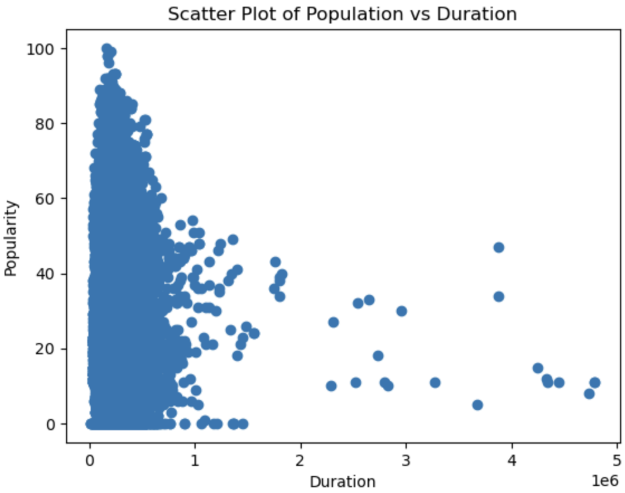
DS-GA-1001-1-001 Final Project

Group 23: Juhua Huang, Dennis Hu, Xinyi Yang

Dec. 2023

1. Is there a relationship between song length and popularity of a song? If so, is it positive or negative?
Model: T-test
T-Statistic: -3.5071164046858088
P-Value: 0.00045337282123505463

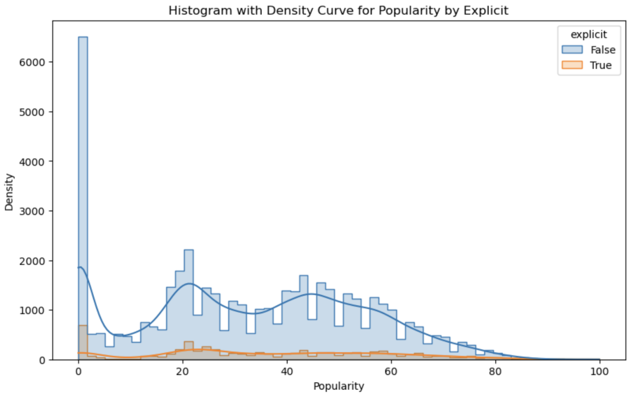
We use T-test for this question, with a t-statistic of -3.507 and a p-value of approximately 0.00045, provide strong statistical evidence to support the conclusion that there is a significant negative relationship between popularity and duration. The negative value of the t-statistic indicates that as duration increases, popularity tends to decrease, or vice versa. The very low p-value, which is well below the conventional threshold of 0.05, suggests that the probability of observing such a strong negative relationship by chance is extremely low. Therefore, we can assert that the observed negative correlation between popularity and duration is statistically significant. Shorter songs tend to have higher popularity.



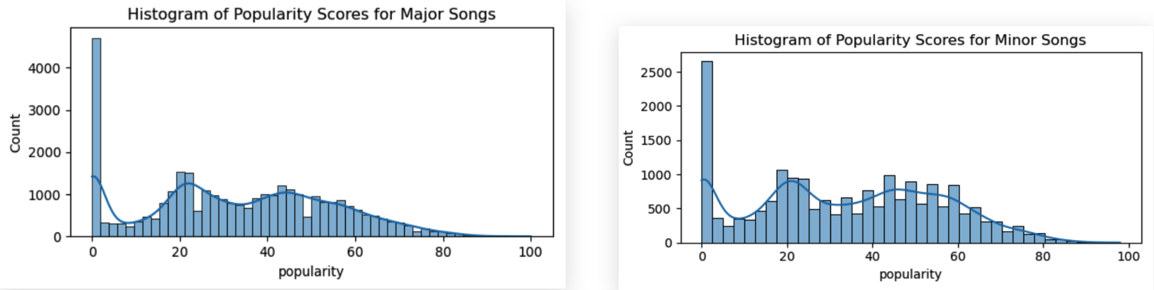
2. Are explicitly rated songs more popular than songs that are not explicit?
Model: T-test
T-Statistic: 9.833
P-Value: 4.250551719953934e-23

Based on the results of the T-test, with a t-statistic of 9.4955 and an extremely small p-value (approximately 1.48e-21). We conclude that the high t-statistic indicates a substantial difference in the mean popularity scores between the two groups, with explicit songs having higher popularity on average.

The p-value is far below any standard significance level (like 0.05 or 0.01), show that the probability of observing such a marked difference by chance is extremely low. This reinforces the strength of the evidence against the null hypothesis, which posited no difference in popularity between explicit and non-explicit songs.



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

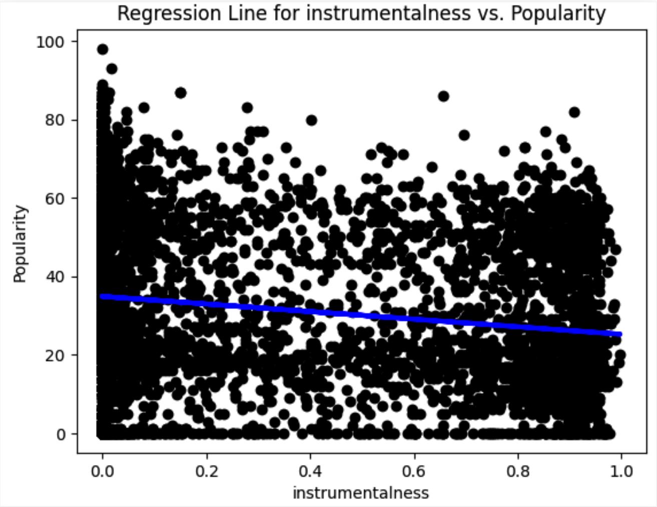


Model: Mann-Whitney U test
U-statistic: 309702373.0
P-value: 0.99999
Median for major song: 32.0
Median for minor song: 34.0

The histograms show the distribution of popularity scores for major and minor songs, which is not normal distribution. There is a higher concentration of lower scores and a tail extending towards the higher scores. The Mann-Whitney U test is a non-parametric test that does not assume normality in the data, thus, we use Mann-Whitney U test instead of T-test for this question.

The results show a p-value = 2.02e-06. indicates that there is a statistically significant difference in Spotify popularity scores between songs in major keys and songs in minor keys. Moreover, songs in a minor key have a slightly higher median popularity score compared to those in a major key, indicating that minor key songs tend to be more popular among Spotify listeners in this sample.

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?



Model: Simple linear regression model
RMSE of instrumentalness: 21.483
R2 of instrumentalness: 0.023
The best feature predicting popularity is: instrumentalness

We used simple linear regression models to evaluate every feature to predict song popularity. The feature 'instrumentalness' is the most significant predictor of song popularity. The R square value given is 0.023 and the RMSE is 21.483, which means that the proportion of the variance that can be accounted for by the model is only 0.023 and this single predictor can not fit the data well.

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?

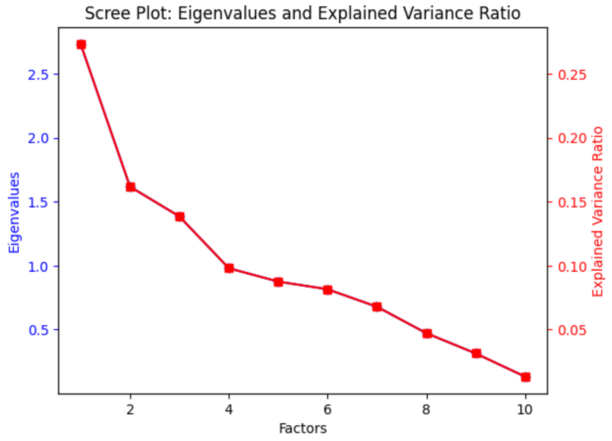
After building a multiple linear regression model with all 10 given features, the resulting RMSE is 21.15, which is actually a downgrade from the best model from question 4.

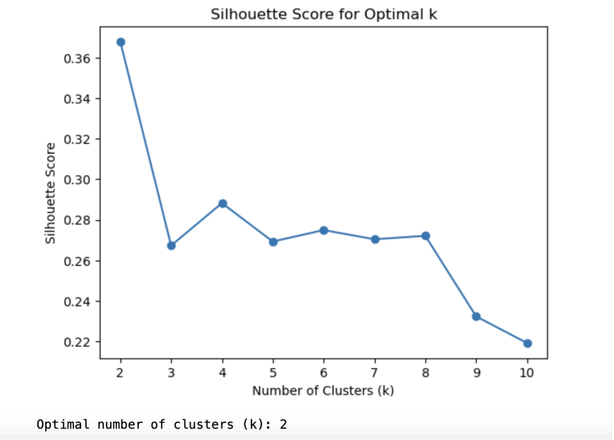
After regularization with the Ridge regression model (and comparing the alpha values from [0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0]), we found that the best alpha value was 1, and the resulting RMSE was 21.15, virtually no improvement at all. This can be explained by the fact that the original model is not overfitting before regularization.

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?

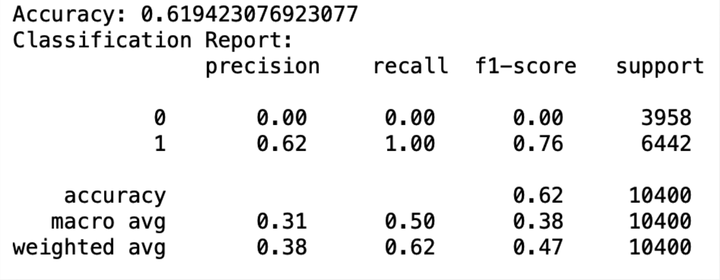
Using the explained variance ratio, we decided to keep 4 principle components. Together, these 4 account for ~70% of the variance. Using these components, we used the silhouette method (find the maximum sum of silhouette score with different k) and finally settled on 2 clusters. As can be seen from the scatterplot, these clusters are not satisfactory and do not reasonably correspond to the number of genre labels(52) in column 20 of the data, which may happen when different genres exhibit similar characteristics.





```
Optimal number of clusters (k): 2
```

7. Can you predict whether a song is in a 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?

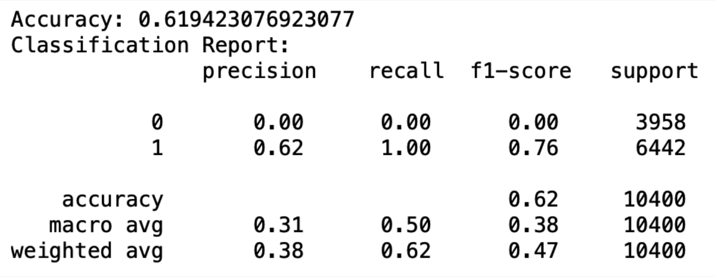


```
Accuracy: 0.6194230769230777
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      3958
           1       0.62      1.00      0.76      6442

    accuracy                           0.62     10400
   macro avg       0.31      0.50      0.38     10400
weighted avg       0.38      0.62      0.47     10400
```

We answer this question by applying a logistic regression model and SVM model.

For logistic regression model, we have an overall accuracy of 61.94%. It predicted the major key class (label 1) with 62% precision, which is reflective of the overall accuracy. However, we notice that the model completely failed to predict the minor key class (label 0), as indicated by 0% precision and recall for this category. The AUC is 0.5, which is about random guess.

The results indicate a critical imbalance in the model's predictive capability, heavily biased towards predicting songs as being in a major key regardless of their actual class.



```
Accuracy: 0.619423076923077
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      3958
           1       0.62      1.00      0.76      6442

    accuracy                           0.62     10400
   macro avg       0.31      0.50      0.38     10400
weighted avg       0.38      0.62      0.47     10400
```

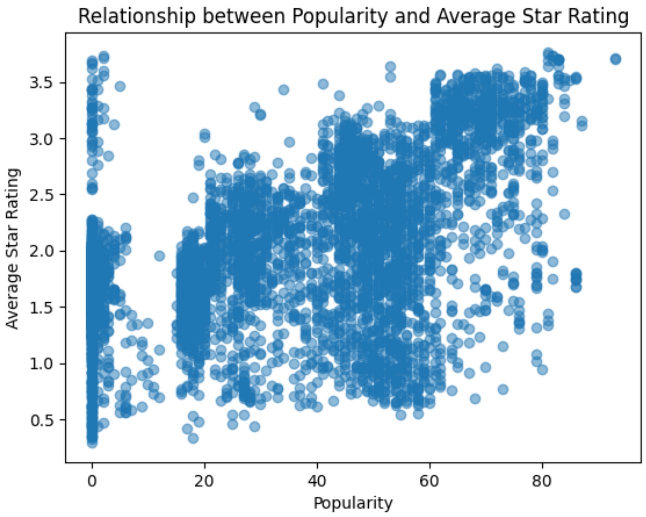
The SVM model achieved an accuracy of 61.94%. However it has the same problem: the model completely failed to identify any songs in the minor key (label 0), as evidenced by zero precision and recall for this class. In contrast, it predicted the major key (label 1) with a precision of 62% and a recall of 100%, indicating that it classified almost all songs as major. This implies that the single feature cannot product mode well.

8. Can you predict the genre by using the 10 song features from question 6 directly or the principal components you extracted in question 9 with a neural network? How well does this work?

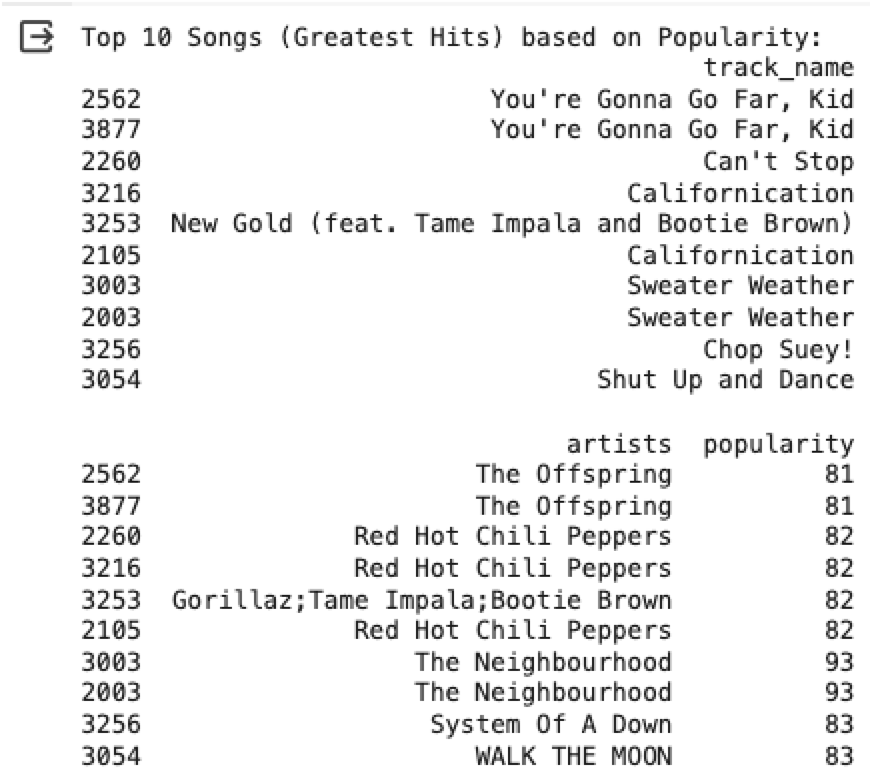
Before feeding the data into the model, preprocessing steps are applied. The categorical target variable, 'track_genre', is encoded using Label Encoding. Additionally, the feature values are standardized using StandardScaler to ensure uniform scales across variables. The neural network architecture is constructed using the Sequential model from Keras. It consists of three layers: an input layer with 64 neurons and ReLU activation, a hidden layer with 32 neurons and ReLU activation, and an output layer with a softmax activation function to handle multiclass classification, considering the unique genres in the dataset. The model is compiled with the Adam optimizer and sparse categorical cross entropy loss function. Subsequently, the model is trained on the training set for 20 epochs with a batch size of 32 and a validation split of 20%. Finally, the model is evaluated on the test set. For this multi-label task, the accuracy is 0.2789. After using OVR strategy(AUC calculates the current label vs all other 51 labels) and summarizing the performance across all classes equally, the unweighted mean of the AUC scores is 0.8869, which tells us that the neural network works well.

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 the 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), based on the popularity-based model?

a) For this part of the question, we first conducted element-wise removal of NaNs and then did a left join on the two datasets given, merging the two on track name so that the popularity scores of the first 5000 songs can be accessed. A scatter plot of the rudimentary distribution of the relationship can be seen as follows:



To investigate whether there is indeed correlation between popularity and average rating, we choose the Spearman correlation coefficient as a measure, since star ratings data does not necessarily assume linearity. The result is Spearman Correlation: 0.543 with a P-Value of nearly 0, which tells us that there is moderate positive correlation between popularity and average rating at the given significance level.



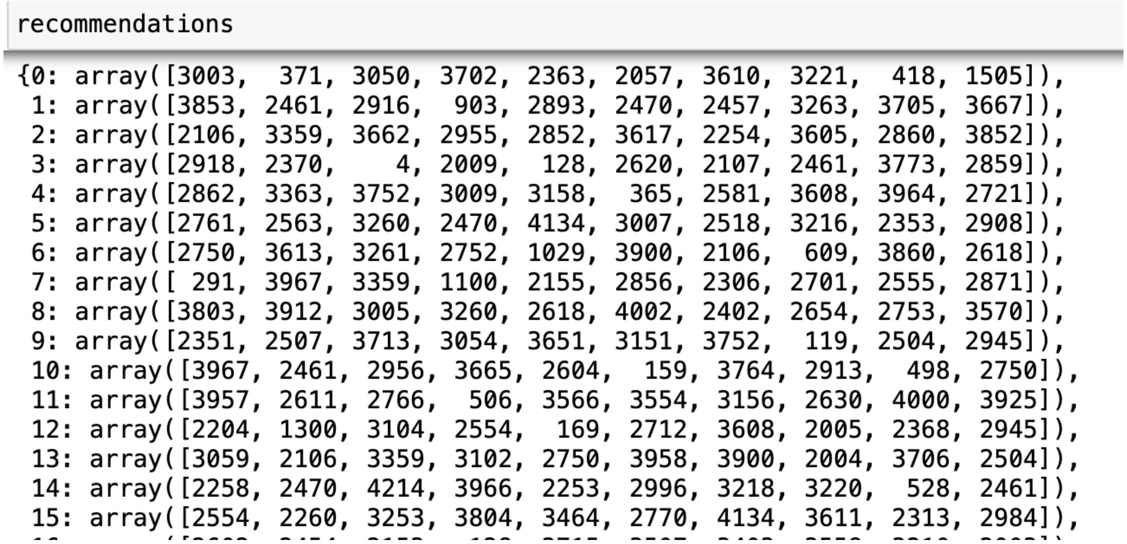
```
Top 10 Songs (Greatest Hits) based on Popularity:
                                         track_name
2562                         You're Gonna Go Far, Kid
3877                         You're Gonna Go Far, Kid
2260                                       Can't Stop
3216                                  Californication
3253  New Gold (feat. Tame Impala and Bootie Brown)
2105                                  Californication
3003                                  Sweater Weather
2003                                  Sweater Weather
3256                                       Chop Suey!
3054                               Shut Up and Dance

                                    artists  popularity
2562                          The Offspring          81
3877                          The Offspring          81
2260                  Red Hot Chili Peppers          82
3216                  Red Hot Chili Peppers          82
3253  Gorillaz;Tame Impala;Bootie Brown          82
2105                  Red Hot Chili Peppers          82
3003                       The Neighbourhood          93
2003                       The Neighbourhood          93
3256                        System Of A Down          83
3054                           WALK THE MOON          83
```

b) Based on the popularity model, which is ranking the top 10 most highly rated items in the rating dataset. It is worthy of noting that the results are highly correlated with the popularity scores assigned by Spotify, but there is not a 100% match. This tells us that in general most highly rated songs will be popular ones, but not all popular songs will be highly rated.

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?

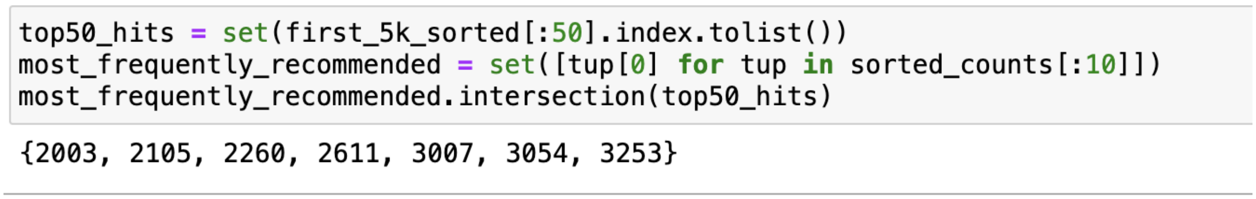
The ratings dataset was preprocessed with handling missing values element-wise. It was switched into a dataframe that contains user_ids, movie_ids and their mapping ratings as three different columns, from which we created a sparse user_item matrix. Then we trained an ALS model from the matrix and generated movie recommendations for a specific user using the implicit library. We set the factors size as 50, which determines the number of the latent factors in the matrix factorization. Besides, we use a regularization strength of 0.01, and undergo 50 iterations during the optimization process.

To evaluate the performance of our model, we choose the mean average precision, calculating the mean value of average precision for all the 10000 users. Our MAP is 0.3445, which indicates that we create a good personal mixtape recommender system. From the intersection of the top 50 hits using the popularity rank and the most 10 frequently recommended movies for all the 10k user, we conclude that it is more likely for a popular movie to be recommended for a user, but our algorithm also includes some personal preferences based on the latent factors learned from the ratings data.



```
recommendations
```

```
{0: array([3003,  371, 3050, 3702, 2363, 2057, 3610, 3221,  418, 1505]),
 1: array([3853, 2461, 2916,  903, 2893, 2470, 2457, 3263, 3705, 3667]),
 2: array([2106, 3359, 3662, 2955, 2852, 3617, 2254, 3605, 2860, 3852]),
 3: array([2918, 2370,    4, 2009,  128, 2620, 2107, 2461, 3773, 2859]),
 4: array([2862, 3363, 3752, 3009, 3158,  365, 2581, 3608, 3964, 2721]),
 5: array([2761, 2563, 3260, 2470, 4134, 3007, 2518, 3216, 2353, 2908]),
 6: array([2750, 3613, 3261, 2752, 1029, 3900, 2106,  609, 3860, 2618]),
 7: array([ 291, 3967, 3359, 1100, 2155, 2856, 2306, 2701, 2555, 2871]),
 8: array([3803, 3912, 3005, 3260, 2618, 4002, 2402, 2654, 2753, 3570]),
 9: array([2351, 2507, 3713, 3054, 3651, 3151, 3752,  119, 2504, 2945]),
 10: array([3967, 2461, 2956, 3665, 2604,  159, 3764, 2913,  498, 2750]),
 11: array([3957, 2611, 2766,  506, 3566, 3554, 3156, 2630, 4000, 3925]),
 12: array([2204, 1300, 3104, 2554,  169, 2712, 3608, 2005, 2368, 2945]),
 13: array([3059, 2106, 3359, 3102, 2750, 3958, 3900, 2004, 3706, 2504]),
 14: array([2258, 2470, 4214, 3966, 2253, 2996, 3218, 3220,  528, 2461]),
 15: array([2554, 2260, 3253, 3804, 3464, 2770, 4134, 3611, 2313, 2984]),
```



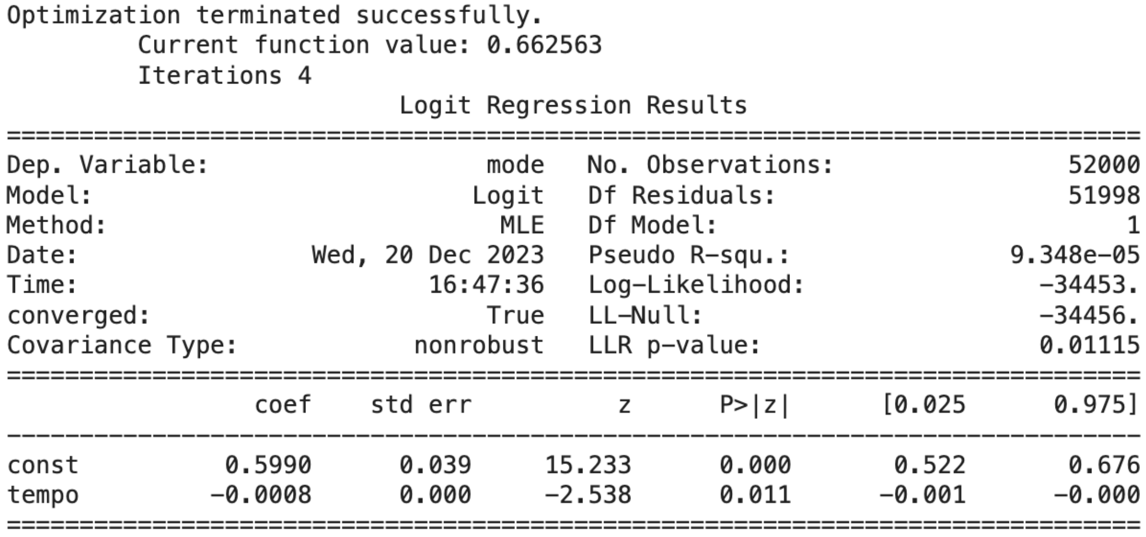
```
top50_hits = set(first_5k_sorted[:50].index.tolist())
most_frequently_recommended = set([tup[0] for tup in sorted_counts[:10]])
most_frequently_recommended.intersection(top50_hits)
```

```
{2003, 2105, 2260, 2611, 3007, 3054, 3253}
```

11. Extra Credit: Are songs with a higher tempo more likely than songs with a lower tempo to be in major key?

The rationale behind investigating this relationship lies in the potential connection between musical characteristics (specifically tempo) and the tonality of a song. Understanding this relationship can provide insights into the patterns of musical composition and contribute to our knowledge of how tempo might influence the emotional or aesthetic qualities of a piece.

The logistic regression model was built using a binary dependent variable (mode), where 1 represents a major key, and 0 represents a minor key. The independent variable was tempo, and the logistic regression results indicate that there is a statistically significant association between tempo and the likelihood of a song being in a major key. The coefficient for the tempo variable is -0.0008, with a p-value of 0.011. This suggests that, holding other factors constant, as the tempo increases, the odds of a song being in a major key decrease. The pseudo R-squared value is very small (9.348e-05), indicating that the model explains a minimal amount of variability in the mode variable. The Wald test for the tempo variable (LLR p-value: 0.01115) further supports the significance of the tempo in predicting the likelihood of a song being in a major key. Overall, the findings suggest a modest but statistically significant relationship between tempo and the tonality of a song.



```
Optimization terminated successfully.
         Current function value: 0.662563
         Iterations 4
                           Logit Regression Results                           
==============================================================================
Dep. Variable:                   mode   No. Observations:                52000
Model:                          Logit   Df Residuals:                    51998
Method:                           MLE   Df Model:                            1
Date:                Wed, 20 Dec 2023   Pseudo R-squ.:               9.348e-05
Time:                        16:47:36   Log-Likelihood:                -34453.
converged:                       True   LL-Null:                       -34456.
Covariance Type:            nonrobust   LLR p-value:                   0.01115
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.5990      0.039     15.233      0.000       0.522       0.676
tempo         -0.0008      0.000     -2.538      0.011      -0.001      -0.000
==============================================================================
```