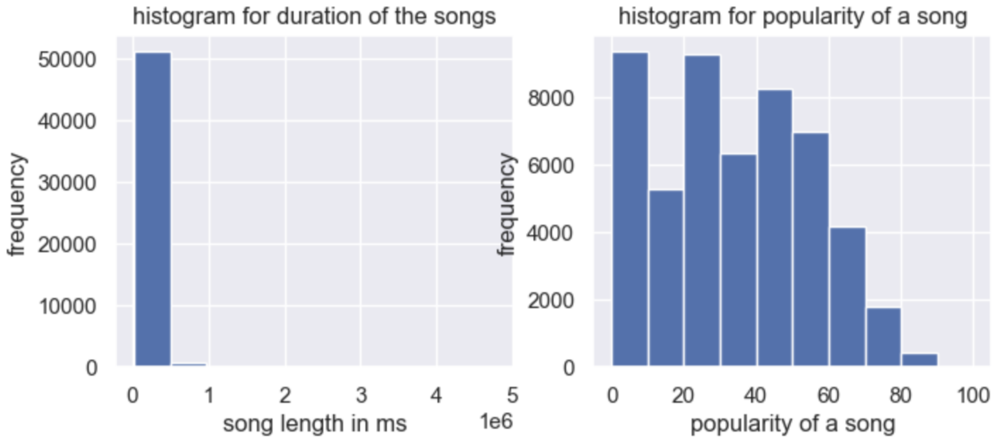
Group DS14
Erchi Zhang, Joz Zhou

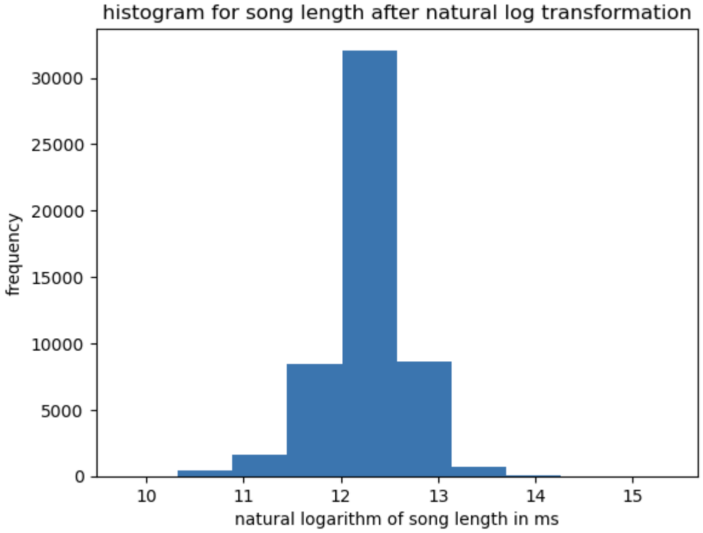
Capstone Project

Project link on github:https://github.com/Archertakesitez/capstone_project_1001
This is Joz Zhou and Erchi Zhang from group DS14. We splitted work equally, each focusing on tackling half of the problems. During the process, we used ChatGPT for code fixing as well as generating some helpful insights of our data. Random seed was set to 14276662 (N-ID of Joz Zhou). In our initial assessment of the "Spotify 52K Songs" dataset, we observed a high level of data integrity and quality. The dataset was devoid of missing values and duplicate records (that we will discuss later), so we did not encounter many issues while working on it. Furthermore, most of the numerical and categorical variables present appropriate data types and distributions that did not necessitate universal preprocessing or transformation. Although some of the variables are skewed and may need normalization as well as standardization for later use, given the dataset's robustness in general, we decided to address data manipulation on a case-by-case basis, tailored to the specific requirements of each question. This approach allows us to preserve the dataset's original integrity and ensures that any modifications are contextually relevant and purpose-driven. However, for the “starRatings.csv” file, we manually extracted the first 5000 songs and assigned them as column names for the data frame for more data clarity. In user ratings datasets, there are a lot of missing values, but we think a missing value typically indicates that a user has not rated a particular song. This is different from a missing value due to data collection errors. Imputing such missing values could introduce significant bias, as we'd been assuming a level of preference (or lack thereof) that the user has not explicitly expressed, and therefore, we choose not to impute the dataset universally.

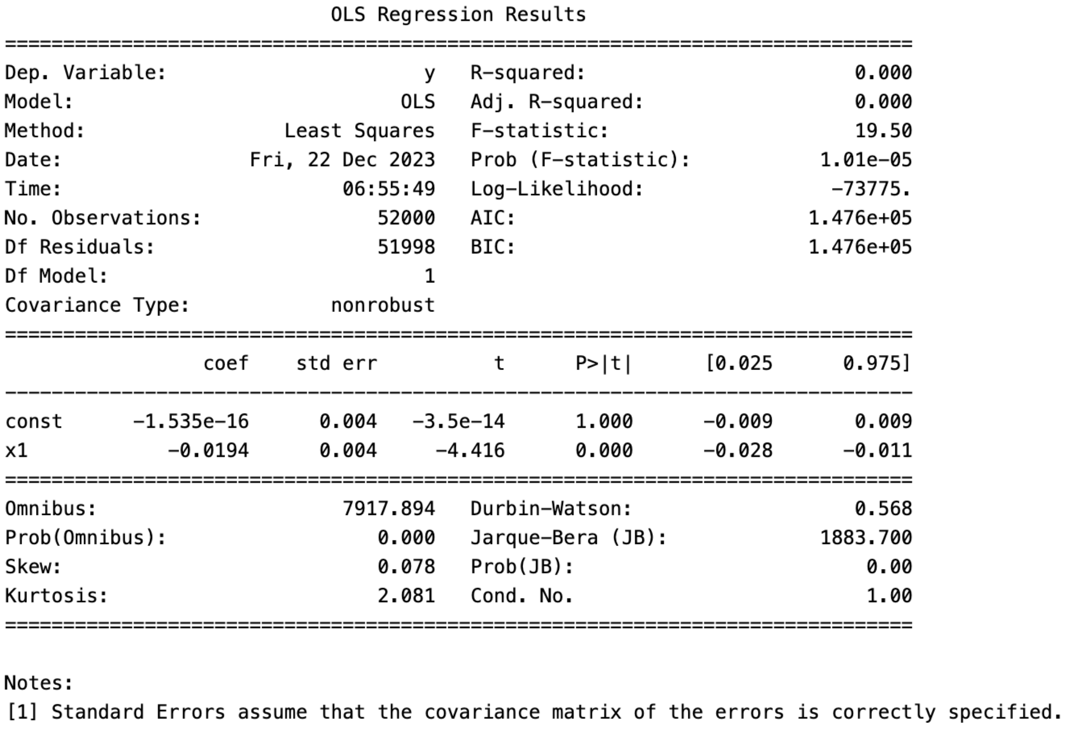
Q1:
For this question, song length here is associated with the ‘duration’ column, and popularity of a song is associated with the ‘popularity’ column from ‘spotify52kData.csv’ file.
First, we checked if there are any missing values in the ‘popularity’ and ‘duration’ column, and it turned out that there are no missing values in any of these columns. Hence, we do not need to drop rows or do imputations for handling missing values. Then, we plotted the histograms for both the song length (‘duration’) and popularity of a song (‘popularity’), shown as below.



From these graphs, we have noticed that the song lengths data are highly skewed. We calculated its skewness and the result is 11.57, indicating that its distribution is extremely positively skewed. Hence, we performed natural logarithmic transformation on it, reducing its skewness to -0.39. The histogram of song length data after natural logarithmic transformation is shown below.



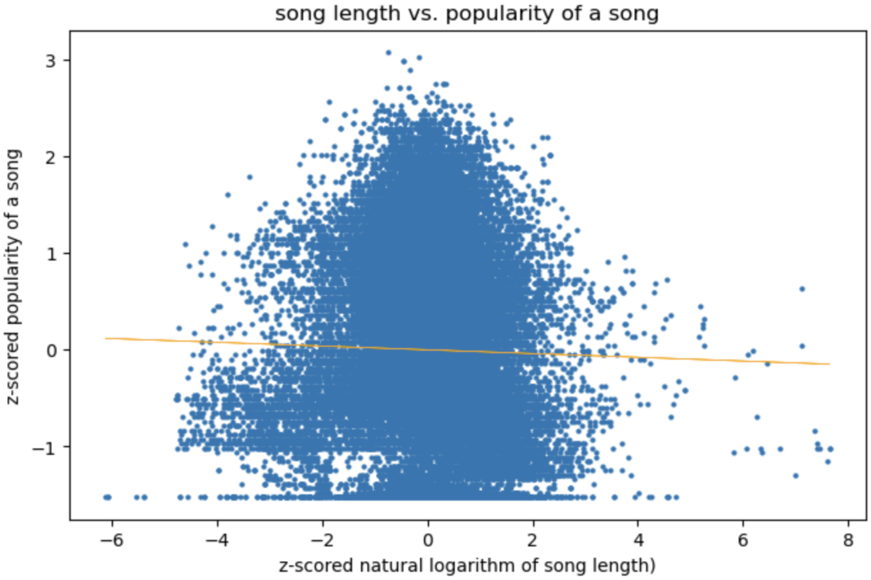
Then, we have standardized the log-transformed song length and popularity of a song by z-scoring, so that they are more comparable and interpretable, making both variables more suitable for the collinearity test. We ran a simple linear regression for these z-scored variables using the statsmodels package, and determined their relationship by the returned variable coefficient and its associated p-value indicating the significance of the collinearity. Below is the summary of our linear regression model.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     19.50
Date:                Fri, 22 Dec 2023   Prob (F-statistic):           1.01e-05
Time:                        06:55:49   Log-Likelihood:                -73775.
No. Observations:               52000   AIC:                         1.476e+05
Df Residuals:                   51998   BIC:                         1.476e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -1.535e-16      0.004   -3.5e-14      1.000      -0.009       0.009
x1            -0.0194      0.004     -4.416      0.000      -0.028      -0.011
==============================================================================
Omnibus:                     7917.894   Durbin-Watson:                   0.568
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1883.700
Skew:                           0.078   Prob(JB):                         0.00
Kurtosis:                       2.081   Cond. No.                         1.00
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

According to the table, the coefficient of x1 (song length) is -0.02 which is negative, and its associated p-value is 0, which is smaller than our alpha of 0.05, in favor of the alternative hypothesis that there is a statistically significant relationship between song length and popularity of a song.

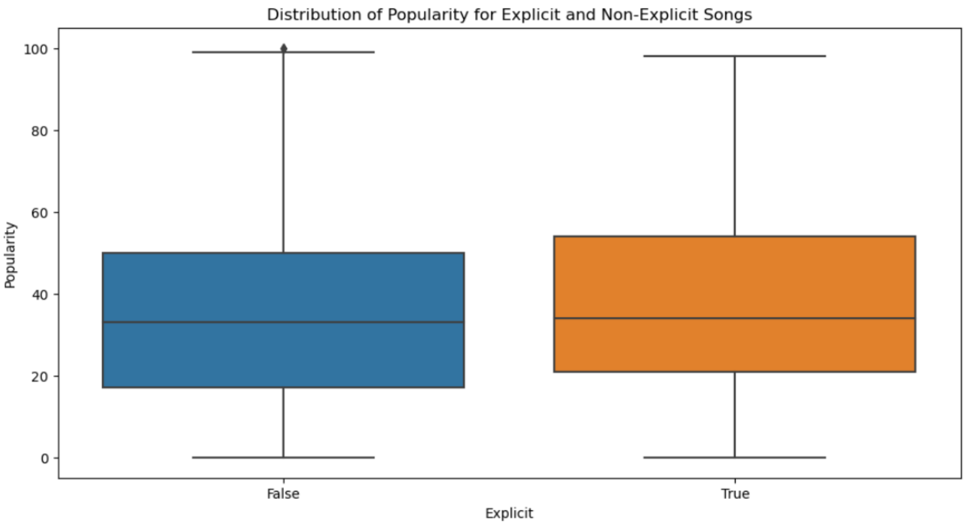
We have then made a scatter plot for the z-scored natural logarithm of song length vs. z-scored popularity of the song and drawn the linear regression line on it, shown below.



According to the graph, the negative linear relationship between song length and popularity of the song is notable. As a result, we can conclude that there is a significant negative linear relationship between song length and popularity of a song i.e. when song length increases, the popularity decreases.

Q2:

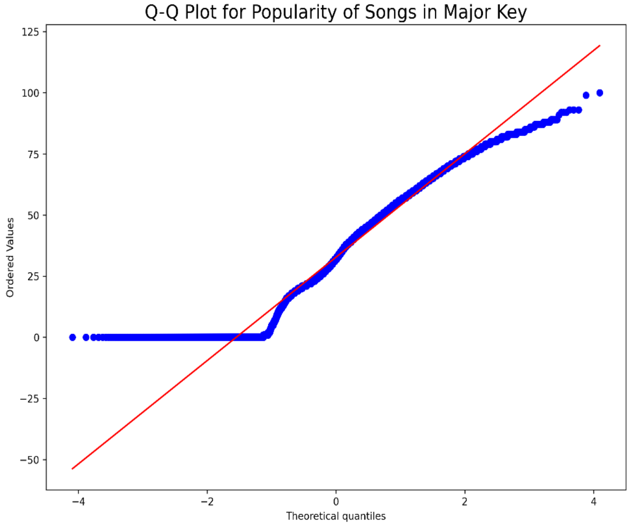
In our analysis to determine whether explicitly rated songs are more popular than non-explicit songs on Spotify, we conducted Welch's t-test, which does not assume equal variances since our Levene’s test for homogeneity of variance gives us a statistically significant p-value which indicates that the variances in the two groups are not equal. Considering the other two assumptions of independence and normality, we think that it is satisfied as the two different groups should not have dependent ratings in essence, and we also relied on the Central Limit Theorem for normality given that we have a relatively large sample size. We also visualized a boxplot that illustrates the distribution of popularity scores for both explicit and non-explicit songs. It seems that explicit songs might have a slightly higher median popularity, but the difference is not very pronounced.

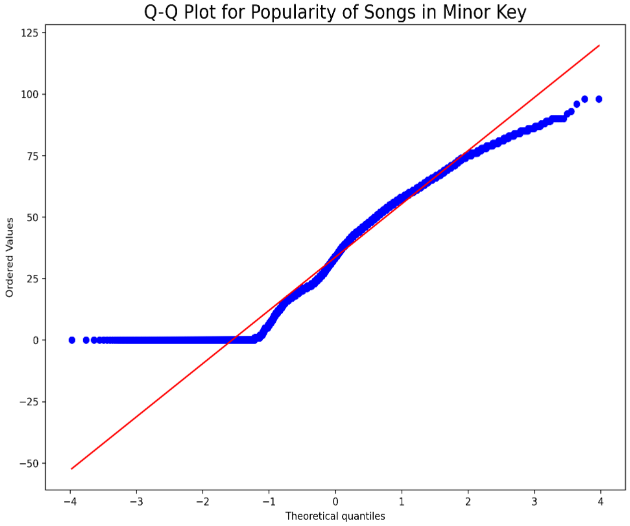


The actual test(explicit vs non-explicit) resulted in a statistically significant difference in mean popularity scores (t-statistic ≈ 9.50, p-value ≈ 2.95e−21), and we would therefore reject our null that there is no statistically significant difference in mean popularity scores, indicating that explicit songs are indeed more popular than non-explicit songs.

Q3:

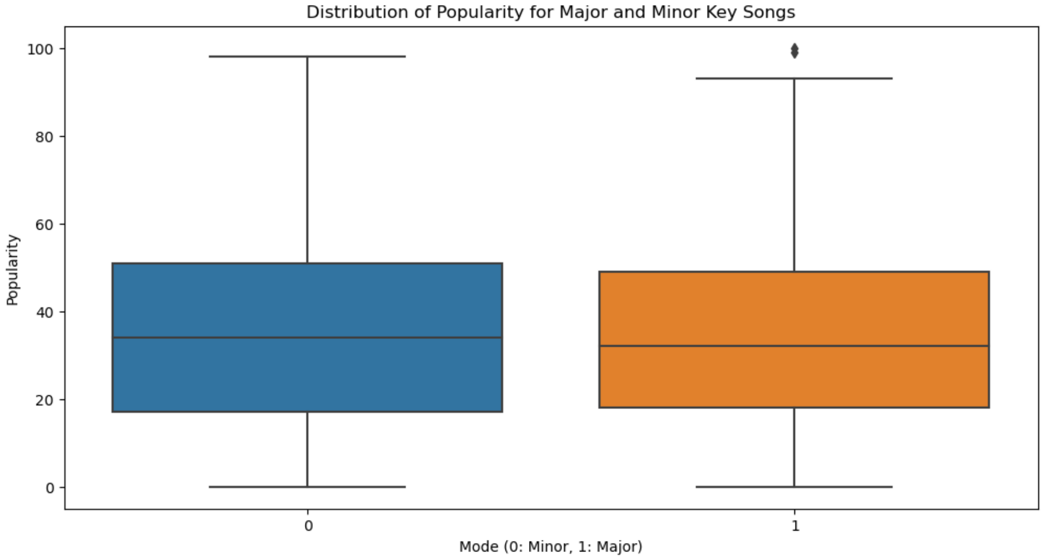
In our investigation to assess whether songs in a major key are more popular than those in a minor key on Spotify, we employed Welch's t-test again, appropriate for scenarios with unequal variances between groups. This approach was chosen based on the outcome of Levene's test for homogeneity of variance, which yielded a significant p-value (6.42e−13), indicating a disparity in variances between the major and minor key song groups. For the assumptions of independence and normality: the independence assumption holds as the songs in major and minor keys are distinct and unrelated groups by nature; for normality, while the Central Limit Theorem supports our assumption due to the large sample size, we further validated it through Q-Q plots:





These plots revealed that both major and minor key songs' popularity distributions approximately followed a normal distribution, especially in their central parts, despite some deviations at the tails.

Given the large sample size, the minor deviations from normality seen in the Q-Q plots are not overly concerning for the application of the robust t-test. We also plotted distributions of popularity scores for both major and minor key songs using a boxplot which provides a comparative view of the two groups but does not indicate a stark difference in their median popularity :

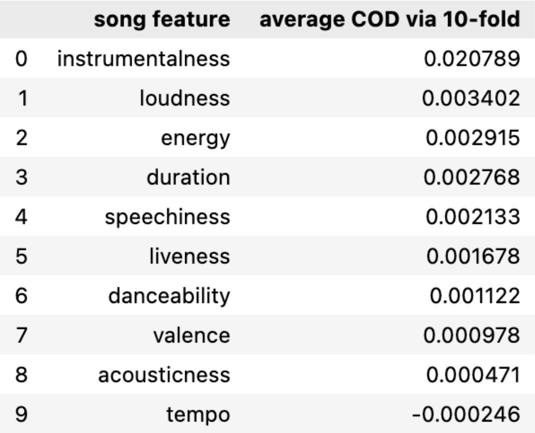


The one-sided Welch's t-test comparing the popularity of songs in major versus minor keys showed a statistically significant difference (t-statistic ≈ -4.79, p-value ≈ 0.9999991). Based on this result, we fail to reject the null hypothesis that major key songs are not more popular than minor key songs. The negative value of the t-statistic(and also the approximately 0 p-value for the two-sided test) indicates that, in fact, songs in a minor key tend to be more popular than those in a major key. This finding is quite counterintuitive in our opinion, offering an interesting perspective on musical preferences as reflected in the Spotify dataset.

Q4:

For this question, we first checked if there are any missing values within the columns of the given features, and it turned out there are no missing values, which means we do not have to do missing values handling. Then, we iterate through the ten columns associated with the ten given features, and for each of the column's values, we do a simple linear regression of feature column vs. popularity column with 10-fold cross validation. We do not normalize the variables because linear regression does not assume normal distribution. We also do not perform transformation on data to reduce skewness, because we have noticed that the average R^2 resulting from 10-fold cross validation of linear regression of some highly skewed columns have decreased after we transformed its data from skew to symmetric. For example, we observed that song length is highly positively skewed with a skewness of 11.57, so we applied natural log transformation to it and reduced its skewness to -0.39. However, it turned out that although its skewness decreased by 11.96, its average R^2 from 10-fold cross validation of linear regression dropped from 0.0028 to 0.0001, which means our transformation has harmed the regression model prediction result. Hence, to ensure fairness, we are not performing skewness reduction for any variables.

In each iteration, we have calculated the average COD (R^2) of the 10-fold cross validation for the simple linear regression model for evaluation purposes. We sorted the obtained 10 average COD values from the highest to the lowest and made a table of it, as shown below.



	song feature	average COD via 10-fold
0	instrumentalness	0.020789
1	loudness	0.003402
2	energy	0.002915
3	duration	0.002768
4	speechiness	0.002133
5	liveness	0.001678
6	danceability	0.001122
7	valence	0.000978
8	acousticness	0.000471
9	tempo	-0.000246

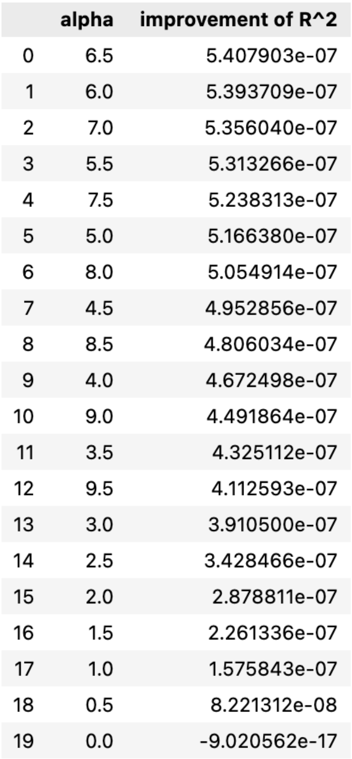
Higher COD means higher variance that can be accounted for by our model, and thus means predicting popularity better in this case. From the table we can conclude that the feature ‘instrumentalness’ predicts popularity the best, as the simple linear regression model associated with it has the highest average COD value of 0.021——which means 2.1% variance can be accounted for by our model.

Even though ‘instrumentalness’ predicts popularity the best, its average COD value of 0.021 is still very low, as an acceptable COD value should be greater than 0.5 in general. Hence, the model associated with only using ‘instrumentalness’ as the single predictor is not very good.

Q5:

We applied a multiple linear regression model to use all ten features in question 4 to predict popularity, using 10-fold cross validation and average R^2 for evaluation. The mean of R^2 from 10-fold cross validation is 0.04715, showing that this model cannot predict popularity very well, as only 4.7% variance can be accounted for by our model. Nonetheless, it is still much better than the models from question 4, as its average R^2 of 0.04715 is 0.026 more than the average of R^2 from the 10-fold cross validation of the best predictor(instrumentalness) in question 4, achieving a 126.83% improvement. This dramatic improvement can be accounted for by the fact that more than one predictor matters (i.e. song length is not the only predictor among the ten features that is related to popularity), and not all of the 10 features are correlated. Hence, when using all of the ten features, as more of the predictors that influence popularity are added, we can predict the popularity of a song better.

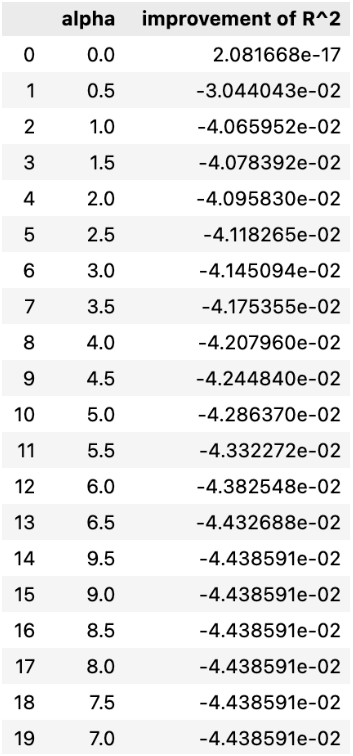
Then, we regularize our model via ridge regression. We iterate over different alpha values (0 to 10 with step of +0.5) and in each iteration, we perform ridge regression using that alpha value, and run a 10-fold cross validation and compute its average R^2. The 10-fold cross validation aids us to obtain the optimal alpha within the range we set. We recorded each alpha and its associated average R^2 as well as how much the R^2 has increased compared to the model without regularization. We then sort the alpha based on their improvement of R^2. A table showing the alphas that improved R^2 the most to the least is shown below.



	alpha	improvement of R^2
0	6.5	5.407903e-07
1	6.0	5.393709e-07
2	7.0	5.356040e-07
3	5.5	5.313266e-07
4	7.5	5.238313e-07
5	5.0	5.166380e-07
6	8.0	5.054914e-07
7	4.5	4.952856e-07
8	8.5	4.806034e-07
9	4.0	4.672498e-07
10	9.0	4.491864e-07
11	3.5	4.325112e-07
12	9.5	4.112593e-07
13	3.0	3.910500e-07
14	2.5	3.428466e-07
15	2.0	2.878811e-07
16	1.5	2.261336e-07
17	1.0	1.575843e-07
18	0.5	8.221312e-08
19	0.0	-9.020562e-17

As a result, the optimal alpha is 6.5, by which average R^2 has increased by 5.4e-07, or 0.0011% when comparing to the multiple regression model without regularization, and this is a very slight increase. In conclusion, after regularizing the model via ridge regression, the average R^2 from 10-fold cross validation has extremely slightly increased.

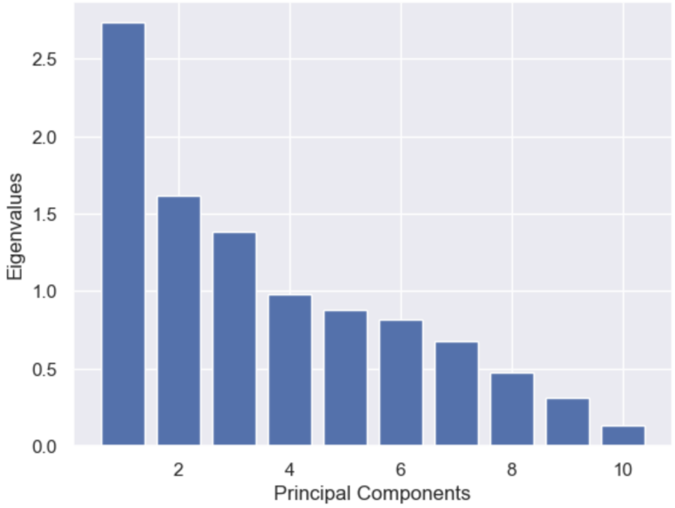
We have also applied Lasso regression to regularize our model, but it turned out its average R^2s from 10-fold cross validation with the alpha between 0 to 10 with +0.5 step are all lower than the average R^2 without applying any regularization. Hence, regularizing using Lasso regression makes our model predictions worse. A table with different alpha values associated with improvement of R^2 is shown below.



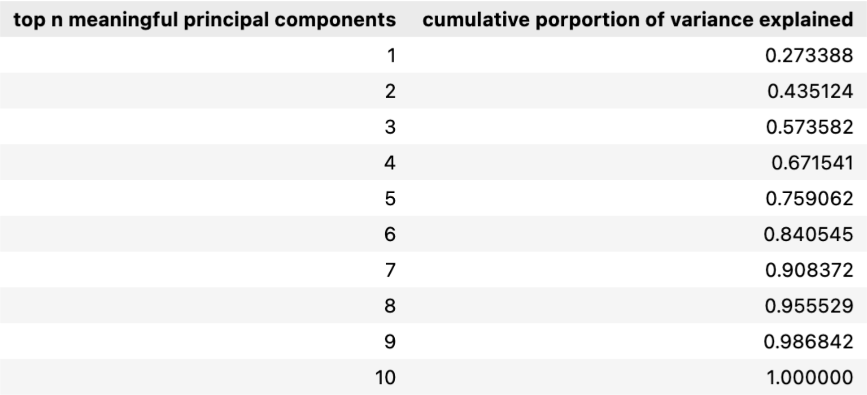
	alpha	improvement of R^2
0	0.0	2.081668e-17
1	0.5	-3.044043e-02
2	1.0	-4.065952e-02
3	1.5	-4.078392e-02
4	2.0	-4.095830e-02
5	2.5	-4.118265e-02
6	3.0	-4.145094e-02
7	3.5	-4.175355e-02
8	4.0	-4.207960e-02
9	4.5	-4.244840e-02
10	5.0	-4.286370e-02
11	5.5	-4.332272e-02
12	6.0	-4.382548e-02
13	6.5	-4.432688e-02
14	9.5	-4.438591e-02
15	9.0	-4.438591e-02
16	8.5	-4.438591e-02
17	8.0	-4.438591e-02
18	7.5	-4.438591e-02
19	7.0	-4.438591e-02

Q6:

Firstly, we z-scored the data to standardize our dataset features on a unit scale with mean = 0 and standard deviation = 1. This is necessary before we implement PCA, because PCA is sensitive to the scale of features and thus assumes the input data to be standardized. Then, we applied PCA to the z-scored data and plotted a graph of principal components with their associated eigenvalues, shown as below.



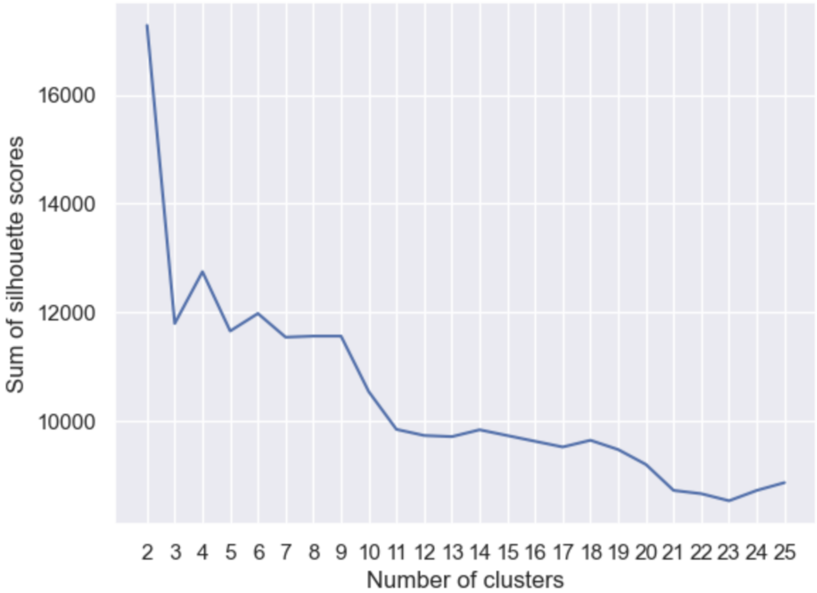
Assume at least 70% variance should be explained by our selected meaningful principal components. We make a table for the principal components and their associated cumulative proportion of variance explained, in the order of their proportion of variance explained from the greatest to the lowest, as shown below.



top n meaningful principal components	cumulative porportion of variance explained
1	0.273388
2	0.435124
3	0.573582
4	0.671541
5	0.759062
6	0.840545
7	0.908372
8	0.955529
9	0.986842
10	1.000000

According to the table, we extract 5 meaningful principal components, as the top 5 meaningful principal components explain 0.759, or 75.9% of total variance. We transformed our z-scored data using the top 5 meaningful principal components.

We have determined to use the K-means clustering method, because it is simple, fast, and we assume our data has 'spherical clusters', as song data do not usually have an uncommon shape such as spiral. We managed to find the optimal number of clusters by applying silhouette method on the transformed data, given that the greatest sum of silhouette scores is associated with the optimal number of clusters. We looped over different numbers of clusters—from 2 to 25—and plotted the sum of silhouette scores by different numbers of clusters, shown below.



According to this graph, as 2 clusters give the greatest sum of silhouette scores, the optimal number of clusters is two.

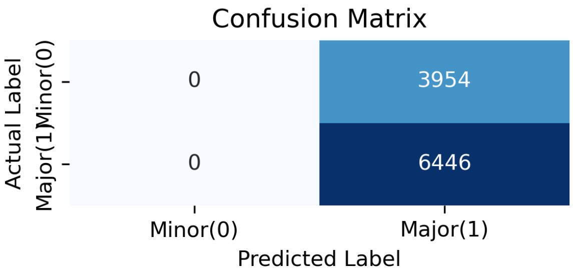
We then performed k-means clustering with n_clusters = 2 for the transformed data, which divided the data into two clusters. To determine whether the clusters reasonably correspond to the genres labels at column 20 from the dataset, we firstly encoded the genre labels (column 'track_genre') from strings to numeric values. There are 52 unique genres in total, so the numeric values associated with the unique genres are 0 to 51. Then, we counted the frequencies of each genre appearing in each of the two clusters given by the k-means (making two frequency lists, each associated with one cluster), and applied a chi-squared test for this two groups of frequencies to see whether the two clusters given by the k-means are really two different clusters, that is, representing two significantly different “big” genres. The null hypothesis is that two groups of data have the same frequencies of each category. The reason we chose the chi-squared test to spot the two clusters’ differences is that we need a significant test on categorical data, and the chi-squared test is designed for categorical data—it can compare two groups of data consisting of frequencies of each category. Moreover, the frequencies would not be too small (lower than 13) for each category in our two clusters, so we have satisfied the requirements for using chi-squared test. Since chi-squared test requires the sum of frequencies to be the same for the two input frequency lists, and our sums of frequencies of each frequency list are not the same, we have divided the values of the frequency list of one cluster by its sum of frequencies, and then multiplied it by the sum of frequencies of another cluster’s frequency list. In mathematics, it can be shown as:

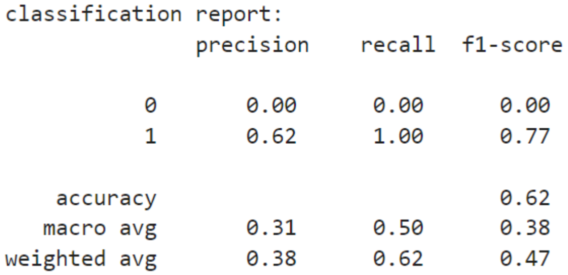
```
frequencies from cluster x = (frequencies from cluster x)*(sum of frequencies from cluster y)/(sum of frequencies from cluster x)
```

By this technique, the two frequency lists achieved the same sum of frequencies, and thus can be used for the chi-squared test. As a result, we have obtained a p-value of 0.0 from the chi-squared test, rejecting the null hypothesis that our two clusters have the same relative frequencies for the categories. This shows that our two clusters are associated with two significantly different groups of data, differing by genre labels. Hence, we can conclude that our two clusters from k-means do reasonably correspond to the genre labels in column 20 of the data.

Q7:

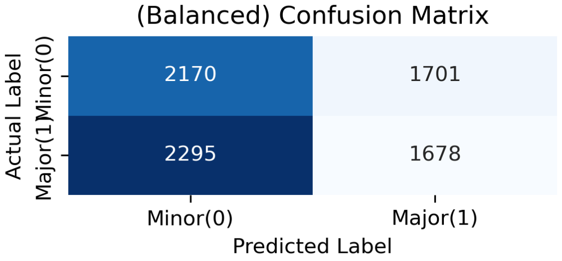
In this analysis, we aimed to predict whether a song is in a major or minor key based on its valence initially using logistic regression. After selecting 'valence' as the predictor and 'mode' as the target variable from the Spotify dataset, we split the data into training and testing sets, ensuring randomization with the specified seed. The logistic regression model was then trained on the training set and its performance was evaluated on the test set, revealing an accuracy of approximately 62%. While we see a good recall of 100% for class 1, the recall for class 0(or specificity in a binary classification setting) is precisely at 0%. This is because if we look at the confusion matrix, our model never predicts class 0.

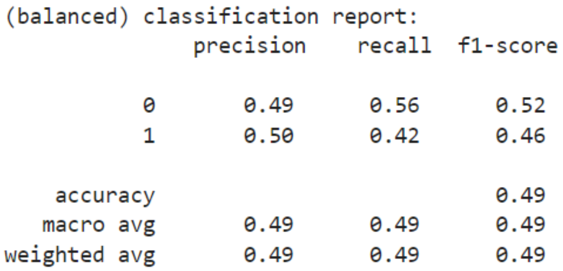




```
classification report:
              precision    recall  f1-score

           0       0.00      0.00      0.00
           1       0.62      1.00      0.77

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

The classification results suggest that while the model has an overall moderate accuracy, it is heavily biased and not effectively distinguishing between the two classes. This bias was likely due to a class imbalance we observed in the dataset, where approximately 62% of songs were in the major key, and only about 38% were in a minor key. To address this issue, we employed an under-sampling strategy, where we randomly reduced the number of major key songs in the dataset to match the number of minor key songs. After under-sampling, we trained a new logistic regression model on this balanced dataset with the results below:



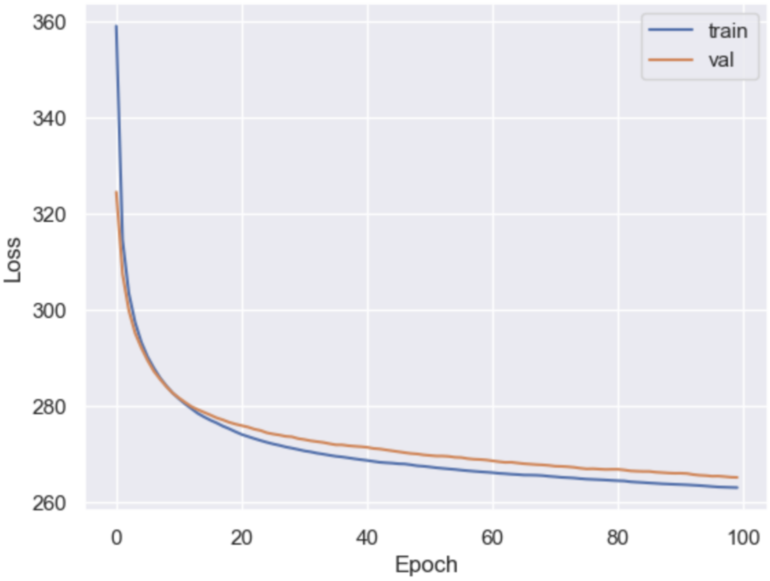

```
(balanced) classification report:
              precision    recall  f1-score

           0       0.49      0.56      0.52
           1       0.50      0.42      0.46

    accuracy                           0.49
   macro avg       0.49      0.49      0.49
weighted avg       0.49      0.49      0.49
```

The new model achieved an accuracy of approximately 49%, a decrease from the initial model's accuracy of 62%. This drop in accuracy indicates that while the model was no longer biased towards the majority class, its overall predictive capability was only comparable to random guessing. The precision and recall for both classes were roughly equal, demonstrating that the model was now treating both major and minor key songs more equitably. However, the values of these metrics suggested that the model's ability to correctly classify songs based on valence alone was limited. We also attempted using other models on our balanced dataset, even though some showed improvements like 53% accuracy when using the random forest model, it is still only slightly better than just randomly guessing. Our results suggested that our current model's reliance on valence alone as a predictor might be insufficient, and maybe incorporating additional features could potentially improve the model's predictive capabilities.

Q8:

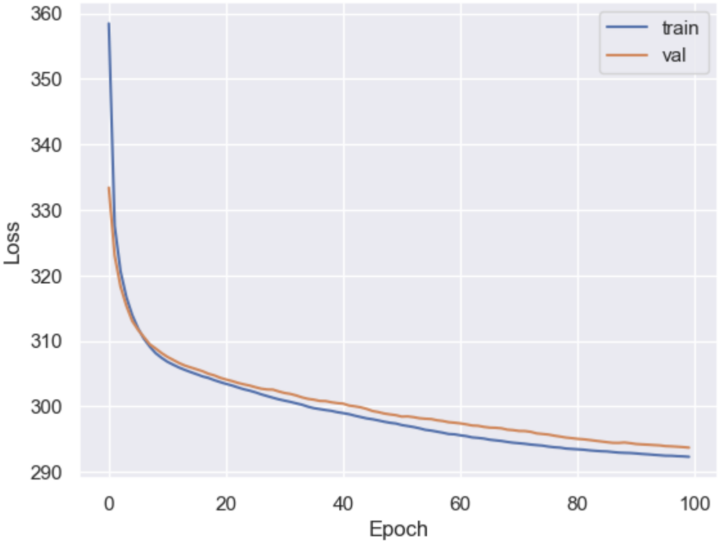
Firstly, we build a MLP, which is a fully-connected feedforward network built from linear units and activation functions. It learns through backpropagation via stochastic gradient descent, and can approximate nonlinear functions. Since we need to predict genres, this neural network is good for us as it introduces nonlinearity and can thus predict categorical data. We used cross-entropy loss instead of L2 loss because cross-entropy loss is generally less prone to overfitting in classification settings. We use the encoded numeric values for the genre labels generated from Question 6 as our dependent variables (y). While training our models, we set the number of epochs to be 100, learning rate to be 0.0005, and batch_size to be 100. These hyperparameters make our model work well. We set the number of classes to be 52, as there are 52 genres in total.

We first managed to train the model using the 10 song features from question 4 directly. Before making these 10 features as our independent variables (x), we apply z-scoring to standardize them, as mapping our data to a standardized scale improves the performance of our neural network, making it more stable during learning. We set the number of features to be 10 because there are 10 features in total. After training and evaluating our model, we plotted its learning curve as below.



We can tell from the graph that both training loss and validation loss decreases to stability while epoch gets greater, which indicates our model has a good fit. The accuracy of our trained model on validation data is 0.252, meaning 25.2% predictions are correct. It seems low, but given there are 52 labels in total, 25.2% accuracy is an acceptable performance.

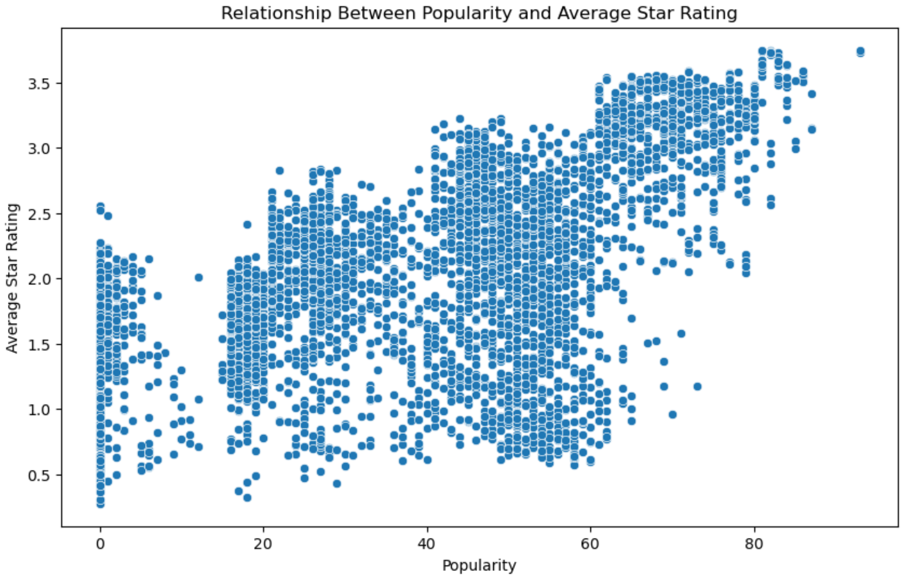
Then, we trained the model using the 5 principal components we extracted in question 6. Since the data associated with those principal components are already z-scored before transformation, we do not have to standardize them again. We set the number of features to be 5 since there are 5 principal components in total. After training and evaluating our model, we plotted its learning curve as below.



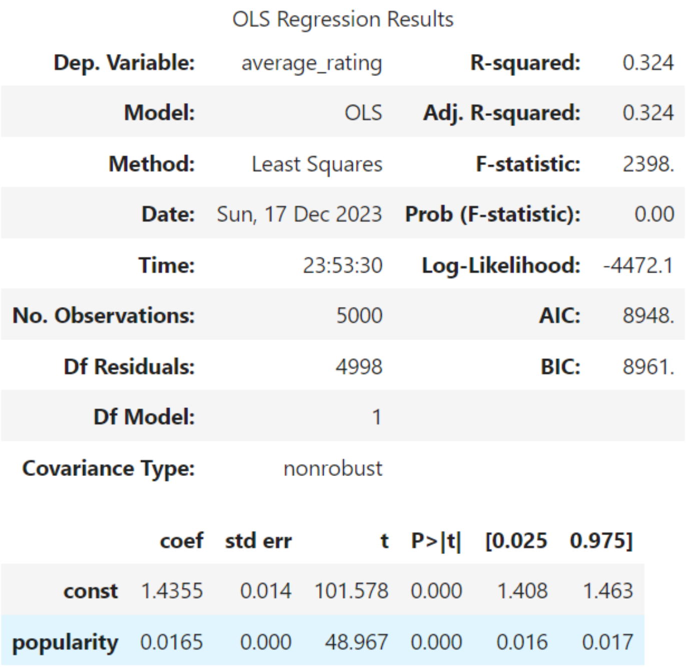
We can tell from the graph that both training loss and validation loss decreases to approach stability while epoch gets greater, but they do not become as stable as the learning curve from the previous graph (predicting with 10 song features), and both training loss and validation loss are notably larger than those from the previous graph at the same epochs. The accuracy of this trained model on validation data is 0.182, meaning 18.2% predictions are correct. It is still acceptable, as there are 52 labels in total. However, It is notably lower than the 25.2% accuracy obtained from the previous model. As a result, we can conclude that, when using the neural network to predict genre, using the 10 song features from question 4 makes better predictions than using the 5 principal components extracted from question 6, though both models are acceptable.

Q9:

a) In analyzing the relationship between popularity and average star ratings for the first 5,000 songs in the Spotify dataset, we initially computed the average rating and visualized the data with a scatter plot, revealing a general trend of increasing average ratings with higher popularity.



Recognizing the ordinal nature of star ratings, we opted for Spearman's rank correlation test instead of Pearson's, as the former non-parametric test is more appropriate for ordinal data and has fewer assumptions. Spearman's correlation coefficient of approximately 0.543, with a p-value effectively close to 0.0, indicated a moderate and statistically significant positive monotonic relationship. To further validate this relationship, we conducted an Ordinary Least Squares (OLS) regression, which yielded a statistically significant positive coefficient of 0.0165 for popularity, confirming its predictive value for average ratings.



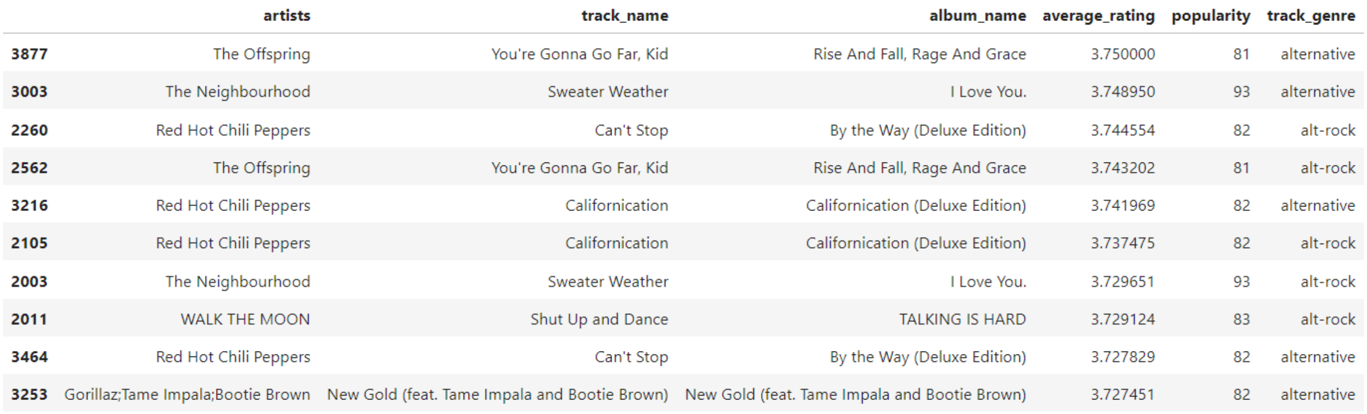
OLS Regression Results

Dep. Variable:	average_rating	**R-squared:**	0.324
Model:	OLS	**Adj. R-squared:**	0.324
Method:	Least Squares	**F-statistic:**	2398.
Date:	Sun, 17 Dec 2023	**Prob (F-statistic):**	0.00
Time:	23:53:30	**Log-Likelihood:**	-4472.1
No. Observations:	5000	**AIC:**	8948.
Df Residuals:	4998	**BIC:**	8961.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.4355	0.014	101.578	0.000	1.408	1.463
popularity	0.0165	0.000	48.967	0.000	0.016	0.017

However, with an R-squared value of 0.324, it became evident that while popularity is a meaningful predictor, it doesn't fully account for the variability in average ratings, suggesting the potential influence of other unaccounted factors.

b) In Part b, we sought to identify the "greatest hits" among the first 5,000 songs in the Spotify dataset, based on a popularity model defined by average star ratings. Our approach involved ranking these songs according to their average ratings and selecting the top 10 to represent the most popular or "greatest hits":

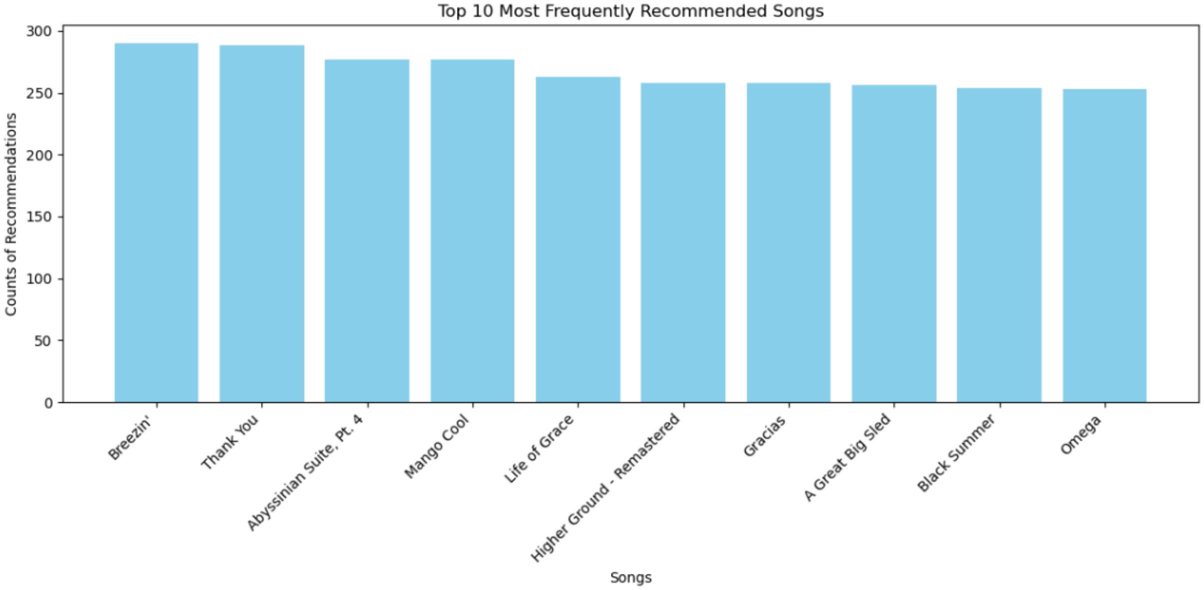


	artists	**track_name**	**album_name**	**average_rating**	**popularity**	**track_genre**
3877	The Offspring	You're Gonna Go Far, Kid	Rise And Fall, Rage And Grace	3.750000	81	alternative
3003	The Neighbourhood	Sweater Weather	I Love You.	3.748950	93	alternative
2260	Red Hot Chili Peppers	Can't Stop	By the Way (Deluxe Edition)	3.744554	82	alt-rock
2562	The Offspring	You're Gonna Go Far, Kid	Rise And Fall, Rage And Grace	3.743202	81	alt-rock
3216	Red Hot Chili Peppers	Californication	Californication (Deluxe Edition)	3.741969	82	alternative
2105	Red Hot Chili Peppers	Californication	Californication (Deluxe Edition)	3.737475	82	alt-rock
2003	The Neighbourhood	Sweater Weather	I Love You.	3.729651	93	alt-rock
2011	WALK THE MOON	Shut Up and Dance	TALKING IS HARD	3.729124	83	alt-rock
3464	Red Hot Chili Peppers	Can't Stop	By the Way (Deluxe Edition)	3.727829	82	alternative
3253	Gorillaz;Tame Impala;Bootie Brown	New Gold (feat. Tame Impala and Bootie Brown)	New Gold (feat. Tame Impala and Bootie Brown)	3.727451	82	alternative

During the analysis, we observed that certain songs appeared multiple times. Notably, these repetitions mostly involved songs with a popularity score of 0, suggesting they might originate from smaller or less mainstream music creators. This led us to a key consideration: the possibility of different versions or releases of the same song being listed under identical track and album names, a scenario not uncommon in music datasets, particularly involving independent artists. The similar average ratings across these repeated entries further supported this notion, indicating that listeners generally perceived these versions similarly. Consequently, we decided to treat these repeating songs as distinct entries, acknowledging the likelihood that they represent different versions or releases not explicitly differentiated in the dataset.

Q10:

User-Based Collaborative Filtering Method and Results: In our exploration of collaborative filtering for recommendation systems, we first implemented a user-based collaborative filtering approach. This method involved calculating cosine similarities between users based on their shared ratings and then predicting ratings for unrated items by leveraging ratings from similar users. The predicted ratings were used to generate top 10 song recommendations for each user:



We observed that our recommendations were very different from the "greatest hits" in Q9. When evaluating this model, we set a relatively high threshold (3 out of 4) to define 'likes' and observed an average recall of 0.00317 and precision of 0.05814. These metrics indicated that while a small fraction of the recommended items were relevant (precision), the system was only able to capture a very small portion of all relevant items (recall). We believe one significant factor contributing to these low scores is the sparse nature of the dataset, where many users had rated only a few songs, making it challenging to find accurate user similarities and predictions.

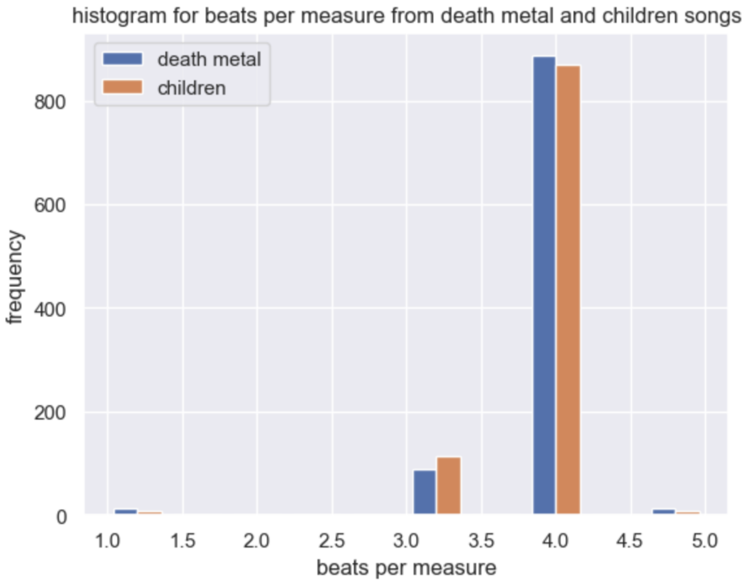
Limitations and challenges: The user-based collaborative filtering method faced limitations primarily due to the sparsity of the dataset. With users rating only a limited number of songs, finding reliable similar users to base predictions on was challenging. Furthermore, the conservative approach in defining 'likes' also contributed to the lower performance metrics, as it restricted the number of items considered relevant.

Attempting More Advanced Methods (SVD++): Recognizing these limitations, we turned to a more advanced method, SVD++ (Singular Value Decomposition Plus Plus). SVD++ is sophisticated in its ability to incorporate both explicit feedback (actual ratings) and implicit feedback (derived from user behavior, such as unrated items), making it particularly suitable for datasets with lots of missing values. However, the complexity and computational intensity of SVD++ posed significant challenges, as each epoch would take around 10 minutes to train on a single RTX 4090 GPU. Despite these constraints, we experimented around and computed the RMSE of the SVD++

model, achieving an RMSE of around 1.07 with n_factors=10, n_epochs=3, and lr_all=0.01. Notably, an RMSE of approximately 1 on a rating scale of 0-4 is quite respectable, suggesting that the model's predictions were reasonably close to the actual ratings. Additionally, we observed that increasing the number of epochs and factors did not substantially enhance the model, which could indicate a swift convergence to an optimal solution or an intrinsic characteristic of the dataset limiting further improvements.

Extra Credit: *Death metal and Children are two genres with great differences. The former is known for harshness, while the latter is known for harmony. We want to investigate whether the beats per measure differ between death metal and children's music.*

Death metal songs are associated with ‘death-metal’ within the ‘track_genre’ column, children's songs are associated with ‘children’ within the ‘track_genre’ column, and beats per measure is associated with the ‘time_signature’ column. We have extracted two groups of data: beats per measure values from death metal songs, and beats per measure values from children’s songs. We have plotted their histograms on a single graph, shown as below.



From this graph, there is no notable difference between histograms of beats per measure from both genres. As a result, we applied Mann Whitney U test on them to test their difference with the null hypothesis that the distribution underlying beats per measure from death metal is the same as the distribution underlying beats per measure from children songs. We chose to perform Mann Whitney U test because it is not reasonable to reduce the beats per measure values to sample means, as beats per measure is ordinal data, which is not continuous; furthermore, we have two groups to compare in the design. In this case, Mann Whitney U test can handle ordinal data and can test the difference between two data. After running Mann Whitney U test, we got a p-value of 0.107, which is greater than our alpha of 0.05, which failed to reject the null hypothesis, supporting the alternative hypothesis that the distribution underlying beats per measure from death metal is different from that from children’s songs. Hence, death metal songs are significantly distinct from children’s songs in terms of beats per measure.