
Histogram of Speechiness



The histogram of speechiness feature indicates there are no songs with total spoken words. This indicates the absence of any podcasts or any other speech related music streaming in the data. This distribution is right-skewed.

Figure-9

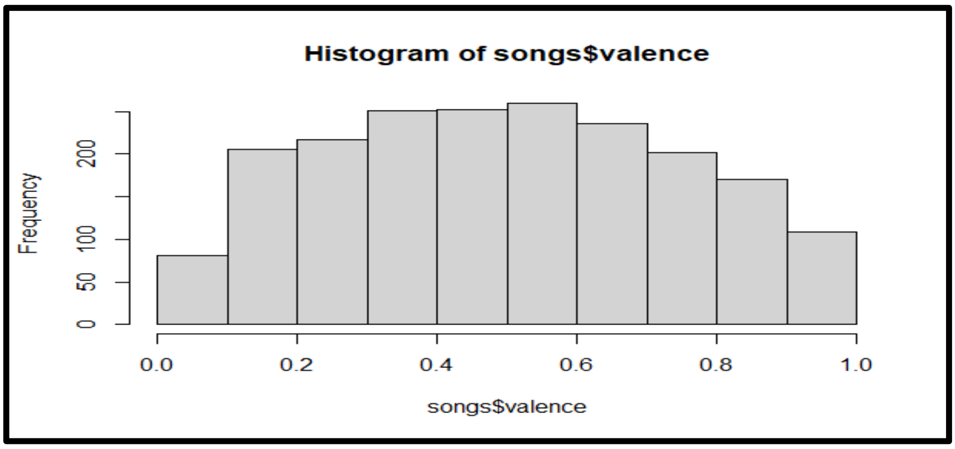
Histogram of Tempo



The histogram of tempo is normally distributed. This indicates there are songs with different ranges of tempo.

Figure-10

Histogram of valence



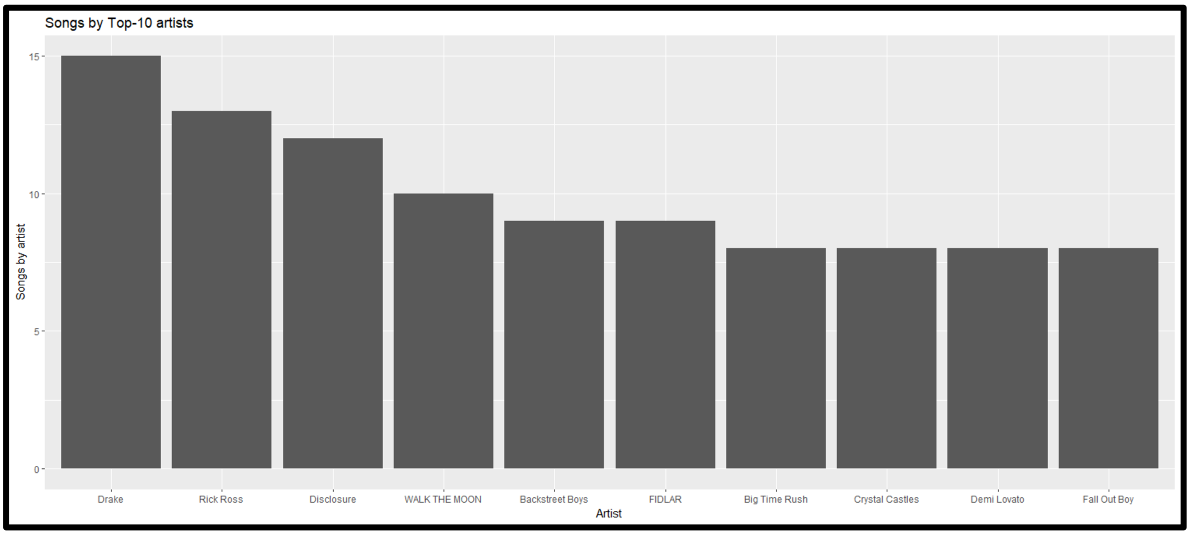
Valence is the musical positivity indicator. The distribution of valence feature is normally distributed. This indicates there are all kinds of songs suitable for different moods of the listeners.

These data distributions of individual features give us a closer look at the behaviour of the data. Now we understand how the data is distributed? is there any imbalance in the feature distribution? What characteristics are evenly distributed? We can make better explorations and determine the contextual understanding of the statistical results obtained later in the project.

As part of the exploratory data analysis, let's explore the popularity of the artists based on their songs' frequency in the Spotify dataset. This can be illustrated better with a bar chart of songs count against the artists as shown below.

Figure-11

Bar chart of Top-10 artists based on Songs count.

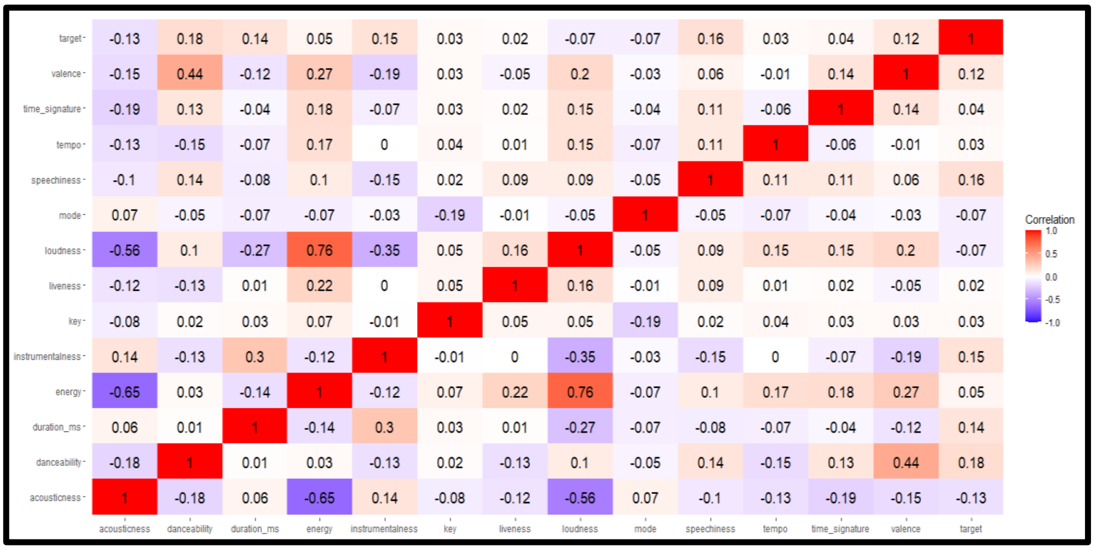


From the above bar chart, we can observe that "Drake" is the most popular artist with 15 songs on Spotify dataset followed by "Rick Ross" in the second position with 13 songs.

Correlation Matrix

Figure-12

Heat Map of Correlation Matrix between Spotify data features



	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	target
target	-0.13	0.18	0.14	0.05	0.15	0.03	0.02	-0.07	-0.07	0.16	0.03	0.04	0.12	1
valence	-0.15	0.44	-0.12	0.27	-0.19	0.03	-0.05	0.2	-0.03	0.06	-0.01	0.14	1	0.12
time_signature	-0.19	0.13	-0.04	0.18	-0.07	0.03	0.02	0.15	-0.04	0.11	-0.06	1	0.14	0.04
tempo	-0.13	-0.15	-0.07	0.17	0	0.04	0.01	0.15	-0.07	0.11	1	-0.06	-0.01	0.03
speechiness	-0.1	0.14	-0.08	0.1	-0.15	0.02	0.09	0.09	-0.05	1	0.11	0.11	0.06	0.16
mode	0.07	-0.05	-0.07	-0.07	-0.03	-0.19	-0.01	-0.05	1	-0.05	-0.07	-0.04	-0.03	-0.07
loudness	-0.56	0.1	-0.27	0.76	-0.35	0.05	0.16	1	-0.05	0.09	0.15	0.15	0.2	-0.07
liveness	-0.12	-0.13	0.01	0.22	0	0.05	1	0.16	-0.01	0.09	0.01	0.02	-0.05	0.02
key	-0.08	0.02	0.03	0.07	-0.01	1	0.05	0.05	-0.19	0.02	0.04	0.03	0.03	0.03
instrumentalness	0.14	-0.13	0.3	-0.12	1	-0.01	0	-0.35	-0.03	-0.15	0	-0.07	-0.19	0.15
energy	-0.65	0.03	-0.14	1	-0.12	0.07	0.22	0.76	-0.07	0.1	0.17	0.18	0.27	0.05
duration_ms	0.06	0.01	1	-0.14	0.3	0.03	0.01	-0.27	-0.07	-0.08	-0.07	-0.04	-0.12	0.14
danceability	-0.18	1	0.01	0.03	-0.13	0.02	-0.13	0.1	-0.05	0.14	-0.15	0.13	0.44	0.18
acousticness	1	-0.18	0.06	-0.65	0.14	-0.08	-0.12	-0.56	0.07	-0.1	-0.13	-0.19	-0.15	-0.13

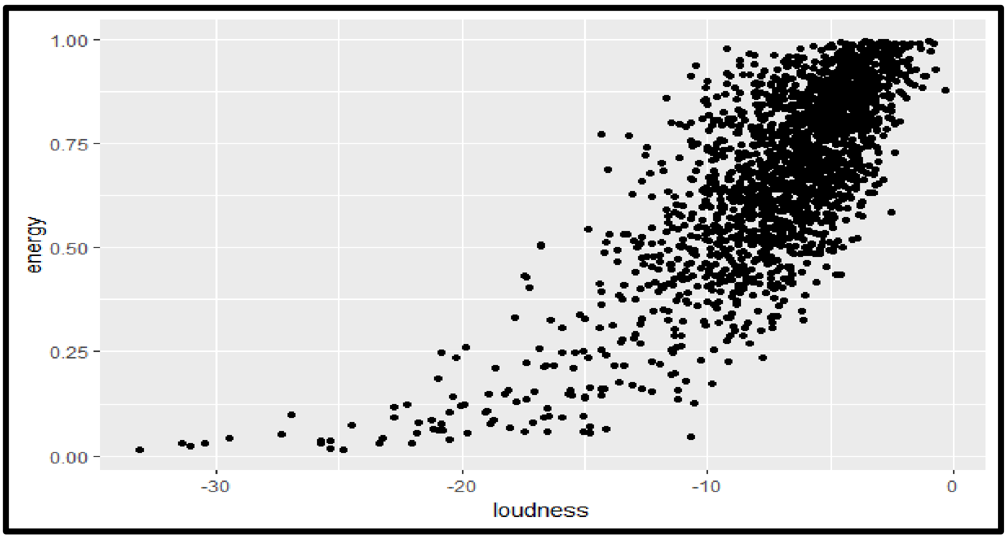
In the next step of Exploratory Data Analysis, we want to explore the relationships between the individual features. By doing so, we can understand which of the independent variables are highly correlated and which of them doesn't have any correlation among them. To draw these insights, we have plotted a correlation matrix among all numerical parameters of the Spotify songs data as shown in the above heat map. The heat map indicates the positive correlation with bright shade of "red" colour and negative correlation with bright shade of "blue" colour.

From the heat map, we can observe that the energy and loudness variable have highest positive correlation of (0.76) whereas the acousticness feature has significant negative correlation with both energy (-0.65) and loudness (-0.56). This means that the as the loudness increase the energy of the song will increase. However, with the increase in acousticness, both loudness and energy will drop.

To explore these strong correlations, we have plotted scatter plots between the highly correlated features to visualize their behaviour. The scatter plots and their relevant insights are as follows.

Figure-13

Scatter Plot drawn between energy and loudness features.



The above scatter plot illustrates the positive correlation (coeff. of variation = 0.76). We can see as the loudness is increasing, the energy also increases. Moreover, we can observe that most of the songs are highly energetic and loud songs.

Additionally, we will draw the scatter plots for acousticness and loudness, acousticness and energy to observe and explore the negative correlation among them.

Figure-14

Scatter Plot drawn between acousticness and energy.

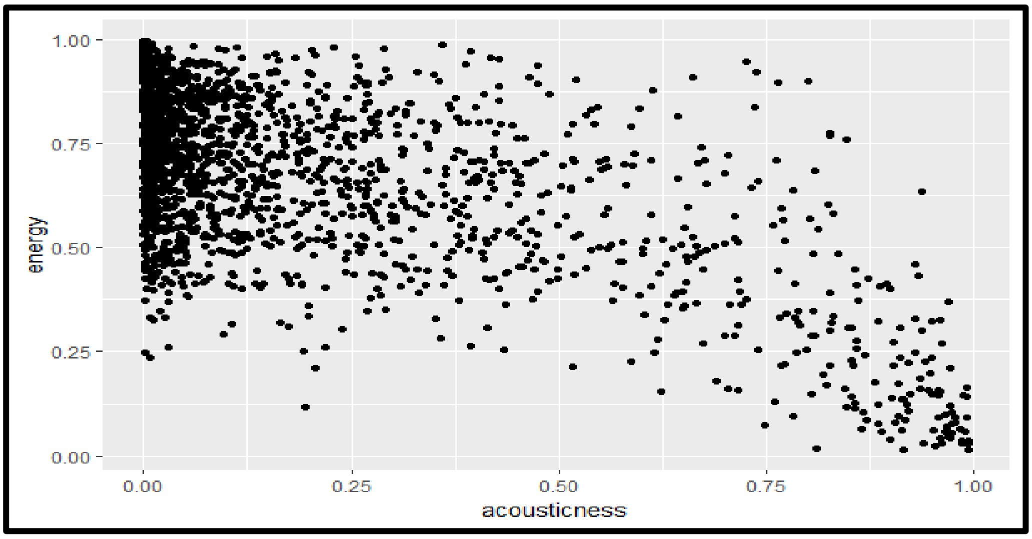
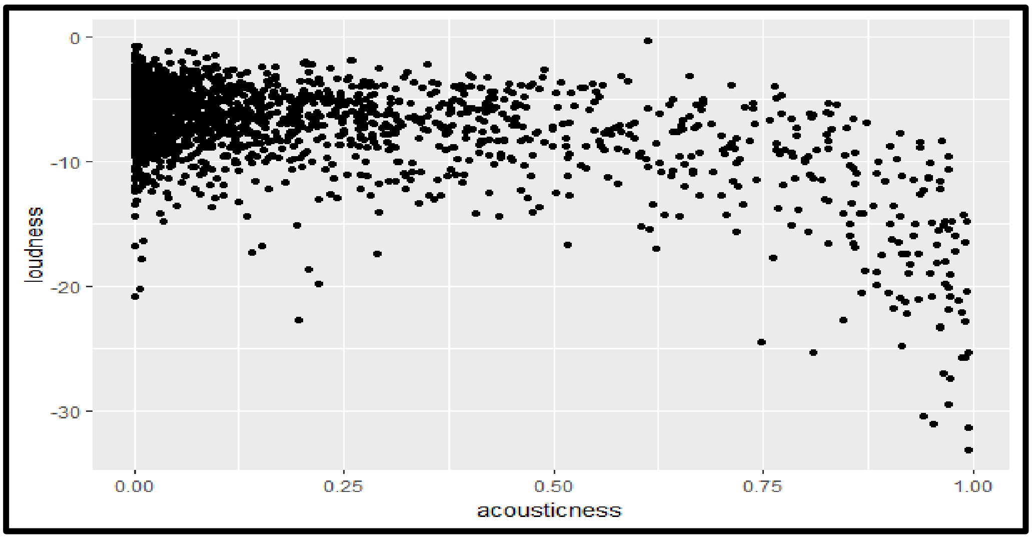


Figure-15

Scatter Plot drawn between acousticness and loudness.

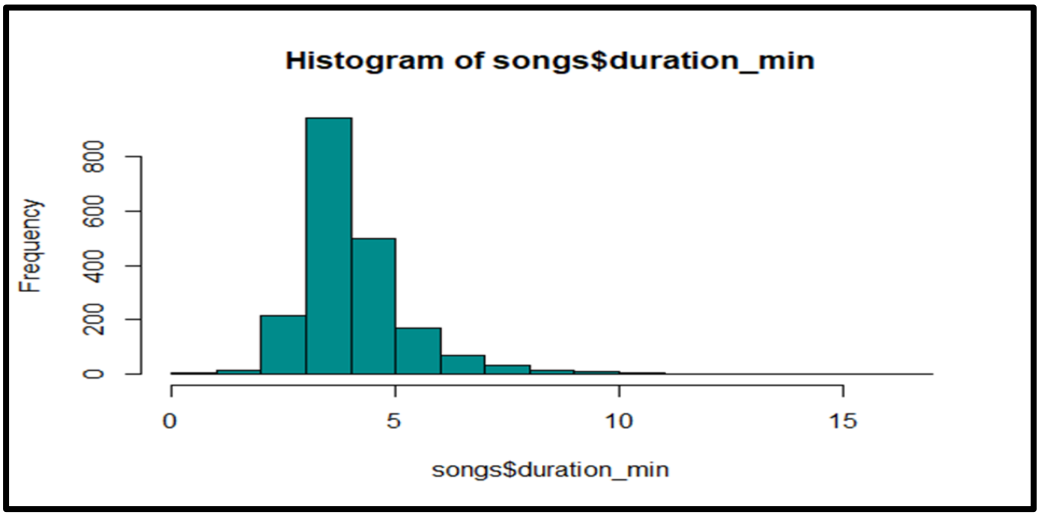


The negative correlations between acousticness and energy (COV = -0.65) & acousticness and loudness (COV = -0.56) are obvious from the above scatter plots. As the acousticness is increasing the energy is dropping gradually whereas the drop in loudness is small until 0.75 but dropped suddenly beyond acousticness = 0.75.

In the last step of exploratory data analysis, we want to explore the typical song durations for all the songs in the data. We plot the histogram of all the songs durations as shown below.

Figure-16

Histogram of song durations in minutes



From the above song durations, we can see that most of the songs are of duration 3-4 minutes. However, there are some songs with a duration beyond 10 minutes, which will be considered as outliers and handled later in the data pre-processing.

Data Pre-Processing

Before building the models, we need to ensure that our data is pre-processed to remove any unwanted elements exist in our data. The major unwanted elements in model building are the outliers. Outliers are the data points that lies away from the typical range of data points. It is important to find these outliers and handle them in a proper way as they will influence and distort the model performance.

There are lot of methods to identify outliers. The most frequently used methods to observe the outliers visually are box plots and density plots. Based on the exploratory data analysis insights, we found that there are 5 variables with outliers. These variables are duration_min, Speechiness, tempo, liveness, and energy. We have created box plots and density plots to observe the outliers and their distribution among the data points.

Figure-17

Box plot and Density plots for duration in minutes.

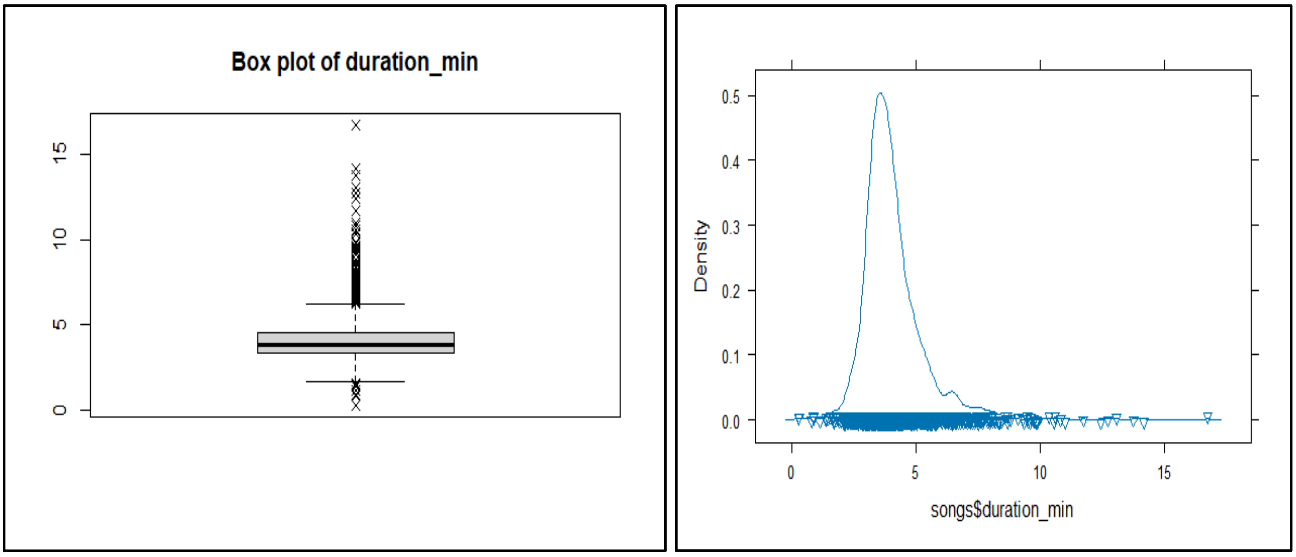


Figure-18

Box Plot and Density plots for Speechiness.

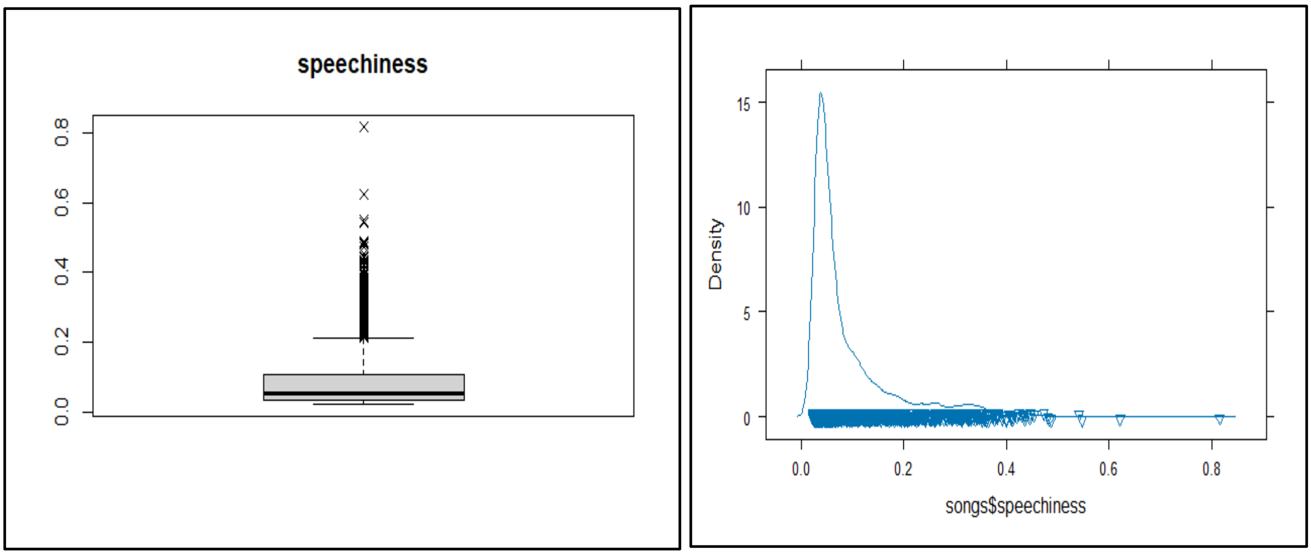


Figure-19

Box Plot and Density plots for Tempo.

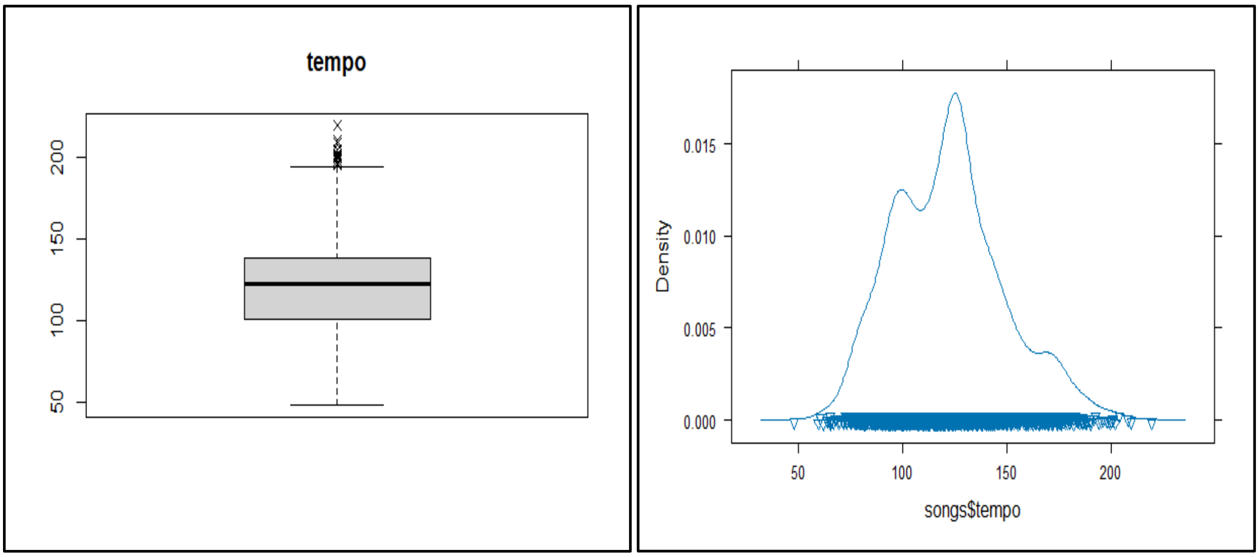


Figure-20

Box Plot and Density plots for Liveness.

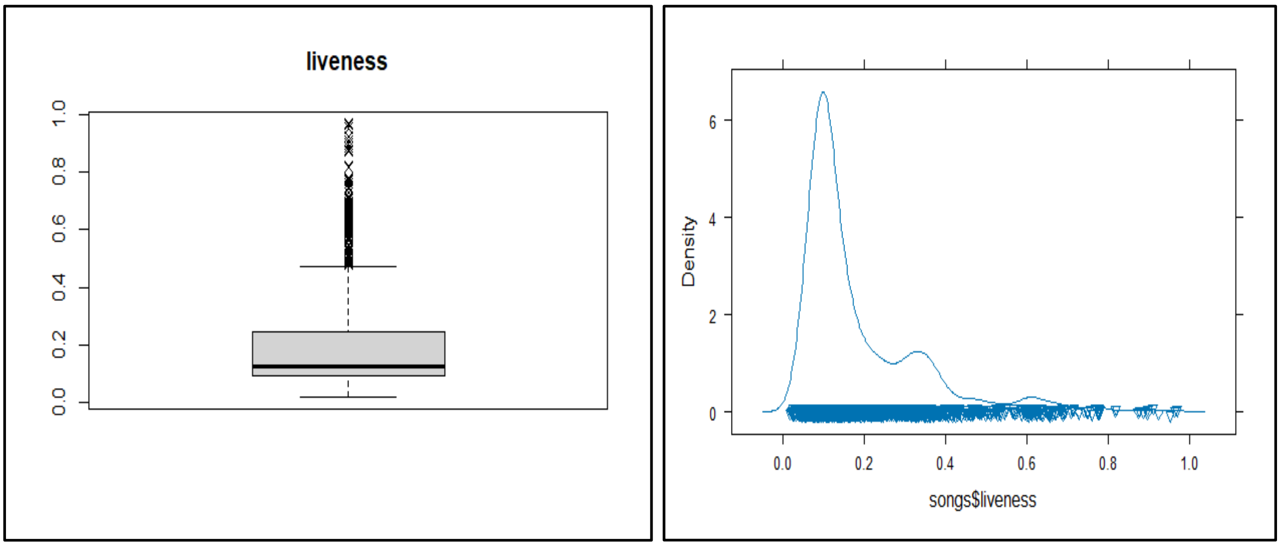
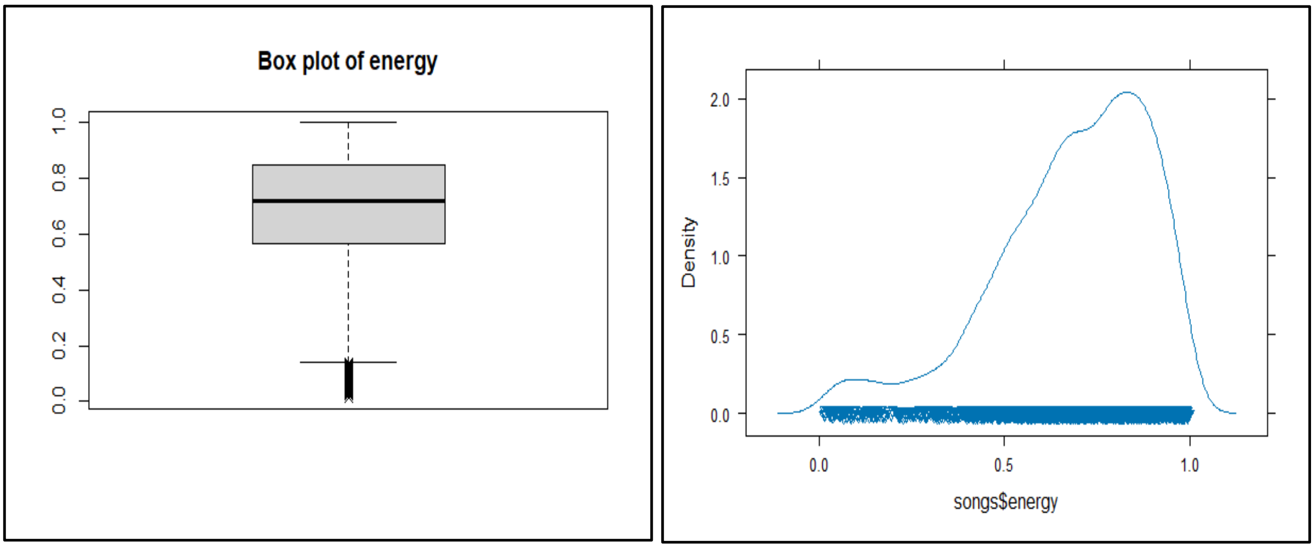


Figure-21

Box Plot and Density Plots for Energy.



From the above Box and Density plots we can observe the presence of outliers in duration, speechiness, tempo, liveness, and energy. However, these outliers appear to be part of the distribution as seen in the density plots. Removing the outliers based on visual representation leads to subjective exclusion. Hence, we will deal with these outliers numerically after standardizing the data. Therefore, we are retaining the observed outliers for now as part of the data.

Feature Engineering

Greater the features, bigger the complexity. Each feature adds a dimension to the model interpretation and addition of more dimensions increases the complexity and hinders the comprehension of model results. Hence, data features should be modified, selected, and reduced in a way that a subset of features should provide us as much information as the original data with all features. This process of handling the features for better model performance and optimization is knowns as "Feature Engineering".

As part of this project, our original dataset has 16 features at this point of time. We want to reduce as many as possible features for building the model. The feature engineering can be obtained in multiple ways such as feature modification, feature selection, and feature reduction.

Feature Modification: The original feature duration_ms is the duration of songs expressed in milli seconds. These values are in the order of hundreds of thousands whereas other features are expressed in lower ranges. Hence, for the convenience of model building, we have converted the duration from milli seconds to the minutes. This allowed easy interpretation for analysis and model building.

Feature Selection: When performing supervised learning, we can identify which features are more contributing towards the prediction of target using feature selection methods such as Forward selection, backward selection, and stepwise selection. However, as our project objective deals with unsupervised clustering models, there is no target variable to evaluate the individual feature contributions. Hence, we have not performed any feature selection for the given data.

Feature Reduction: Feature Reduction is one of the major steps in the feature engineering process. In this, we try to reduce the number of features by eliminating the features that are not useful for model building based on certain criteria such as low variance, high correlation, or Principal Component Analysis (PCA). We applied feature reduction strategies to drop unwanted features thereby reducing the complexity for model building. The feature reduction strategies are explained as below.

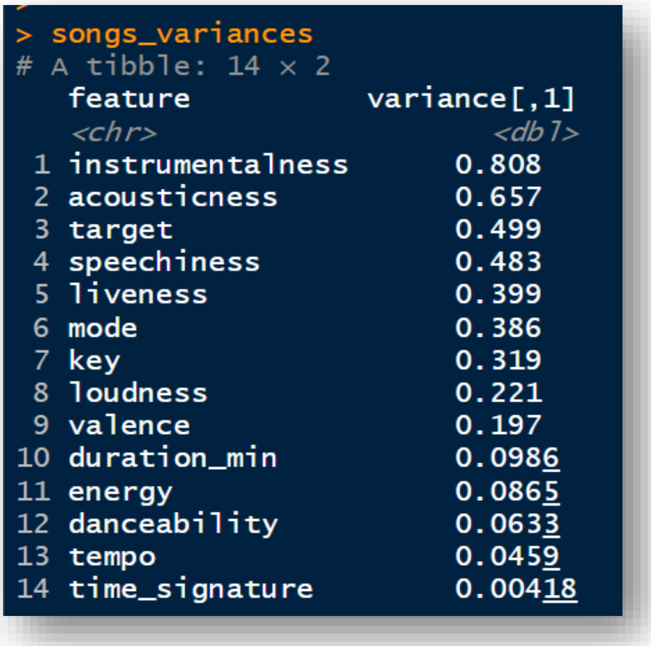
a. Feature Reduction Using the Low Variance Filter

Low variance indicated the absence of variability in the data. If any feature exhibits low variance, it means that the feature contains almost similar values throughout the data observations. These kinds of low-variance features don't provide much valuable information and doesn't help much in the model building. Hence, low variance features must be eliminated.

The variances of all features can be seen in the below screenshot of results.

Figure-22

Variance Values of Individual Features.



```
> songs_variances
# A tibble: 14 × 2
   feature          variance[,1]
   <chr>                   <dbl>
 1 instrumentalness      0.808
 2 acousticness          0.657
 3 target                0.499
 4 speechiness           0.483
 5 liveness              0.399
 6 mode                  0.386
 7 key                   0.319
 8 loudness              0.221
 9 valence               0.197
10 duration_min          0.0986
11 energy                0.0865
12 danceability          0.0633
13 tempo                 0.0459
14 time_signature        0.00418
```

When we evaluated the variances of individual features, we found that time_signature variable has a very low variance of 0.00418. This indicates there is not much variation in the data values of time_signature column. Hence, we dropped the time_signature feature as it will not add value to the model.

b. Feature Reduction using High Correlation Filter

High Correlation between two features indicates the dependency between the features. This causes multi-collinearity among the features, which creates redundancy of data features. When 2 features are highly correlated, they both represent the same information redundantly. Hence, if any of 2 features exhibits high correlation one of the two features should be dropped to avoid redundancy.

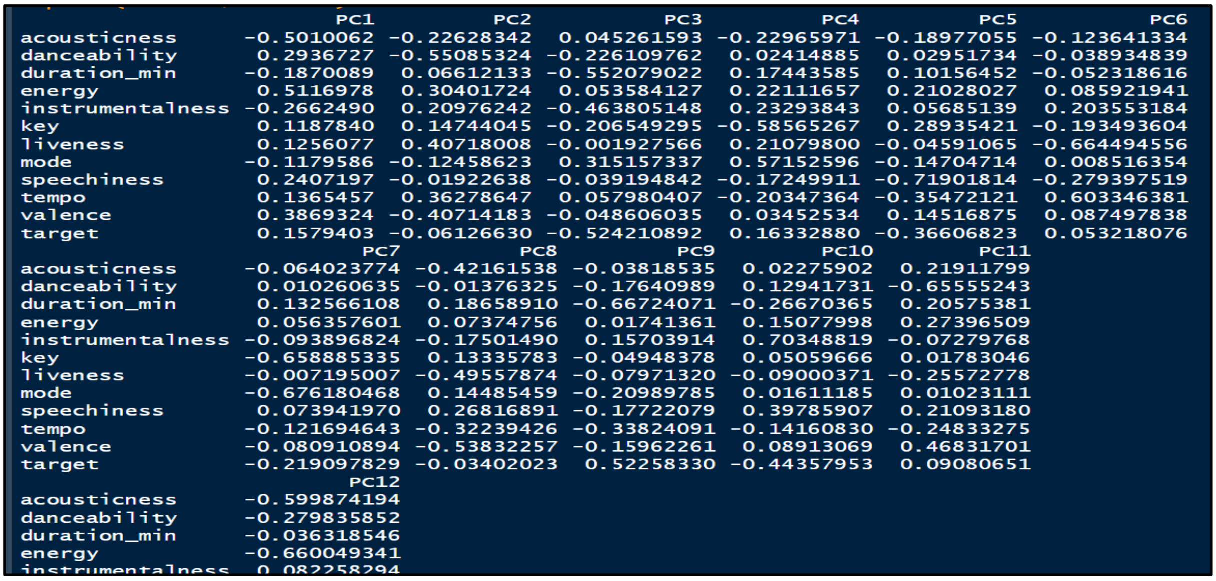
Previously, during the exploratory data analysis (EDA), from the correlation matrix of features, we have seen that loudness and energy are highly correlated with coefficient of variation of 0.76. In general, an absolute correlation coefficient of >0.7 among two or more predictors indicates the presence of multicollinearity. Hence, we wanted to keep either one of the loudness or energy variable. Hence, we are dropping the loudness feature and keeping the energy feature in the data.

c. Feature Reduction by Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique used for dimensionality reduction. It transforms a dataset into a new set of uncorrelated variables called principal components. These components capture the maximum variance in the data. By selecting a subset of the principal components, PCA helps reduce the dimensionality of the data while retaining as much information as possible. It is achieved by finding orthogonal directions in the data that represent the most significant features, allowing for efficient representation and visualization of the dataset. We have performed principal component analysis on our Spotify songs dataset. The principal components were derives as shown below.

Figure-23

Principal Components generated after PCA.

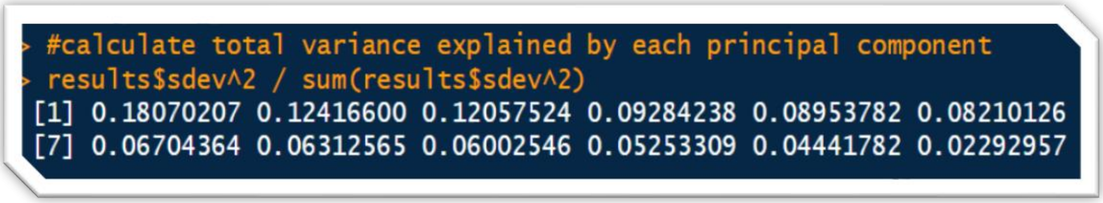


```
                        PC1         PC2          PC3         PC4         PC5          PC6
acousticness     -0.5010062 -0.22628342  0.045261593 -0.22965971 -0.18977055 -0.123641334
danceability      0.2936727 -0.55085324 -0.226109762  0.02414885  0.02951734 -0.038934839
duration_min     -0.1870089  0.06612133 -0.552079022  0.17443585  0.10156452 -0.052318616
energy            0.5116978  0.30401724  0.053584127  0.22111657  0.21028027  0.085921941
instrumentalness -0.2662490  0.20976242 -0.463805148  0.23293843  0.05685139  0.203553184
key               0.1187840  0.14744045 -0.206549295 -0.58565267  0.28935421 -0.193493604
liveness          0.1256077  0.40718008 -0.001927566  0.21079800 -0.04591065 -0.664494556
mode             -0.1179586 -0.12458623  0.315157337  0.57152596 -0.14704714  0.008516354
speechiness       0.2407197 -0.01922638 -0.039194842 -0.17249911 -0.71901814 -0.279397519
tempo             0.1365457  0.36278647  0.057980407 -0.20347364 -0.35472121  0.603346381
valence           0.3869324 -0.40714183 -0.048606035  0.03452534  0.14516875  0.087497838
target            0.1579403 -0.06126630 -0.524210892  0.16332880 -0.36606823  0.053218076
                          PC7         PC8         PC9        PC10        PC11
acousticness     -0.064023774 -0.42161538 -0.03818535  0.02275902  0.21911799
danceability      0.010260635 -0.01376325 -0.17640989  0.12941731 -0.65555243
duration_min      0.132566108  0.18658910 -0.66724071 -0.26670365  0.20575381
energy            0.056357601  0.07374756  0.01741361  0.15077998  0.27396509
instrumentalness -0.093896824 -0.17501490  0.15703914  0.70348819 -0.07279768
key              -0.658885335  0.13335783 -0.04948378  0.05059666  0.01783046
liveness         -0.007195007 -0.49557874 -0.07971320 -0.09000371 -0.25572778
mode             -0.676180468  0.14485459 -0.20989785  0.01611185  0.01023111
speechiness       0.073941970  0.26816891 -0.17722079  0.39785907  0.21093180
tempo            -0.121694643 -0.32239426 -0.33824091 -0.14160830 -0.24833275
valence          -0.080910894 -0.53832257 -0.15962261  0.08913069  0.46831701
target           -0.219097829 -0.03402023  0.52258330 -0.44357953  0.09080651
                         PC12
acousticness     -0.599874194
danceability     -0.279835852
duration_min     -0.036318546
energy           -0.660049341
instrumentalness  0.082258294
```

After generating the principal component, we wanted to evaluate how much variance each principal component is capturing. This would help us in deciding how many components can be selected such that they can capture maximum variance. The variance captured by each principal component is as shown below.

Figure-24

Variances Captured by each Principal Component after PCA.



```
> #calculate total variance explained by each principal component
> results$sdev^2 / sum(results$sdev^2)
 [1] 0.18070207 0.12416600 0.12057524 0.09284238 0.08953782 0.08210126
 [7] 0.06704364 0.06312565 0.06002546 0.05253309 0.04441782 0.02292957
```

From the above screenshot, we can see that, principal component-1 (PC1) capturing only 18% of variance, principal component-2 (PC2) holding only 12% of variance and so on. To explain 88% of the variance, we need to consider 9 PCA components. Considering 9 features for only 88% of variance is not affordable. In this scenario, the principal component analysis (PCA) is not allowing feature reduction significantly. Hence, we are going with the actual data for clustering analysis.

Final Data Preparation for Model Building

Now that we performed all preparatory work on data, data is available for model building. However, the features in the original data are in different order of values. This can cause bias in the model due to various numerical value ranges. To avoid this to happen, we can bring all data values of all features into a uniform order by applying standardization.

There are essentially 2 kinds of standardization techniques available. They are.

1. Min-max normalization = (x – min(x))/range(x)
2. Z-score standardization = (x – mean(x))/sd(x)

As the original data is having a lot of observations and contains outliers in them, we have considered to apply Z-score standardization on our data points using scale () function. This Z-score standardization transforms all data values into the distance of the data points from the mean in terms of standard deviation. Hence, we will get numbers ranging from negative finite values to positive finite values depends on their position on the either side of mean value. Now the data appears to be uniform.

Now that we have scaled the data, it is easy to deal with the outliers numerically. Typically, after scaling or standardization, any values that are 3 standard deviations away from mean = 0 will be considered as outliers. In the scaled data, any values above +3 or below -3 will be identified as outliers. Hence, we dropped the rows with all outlier values in duration_min, speechiness, tempo, liveness, and energy variables. At the end, we were left with 1840 observations after the outlier omission.

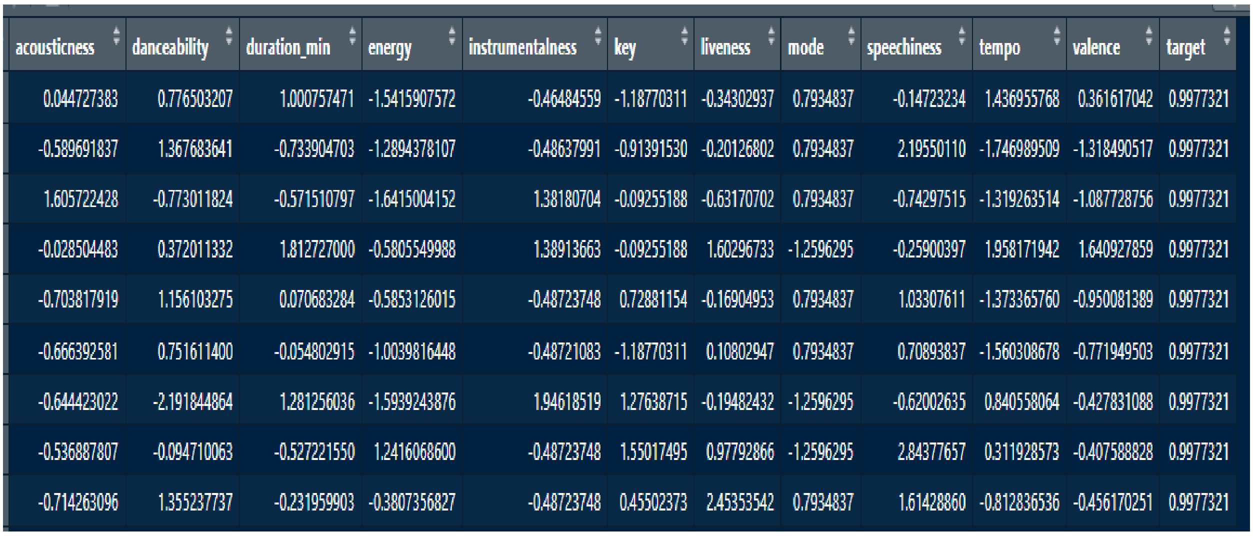
As we can only build clustering models using the numerical data features, we don't need the character data type columns. Hence, the character columns, such as song_title & artist, were dropped for further analysis, finally leaving 12 numeric features in the final dataset.

Summary

As part of this introduction to our project, first, we defined the Problem statement and the abstract approach for solving the business problem. Second, we have selected the appropriate dataset with sufficient observations and features. Third, we cleaned and pre-processed the data for further analysis. Fourth, we performed exploratory data analysis to find the patterns and insights of all the data using statistical analysis and visualizations. Fifth, we applied feature engineering strategies on the dataset to reduce the dimensionality problem. Finally, we refined the data by standardizing, removing outliers, and omitting unnecessary columns. After all operations were performed the dataset look as shown below.

Figure-25

Final Dataset Ready for Model Building.



acousticness	danceability	duration_min	energy	instrumentalness	key	liveness	mode	speechiness	tempo	valence	target
0.044727383	0.776503207	1.000757471	-1.5415907572	-0.46484559	-1.18770311	-0.34302937	0.7934837	-0.14723234	1.436955768	0.361617042	0.9977321
-0.589691837	1.367683641	-0.733904703	-1.2894378107	-0.48637991	-0.91391530	-0.20126802	0.7934837	2.19550110	-1.746989509	-1.318490517	0.9977321
1.605722428	-0.773011824	-0.571510797	-1.6415004152	1.38180704	-0.09255188	-0.63170702	0.7934837	-0.74297515	-1.319263514	-1.087728756	0.9977321
-0.028504483	0.372011332	1.812727000	-0.5805549988	1.38913663	-0.09255188	1.60296733	-1.2596295	-0.25900397	1.958171942	1.640927859	0.9977321
-0.703817919	1.156103275	0.070683284	-0.5853126015	-0.48723748	0.72881154	-0.16904953	0.7934837	1.03307611	-1.373365760	-0.950081389	0.9977321
-0.666392581	0.751611400	-0.054802915	-1.0039816448	-0.48721083	-1.18770311	0.10802947	0.7934837	0.70893837	-1.560308678	-0.771949503	0.9977321
-0.644423022	-2.191844864	1.281256036	-1.5939243876	1.94618519	1.27638715	-0.19482432	-1.2596295	-0.62002635	0.840558064	-0.427831088	0.9977321
-0.536887807	-0.094710063	-0.527221550	1.2416068600	-0.48723748	1.55017495	0.97792866	-1.2596295	2.84377657	0.311928573	-0.407588828	0.9977321
-0.714263096	1.355237737	-0.231959903	-0.3807356827	-0.48723748	0.45502373	2.45353542	0.7934837	1.61428860	-0.812836536	-0.456170251	0.9977321

Plan Ahead

With the final data available as shown above, will proceed into the model building phase. We will build clustering models using K-means clustering algorithms and Hierarchical clustering algorithms. Once, the models were created an optimized. We will conduct performance evaluation tests to verify the similarities and dissimilarities within and between the clusters. After thorough performance evaluation, we will compare both clustering models to identify the best model. This finally selected model along with performance results will be demonstrated at the end of the project to the key stakeholders (our peers). The finally selected model will be the assistive model for Spotify song recommendations algorithms for suggesting the users more songs from the clusters they are listening from. Thus, in this manner, the clustering models resolves the problem statement mentioned at the beginning of this report.

References

- Marylou. (2023, April 25). *Song Recommender ML Analysis.* Retrieved from Kaggle.com: https://www.kaggle.com/code/marylou22/song-recommender-ml-analysis
- De Dios Santos, J. (2018, May 25). Is my Spotify music boring? An analysis involving music, data, and machine learning. Medium. https://towardsdatascience.com/is-my-spotify-music-boring-an-analysis-involving-music-data-and-machine-learning-47550ae931de
- How Spotify’s Algorithm Works? A Complete Guide to Spotify Recommendation System [2022] | Music Tomorrow Blog. (n.d.). https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022