DS-GA 1001 Intro to Data Science - **Capstone Project Report**

Group 31

Antonio Vela Gartner
Shuyan Liu
Saakshi More

Member Contribution:

Question 1 was discussed in group and finished/reported by Antonio
Questions 2 to 7 were answered in group and the reports for these were written in group
Questions 8 to 10 were answered by Shuyan and Saakshi

Section 1 - Data processing:

Data Cleaning:

Before starting the project, we ensured that the song dataset was clean in that it did not contain any missing values.

Data Transformation:

We plotted the distribution of all audio features in the song dataset and realized that all features except 3 (duration, loudness, and tempo) are defined in the range 0 to 1. Moreover, the duration data was extremely right-skewed. Thus, we transformed our data through the following steps:

- Replace duration with its logarithmic scale
- Perform min-max normalization on loudness, tempo, and the log of duration to limit their range between 0 and 1’’

Section 2 - Data Analysis and Machine Learning:

1) Is there a relationship between song length and popularity of a song? If so, is it positive or negative?

As noted above, we observed that the duration data was heavily skewed to the right, indicating that there are a few very long songs. To include these songs in our analysis we performed a log transformation, which made the distribution more normal-like. Then, on this data we performed a simple linear regression against popularity. The model had a beta of around -1 and an R^2 of 0.0004. These numbers suggest that duration is negatively correlated with popularity but it is a minor factor, and as such does not explain much of the variance in song popularity. Then, to confirm that the slope is significant we performed a T-test in which the null hypothesis was that there is no correlation between popularity and log-duration. The p-value of the test was 1e-5, which is lower than the threshold, so we rejected the null hypothesis. All-in-all it is not surprising that longer songs are less popular, especially taking into account that popularity is based on play times, which of course are harder to accumulate the longer the song is.

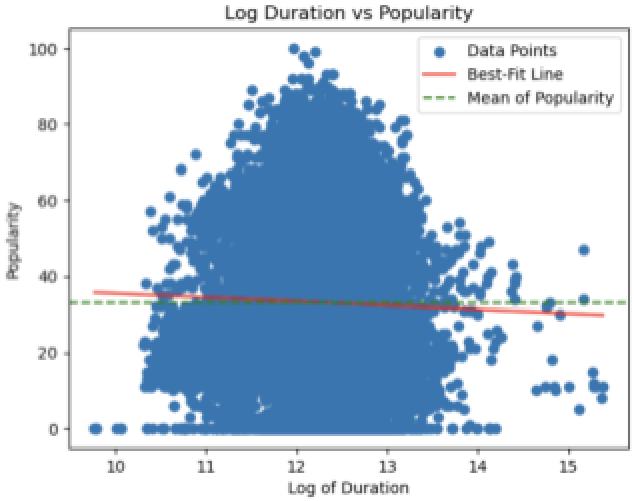


Figure 1. Scatterplot of log-duration vs. popularity and fitted regression model

2) Are explicitly rated songs more popular than songs that are not explicit?

To perform our analysis, we split the dataset into 2 groups: explicitly rated songs and non-explicit songs. We stored the corresponding popularity values as well. We plotted the distribution of both these groups to eyeball whether we could perform a t-test on the data (see figure 2a). To get a better sense of the data, we plotted a second histogram of 'popularity' where 'explicit' is True and False with the y-axis representing the percentage within each group (see figure 2b). On this histogram, we also marked the median popularity values for each group and observed that the median popularity was higher for songs that were marked true. To determine the statistical significance of this observation, we had to choose which inference test we could perform. It was reasonable to reduce the data to sample means since we made a linearity assumption for popularity. However, the visualizations were sufficient to deduce that the data was not normally distributed, meaning that we could not use t-tests. Thus, we performed a Mann-Whitney test to confirm the statistical significance of the median difference between the two groups. We set the alternative hypothesis to “greater” since we had to check if explicitly rated songs were **more** popular than songs that were not explicit. The p-value for the Mann-Whitney test statistic was to the order of **1e-11**, much lower than the standard alpha value of 0.05. Hence, we rejected our null hypothesis, which stated that explicitly rated songs are not more popular than non-explicit songs. This indicated that the difference in the median of the two groups was indeed significant.

To further strengthen our result, we calculated Cohen's d to determine the effect size and obtained a value of 0.116. The score reflected a small effect size, which is expected since the median difference we obtained (as can be seen in figure 2b) was not very large.

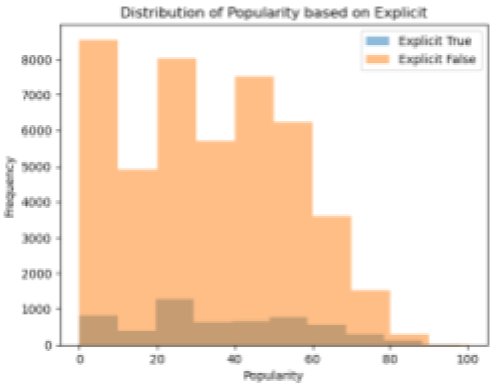


Figure 2a. Count Distribution of Popularity for Explicit v/s Non-explicit songs

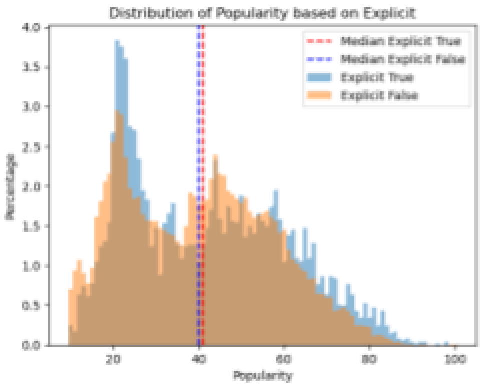


Figure 2b. Percentage Distribution of Popularity for Explicit v/s Non-explicit songs

Although the data was heavily skewed, we decided to not transform it since this skewedness reflected the true nature of the underlying phenomenon.

3) Are songs in major key more popular than songs in minor key?

For this question, we focused on 'popularity' and 'mode' features of the data. Mode can take up 2 value - "major key" and "minor key". As was the case with the previous question, we plotted two histograms to answer this question: a frequency distribution and a percentage distribution of popularity colored by 'mode' (figures 3a and 3b respectively). In dealing with popularity, we did not see a normal distribution for this data as well and decided to do a Mann-Whitney test. Achieving a p-value of **0.99**, we failed to reject the null hypothesis that major key songs are not more popular than minor key songs. Thus, we concluded that major key songs are not any more popular than minor key songs. Similar to the previous question, despite the skewed distribution of the data, we did not transform it since it reflects the true nature of the data. Instead we used the percentage histogram to give us more insight into the data, if required, and visually, we can see that the distributions for the two groups are not that different. In short, Key does not affect the popularity of songs.

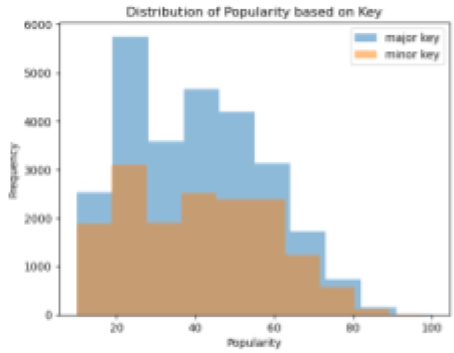


Figure 3a. Count Distribution of Popularity for Major key v/s Minor key songs

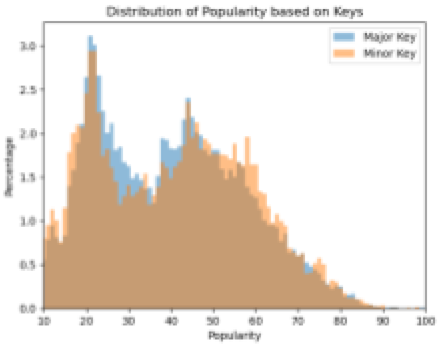
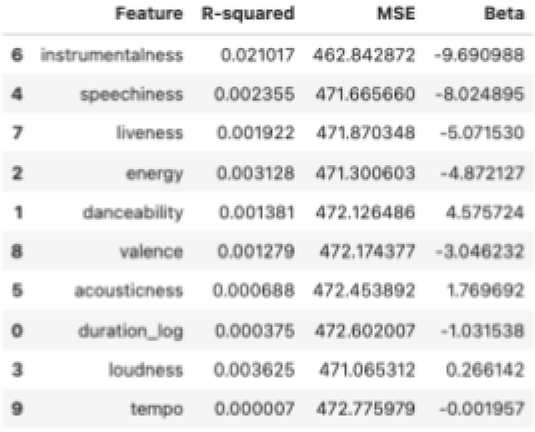


Figure 2b. Percentage Distribution of Popularity for Major key v/s Minor key songs

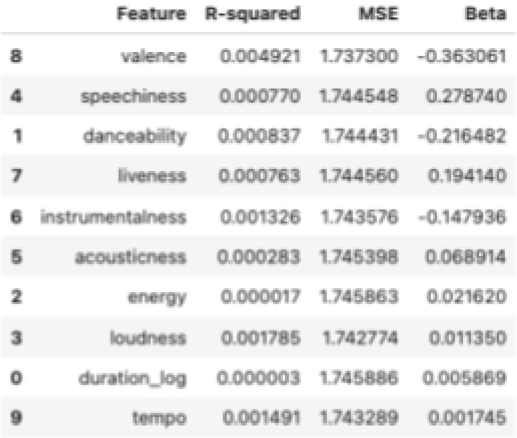
4) Which of the following 10 song features: duration, danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence and tempo predicts popularity best? How good is this model?

For this question, we built linear regression models and recorded the RMSE score, R^2 value and the beta coefficient for each feature as a predictor. We decided to test models with and without popularity log transformation to uncover potential exponential or multiplicative relationships.



	Feature	R-squared	MSE	Beta
6	instrumentalness	0.021017	462.842872	-9.690988
4	speechiness	0.002355	471.665660	-8.024895
7	liveness	0.001922	471.870348	-5.071530
2	energy	0.003128	471.300603	-4.872127
1	danceability	0.001381	472.126486	4.575724
8	valence	0.001279	472.174377	-3.046232
5	acousticness	0.000688	472.453892	1.769692
0	duration_log	0.000375	472.602007	-1.031538
3	loudness	0.003625	471.065312	0.266142
9	tempo	0.000007	472.775979	-0.001957

Table 4a. Results for linear model to popularity



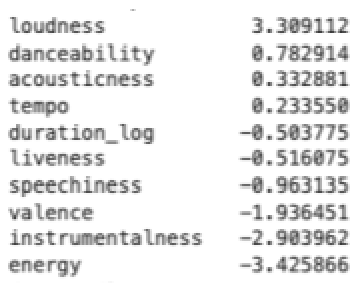
	Feature	R-squared	MSE	Beta
8	valence	0.004921	1.737300	-0.363061
4	speechiness	0.000770	1.744548	0.278740
1	danceability	0.000837	1.744431	-0.216482
7	liveness	0.000763	1.744560	0.194140
6	instrumentalness	0.001326	1.743576	-0.147936
5	acousticness	0.000283	1.745398	0.068914
2	energy	0.000017	1.745863	0.021620
3	loudness	0.001785	1.742774	0.011350
0	duration_log	0.000003	1.745886	0.005869
9	tempo	0.001491	1.743289	0.001745

Table 4b. Results for linear model to log-popularity

By sorting the final results based on the absolute values of the Beta coefficients in descending order, we identified which features had the most significant impact on predicting song popularity. First, it was clear that taking the log-popularity decreased the amount of variance explained by the model (Table 4b). For the models built with popularity **'Instrumentalness'** had the highest coefficient yet it only explained 2% of the data variability. The R^2 values for the rest of the predictors were even lower and thus did not allow conclusive predictions. As in question 1, the information contained in these 10 features is not enough to make predictions on popularity of a song.

5) Building a model that uses *all* of the song features mentioned in question 1, how well can you predict popularity? How much (if at all) is this model improved compared to the model in question 4). How do you account for this? What happens if you regularize your model?

Using the 10 features in question 4 as predictors, we built a multiple linear regression model to predict popularity. The feature 'duration' has a distribution that is right skewed, to optimize our model performance, we log transformed data in this column. Three of the feature 'loudness', 'tempo', and 'Log duration' were then normalized between 0 and 1 to match the scale of other features. Now all of our predictors are on the same scale from 0 to 1. After fitting our data into the model, we yielded a R^2 of 0.05. MSE for our model is 446.5. Overall our model performed very poorly on predicting the popularity of the song. Here we also tried a log transformed ‘popularity’ which yielded similar results. Our model performed poorly, the same as question 4. We hypothesized that multicollinearity was making the model overfit the noise. (e.g. ‘speechiness’ and ‘instrumentalness’ are highly correlated.) Therefore, we tried Lasso regularization to remove these predictors. The result shows that the model’s performance did not improve at all. With the same MSE and R². The beta of our predictors did not get penalized significantly so our initial hypothesis of multicollinearity is not the reason for our model’s poor performance, rather the model is underfitting the data. In this model the features with largest beta absolute value were instrumentalness, energy, and loudness (Table 7.) It is interesting to note that loudness had a similar beta in the single regression model, as opposed to instrumentalness which has a smaller beta in the multiple regression model. This indicates that in the single regression model some collinearity between features was not accounted for.



```
loudness             3.309112
danceability         0.782914
acousticness         0.332881
tempo                0.233550
duration_log        -0.503775
liveness            -0.516075
speechiness         -0.963135
valence             -1.936451
instrumentalness    -2.903962
energy              -3.425866
```

Table 7. Betas for multiple linear regression model

6) When considering the 10 song features in the previous question, how many meaningful principal components can you extract? What proportion of the variance do these principal components account for? Using these principal components, how many clusters can you identify? Do these clusters reasonably correspond to the genre labels in column 20 of the data?

For this question, we performed PCA on the 10 song features and using a 95% cut off threshold, we were able to extract 8 meaningful principal components. To determine the number of the clusters, we used the silhouette method with K-means clustering, k=2 was identified as the optimal cluster number, however this is the method’s cluster number bottom limit. By looking at the data projected in the first to principal components colored according to these clustering it was clear that there are no real meaningful clusters. For completion, we also performed K-means clustering with k=52 which is the number of music genres reported. It was not surprising that this gave a very low silhouette score. In short, the data contained in these 10 song features is not enough to cluster songs by genre.

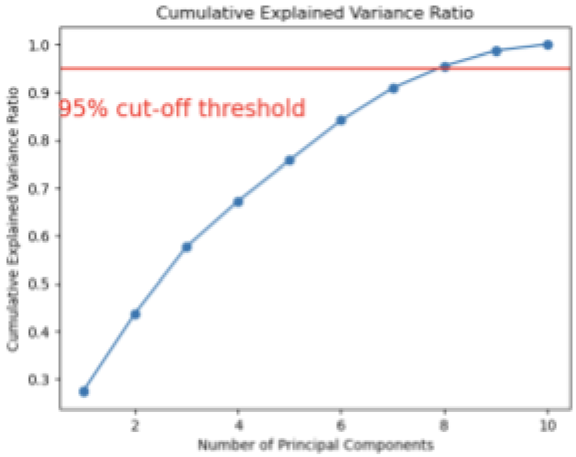


Figure 6a. Cumulative explained variance by principal components

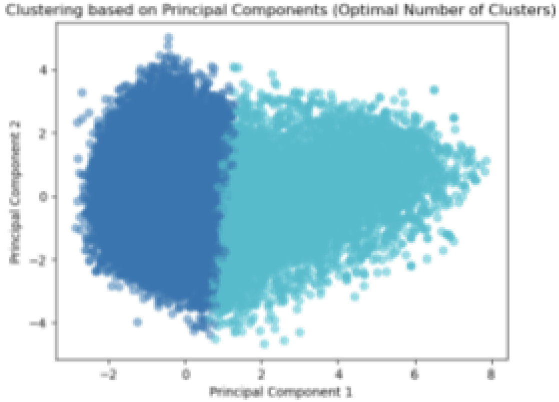


Figure 6b. K Means Clustering with k=2

7) Can you predict whether a song is in major or minor key from valence using logistic regression or a support vector machine? If so, how good is this prediction? If not, is there a better one?

For this question, we used column ‘mode’ in our data which contains major and minor for each of the songs. This will also be our outcome predicted by our predictor ‘valence’. The ‘mode’ data is unbalanced with 32391 values as major and 19609 as minor. Undersampler is used to address this problem. After the preprocessing of our data using undersampler, we first fit our data into a logistic regression model. The model yielded an accuracy of 0.503 and an AUC score of 0.5. This result indicates that our model’s prediction based on predictor ‘valence’ performed poorly, and no better than random guessing. We then fit our data into SVM and the result is almost identical. The SVM model yielded an accuracy of 0.503 and AUC of 0.5. Below figure also showed how the ROC curve looked for our models. Both models failed to make proper predictions. To explore further if other classification models would give us a better prediction, we used random forest model. The random forest model yielded a slightly better result with an accuracy of 0.52 and AUC score 0.53. Although the models performance was better than the SVM and logistic regression model, it still failed the task of predicting ‘mode’ with ‘valence’. Intuitively, we think there will not be a classification model that can yield any better result since we only have one predictor and it is hard to overcome underfitting.

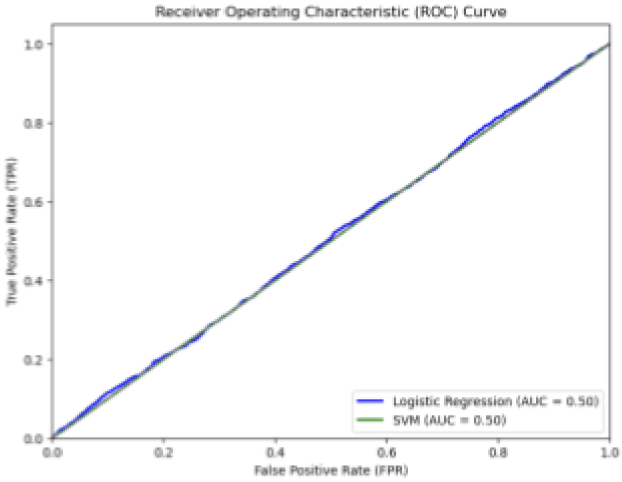


Figure 7. ROC Curve for Logistic Regression and SVM models

8) Can you predict genre by using the 10 song features from question 4 directly or the principal components you extracted in question 6 with a neural network? How well does this work?

For this question, we firstly used 10 features we had from question 4, our target for this question will be ‘track_genre”. Missing values from the selected features and target are dropped. To ensure all of our features are on the same scale, we normalized ‘duration_log’, ‘ loudness’ and ‘temp’ using the MinmaxScaler. StandardScaler is used to standardize the training and test feature sets. The target variable 'track_genre' is encoded using LabelEncoder. A sequential neural network model is constructed with three Dense layers. The first two layers use ReLU activation, while the output layer uses softmax. The number of neurons in the output layer corresponds to the number of unique genres in the target variable. After fitting the data, we yielded an accuracy of 0.28 and macro averaged F1 score of 0.26 for our model. This indicates that our model’s performance is not ideal and failed to predict the genre of the song using the neural network model. To see if the performance can be improved using the 8 principle component we extract earlier, we ran the model again and the Macro Averaged F1 score is even lower, 0.24, indicating that both of these approaches are not sufficient to make proper prediction.

9) In recommender systems, the popularity based model is an important baseline. We have a two part question in this regard: a) Is there a relationship between popularity and average star rating for the 5k songs we have explicit feedback for? b) Which 10 songs are in the “greatest hits” (out of the 5k songs), on the basis of the popularity based model?

For this question, we handled missing values in our datafile ‘starRatings’ with a 50/50 blend approach. We took the average of each column and row and each weighted 50% to fill the missing values. We first checked the correlation between popularity and average star rating of our songs, we had a correlation coefficient 0.57 (Figure 9a), indicating there is a moderate positive relationship between our two variables. We then ran the test again to check the correlation between the popularity and our imputed start ratings. It gave us the same result of 0.57 correlation coefficient. (Figure 9b) We then ran the same data sets using spearman’s rank. Both average star rating and imputed start ratings yielded similar results of p-value 0.0 and correlation coefficient 0.54. We reject the null hypothesis and conclude that observed correlation is statistically significant. A moderate positive correlation is recognized based on our data that songs with higher ratings will tend to have higher popularity. OLS regression is also performed to confirm this relationship. With R^2 of 0.324. To answer the b part of the question, after dropping duplicated song in our data file, we sorted songs based on popularity and top 10 were select as our 10 songs of ‘’greatest hits’’ (Table 9c)

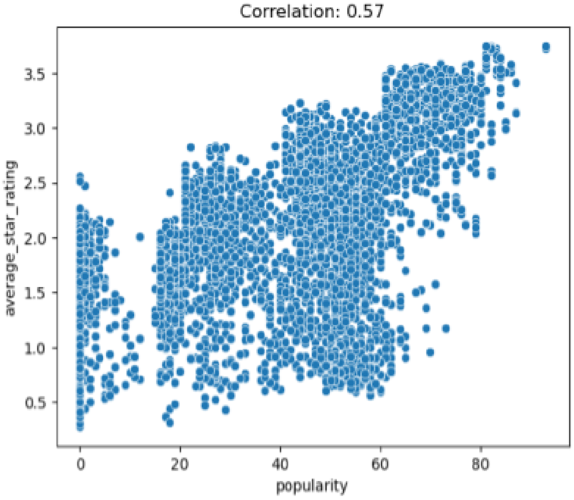


Figure 9a. Correlation plot of popularity vs. average star rating

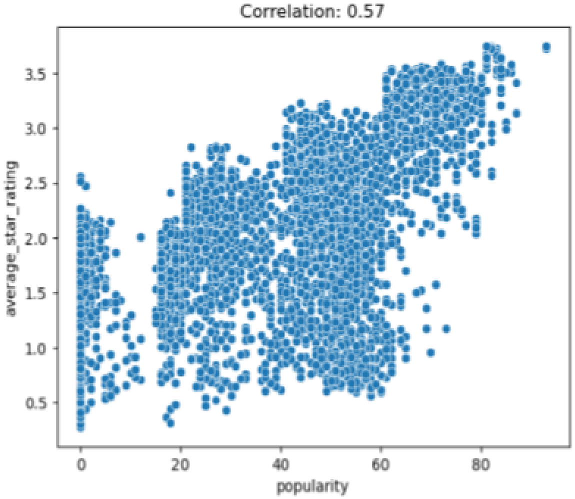
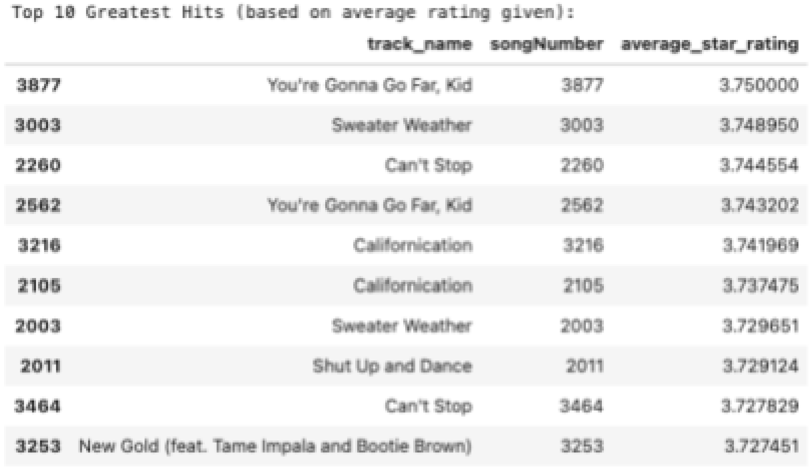


Figure 9b. Correlation plot of popularity vs. imputed average star rating



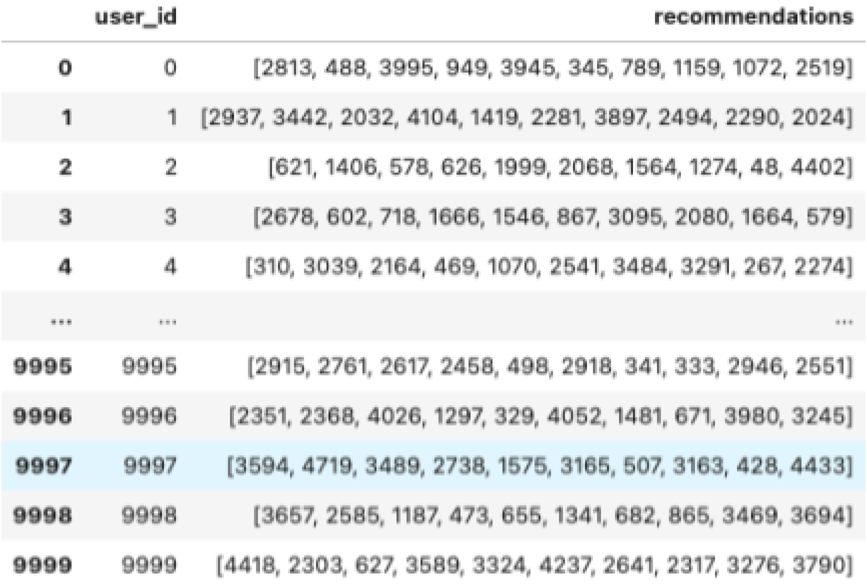
```
Top 10 Greatest Hits (based on average rating given):
```

	track_name	songNumber	average_star_rating
3877	You're Gonna Go Far, Kid	3877	3.750000
3003	Sweater Weather	3003	3.748950
2260	Can't Stop	2260	3.744554
2562	You're Gonna Go Far, Kid	2562	3.743202
3216	Californication	3216	3.741969
2105	Californication	2105	3.737475
2003	Sweater Weather	2003	3.729651
2011	Shut Up and Dance	2011	3.729124
3464	Can't Stop	3464	3.727829
3253	New Gold (feat. Tame Impala and Bootie Brown)	3253	3.727451

Table 9c. Top 10 "Greatest Hits"

10) You want to create a "personal mixtape" for all 10k users we have explicit feedback for. This mixtape contains individualized recommendations as to which 10 songs (out of the 5k) a given user will enjoy most. How do these recommendations compare to the "greatest hits" from the previous question and how good is your recommender system in making recommendations?

For this question we detail the application of collaborative filtering via matrix factorization to develop the "personal mixtape" recommendation system for all 10k users. Our objective is to recommend a list of the top 10 songs to each user based on their implicit preferences and historical interactions. We employed Truncated Singular Value Decomposition (SVD), to decompose the user-item interaction matrix into latent factors. We chose 17 as the number of latent factors matched with our 17 features.We transformed the original sparse matrix into a latent user feature matrix using TruncatedSVD. This transformation allowed us to predict ratings for songs that users had not yet interacted with. For each user, we predicted the ratings of songs they had not rated and selected the top 10 songs with the highest predicted ratings. The predictions were stored in a DataFrame for ease of analysis and interpretation. (Table 10a) In the evaluation of our song recommender system, we utilized Mean Average Precision (MAP) as the metric to assess the alignment between the system's recommendations and users' actual song ratings. Surprisingly, the MAP score was calculated to be 0, indicating no overlap between the recommended songs and the songs users actually rated. Suggesting our model failed recommending anything that can interest our user.This unexpected result suggests several potential issues: a mismatch in user or song IDs between our recommendation and rating datasets, or an overly stringent evaluation method. The zero MAP score underscores the need for a thorough review of both our data alignment and the recommendation algorithm, highlighting the need for refinement to better capture user preferences.



	user_id	recommendations
0	0	[2813, 488, 3995, 949, 3945, 345, 789, 1159, 1072, 2519]
1	1	[2937, 3442, 2032, 4104, 1419, 2281, 3897, 2494, 2290, 2024]
2	2	[621, 1406, 578, 626, 1999, 2068, 1564, 1274, 48, 4402]
3	3	[2678, 602, 718, 1666, 1546, 867, 3095, 2080, 1664, 579]
4	4	[310, 3039, 2164, 469, 1070, 2541, 3484, 3291, 267, 2274]
...	...	...
9995	9995	[2915, 2761, 2617, 2458, 498, 2918, 341, 333, 2946, 2551]
9996	9996	[2351, 2368, 4026, 1297, 329, 4052, 1481, 671, 3980, 3245]
9997	9997	[3594, 4719, 3489, 2738, 1575, 3165, 507, 3163, 428, 4433]
9998	9998	[3657, 2585, 1187, 473, 655, 1341, 682, 865, 3469, 3694]
9999	9999	[4418, 2303, 627, 3589, 3324, 4237, 2641, 2317, 3276, 3790]

Table 10a. Recommendation dataframe based on collaborative filtering

Extra Credit: Whether album names with one word are associated with higher popularity?

In this question we want to find out if albums that contain only one word are associated with higher popularity than albums that has more than one word. We first created a boolean column where album that contains more than one word will be marked as 1 otherwise 0. We then fit our model into an logistic regression model to see if we can predict if album name is one word or not using popularity. The result yielded an accuracy of 0.8 and AUC score of 0.61. This result suggests that our model can somewhat distinguish between one word and more than one word albums but the predicting power is not strong. Further analysis with a balanced approach and perhaps additional features is recommended for a more accurate interpretation.

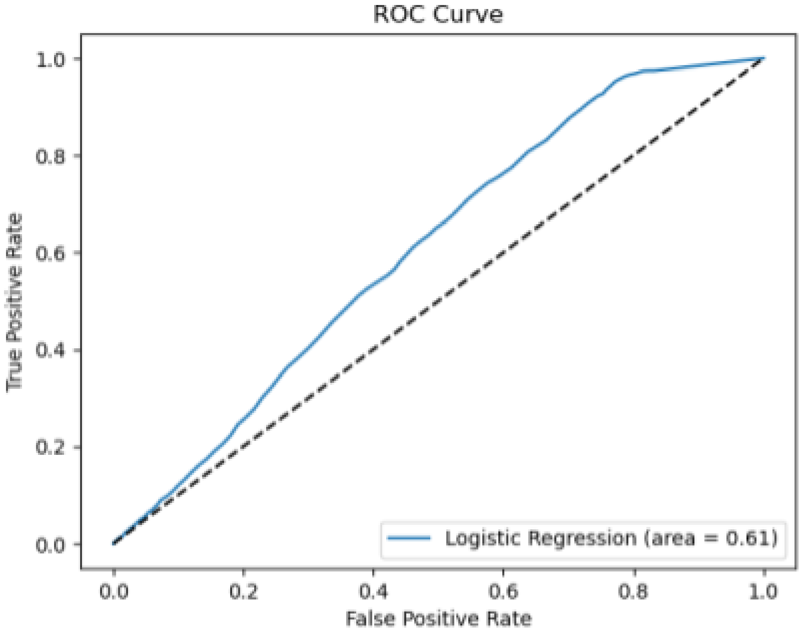


Figure 11. ROC Curve for the Logistic Regression