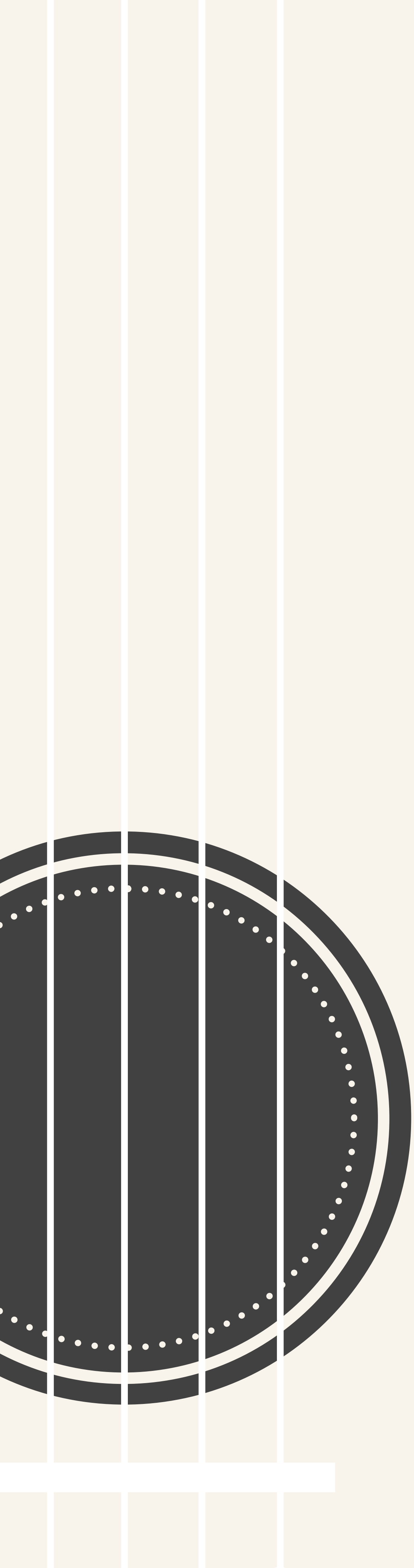


The Hit Factor

Lim Xi Chen Terry A0199513W
Loke Kay Chi A0200865J
Nigel Issac Chua Kiat Fang A0206089B
Tan Jie Yi A0206383H
Wong Zea Teng A0204680L
Yang Zhuolin A0180103Y
Zhong Zhaoping A0194519U

DSA3101 AY 21/22
Group 13 Assignment 1 Report

Project Contributions

Lim Xi Chen Terry

- Provided recommendations
- Performed data scraping
- Performed preliminary studies (SMOTE + Regression Trees)
- Performed EDA + Factor Analysis + Decision Tree + Analysis of results
- Prepared & Presented slides

Loke Kay Chi

- Provided recommendations
- Performed data cleaning + PCA
- Drafted + vetted report
- Designed report layout

Nigel Issac Chua Kiat Fang

- Provided recommendations
- Tried Factor Analysis + NLP + Logistic Regression

Tan Jie Yi

- Provided recommendations
- Performed data scraping
- Performed EDA + Data cleaning + Decision Tree + Analysis of results
- Prepared slides

Wong Zea Teng

- Provided recommendations
- Performed data scraping
- Performed Data cleaning + PCA + Decision Tree
- Prepared & Presented slides

Yang Zhuolin

- Provided recommendations
- Performed data scraping
- Performed Data cleaning + PCA + Decision Tree
- Prepared & Presented slides

Zhong Zhaoping

- Provided recommendations
- Drafted report

Abstract

Every year, millions of songs are being released, but not all songs become popular. Like any craft, music production requires a lot of time and effort, as a song has to go through several iterations of recording and fine-tuning before it can be released. In addition, artists also invest a lot of money into their music, be it for promotional materials, scheduling world tours, or as merchandise for their fans. While some songs make it to the top hits, others remain unheard of.

On Spotify, the payout range averages to $0.004 per stream, and for an artist to earn $4,000, they would require approximately a million listens on Spotify, which is only achievable by the biggest hits.[1] Do popular songs share similar musical traits, or are there external factors not even related to the composition of a piece that makes it popular? By knowing the traits of popular music, production companies can make more informed decisions on their budgeting and many more factors that go into the music production.

In this project, we adopted several statistical learning methods to gain information about the different song features in hopes to find out which feature(s) contribute(s) largely to a song's popularity.

Introduction

Music trends change all the time. Artists themselves are delving into unknown musical territories and experimenting with different tunes in hopes of producing fan-favourite songs. The American Music Awards have seen various artists from different musical genres grabbing home the coveted "Artist of the Year" title, and as decades come and go, we see new artists being nominated. Akin to biological evolution, music that is fresh and inviting will often be rewarded. Unless the artist is able to evolve to cater to the public's musical preferences, it is out with the old and in with the new.

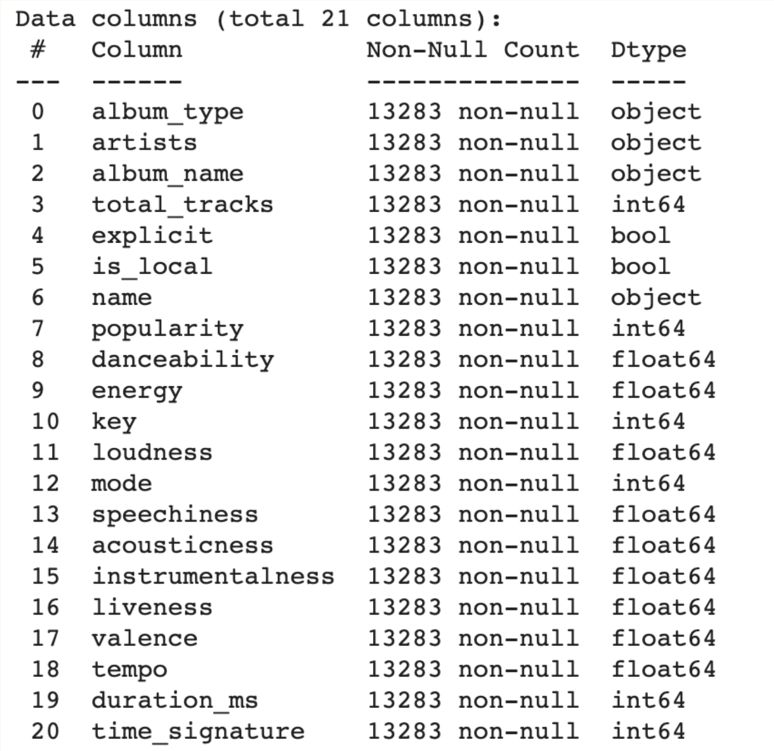
What makes a song popular? Is it having a catchy tune that is both compelling and sells well with the audience? Does it carry a political reference, like artist H.E.R.'s "I Can't Breathe", which won a Grammy for Song of the Year 2021? In our analysis, we attempt to find out which musical attribute(s) play a large part in influencing the popularity of a song through modelling a decision tree.

[1] "How much does Spotify pay per stream? Streaming payouts comparison [2021]". Retrieved 2021-09-15. https://freeyourmusic.com/blog/how-much-does-spotify-pay-per-stream

Dataset

Spotify is currently the most popular audio streaming platform, housing over 70 million tracks.[2] We decided to scrape our own data, rather than obtain well-prepared datasets from platforms like Kaggle, as we were able to scrape variables of our choosing to make our analysis more interesting. Our dataset was scraped from Spotify's database using their open API[3], and consisted of 21 variables (features): Album type, Artists, Album name, Total tracks, Is the song explicit?, Is the song local?, Song name, Song popularity, Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Duration (in milliseconds), Time Signature.

A total of 13,500 random songs (observation) from 2010 to 2021 were collected. The randomness, coverage and representativeness of our dataset have met the research standards to yield convincing statistical learning analysis. After performing some pre-analysis, no missing values were found in our dataset (see Figure 1.0).



```
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   album_type        13283 non-null  object
 1   artists           13283 non-null  object
 2   album_name        13283 non-null  object
 3   total_tracks      13283 non-null  int64
 4   explicit          13283 non-null  bool
 5   is_local          13283 non-null  bool
 6   name              13283 non-null  object
 7   popularity        13283 non-null  int64
 8   danceability      13283 non-null  float64
 9   energy            13283 non-null  float64
 10  key               13283 non-null  int64
 11  loudness          13283 non-null  float64
 12  mode              13283 non-null  int64
 13  speechiness       13283 non-null  float64
 14  acousticness      13283 non-null  float64
 15  instrumentalness  13283 non-null  float64
 16  liveness          13283 non-null  float64
 17  valence           13283 non-null  float64
 18  tempo             13283 non-null  float64
 19  duration_ms       13283 non-null  int64
 20  time_signature    13283 non-null  int64
```

Figure 1.0

[2] Spotify For the Records. Retrieved 2021-09-08. https://newsroom.spotify.com/company-info/

[3] Spotify Audio Features & Analysis Documentation. Retrieved 2021-09-09. https://developer.spotify.com/documentation/web-api/reference/#category-tracks

Danceability

How suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Energy

A perceptual measure of intensity and activity, ranging from 0.0 to 1.0. Typically, highly energetic tracks like death metal scores high on the scale, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

Key

Standard pitch notation.[4]

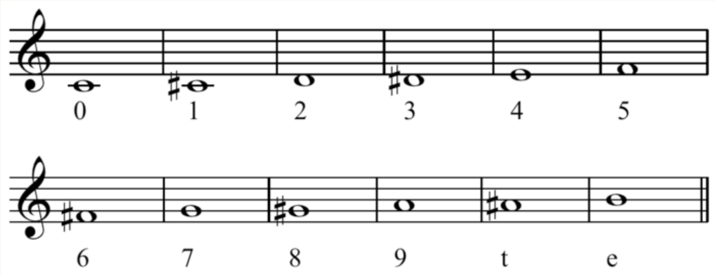


Figure 1.1

Loudness

The overall loudness of a track in decibels (dB), ranging from -60 to 0 dB. Values are averaged across the entire track.

Mode

The modality (major or minor) of a track, or the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0. Generally, major tracks sound bright and happy, whereas minor tracks sound dark and gloomy.

[4] Wikipedia Pitch Class. Retrieved 2021-09-09. https://en.wikipedia.org/wiki/Pitch_class

Speechiness

Detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

Acousticness

Confidence measures from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents a high confidence that the track is acoustic.

Instrumentalness

Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

Liveness

Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.

Valence

A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Tempo

The overall estimated speed or pace of a track in beats per minute (BPM).

Time Signature

A notational convention to specify how many beats are in each bar (or measure).

Methodology

This project aims to explain the influence of different features and characteristics of music on its popularity, through the implementation of different statistical learning methods to provide a broader and constructive support for popularity analysis. We first conducted exploratory data analysis (EDA), followed by factor analysis to provide us with a preliminary summary of the dataset.

Next, we decided to adopt decision tree-based models to explore the extent to which each feature affects the popularity of a song. We decided not to utilise simple linear regression models as we assumed our data to be highly non-linear. Compared with a linear regression model, we believed that a decision tree model can better handle the non-linearity and collinearity in our data.[5]

Lastly, it was a matter of finding the model that gave us the best results, and analysing them to generate insights. In the following segments, we attempt to describe our EDA and findings.

[5] Varghese, D. (2018). "Comparative Study on Classic Machine Learning Algorithms". https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222

EDA

We noticed that 14% of our dataset have a popularity score of 0 (see Limitations for further explanation), and hence decided to exclude them from our data. Figure 2.0 and Figure 2.1 represents the popularity distribution before and after removing the 0 scores respectively.

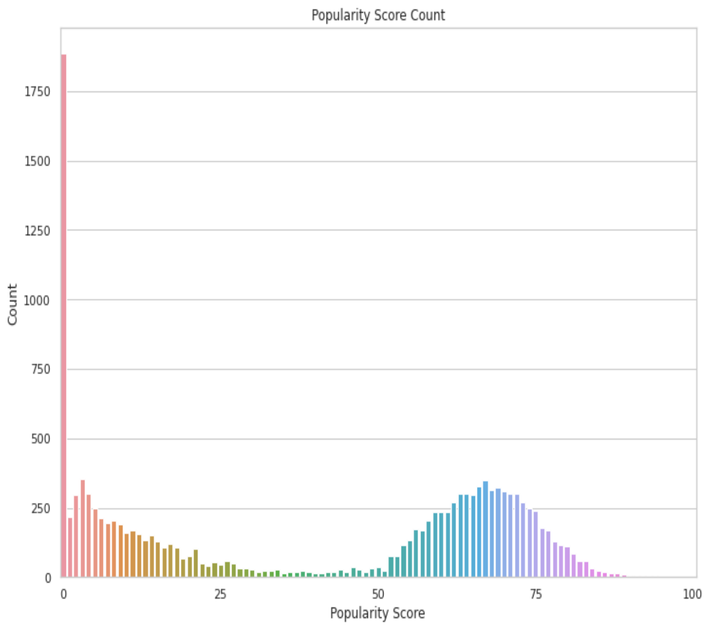


Figure 2.0

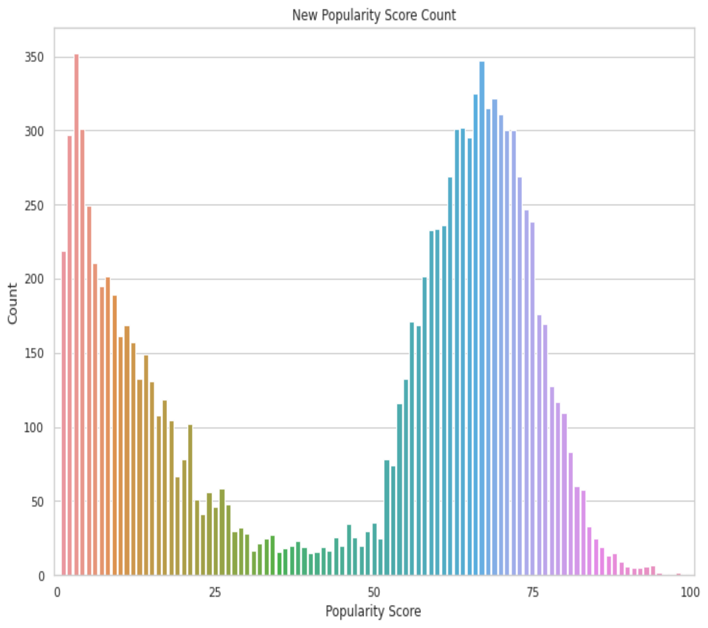


Figure 2.1

Correlation Between Attributes

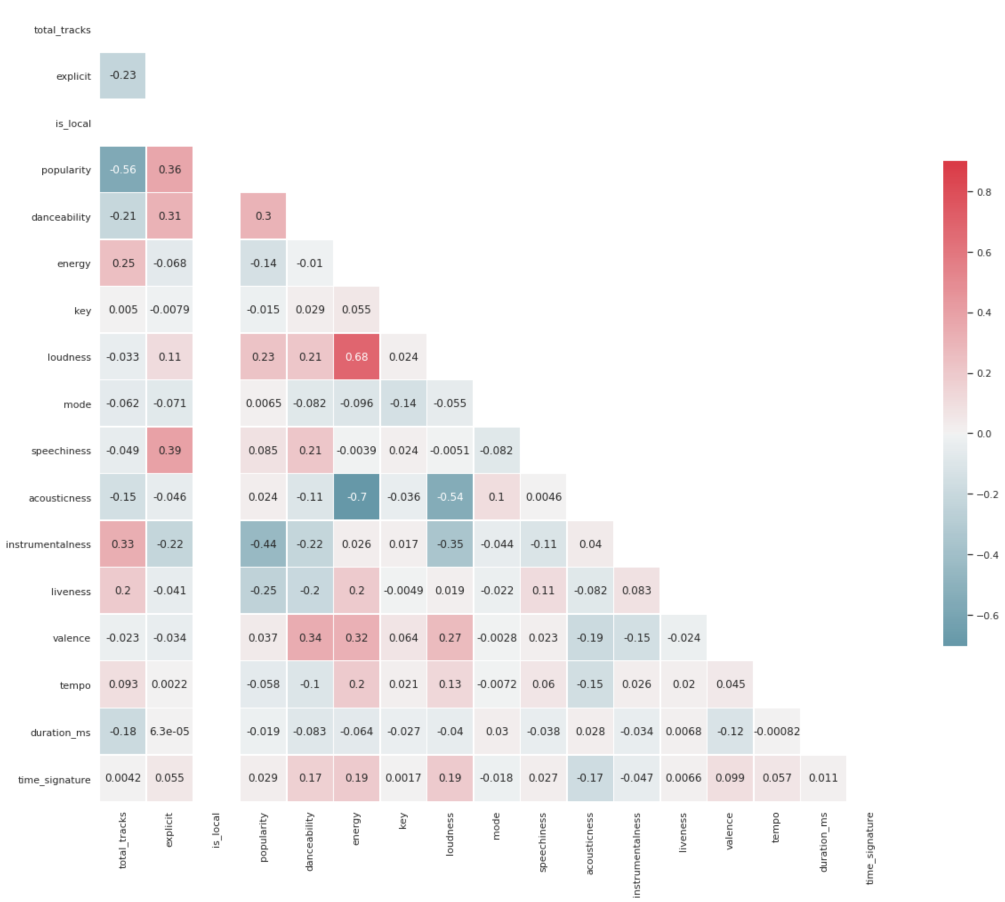


Figure 2.2

Looking at the pairwise correlation matrix, we can see that the attributes are not highly correlated, with the highest being 0.68 between loudness and energy. Thus, principal component analysis may not be very useful in this case.

Removing Certain Features

Time signatures are notations to specify how many beats are there in each measure. From our EDA, we can see that time signatures of 4 (i.e. four crotchet or quarter note beats in a measure) are most common (see Figure 2.3), and the mean popularity score did not vary much among the other time signatures (see Figure 2.4). Thus, we decided to drop the *Time signature* feature from our dataset. Following, we dropped the *Is the song local* feature from our dataset, as the songs that we scraped all turned out to not have originated from Singapore (i.e. all False). We also dropped three more column features that are not necessary for modelling, namely *Artists*, *Album name* and *Song name*.

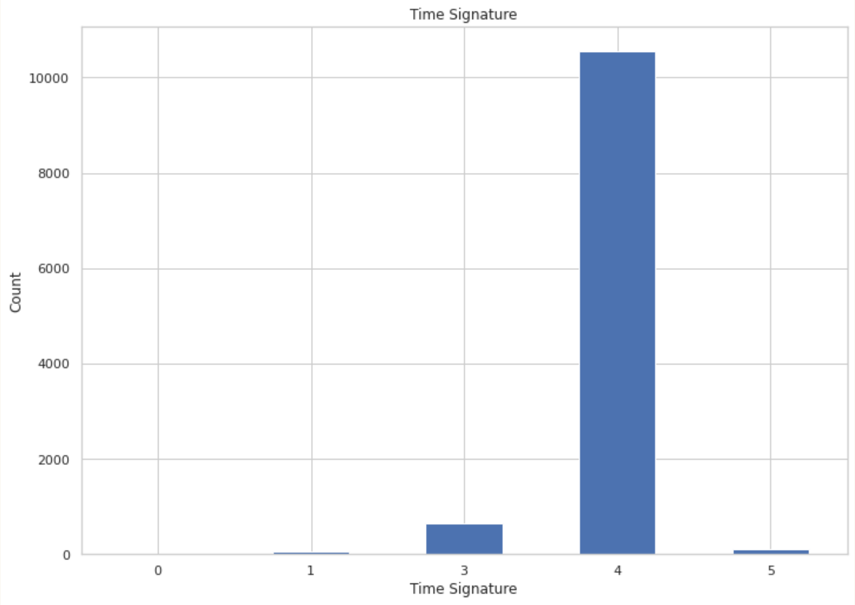


Figure 2.3

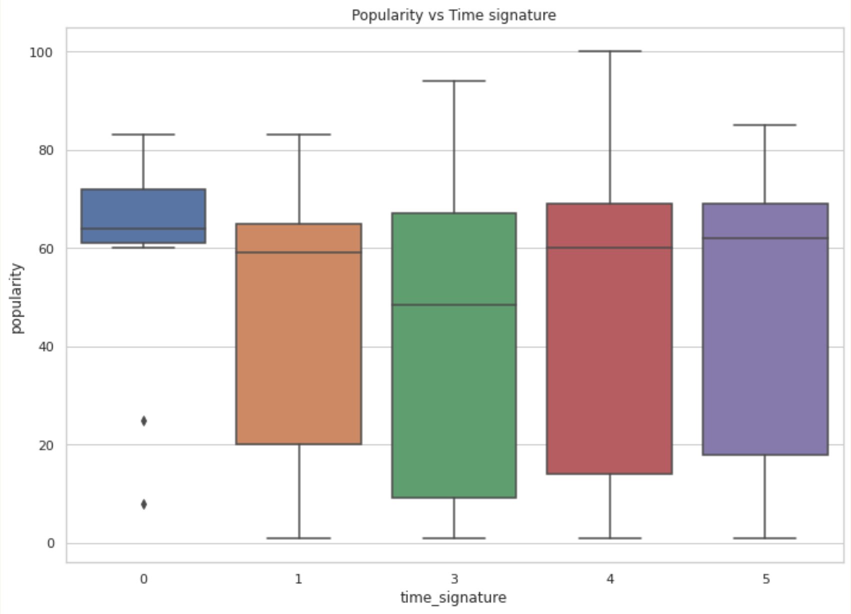


Figure 2.4

Skewed Data Features

From our EDA, we noticed that features such as *Speechiness*, *Acousticness*, *Instrumentalness* and *Liveness* are right-skewed, while features like *Danceability*, *Energy* and *Loudness* are left-skewed. Figure 2.5 illustrates the distribution of the speechiness attribute of our dataset.

Speechiness is a measure of how close a song comes to being speech-like, hence, it would make sense for the majority of our dataset to have low speechiness. The same can be argued for acousticness, instrumentalness and liveness. On the other hand, in this era of pop music, we would expect that loud and energetic songs appeal better to the general audience, hence, having our dataset to be left-skewed for features like danceability or energy is not surprising.

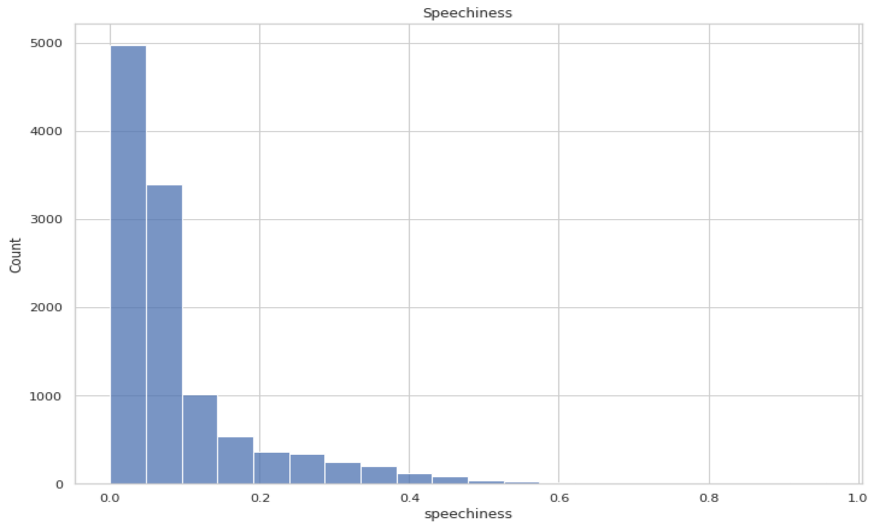


Figure 2.5

Duration (in minutes)

The data for the duration of songs turns out to be extremely right-skewed, which is not surprising given that most songs are approximately five minutes long. When plotted against the popularity score, we noticed that songs with longer durations tend to have lower popularity scores (see Figure 2.7), hence, we decided not to remove this feature from our dataset as it might generate useful insights.

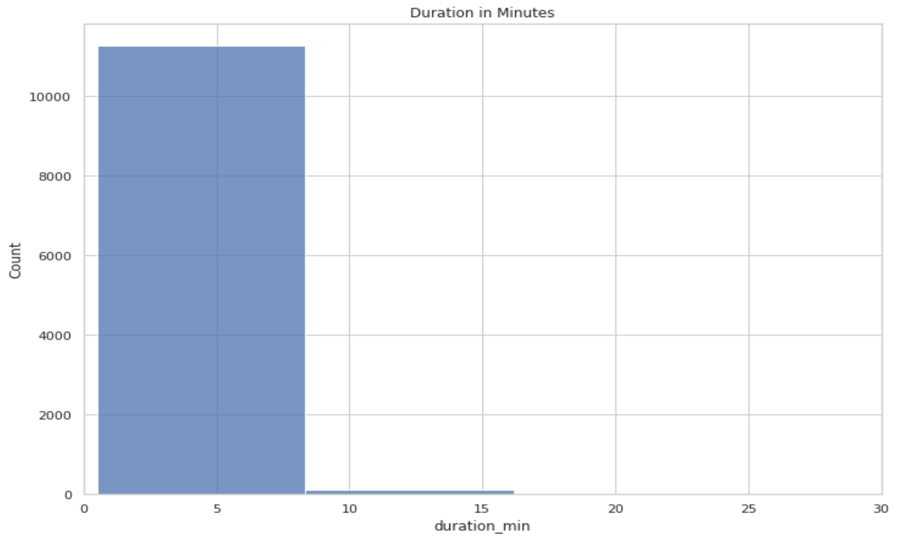


Figure 2.6

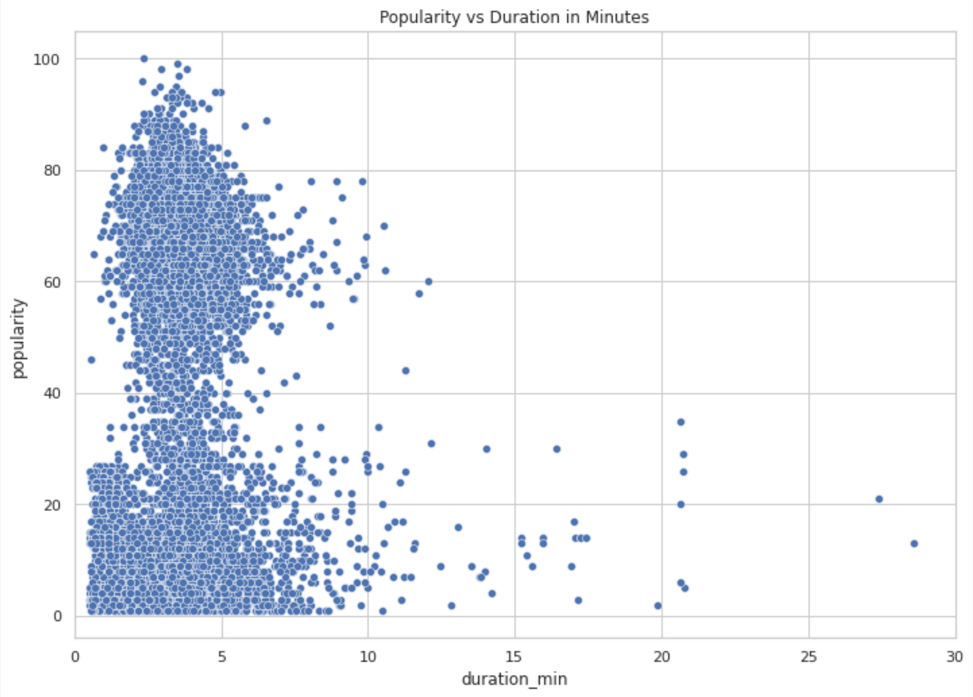


Figure 2.7

Findings

Factor Analysis

We used the Bartlett's Sphericity Test and the Kaiser-Meyer-Olkin (KMO) Test to do a preliminary check on whether our dataset is suitable for factor analysis. The Bartlett's Sphericity Test tests on whether the observed variables intercorrelate using the observed correlation matrix against the identity matrix. The KMO Test measures the suitability of our data for factor analysis by estimating the proportion of variance among all observed variables. A considerably low p-value of 0.0 was obtained for the Bartlett's Sphericity Test, and a score of 0.655 was obtained for the KMO test. From the above two metrics, since the p-value was less than 0.05 and the KMO score was more than 0.5, this indicated that factor analysis may be useful with our data. However, despite the metrics showing decent justification of conducting factor analysis, further analysis showed otherwise.

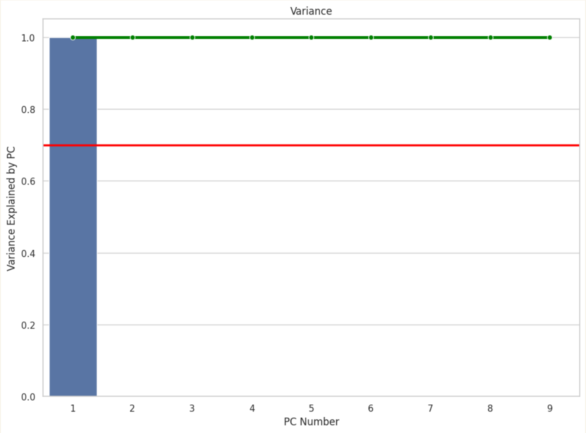


Figure 3.0

From Figure 3.0, the horizontal line remained at 1.0 demonstrates that the number of principal components has no effect on the proportion of variance explained.

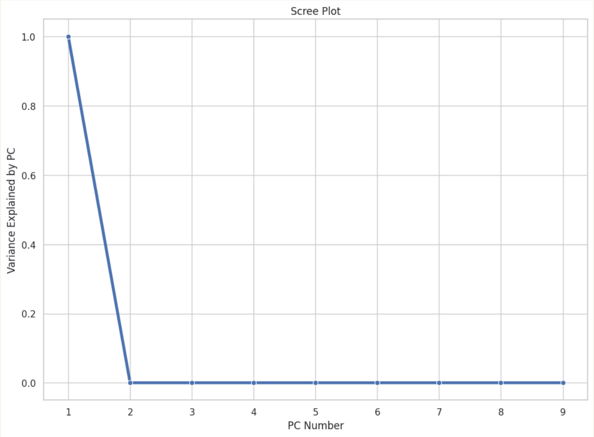


Figure 3.1

From Figure 3.1, we decided to perform factor analysis on two clusters. However, when examining the variables within each principal component, little connections can be extracted from the variables since the variables were quite "different" from each other. Furthermore, we obtained a Cronbach's Alpha score of less than 0.7, which indicated that factor analysis may not be ideal.

To conclude, factor analysis was not suitable to provide us with useful and insightful information, and hence, we decided to adopt other models.

Model Experiments

We conducted an initial simple linear regression to serve as a benchmark. With the existence of non-linearity and collinearity, we expected that some variance cannot be explained by the linear regression model. With a R^2 test statistic of 0.5554, and a high test RMSE of 18.8309, we can confirm our expectation. Hence, we decided to adopt regression tree models.

We tried out several regression tree models, namely the original decision tree regression model, Extreme Gradient Boosting (XGB) regression model, Light Gradient Boosting (LGB) regression model, and the random forest model. Among these different regressors, the XGBRegressor has outperformed other regressors with the lowest test RMSE of 15.972.

Model	Parameters	RMSE (Train)	RMSE (Test)
Linear Regression (Vanilla)	-	19.322	18.831
DecisionTreeRegressor (Vanilla)	-	-	25.416
XGBRegressor (GridSearch)	{'colsample_bytree': 0.7, 'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 1000}	4.834	15.972
LGBMRegressor (GridSearch)	{'colsample_bytree': 0.8, 'max_depth': 9, 'min_child_weight': 0.1, 'subsample': 0.6}	3.800	15.476
LGBMRegressor (Randomised Search)	{'colsample_bytree': 0.8335107260075492, 'max_depth': 11,	3.587	15.511

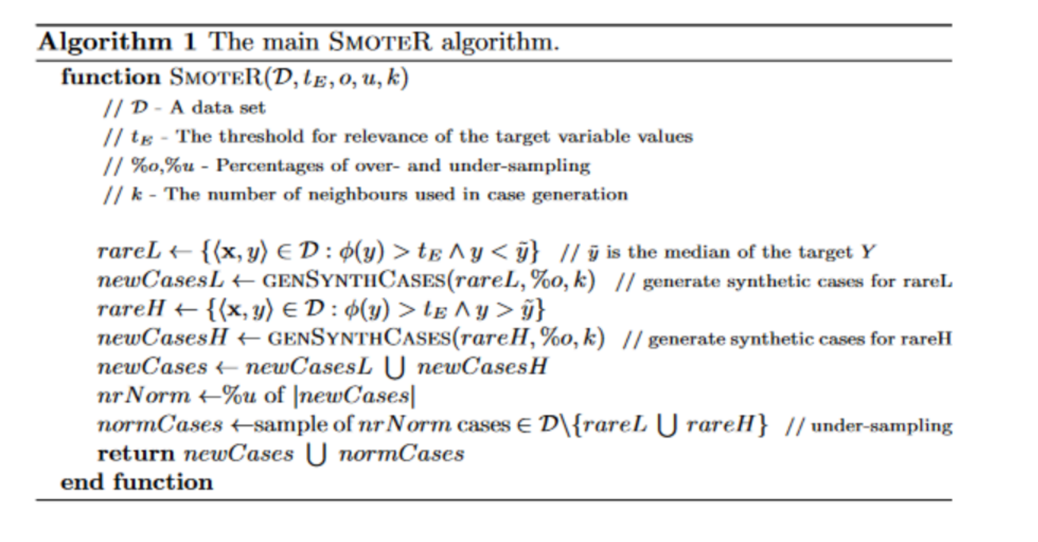
	'min_child_weight': 9.593483475672347, 'subsample': 0.8082607886651456}		
RandomForestRegressor (GridSearch)	{'bootstrap': True, 'max_depth': 10, 'max_features': 5, 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 1000}	13.873	15.814

Improvements

SmoteR

After further review of our data, we found that there was an imbalance of the distribution of the *population* feature. We decided to run the SmoteR, which is a variant of the Synthetic Minority Oversampling TEchnique (SMOTE) algorithm to address data imbalance.[6] SMOTE is used to address classification problems with imbalance class distribution by generating new samples of the minority class through interpolating several existing data points from the minority classes. SmoteR is used to address regression problems instead. In classification problems, the number of classes are distinct and limited, which makes it easy to differentiate the minority class from the majority. However, for regression, there is a potentially infinite number of values of the target variable. Hence, there is a need to use a relevance function and a user-specified threshold to sieve out the minority class.

This relevance function and user-specified threshold would result in the set D (see Figure 4.0). The algorithm would over-sample the observations in D and under-sample the remaining cases, resulting in a new training set with a more balanced distribution of the values.[7]



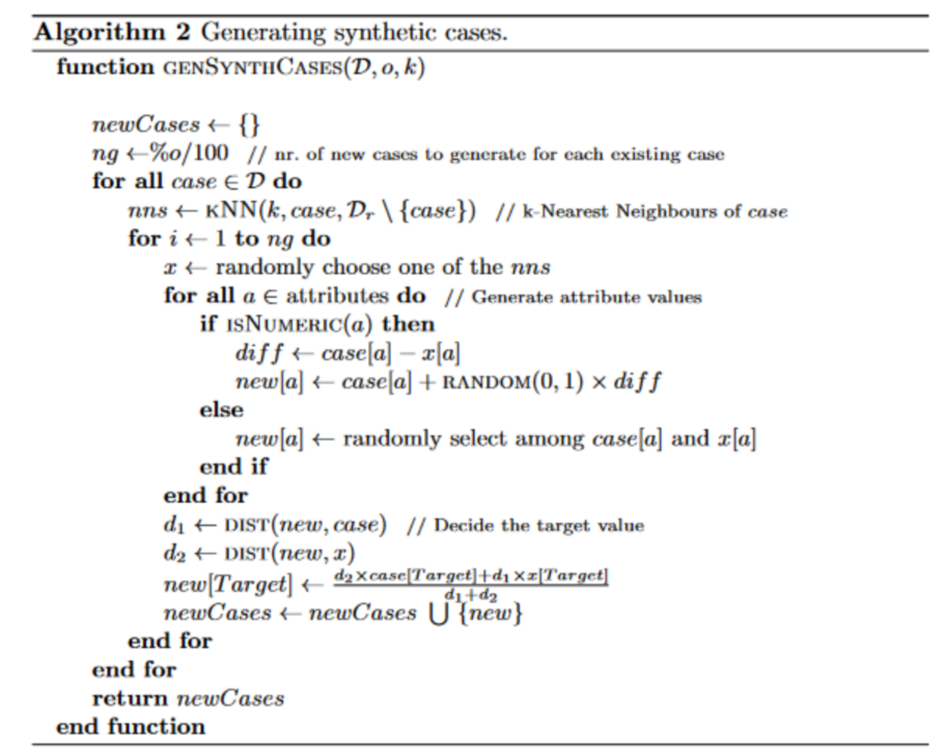
```
Algorithm 1 The main SmoteR algorithm.
  function SmoteR($\mathcal{D}, t_E, o, u, k$)
      // $\mathcal{D}$ - A data set
      // $t_E$ - The threshold for relevance of the target variable values
      // $\%o, \%u$ - Percentages of over- and under-sampling
      // $k$ - The number of neighbours used in case generation

      $rareL \leftarrow \{\langle \mathbf{x}, y \rangle \in \mathcal{D} : \phi(y) > t_E \land y < \tilde{y}\}$   // $\tilde{y}$ is the median of the target $Y$
      $newCasesL \leftarrow$ genSynthCases($rareL, \%o, k$)   // generate synthetic cases for rareL
      $rareH \leftarrow \{\langle \mathbf{x}, y \rangle \in \mathcal{D} : \phi(y) > t_E \land y > \tilde{y}\}$
      $newCasesH \leftarrow$ genSynthCases($rareH, \%o, k$)   // generate synthetic cases for rareH
      $newCases \leftarrow newCasesL \bigcup newCasesH$
      $nrNorm \leftarrow \%u$ of $|newCases|$
      $normCases \leftarrow$sample of $nrNorm$ cases $\in \mathcal{D} \backslash \{rareL \bigcup rareH\}$   // under-sampling
      return $newCases \bigcup normCases$
  end function
```

Figure 4.0

[6] Torgo, L., Ribeiro, R.P., Pfahringer, B. and Branco, P. (2013). "SMOTE for Regression". *ResearchGate*. 8-10. doi: 10.1007/978-3-642-40669-0_33.

[7] Kaggle code: "Addressing Extreme Rare Cases with SmoteR for Regression". https://www.kaggle.com/aleksandradeis/regression-addressing-extreme-rare-cases/comments



Algorithm 2 Generating synthetic cases.

```
function GENSYNTHCASES($\mathcal{D}, o, k$)

    $newCases \leftarrow \{\}$
    $ng \leftarrow \%o/100$  // nr. of new cases to generate for each existing case
    for all $case \in \mathcal{D}$ do
        $nns \leftarrow$ KNN($k, case, \mathcal{D}_r \setminus \{case\}$)  // k-Nearest Neighbours of case
        for $i \leftarrow 1$ to $ng$ do
            $x \leftarrow$ randomly choose one of the $nns$
            for all $a \in$ attributes do  // Generate attribute values
                if ISNUMERIC($a$) then
                    $diff \leftarrow case[a] - x[a]$
                    $new[a] \leftarrow case[a] +$ RANDOM$(0, 1) \times diff$
                else
                    $new[a] \leftarrow$ randomly select among $case[a]$ and $x[a]$
                end if
            end for
            $d_1 \leftarrow$ DIST($new, case$)  // Decide the target value
            $d_2 \leftarrow$ DIST($new, x$)
            $new[Target] \leftarrow \frac{d_2 \times case[Target] + d_1 \times x[Target]}{d_1 + d_2}$
            $newCases \leftarrow newCases \bigcup \{new\}$
        end for
    end for
    return $newCases$
end function
```

Figure 4.1

Pruning The Tree

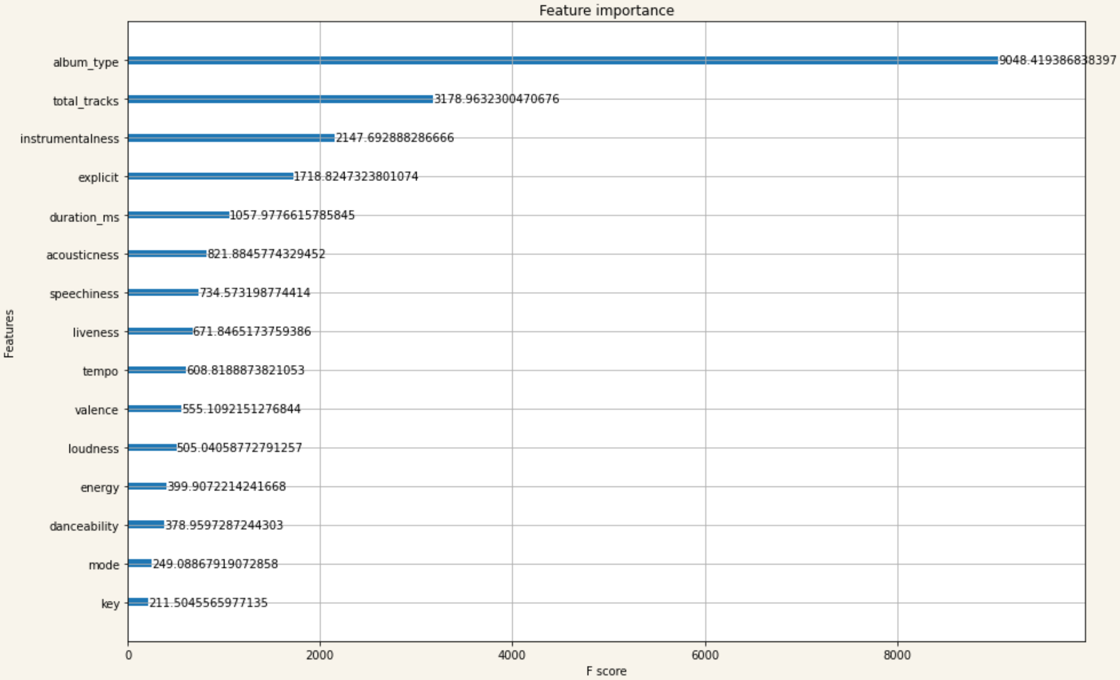


Figure 4.2

Looking at Figure 4.2, we then decided to prune our tree by removing the features that hold lesser importance. This decreases the likelihood of the tree overfitting, and allows lesser features to express the tree.

We repeated the process of obtaining the least important feature and removing it, followed by refitting and re-evaluating the model until we were left with two features. Figure 4.3 illustrates the change in the train and test RMSE score against the number of features. Figure 5.0 in the following segment illustrates the top 12 most important features.

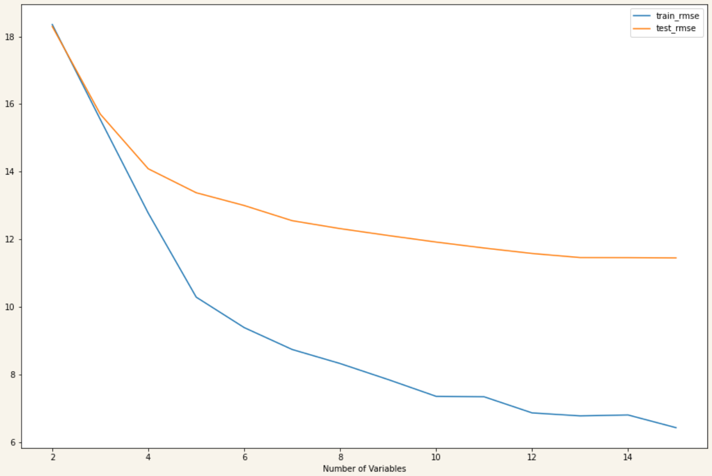


Figure 4.3

Final Model

Model	R^2		RMSE		MAE		MAPE	
	Train	Test	Train	Test	Train	Test	Train	Test
XGBRegressor	0.689	0.628	15.690	17.236	11.005	12.010	1.274	1.484
GridSearchCV + XGBRegressor	0.971	0.681	4.834	15.972	3.206	10.724	0.262	1.211
GridSearchCV + smoteR + XGBRegressor	0.923	0.753	6.422	11.449	3.920	7.249	0.316	0.649
Final Model	0.909	0.747	7.002	11.595	4.272	7.334	0.343	0.652

Explanation

In decision tree learning, information gain is the improvement in accuracy brought by a feature to the branches it is on. Using information gain as the importance matrix, we were able to identify which features would have the greatest gain and therefore affect popularity the most. In Figure 5.0, the feature importance is plotted in descending order. Features such as *Album type*, *Total tracks*, *Explicit* and *Instrumentalness* have high importance.

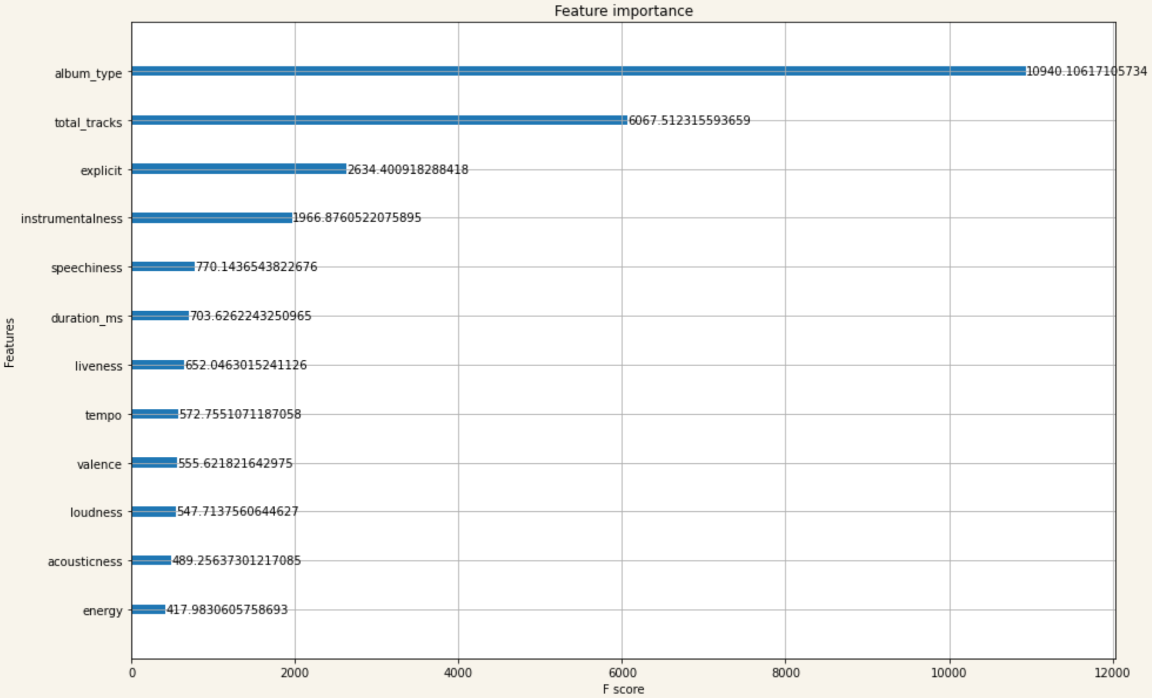


Figure 5.0

Album Type

Album type is a categorical variable that contains three categories, namely album, compilation and single. From Figure 5.1, a clear distinction in popularity with different album types can be observed.

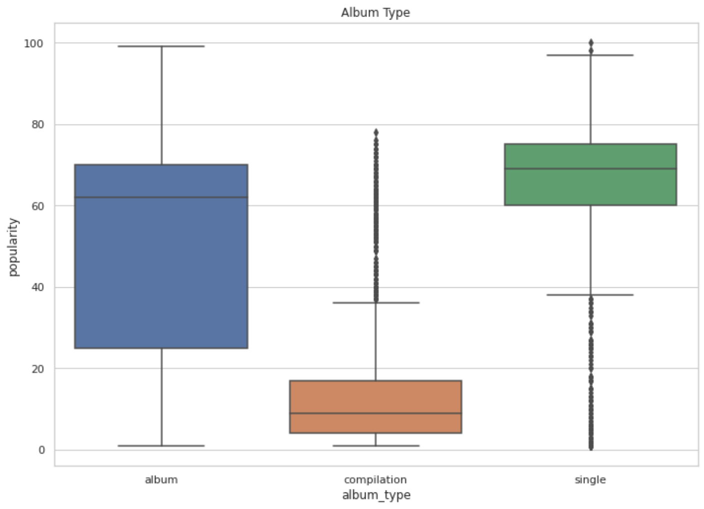


Figure 5.1

A compilation generally has a lower popularity score compared to an album or a single, presumably because they consist of songs previously released, or even unreleased, by one or even multiple artists. We can draw the parallel of a compilation album to a "best of" album, where an artist may choose to release a compilation of their most popular songs. Lacking the element of "freshness" may have contributed to compilation songs having a lower popularity score. In addition, a compilation from multiple artists of similar genre may have diluted its popularity score.

Since album type is a categorical variable with only three possible values, this feature would only be used once or twice in each tree relative to other continuous variables, which might appear much more often on different levels of the trees. As such, when we use the "gain" as the importance metric, it would result in album type having a high importance.

Total Tracks

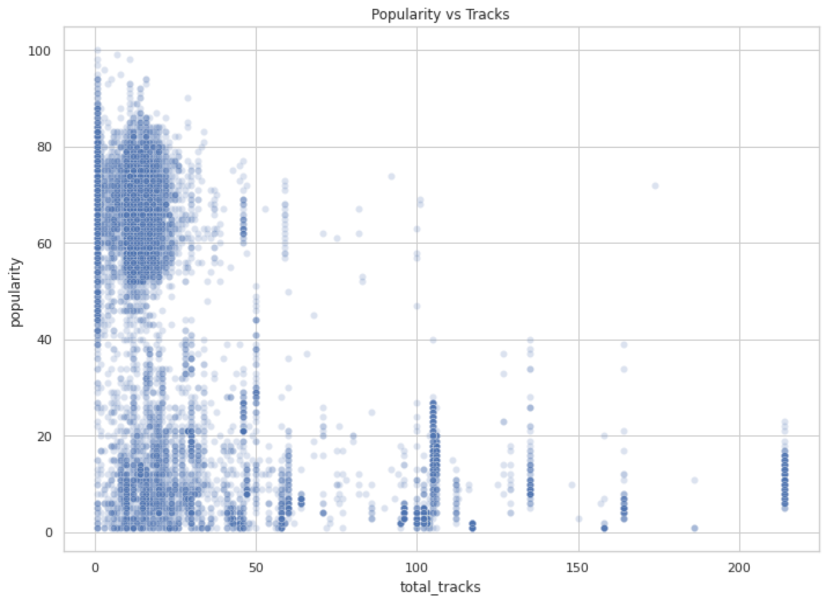


Figure 5.2

Referring to the scatter plot in Figure 5.2, there is no clear trend on how the total number of tracks will impact the popularity of a song, especially with albums having a low number of tracks. However, it is clear that the highest possible popularity generally decreases as the total number of tracks in the album increases. Hence, to achieve a higher popularity score, we suggest that artists do not include too many tracks in a single album, as the old saying goes, it would be more strategic for an artist to value quality over quantity.

Explicit

The explicit feature tells us whether a song is explicit or not. Explicit songs are songs that contain potentially offensive material, for example, foul language, sexual content, or depiction of violence. From Figure 5.3, explicit songs on average have a higher popularity score than non-explicit songs. Even though explicit songs were the minority from our dataset, this finding can be supported by additional data scraped from the Weekly Billboard Top 100 songs (data was scraped on the 11th September 2021), where in Figure 5.4, the mean popularity score of explicit songs are higher than non-explicit songs even if the proportion differs. Since most radios and family-friendly streaming services have censored the explicit words in a song, it would make sense that explicit songs have a higher mean popularity as it offers a new take on a song. In other words, explicit songs offer a wide mirage of suggestive imagery, and appeals to the younger generation.

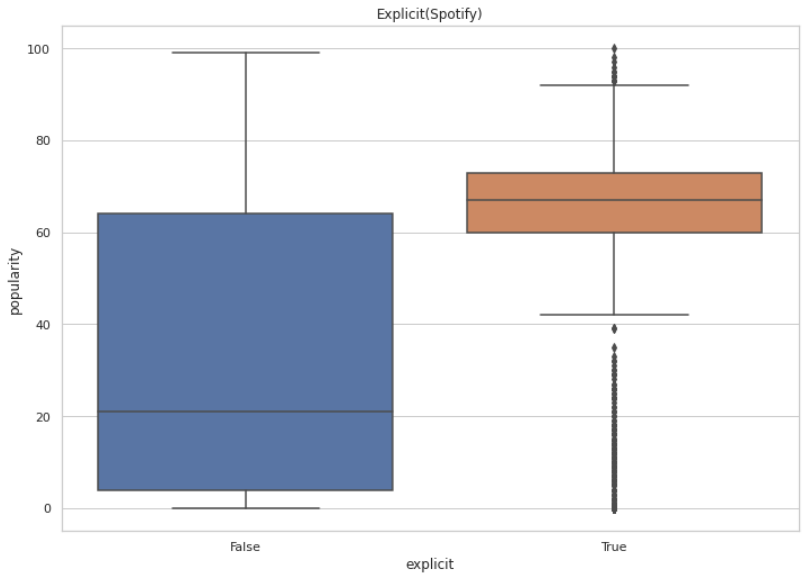


Figure 5.3

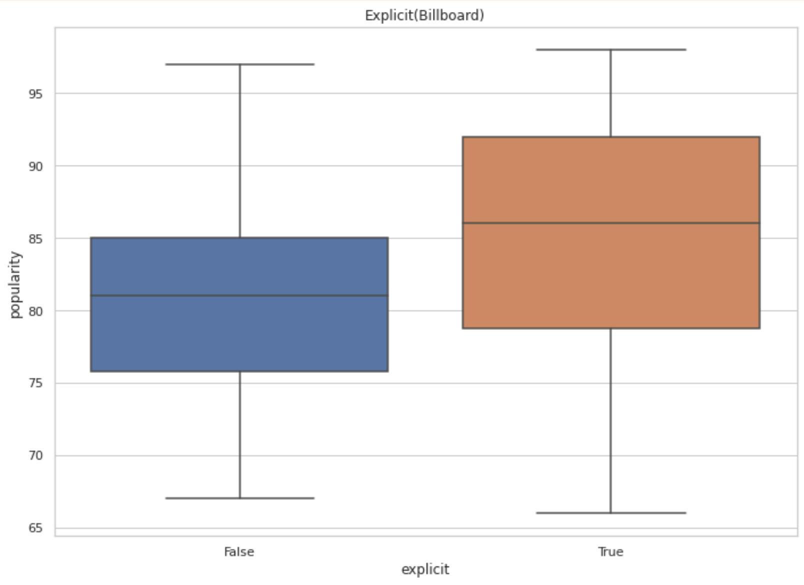


Figure 5.4

Instrumentalness

Instrumentalness is a measure of how vocal a song is. The only trend visible is that songs that have high instrumentalness (0.8 to 1) have a higher likelihood of having low popularity score. This implies that songs with a high vocal content are preferred.

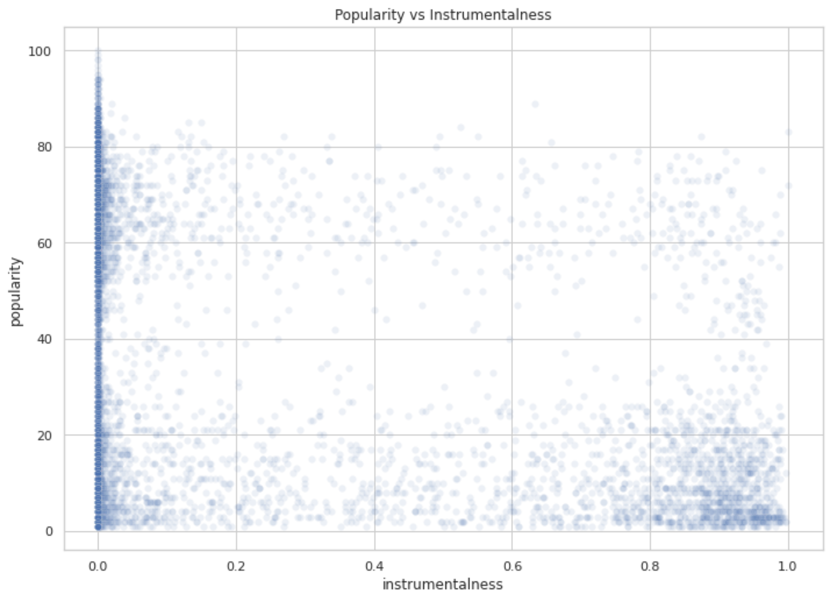


Figure 5.5

Limitations

Spotify only allowed us to scrape 1000 songs per year, and the data gathered was not structured in an orderly manner, thus we had to clean and manipulate the data into a tidy DataFrame.

In addition, we initially thought that a popularity score of 0 indicates that that particular song is not popular at all, hence, we included those songs in hopes that they would provide insight on the attributes of unsuccessful songs. We were mistaken. Echo Nest, now owned by Spotify, is a music intelligence and data platform for developers and media companies. Echo Nest is also responsible for the computations of a song's attributes, but unfortunately, this formulation is not made available to the general public. In the Echo Nest API, a popularity score of 0 is meant to indicate that that particular song has no popularity score calculated for it.[8]

With regards to the skewness and imbalance distribution of the popularity response variable, we tried to reduce the skewness using SmoteR, but nevertheless the skewness is still present. This greatly affected the performance of the model.

[8] Xue, A. and Dupoux, N. "Predicting A Song's Commercial Success Based on Lyrics and Other Metrics".

Future Works

Due to time constraints, we did not manage to try other machine learning models such as Natural Language Processing (NLP) to examine the relationship between song lyrics and their popularity scores.

Secondly, it would be interesting to analyse other similar datasets, like the Billboard Top 100 Songs dataset, to further validate our insights. Datasets that include revenue gain and money invested in developing songs could also be used alongside with our models to make meaningful predictions, such as the optimal returns from investing in a song.

Lastly, it would be desirable to obtain the popularity scores of various artists, perhaps based on how many social media followers they have as an attribute to predict the popularity of a song, as we suspect that there are still external features that may have influenced the popularity of a song.

Conclusion

We have utilised several models to find out which musical attribute(s) play a large part in influencing the popularity of a song, and have attempted to discuss and explain our findings using visuals.

Through our analysis, we conclude that an album or a single is more desirable as compared to a compilation, and an album should not contain too many tracks within. We have also discussed why explicit songs generally have higher popularity scores, but ultimately it is up to an artist's discretion and personal values on whether they are willing to venture into sensitive song writing.

Lastly, we recommend that artists explore reducing the instrumental components in their songs. In other words, songs with high vocal content are preferred, as lyrics, too, are a form of a window to an artist's thoughts.