Junwen Fang, Lyric Li, Rosy Xu

1 Introduction

This report explores the relationship between song features and popularity of songs. Our datasets contain 52000 songs including their features such as song length and explicitness, with popularity of songs ranges from 0 to 100. Additionally, we have a dataset which contains 10,000 users ratings on 5,000 songs rated from 0 to 4. In this report, we will apply Statistical Testings, Linear Regressions, Classification Models, and Recommendation System to predict and recommend songs for users.

In this project, every team member contributes equally to the project. In terms of data preprocessing, we found that there were no null values in the Spotify dataset – each attribute of songs has a value. We also examined the distribution of each column in Spotify by using boxplots for numerical values and pie graphs for categorical variables. We found that duration, loudness, speechiness, instrumentalness, and liveness indicated a large number of outliers. However, song attributes' outliers are in ranges in the provided range as indicated by the description of the dataset, so we kept all data given in Spotify dataset. In this project, we assume that the popularity cannot be reduced to mean because we think two groups with the same popularity mean convey different meanings. ([50,50] and [40,60] are different)

2 Feature Relationship with Popularity

2.1 Song Length (Q1)

To find the relationship between song length and popularity of a song, we first found the corresponding columns, "duration" and "popularity", and transferred the dataframe to numpy. Also, we visualized these two variables in a **scatter plot** as shown in Figure 2.1.1, which showed little correlation between song length and popularity. We also calculated the **Peasrson correlation = -0.05** which is smaller than 0.1. Therefore, it doesn't necessarily have a relationship between song length and popularity.

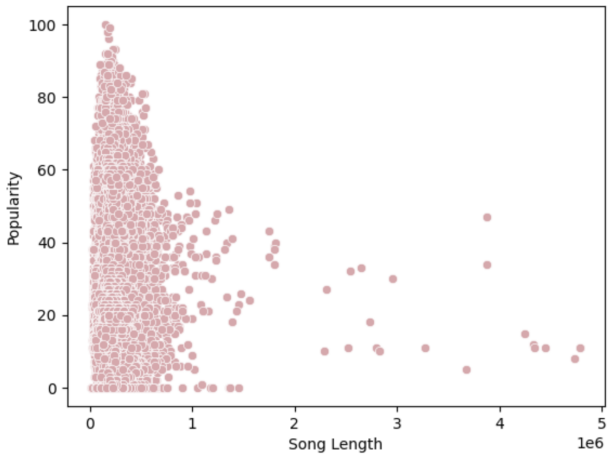


Figure 2.1.1 Scatter Plot of Song Length v.s. Popularity

2.2 Explicitness (Q2)

We utilized the column “explicit” to find explicitness of each song, and transferred the dataframe to numpy. To find if the explicit rating songs are more popular than the implicit rating songs, we first check the median of two groups: median popularity of explicit is **34** and the median popularity of implicit is **33**. Then, we visualized the distribution of explicit and implicit ratings by histogram.

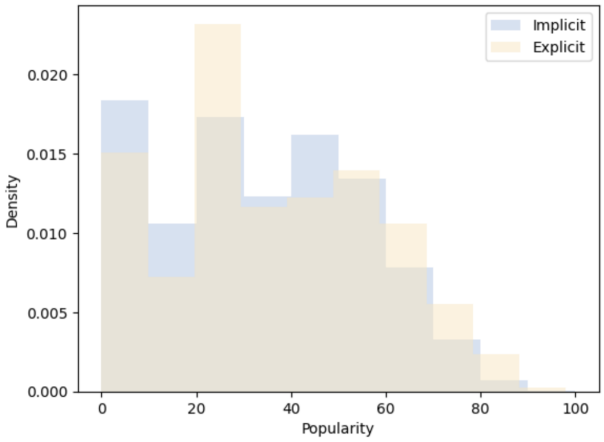


Figure 2.2.1 Histogram of Popularity for Implicit and Explicit Songs

Also, since it is song popularity and it’s meaningless to reduce to mean, we used **One-sided Mann-Whitey U test** to compare the median ratings of explicit and implicit songs. The u-statistics is **1.39e8** and p-value is **1.53e-19**. Because the p-value is less than 0.05, we had enough evidence to **reject null hypothesis** and thus concluded that explicit songs are more popular than implicit songs.

2.3 Major and Minor Key (Q3)

We defined “mode = 1” as “major” and “mode = 0” as “minor”, and transferred the dataframe to numpy. To find if songs in major key are more popular than those in minor keys, we first check the median popularity of two groups: median of major is **32**, median of minor is **34**. Then, we visualized the distribution of major and minor ratings by histogram as shown in Figure 2.3.1. The distribution of two groups seem to be similar to each other.

Also, since it is song popularity and it is meaningless to reduce to mean, we use **One-sided Mann-Whitey U test** to compare the median ratings of major and minor songs. We got a u-statistics of **3.10e8** and **p-value of about 1 (0.99)** > 0.05, so we failed to reject H0 and conclude that songs in major key are **not more popular** than songs in minor key.

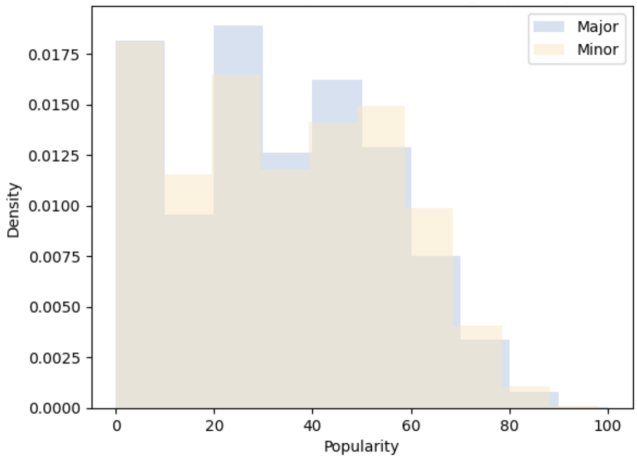


Figure 2.3.1 Histogram of Popularity for Implicit and Explicit Songs

3 Popularity Prediction Model

3.1 Simple Linear Regression (Q4)

After performing a 30%-70% train-test split, we built a simple linear regression model for each of the ten features. We then evaluated these models using Mean Squared Error (MSE) and R-squared (R^2) metrics. The feature "**Instrumentalness**" emerged as the best predictor, with the lowest MSE **(459)** and highest R^2 **(0.018)**. Though the model built by "Instrumentalness" has the best performance among ten models, the R^2 is still relatively low, and the MSE is almost the same as other models.

Song Feature	MSE	R2
Instrumentalness	459.29	0.0179
Duration	465.61	0.0044
Energy	465.92	0.0038
Loudness	466.57	0.0024
Liveness	466.71	0.0021
Speechiness	466.81	0.0019
Acousticness	467.11	0.0012
Valence	466.88	0.0017
Danceability	467.41	0.0006
Tempo	467.73	-0.0001

Table 3.1.1 Table of Simple Linear Regression Model on Each Song Feature, Ordered by R2

3.2 Multiple Linear Regression (Q5)

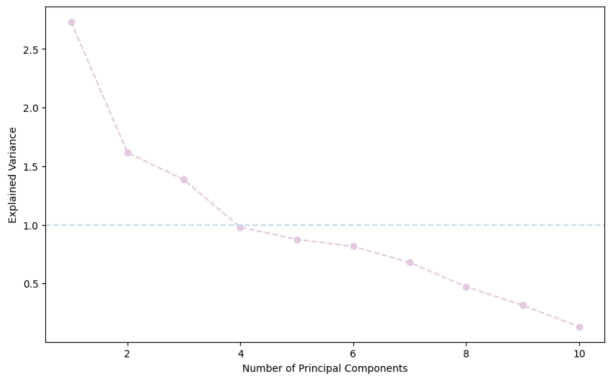
We built a multiple linear regression model using all ten song features to predict song popularity, and utilized MSE and R^2 to evaluate model performance. Our model yielded MSE of **445.86** and R^2 of **0.047**. Compared to the best model in Q4, the multiple linear regression model improved R^2 by **76.5%** and MSE decreased by **2.86%**. The increase in R^2 is **expected** because more features imply higher explainability of variance. The MSE, however, only

decreased little compared to the simple linear regression model, indicating **little improvement** on prediction by multiple linear regression.

We utilized **Lasso Regression** to regularize the multiple linear regression. We utilized **RandomizedSearchCV** to search the best parameter and narrowed alpha in [0.0001, 0.0006, …, 0.9996, 1). The optimal alpha is 0.0006 and yielded an MSE of **445.85** and R^2 of **0.047**. The result of regularization is quite similar to the multiple linear regression model and suggests that the multiple linear regression model is **not overfitting**.

3.3 Multiple Linear Regression with Principal Components Analysis (Q6)

To extract the principal components, we first standardized the data and then applied PCA. As the optimal number of principal components was unknown, we examined the explained variance for each possible number of components to determine the appropriate count. Based on the graph, we selected **three principal components**, as they collectively had an explained variance exceeding 1 according to **Kaiser Criterion** (specifically, 1.384). The three principal components accounted for **57.4%** of the total variance.



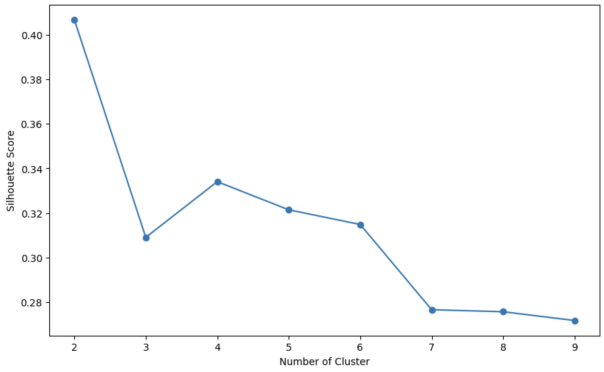


Figure 3.3.1 Explained Variance by Principal Component Figure 3.3.2 Silhouette Score by Number of Cluster

Next, we employed the **K-Means** model to categorize the data. As the optimal number of clusters was unknown, we utilized the **silhouette method** to calculate the silhouette score for each potential cluster count as shown in Figure 3.3.2. We ultimately chose **two clusters**, as this number yielded the highest silhouette score.

Upon comparing the two clusters created by the k-means model with the music track genres, we observed distinct characteristics: Cluster 0 predominantly features music with a relatively stronger beat, while the Cluster 1 consists of tracks that focus more on melody. Though the majority of music genres were classified into Cluster 0, which is also reasonable due to little light music among the track genres.

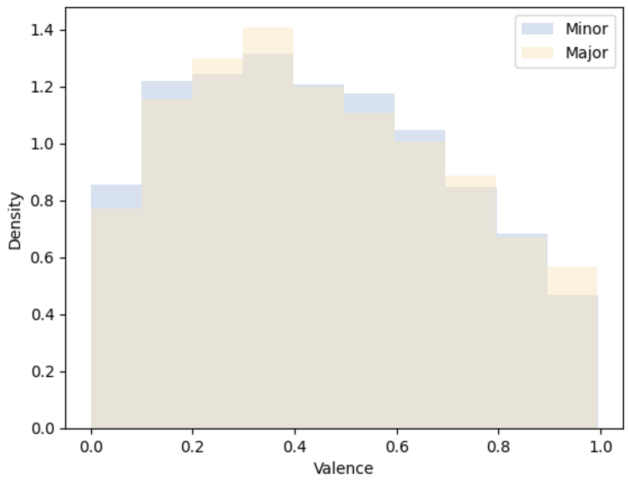
Track Genre	Ratio_0	Ratio_1
classical	0.069	0.931
ambient	0.101	0.899
guitar	0.215	0.785
disney	0.227	0.773
acoustic	0.441	0.559
...	...	...
hardcore	0.991	0.009
death-metal	0.992	0.008
happy	0.994	0.006
drum-and-bass	0.996	0.004
edm	0.997	0.003

Figure 3.3.3 Clustering Result (Ratio_0 represents proportion of track genre classified as group 0)

4 Song Feature Prediction Model

4.1 Major and Minor Key Prediction (Q7)

To predict whether a music track is in a major or minor key by its valence, we employed both logistic regression and a support vector machine. However, due to limitations in our data (no correlation seen between valence and mode as shown in Figure 4.1.1), both models predominantly predicted **every song as being in a major key**, resulting in the same accuracy of around **0.62**, which is only a little bit better than random guess. The model performance is under expected due to low specificity and high sensitivity as shown in Figure 4.1.2. The model can be further improved by several approaches, such as upsampling training data, adding regularization on incorrectly predicting minor classes.



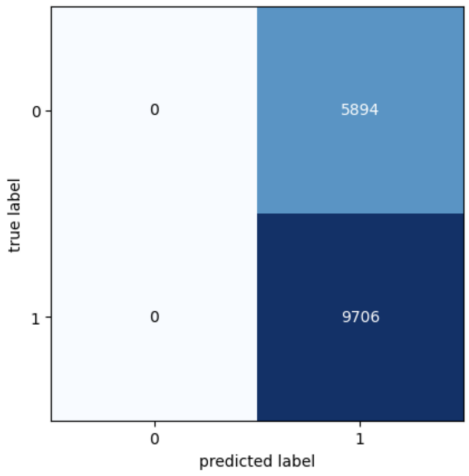


Figure 4.1.1 Valence Distribution for Major and Minor Song Figure 4.1.2 Confusion Matrix for SVM and LR

4.2 Genre Prediction (Q8)

We developed a neural network to predict music genres using the original 10 song features from question 4. The process began with scaling the features and one-hot encoding the genre labels. Then, we split the data into training (70%) and test (30%) sets. The Sequential model architecture included an input layer of **64** neurons (ReLU activation), a hidden layer of **32** neurons (ReLU activation), and an output layer with softmax activation,

corresponding to the number of genre categories. We compiled the model using the Adam optimizer and categorical cross entropy loss, focusing on accuracy as the metric. After training for 100 epochs with a batch size of 32, we evaluated the model on the test set and got the accuracy **28.95%**.

5 Recommendation System Model

5.1 Popularity Based Model (Q9)

To investigate the relationship between popularity and average star rating for the 5k songs, we first calculated the average star ratings for each song by taking average on all ratings for each song. Then, we drew a **scatter plot** to visualize the distribution of average star ratings as shown in Figure 5.1.1. From the graph, we could see that as popularity increases, the average star rating also tends to increase. We then calculated the **Pearson correlation coefficient** between popularity and average star rating, and yielded **0.57**. This suggests a **relatively strong positive** relationship between average star rating and popularity.

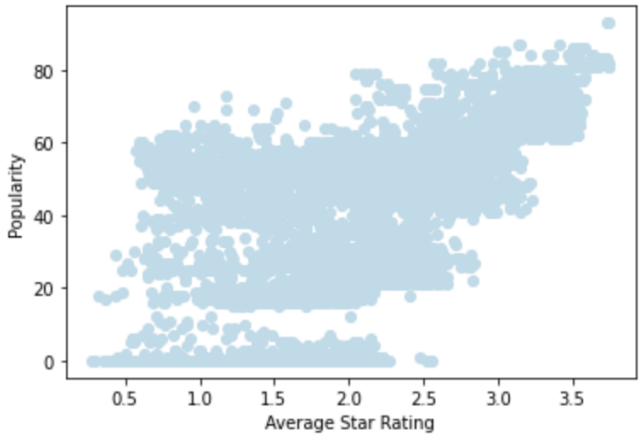


Figure 5.1.1 Scatter Plot of Average Star Rating v.s. Popularity

We then built a recommendation model ordered by the **average star rating** of each song, and chose top 10 songs as the "greatest hits". Table 5.1.2 shows the details of the 10 top songs.

Index	Artists	Track Name
3877	The Offspring	You're Gonna Go Far, Kid
3003	The Neighbourhood	Sweater Weather
2260	Red Hot Chili Peppers	Can't Stop
2562	The Offspring	You're Gonna Go Far, Kid
3216	Red Hot Chili Peppers	Californication
2105	Red Hot Chili Peppers	Californication
2003	The Neighbourhood	Sweater Weather
2011	WALK THE MOON	Shut Up and Dance
3464	Red Hot Chili Peppers	Can't Stop
3253	Gorillaz;Tame Impala;Bootie Brown	New Gold (feat. Tame Impala and Bootie Brown)

Table 5.1.2 Top 10 Recommended Songs Generated by Popularity-Based Model

5.2 Collaborative Filtering Model (Q10)

We imputed the null values by (user's average rating + song's average rating)/2 for each user and each song. We utilized this imputation because the rating is most likely close to the user's average rating or this song's average rating. To build a collaborative filtering model, we used **Singular Value Decomposition (SVD)** to decompose the rating matrix. With SVD, we could predict each user's ratings on each song. We then sorted the predicted ratings and recommended the **top 10 unrated songs** as the personalized recommendation songs for each user.

To evaluate popularity-based model and collaborative filtering model, we will use **precision@10** as our metric. Because we do not have real-time interaction data from users, we will instead use the rated songs to evaluate model performance. In other words, we calculated the actual top 10 song ratings by each rater, and compared the songs with the predicted top 10 rated song ratings from two models to find precision. We found that the average precision@10 for personalized recommendation is about **0.10**, while the average precision@10 for popularity-based recommendation is about **0.01**. As a result, the personalized recommendation has a relatively **better** performance, though still under expectation. And the average precision indicates that we can successfully predict 1 out of 10 songs that the user liked most among all rated songs by using personalized recommendation. We concluded that though both models yielded the low precision, the personalized recommendation had a better outcome that "greatest hits".

6 Extra Credit

In Section 3.1, we have found that instrumentalness predicts popularity best. However, even if we built a simple linear regression with instrumentalness, the model performance is still under expectation. In this question, we were interested in which features **predict average ratings** of 5000 songs the best. By building linear regression models for each song feature, we found that **loudness** predicted average ratings the best, while duration predicted the average ratings the worst, as shown in Table 6.1. Among the top three best features, the loudness and energy has a **positive correlation** with average ratings, while acousticness has a **negative correlation** with average ratings. This indicates the popular trend of loud, intense and synthesized songs.

Feature	MSE	R2
loudness	0.30	0.45
energy	0.37	0.31
acousticness	0.42	0.22
instrumentalness	0.44	0.19
danceability	0.50	0.07
valence	0.51	0.05
liveness	0.54	0.01
tempo	0.54	0.00
speechiness	0.54	-0.00
duration	0.54	-0.00

Table 6.1 Linear Regression Model with Each Song Feature, Ordered by R2