Song Popularity Forecast with Spotify: A Machine Learning Approach

S2568786 **S2604021** **S2500630**

Abstract

This report focuses on predicting the popularity of a song based on historical data. To make the most of versatile features and achieve the best prediction, we execute EDA to explore the potential characteristics and maximize data preprocessing for the following prediction. Visualization is implemented for a better understanding of the dataset. Several models are chosen to compare with each other, the best performance is achieved using Word2vec with logistic regression, SVM and Random Forest (accuracy of 76.2, 74.2 and 73.6 respectively). We further evaluate the logistic regression model on the test set (71.9% accuracy) and determine its ability by performance indicators such as Cohen's Kappa and MCC.

1 Introduction

Spotify is one of the most popular music streaming services. In the second quarter of 2023, the service had 551 million active users [2] listening to a broad spectrum of songs. With such a large user base, the listening habits on Spotify can be viewed as representational for the wider population to some extent. There have been various studies on the prediction of the popularity of songs on Spotify, like [1][6]. These are mostly based on the Top 100 or Top 200 songs of different regions for a given time. Spotify allows to retrieval of metadata for each song with the *Spotify Web API*. This includes several musical features, namely *danceability*, *energy*, *loudness*, *acousticness*, *speechiness*, *instrumentalness* and *valence*.

Four word embedding methods and four models were used in this report, it achieved the best result using the Logistic Regression model and word2vec algorithm.

2 Exploratory Data Analysis

Nearly 0.5 million songs from Spotify's "Top 200" playlists released between January 1, 2017, and May 31, 2023, make up the dataset being analysed in this section. This dataset contains typical information such as song titles, artist details and some musical features, for example, danceability, energy, loudness, and speechiness. We processed the data in such a way that a song is represented in one row and columns representing information about it. In this part, we focused on analyzing the impact of these characteristics and artists on the ranking of songs.

2.1 Data Loading, Overview and Cleaning

The first step was to load the data into a pandas DataFrame and get a brief summary that helped us comprehend the dataset's structure. Then to avoid affecting the data analysis, we checked for null values.

2.2 Distribution of Numerical Features

Next, we examined what values of those features might affect the rank. Figure 1 visualizes the distribution of numerical features of the songs in the top playlist. This indicates that If the features of a song are similar to the values of the high distribution on the graph then it is likely to be a hit. For example, a song with values for Danceability and Energy around 0.7 and values for Speechiness, Acousticness and Instrumentalness are about 0 might become popular based on the historical data.

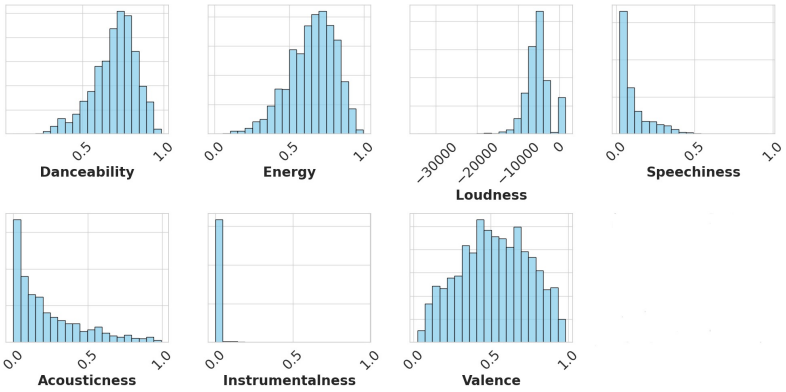
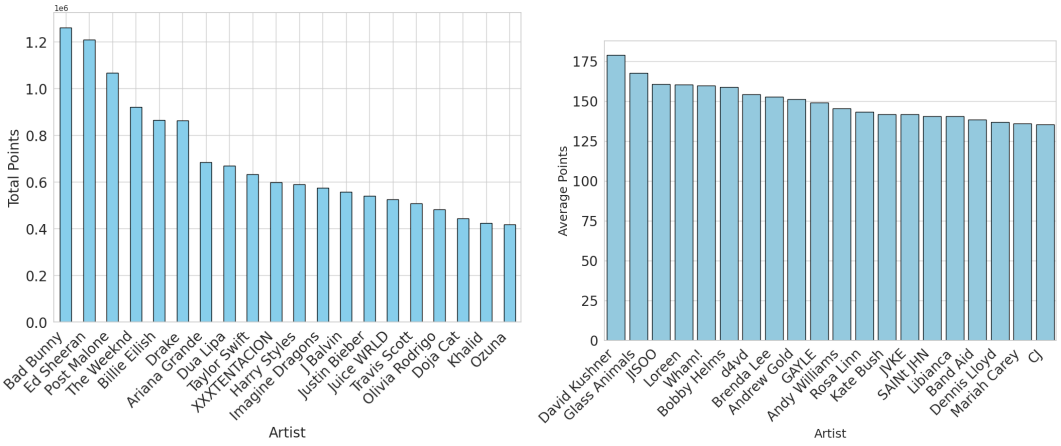


Figure 1: Distribution of Numerical Features

2.3 Average and total points per artist

For the top 20 ranks, we grouped them by artist, summed up the points and plotted the head 20 artists as shown in Figure 2(a). The total score of artists with hit songs between 2017 and 2022 can show which artists are more likely to be hit when they release songs in the future. However, this method can be problematic, an artist may have released several songs, and result in the artist's total score is very high. We then calculated the average number of points each artist would get for a song based on the total number of points they received and the number of songs they had. By comparing the two graphs, we can see that a person with a high total point does not mean they have a high average point in this dataset. Hence, it is possible to see which artists' releases are likely to be a hit via average points. In the following classification part, we used these average points of the singers to group them.



(a) Total point per artist

(b) Average point per artist

Figure 2: Distribution of points of Top 20 artists

3 Preprocessing & Dimensionality Reduction

In this section, we will detail our approaches to explore and prepare the data for future prediction trying to forecast the popularity of songs in the future.

Dimensionality reduction and feature extraction techniques were implemented during the preprocessing phase. This strategy served multiple purposes, including improving the interpretability of machine learning models developed in subsequent stages, reducing training times, enhancing generalization by mitigating over-fitting and, most importantly, learning lower-dimensional manifolds in which the hiding characteristics reside, and facilitating data visualization for data exploration.

The given Spotify dataset has several versatile features such as time series, numeric data including some musical properties, and categorical data, like artists, nationality and continents. Due to the variety of Spotify dataset, we need to preprocess those features separately and carefully consider any potential issues hiding in the data before pre-processing. The preprocessing methods applied and corresponding features will be described in section 3.1.

3.1 Methods

Grouping Label Encoding was implemented on the feature "Artists". Since there are more than 2000 artists in the dataset, we believe the correlation between the average point of an artist and the popularity of a song is strongly relevant, see appendix 9.1, 9.4. We need to find a method to properly process this feature to avoid the huge matrix calculation and keep its sequentiality. We calculated the average point of each artist from the dataset and separated them into 7 groups by average point. After this, we could finally label-encode and numericize this feature and keep its ordinal characteristic. The reduction also helps the following predicting models to converge and prevent the Curse of dimensionality made by other preprocessing methods.

Principal Component Analysis (PCA) was used as a typical linear dimensionality reduction technique on the dataset. Before applying PCA, we normalized the feature "Loudness" to $[0, 1]$ for consistency with other musical properties, and to prevent it from dominating model decision. The cumulative explained variance thresholds of 95% could be reached by having 14 principal components.

One-hot Encoding was one straightforward but powerful method to deal with categorical features, such as "Nationality" and "Continent" in our case. But when the number of categories was too large, it resulted in Curse of dimensionality and made models hard to converge, see appendix 9.2, 9.3. We carefully analysed the mentioned two features and decided to implement one-hot encoding skills on these two since they would not lead to large dimensionality.

TF-IDF (term frequency-inverse document frequency) is a statistical measure that evaluates how relevant a word is to the whole dataset. This is done by multiplying two metrics: how many times a word appears in a dataset, and the inverse document frequency of the word across the dataset. We can transform words into numbers, known as text vectorization. To put it in more formal mathematical terms, the TF-IDF score for the word t in the dataset d f is calculated as follows [5]:

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t,d)$$

$$tf(t,d) = log(1 + freq(t,d)), \quad idf(t,d) = log(\frac{N}{Count_{t \in d}})$$

N is the total number of songs, and $freq(t, d)$ is the frequency of the term in data. $Count_{t \in d}$ is the number of songs in which that term appears in dataset *d*.

TF-IDF enabled us to associate each word in this dataset with a number that represents how relevant each word is. Then songs with relevant terms, such as relevant nationality, would have similar vectors, which was what we were looking for to keep the relation between categories.

Word2Vec is an algorithm popular for learning word embedding and feature extracting by using a shallow neural network model to learn word associations. It constructs an embedding and, here, we implemented it based on *Common Bag Of Words(CBOW)* which took the context of each term as the input and tried to predict the term corresponding to other data-points [4]. We imported Gensim package to achieve the preprocessing for *continent* and *nationality* features by setting the vector size as 100 according to EDA we had done in section 2. Then we obtained the representing vectors for non-numeric data so far.

3.2 Visualization

In elucidating the impacts of diverse preprocessing and dimensionality reduction techniques, we employed the visualization algorithm, Uniform Manifold Approximation and Projection (UMAP)[3], to map each data representation onto a two-dimensional space, which was much easier to recognize the relation between data points. The ensuing projections were subsequently illustrated as scatter plots, as depicted in Figure 3.

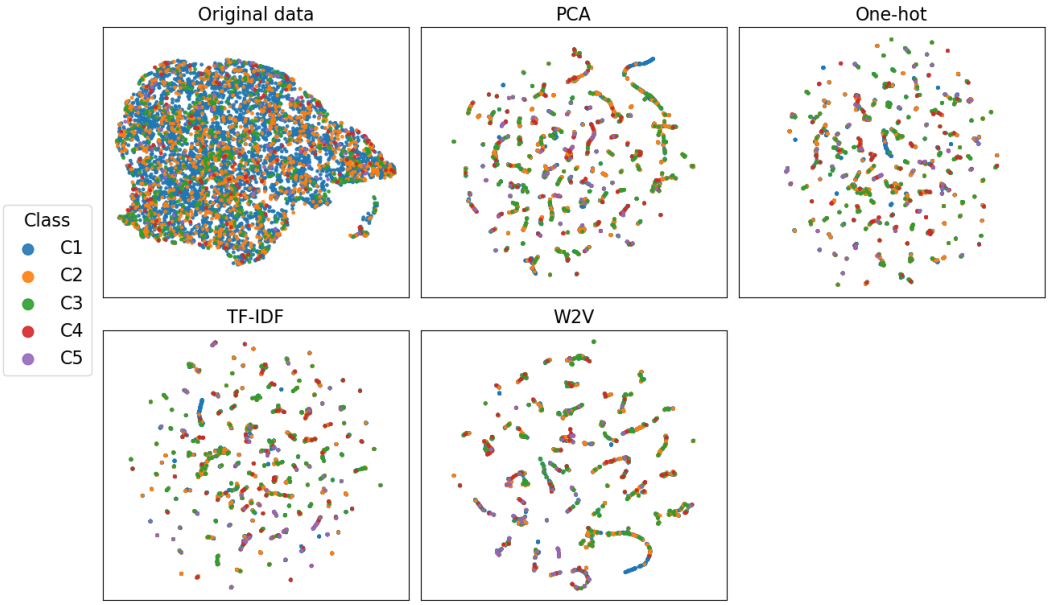


Figure 3: Visualization of Dimensionality Reduction.

The visualization plots have depicted several features for aforementioned preprocessing methods and following predicting models. To kick off, we can see that data points in **original dataset** are highly sparse, which means directly using the raw data possibly not be a decent choice. It is worth noting that it reveals the complexity and diversity of the top 200 dataset. Additionally, some outliers could be found even on the manifolds, so the need for data cleaning and preprocessing is definitely indispensable for this research. For **PCA** method, although the clusters are not clearly enough to be recognized, which is visible since the complexity of dataset, it depicts the continuity between popularity classes and the characteristics of them, such as spacing distribution and the data points density, e.g., class 5(the most popular class) mostly locates in the center. Compared to other preprocess methods, the data points are relatively intensive and more connective to other data points. According to **One-hot encoding**, we can find that the distribution of data points is more sparse but some continuity can be observed in the center. Interestingly, some clusters, such as C1, C4 and C5, are gathered pretty well. The visualization of **IF-IDF** shows the most sparse spacing distribution among all methods. Classes are far away from each other in outer circle, but even closer and more connective in the center. The word frequency-based approach leads to such an extreme case because our dataset is large but with few values of specific features, e.g. "Nationality" or "Continent". *Word2vec* is a word vectorization approach that can relate the similarity between different words. We can find that distinct classes are sequentially connected, and the boundaries between classes on some hyper-shapes. Even though the repeatability is higher than other methods, its dimensionality reduction and classification ability make it still a worth-trying approach.

4 Methodology

The objective is to predict the popularity of songs by categorizing the Spotify dataset into five classes based on Points (or Rank), treating it as a multi-classification problem. This approach allows us to assess a song's popularity across five tiers, streamlining the evaluation task.

4.1 Methods

In this section, we describe our strategy to train the best-performing classifiers for predicting the popularity of songs. To estimate the generalization of performance and prevent overfitting from happening. We implemented k-fold cross-validation for each classifier with k=5 and kept the ratio of the training set to the validation set as 80:20. To best optimize the performance of models, we tried potentially critical parameters in each classifier up to 6 combinations to obtain the most reasonable parameters. Finally, we compared them with each other and evaluated the performance based on the metrics described in section 4.3. The performance during the initial cross-validation was used to estimate the overall generalisation performance.

4.2 Classification Algorithms

We have applied different classification algorithms for reasons, and to help us evaluate our models and compare their pros and cons, and most importantly, to achieve our forecasting target:

1. **Multi-class Logistic Regression**, which is a generalised linear model of binary-logistic regression to handle multiple classes. It directly computes the probability distribution across all classes and predict a possible class by the highest probability and maximum likelihood estimation. Softmax function will be applied to achieve the generalization of probability of all data. However, one of its omnipresent disadvantages is linearity. Logistic Regression could only decide linear boundary between classes, and other characteristics between data points might be ignored.

2. **Support Vector Machine** excels in high-dimensional spaces, which seems an appropriate model in this case. It utilizes a subset of training points (support vectors) for decision-making, ensuring memory efficiency. SVM is versatile, allowing us to use different Kernel functions to do multi-classification. However, drawbacks include potential overfitting with a high feature-to-sample ratio, necessitating careful selection of Kernel functions and regularization terms. Fortunately, although we have up to 200 features sometimes, our samples is much larger than the number of features.

3. **Random Forest** is an ensemble machine learning algorithm that combines multiple decision trees. Each tree is trained on a random subset of the data and considers a random subset of features at each split. It uses a majority vote to combine the predictions for classification problems. Random forests are robust and effectively manage overfitting.

4. **KNN** classifies based on the class of several nearest neighbours and should be very sensitive to the representation of the data and thus might highlight effective dimensionality reduction methods.

4.3 Metrics

According to the Exploratory Data Analysis, it has shown that the class distribution of the top-200 dataset is partially imbalanced, thus using accuracy as an indicator could be straightforward since it is a classification task, but not the most equitable approach.

To improve the efficiency, we first evaluated our four algorithms by their accuracy and chose our best-predicting model to execute the following evaluation. In other words, we implemented reasonable metrics as indicators to evaluate the performance of our final decision model to avoid Accuracy Paradox or relevant imbalanced prediction problems.

Although, for classification models, simply predicting the label that has the highest predicted probability is an appropriate choice. However, this would limit the application of developed tools to the inclusion of further data that may not be balanced. To mitigate the above problems, we evaluated the model by methods which work for imbalanced data such as **Cohen's Kappa** κ and **Matthews Correlation Coefficient** for Multi-Class Classification problem. Lastly, using **Confusion Matrix** to detail the predicting performance.

5 Results

5.1 Algorithms Statistics

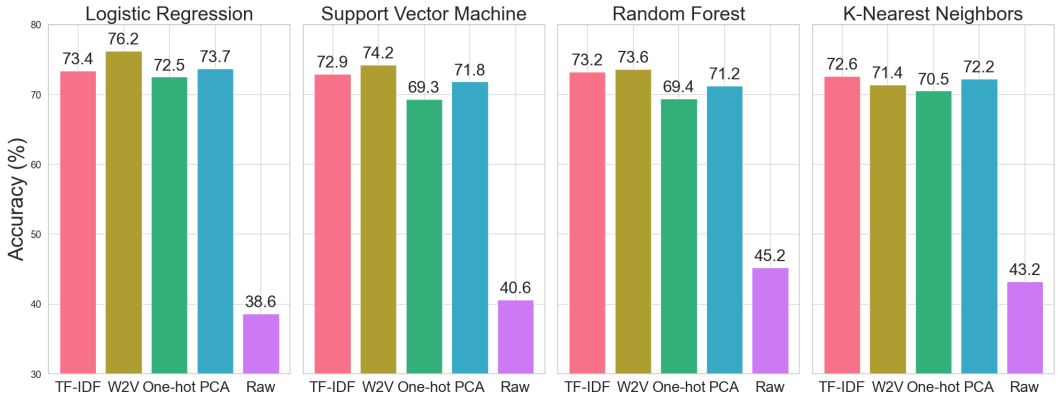


Figure 4: Accuracy of Models on different datasets

5.2 Model Decision and Evaluation

Based on the results, we decided to choose **Logistic Regression** as our model to predict songs' popularity, since it has slightly higher average performance and stability. We then tested its performance on test data. Several indicators had been implemented to illustrate the ability of this model.

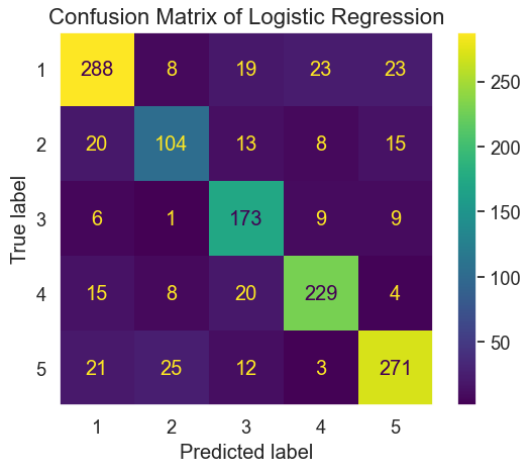


Figure 5: Confusion Matrix

Indicators	Result
Test	0.719
Kappa	0.759
MCC	0.741

Class	MCC
1	0.741
2	0.639
3	0.760
4	0.793
5	0.773

Figure 6: Test score, κ and MCC

6 Conclusions

To conquer the song popularity prediction, we viewed it as a multi-class classification task. 16 combinations of four models and four preprocessing methods led us to achieve reasonable prediction accuracy and performance on training data. We further chose Logistic Regression as the final model and achieved a prediction success rate of 72%, successfully identifying the likely category of popularity among the five groups. We can observe prediction accuracy from confusion matrix and relevant indicators, e.g. κ and MCC, shown as Figure 5 and Figure 6. Confusion matrix details the prediction result, additionally, MCC and κ illustrate the performance on a more objective standpoint. Since the Spotify dataset is huge and complicated, MCC indicator shows class 2 only reached 0.64. Overall, the result of this research is quite acceptable and we both agree we have tried our best on this task!

References

[1] Joshua S. Gulmatico, Julie Ann B. Susa, Mon Arjay F. Malbog, Aimee Acoba, Marte D. Nipas, and Jennalyn N. Mindoro. Spotipred: A machine learning approach prediction of spotify music popularity by audio features. In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, pages 1–5, 2022.

[2] Marie Charlotte Götting. Number of spotify monthly active users (maus) worldwide from 1st quarter 2015 to 2nd quarter 2023. `https://www.statista.com/statistics/367739/spotify-global-mau/`, 2023. [Online; accessed 22-Nov-2023].

[3] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[5] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*. Springer US, Boston, MA, 2010.

[6] Mariangela Sciandra and Irene Carola Spera. A model-based approach to spotify data analysis: a beta glmm. *Journal of Applied Statistics*, 49(1):214–229, 2022.

7 Statement of contribution

S2568786:

1. Abstract Revision
2. Section 3
3. Section 4 + (Models: LR + SVM)
4. Section 5
5. Section 6

S2500630:

1. Abstract
2. Section 2
3. Models in Section 4: Random forest, KNN
4. Grammar and Spelling Check
5. Plots Revised

S2604021:

1. Introduction
2. Grammar and Spelling Check

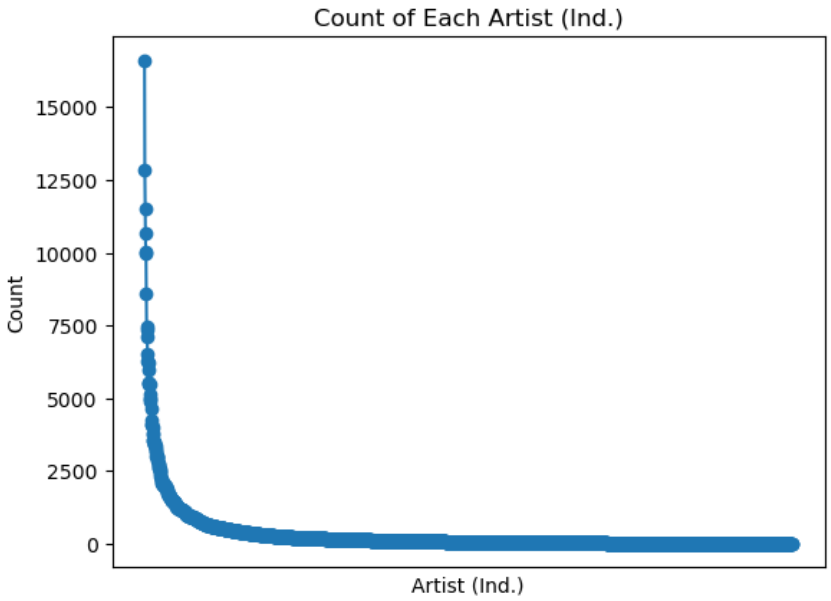
8 Generative AI statement

Some tasks supported by Gen-AI:

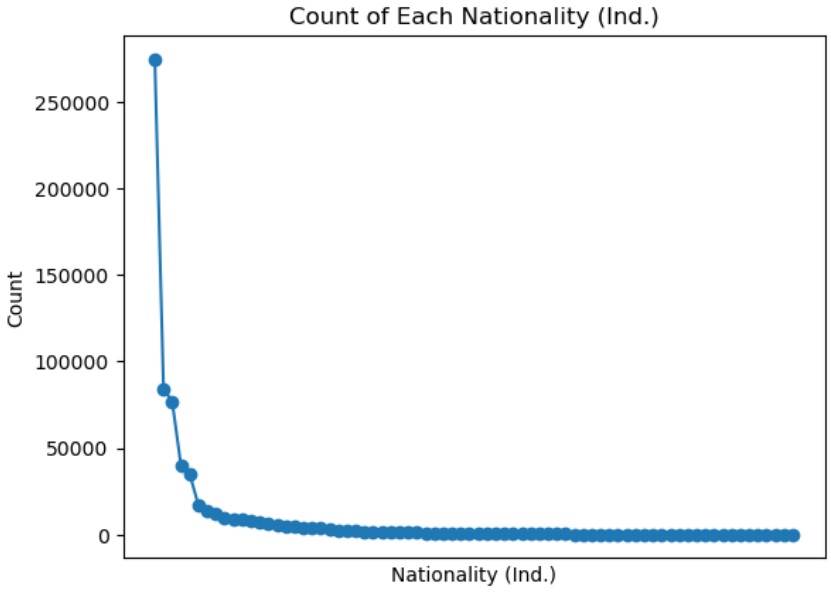
1. Support to make plots.
2. Check spelling and grammar.
3. Paragraphs Partially Revision.
4. Partially Code Revision

9 Appendices

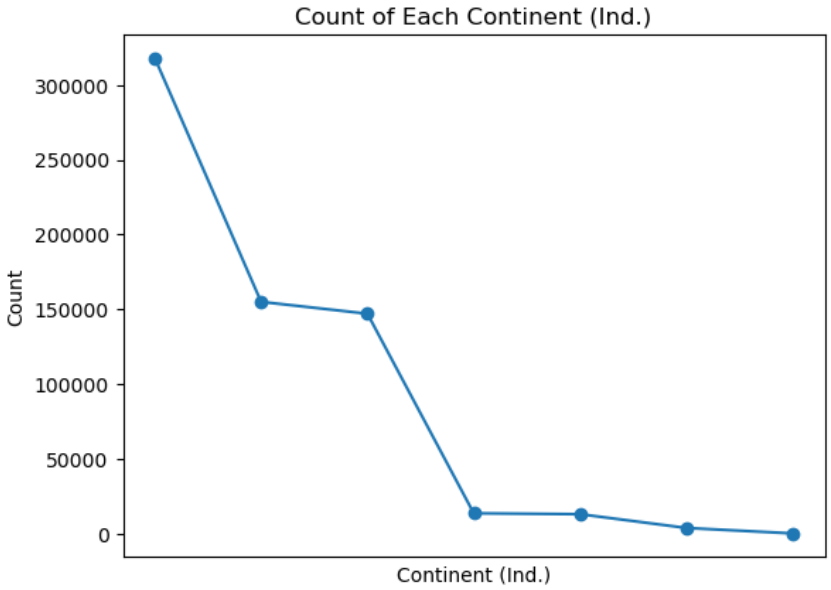
9.1 Frequency of Artist



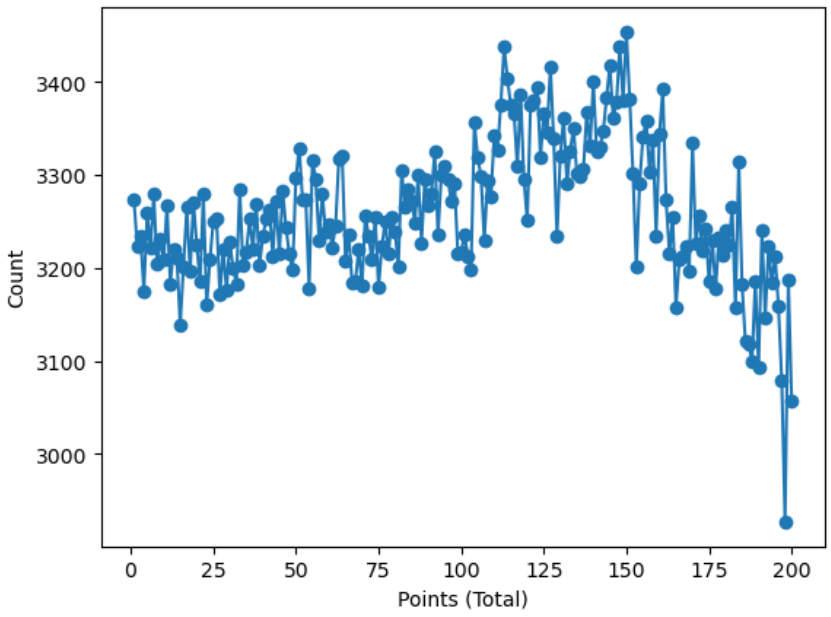
9.2 Frequency of Nationality



9.3 Frequency of Continent



9.4 Distribution of Rank



9.5 PCA 3D Visualization