Spotify Analysis

Abstract

This project utilizes Spotify's global "Top 200" playlists from 2017 to mid-2023, covering approximately 7,780 songs. The primary aim is to identify which songs and artists may gain popularity in 2024 based on historical data trends. We analyzed early future trends across musical features, followed by dimensionality reduction using Principal Component Analysis (PCA) and clustering with a Gaussian Mixture Model (GMM). After assigning each song to the most relevant cluster representing a genre, we performed a Random Forest analysis to determine the characteristics of songs likely to become popular. Our results show that the most important factor influencing a song's popularity is likely the artist. Therefore, we applied several time-series prediction methods, including ARIMA, Prophet, Extreme Gradient Boosting, and Light Gradient-Boosting Machine, to predict which artists are most likely to appear in the Top 20 rankings of the chart in 2024.

1 Introduction

The music industry is constantly changing, users preferences, and song popularity continuously shift in such a random manner. In such a dynamic landscape, being able to predict which songs, artists, or genre might become popular offers priceless insights for artists, producers, and streaming platforms. This project explores a dataset of musical features and popularity metrics to better understand what makes a song deserve the top spots in the chart. By applying techniques such as dimensionality reduction, clustering to identify patterns, and classification to highlight key factors of popular tracks, we aim to uncover the defining elements of hit songs. Additionally, by using time-series prediction model, we project the trend of these important features in the future to offer insights for the industry.

2 Dataset Exploration

2.1 Dataset Description

The raw dataset consisted of 651,936 rows and 20 columns, including the information of song titles, artists, ranks in the chart, points (invert of the daily rank), and musical features (e.g. Loudness, Energy, etc.). Moreover, there are 7780 unique songs and 2925 artists contributing for the whole dataset.

2.2 Initial Observations

In this project, we analyze two forms of training data, one is an aggregated training data to run non-time-series analysis such as clustering and classifying, and the other one is a time-series data to run prediction model analysis. Below is data preparation steps that we conduct:

- **Lower-casing the title and artist names:** During the data exploration process, we find that there are several songs with different letter-casing styles which might cause problem in our training process (e.g. "Boy's a Liar pt. 2" and "Boy's a liar pt. 2")

- **Handling inconsistent musical features on the same song:** The values of song's musical features change on different dates, for example *Thunder*, a song by *Imagine Dragons*. However, we assume that it is an error and we aggregate the musical features using the average value as an alternative.
- **Standardize the musical features value:** To handle the extreme value differences in the musical features (especially Loudness), we apply Scikit-Learn StandardScaler to ensure consistent scaling.

After we clean the data, we visualize the trend of average value of each musical feature over the years to understand more about the dataset. From Figure 1, we could see that there is a seasonality of musical features on the chart. This might be intuitively correct because during certain time people tend to repeat same kind of songs (e.g., Mariah Carey's song during christmas [6]). To understand more about these features, we do a correlation matrix analysis.

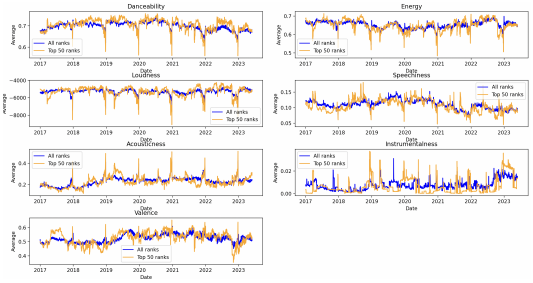


Figure 1: Trend Analysis of Each Musical Feature.

Figure 2 shows us the relation between musical features ranging from -1 to 1, negative value means negative correlation and vice versa [5]. For example, negative correlation between Acousticness and Energy in the correlation matrix reflects a common characteristic in music: acoustic songs tend to be less energetic. Therefore, we decide to do a classification analysis to find out what kind of song is more likely to be popular and identify which factor contributes the most to the popularity. Additionally, we plan to leverage the time-series nature of the dataset by doing a time-series prediction.

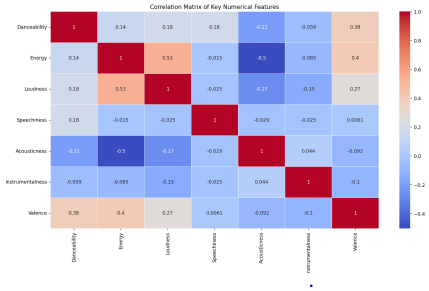


Figure 2: Correlation Matrix of each Song Feature.

3 Dimensionality Reduction and Clustering

3.1 PCA

Since music data contained hundreds of thousands of notes, we started the dimensionality reduction technique over all dataset to analyze and classify songs more effectively. The use of high-dimensional data is computationally inefficient and a well-known phenomenon of overfitting. We used PCA to map the song features into a low-dimensional space and it allows us to extract meaningful variations in the dataset while enabling efficient clustering.

3.2 Clustering

We utilized clustering algorithms based upon features after dimensionality reduction of the data. Clustering was done to check for any natural groups which could lead us to the prediction of song popularity [4]. We try some clustering methods and checked on silhouette score, a metric to assess how close each sample in one cluster to the samples of the neighboring clusters. Higher silhouette scores indicate better-defined clusters.The score ranges from -1 to 1, where "+1" value indicates that the point is well-matched to its cluster, "0" value indicates that the point lies on the boundary between clusters, and "-1" value indicates that the point may have been misclassified.

Following are the clustering methods and their respective silhouette scores on 20% of the dataset: **K-Means** Clustering had a low silhouette score, 0.1965, which indicates poor clustering as it could not form any decent clusters-the data may not provide spherical clusters. By modeling clusters as ellipses rather than as spheres, **GMM** Clustering as shown in Figure 3 is more flexible with a silhouette score of 0.3315. This method fit the best for this dataset compared with the other two methods. We use this method for further analysis and prediction, to cluster songs into groups based on the song audio features. A computationally efficient variant of K-Means, **Mini-Batch K-Means** Clustering performed better than the conventional K-Means with a silhouette score of 0.3114, but could not outperform GMM.

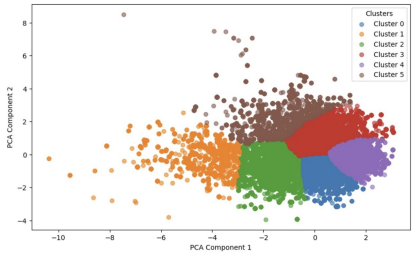


Figure 3: Gaussian Mixture Model (GMM)

4 Feature Engineering

Feature engineering plays an important role in our project, as it helps to increase the predictive power of the dataset by adding domain specific knowledge and also aligning features with models. In this section, we are describing the steps that we followed to extract, optimal narrow down and also feature engineering based on dataset.

The most important feature in our analysis was the construction of the target label `Is_Popular`. This label indicated whether the songs were classified as "popular" or "not popular". This feature added by summing the total points that they gained across playlists. Based on their cumulative points, songs were determined to be "popular" or "not popular", where any song in the top 15 % of the dataset was labeled as 'popular'. These threshold scores were selected to combine the advantages of a very broad popular label while keeping an effective separation between high and average-valued songs. `Is_Popular` thus became our target variable for predictive modeling, allowing us to uncover the elements of popular music.To measure the influence of popular artists on a song, we added another feature called `Popular_Artist` .This is a binary feature, which checks if the total artist points are sufficient to place them in the top 15%, if so they are considered popular artists. Collaborations are a powerful way to merge fan bases and boost the potential traction of one song, so we decided to introduce the feature `Is_Collab`.

Our other goal was to forecast for the top 20 artists of 2024 using time series and we tailored a dataset that would be suited to this purpose. We train the model using Spotify data from January 2017 to December 2022 and test it on data from 2023. We prepared the dataset by aggregating all rows of the same artist on the same day and computed for the mean of all feature values and the sum of the target variable, `points`. Artists who appeared only in 2023 were removed from the dataset to ensure consistent temporal representation. Additionally, we removed artists with fewer than 50 instances before 2023, as such cases lacked sufficient data for reliable forecasting. Lengths of sequences were set to 12 and 24 to train our models, while the threshold was set to 50 to ensure at least half of the

time-series sequence contained valid data points so that higher accuracy could be achieved by a model. This approach also accounted for cases of artists who had very low presence during 2022 but received a huge increase during the year 2023 and thereby inflated the model's predictions to very high aggregated values, such as 60,000 points even when they were lagging behind some of the top artist from 2023. We believe that if the artists were only present in 2023 or present for only less for than 50 days could be a top 20 artist of 2024. So dropping them would not affect the goal of our analysis.

5 Predicting Popular Songs

5.1 Classification Algorithms

The goal of this task is to classify songs into popular and not popular, using different machine learning algorithms and features. `Is_Popular`, the variable that tells us if the song is in Top 15% of songs sorted according to their playlist points and it the target for this classification, and its used to identify patterns and key predictors of popularity. To reach this goal, several classification algorithms were tested:**Logistic Regression , Decision tree,** and **Random Forest**. Firstly, Random Forest is an ensemble method that constructs a multitude of decision trees and combines their outcomes. It is highly proficient at modeling complex, non-linear patterns and is relatively robust against overfitting and well suited for tabular datasets. It functions by dividing features at every node to create the highest purity of classes, and shines when variable-target relationships are non-linear or hierarchical. Moreover,it also offers feature importance which was utilized as a guiding principle to interpret the main predictors in this study [7]. Logistic Regression is using a linear model, where it uses its probabilistic function to guess outcome of binary output. Despite the interpretability with such a model, it is limited in scope as any non-linear relationship or complex feature interaction would require some sort of transformation of features beforehand. Decision Tree is a non-linear algorithm that recursively splits the dataset by features, yielding a tree structure of decision rules, but they are prone to overfitting without regularization. Hyperparameter optimization for Random Forest and Decision Tree algorithms was performed using Optuna, an automatic hyperparameter optimization framework. This made sure that the models were trained with their best configurations for performance. Additionally, to ensure that the model performance metrics are robust and reliable, we used cross-validation here while checking for different splits of data.

5.2 Feature Importance

Based on those three models, the feature importance plots are as follows:

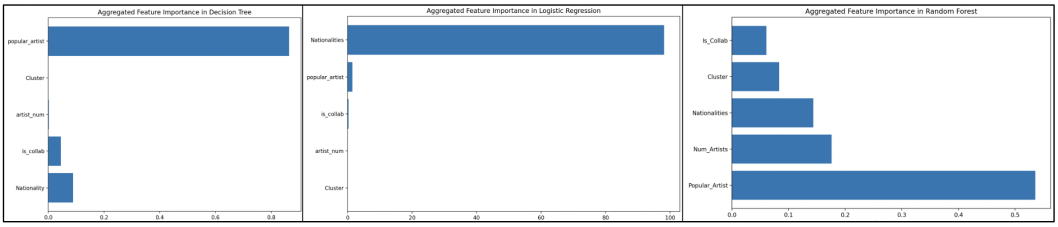


Figure 4: Feature Importance for all three models

The feature importance plots as shown in Figure 4 reveal that *popular_artist* is the most influential feature for the Decision Tree model, while Logistic Regression unexpectedly identifies Nationality as the most important. In contrast, the Random Forest model balances the importance across features like *popular_artist*, Cluster, and *is_collab*, leveraging all available information to outperform the other models.

6 Predicting Popular Artists

Since the musical features are inconsistent for any given song, our project treats the data as time-series, emphasizing daily artist rank and points to predict artists who are likely to rise in popularity in 2024. Using the historical data from 2017 to mid-2023, we forecast the total points each artist

may accumulate in 2024 as a measure of popularity.The sequential nature of the data and the goal to project future values made time-series forecasting the most suitable approach as most of the popular artist were present throughout time series.It allowed us to capture seasonal patterns, and anomalies over time, which would not be possible with static prediction models. We apply **ARIMA, Prophet, XGBoost, and LightGBM** individually to the dataset, applying the strengths of each to the forecasting process. ARIMA models are used to study time series information by examining patterns and correlations, over time periods With the parameters p, d, and q for auto-regressive components, differencing to ensure stationary, and moving averages respectively [2]. We used adfuller for ADF Statistic, p-value to check for non-stationary data, auto-arima for p, d and q, and tune the model for training, predicting and forecasting. Prophet is a flexible time-series forecasting model that captures trends, seasonality, and anomalies. The data was preprocessed by grouping entries by Date and Artist and clipping outliers in the points column using IQR method. Prophet had the capacity to factor in values using regressor but based on our tests models without regressor provided better results. As prophet have built-in features to detect change in trends with changepoints, it resonated with the patterns significantly even without regressors[8]. XGBoost has the ability to model feature interaction in a very complex way, while LightGBM efficiently processes big data with categorical feature support for hierarchical groupings, such as artist-region interactions [1][3]. Each of these models can allow different aspects of the dataset to be explored, minimizing any possibilities of error and providing very accurate predictions related to future music trends. In general ARIMA showed better results for artists with stationary datasets like 'Taylor Swift' as we believe her behavior to release albums on a periodic intervals consistently made it stationary, but this model needed a lot of hyperparameters tuning. On the other hand, Prophet displayed robust adaptability and versatility when handling seasonal patterns and external factors with the need for much tuning.

7 Results and Discussion

7.1 Popular Song Prediction Result

To compare the classifier models, we evaluate each model using *classification_report* function in *sklearn*. We use the weighted average score because our label is unbalanced (we only consider the top 15% rank of all songs to be popular), as in the real world, most songs will not make the top chart.

Classifier	Precision	Recall	F1 Score
Logistic Regression	0.81	0.75	0.77
Decision Tree	0.87	0.72	0.78
Random Forest	0.87	0.92	0.89

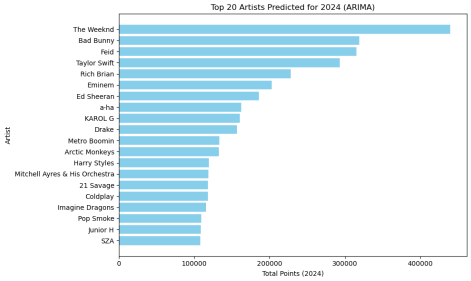
Figure 5: The weighted average scores of each model

Random forest, the most complicated algorithm of our three models, outperforms the others as shown in Figure 5. This means the model successfully capture complex relationships and interactions between features that simpler models like logistic regression or decision trees might miss. By aggregating the predictions of multiple decision trees, random forest reduces *overfitting*, improves generalization, and identifies subtle patterns in the data that contribute to better predictive performance.

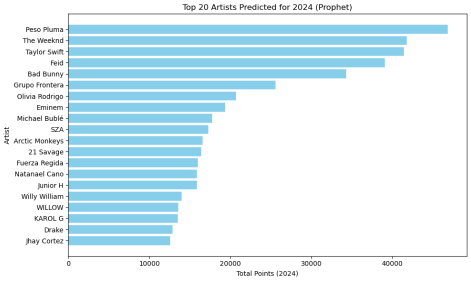
7.2 Popular Artist Prediction Result

Models	MAE	RMSE
ARIMA	46.33	62.17
Prophet	55.75	63.06
XGBoost	28.56	36.62
LightGBM	27.50	32.94

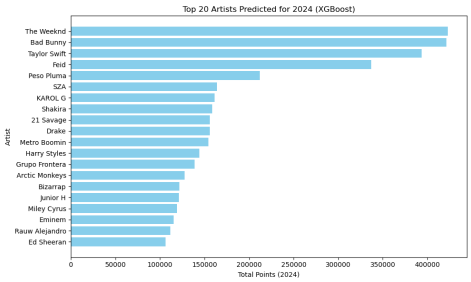
Table 1: Evaluation Metrics



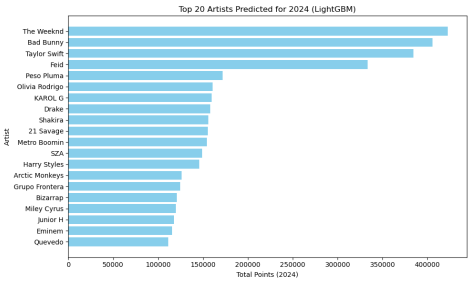
(a) ARIMA



(b) Prophet



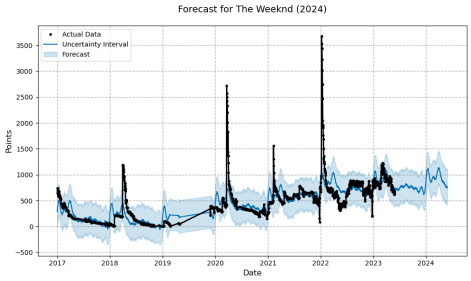
(c) XGBoost



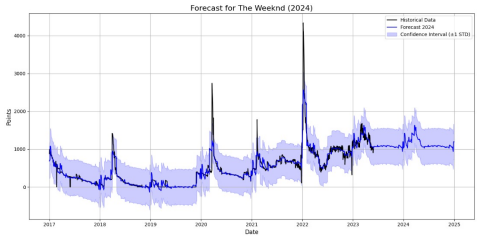
(d) LightGBM

Figure 6: Comparison of Top 20 Artists Across Models

Based on Figure 6 and Table 1, certain artists are prevalent throughout all four model predictions, thus showing a strong correlation between the models. Among these, the LightGBM had the best performance of those based on the evaluation metrics, with a MAE of 27.50 and a RMSE of 32.94. While the Prophet model had the least pleasing results in terms of metrics, its forecast plot for 2024 seemed to fit similar to observed trends compared to the other models as shown in Figure 7. Therefore, it is a strong candidate to capture specific temporal patterns.



(a) The Weeknd Forecast using Prophet



(b) The Weeknd Forecast using LightGBM

Figure 7: Plot Comparison between Prophet and LightGBM Model

8 Conclusion

We managed to achieve a precision, recall and F1 score of 0.87, 0.92, and 0.89, respectively by using Random Forest Classifier. Feature importance analysis shows that the feature popular_artist is a major contributor to make the song popular. Hence, we continue with artist popularity prediction using several time-series analysis models such as ARIMA, Prophet, XGBoost, and LightGBM. The result shows that some artists are likely to be in the Top 20 rankings of the 2024 chart, including The Weeknd, Bad Bunny, Feid, Taylor Swift, 21 Savage, and many more. Moreover, LightGBM has the better MAE and RMSE score, and its forecast visualization seemed to fit similar to observed trends compared to the other models.

.

References

[1] Tianqi Chen and Carlos Guestrin. "XGBoost: A scalable tree boosting system". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[2] Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. 2nd. Retrieved from https://otexts.com/fpp2/. OTexts, 2018.

[3] Guolin Ke et al. "LightGBM: A highly efficient gradient boosting decision tree". In: *Advances in Neural Information Processing Systems*. Vol. 30. Retrieved from https://papers.nips.cc/paper_files/paper/2017/file/f444f6d3815bca9f9e6b1e3de587c203-Paper.pdf. 2017, pp. 3149–3157.

[4] Xu Liu, Ling Zhang, and Zheng Zhang. "Clustering Methods and Applications". In: *Advances in Computer Science Research* 119 (2022). Accessed: 2024-11-1, pp. 81–86. URL: https://www.clausiuspress.com/article/592.html.

[5] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL: probml.ai.

[6] Kalhan Rosenblatt. *These charts show the spread of Mariah Carey's classic Christmas song*. https://www.nbcnews.com/pop-culture/music/these-charts-show-spread-mariah-carey-s-classic-christmas-song-n1283619. Accessed: 2024-11-15. 2021.

[7] Matthias Schonlau and Rosie Yuyan Zou. "The random forest algorithm for statistical learning". In: *The Stata Journal* 20.1 (2020). Accessed: 2024-11-10, pp. 3–29. DOI: 10.1177/1536867X20909688. URL: https://journals.sagepub.com/doi/full/10.1177/1536867X20909688.

[8] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: *The American Statistician* 72.1 (2018), pp. 37–45. DOI: 10.1080/00031305.2017.1380080.