Capstone Project Report

DS-GA 1001 Introduction to Data Science

Group 7

Jiawei Xu, Hongjin Zhu

12/19/23

Preprocess

Following the instructions, we applied a random seed of the N id number of one group member (11993511) and did some dimension reductions before analyzing problems 6, and 10.

Answer to Questions

1) After cleaning the data in the Spotify 52k dataset, we test the correlation between the column “popularity” and “duration”. Since we do not assume linearity in data, we use Spearman Correlation to do the testing. We find that the correlation coefficient is ~ -0.04 and the p-value is ~ 1.81e-17. If we set alpha = 0.05, we see that the p-value is less than alpha. We reject the null hypothesis. Hence, we can claim that there is a negative correlation between 'popularity' and 'duration'.

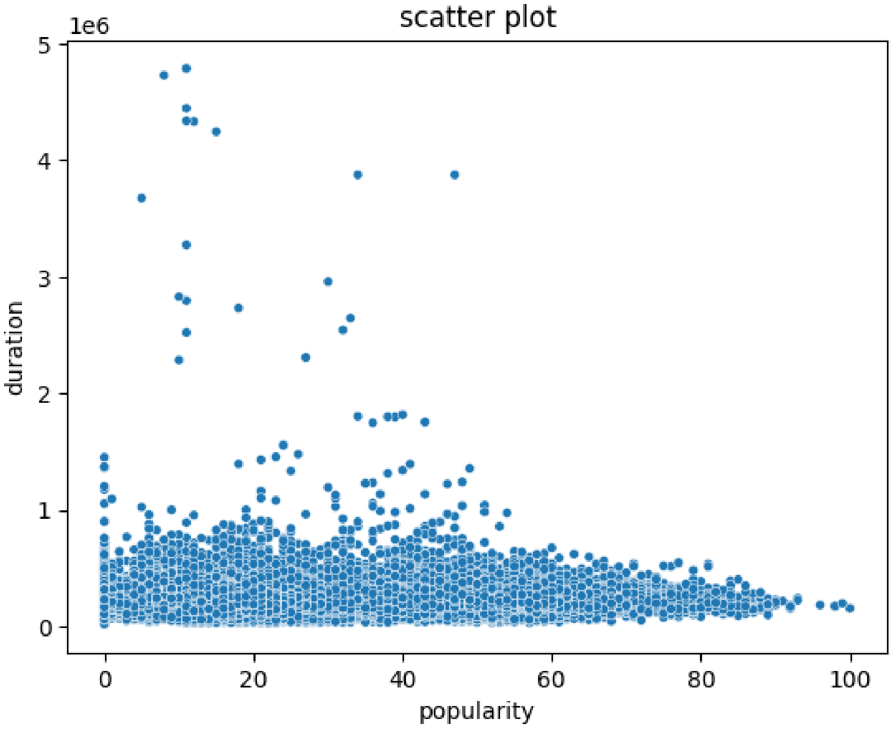


Figure 1.1 Correlation between duration and popularity

2) First we divided the popularity column with explicit or not and calculated the median of each category. The median score of “False” is 32.79, and the median score of “True” is 35.81. Then, we applied a t-test to test whether explicit songs are more popular. We get the P-value of 4.25e-23. With the alpha level set = 0.05. It is obvious that the p-value is below the level significantly. Therefore, we reject the null hypothesis which states that the distribution of popularity for explicitly rated songs is the same as the not explicitly rated songs. Also, we plotted a boxplot of the two classes of explicit and inexplicit-rated songs.

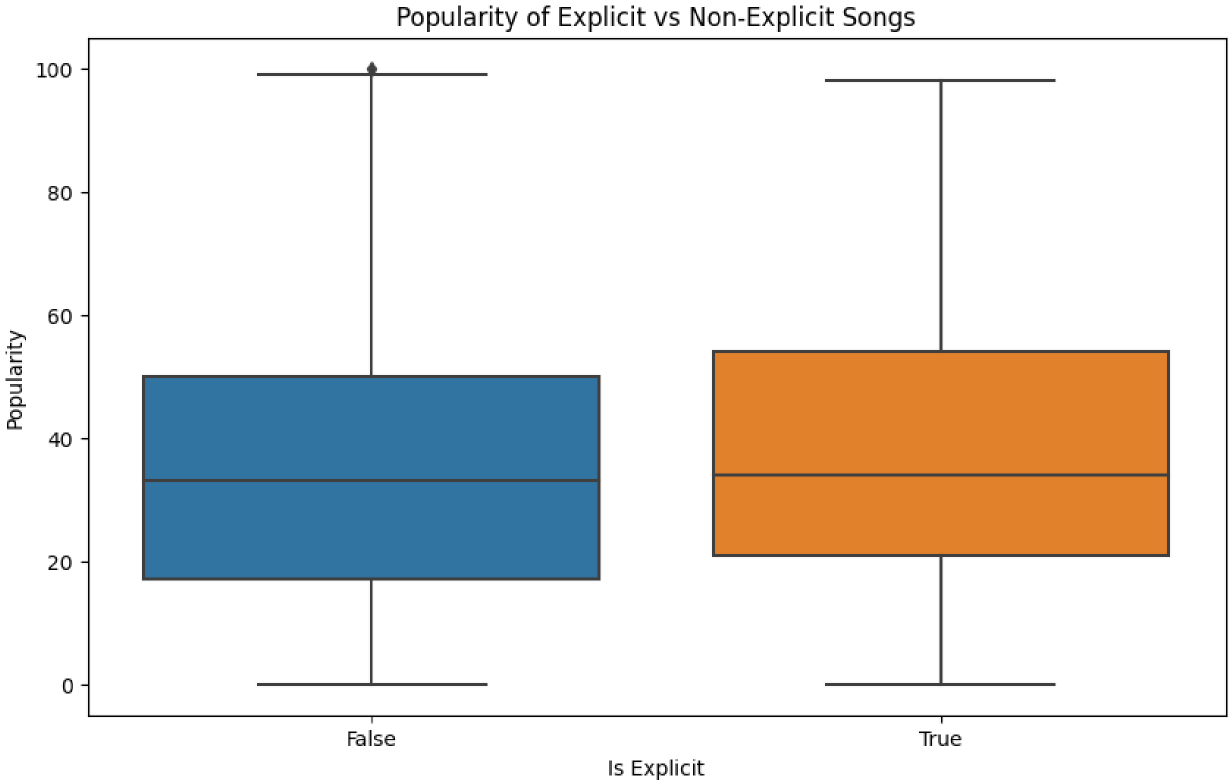


Figure 2.1 Boxplot of popularity of Explicit vs Non-Explicit Songs

3) We split the popularity column based on whether the song is of major key or minor key. We plot out the distribution of the two groups and do a t-test. The distribution can be found in Figure 3.1 (ChatGPT is used to aid aesthetic choices in this figure), and we find the popularity distribution of major and minor key songs are similar. Thus, we believe a t-test is eligible.

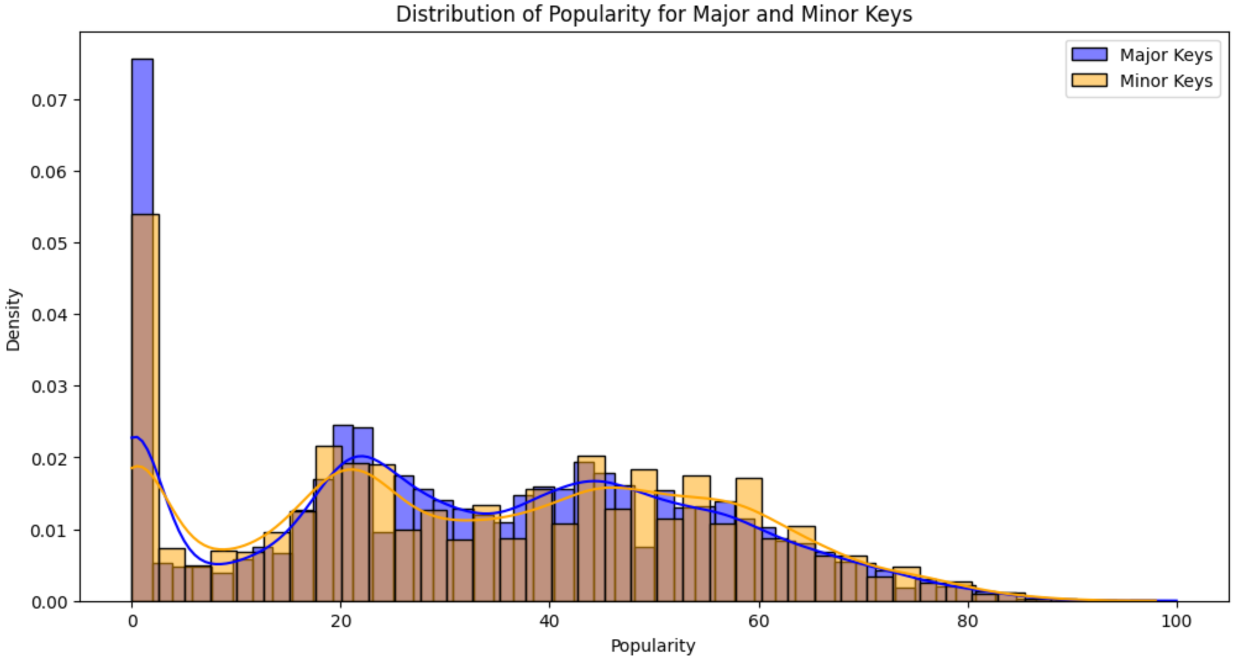


Figure 3.1 Distribution of Popularity

We got a p-value of ~ 0.99, larger than alpha = 0.05. We fail to reject the null hypothesis, and we cannot conclude that major keys are more popular than minor ones.

4) We fit the data into 10 linear regression models. We pick the 10 features respectively as X and popularity as y. To avoid overfitting we use ridge regression and hyperparameter tuning through the grid-search method. We get the following results (ChatGPT is used to aid aesthetic choices in this figure):

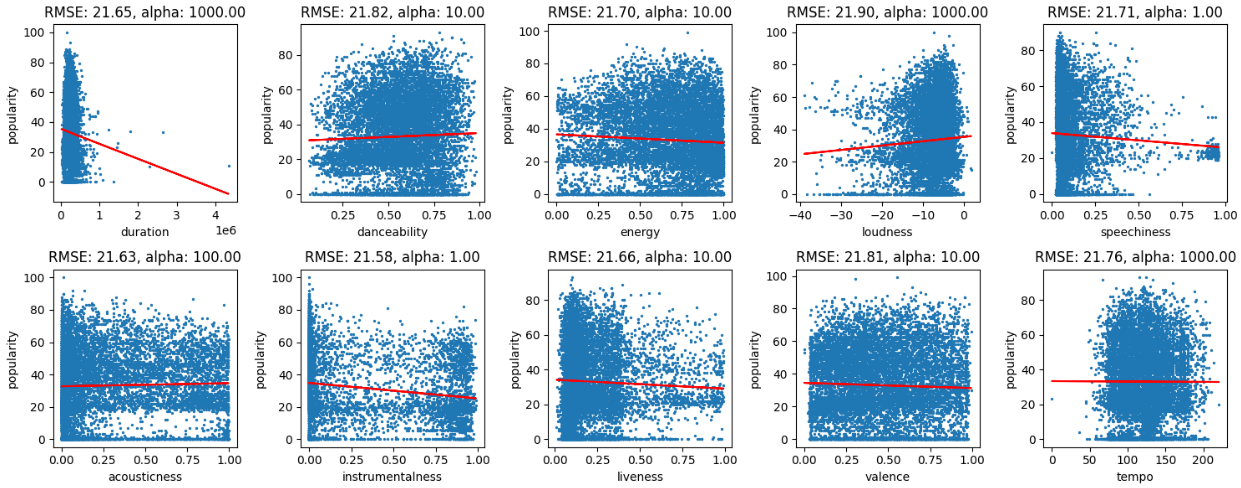
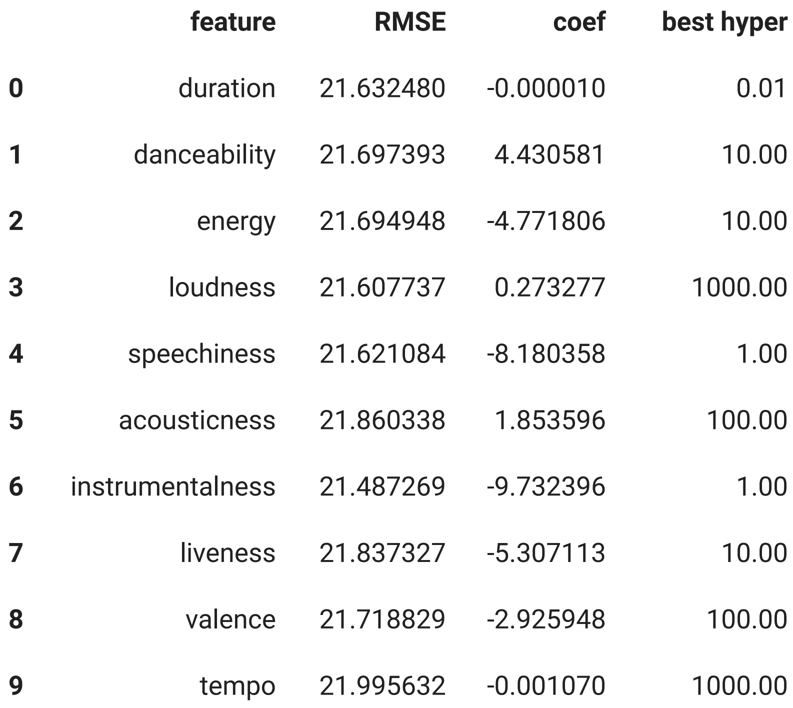


Figure 4.1 Linear regression



	feature	RMSE	coef	best hyper
0	duration	21.632480	-0.000010	0.01
1	danceability	21.697393	4.430581	10.00
2	energy	21.694948	-4.771806	10.00
3	loudness	21.607737	0.273277	1000.00
4	speechiness	21.621084	-8.180358	1.00
5	acousticness	21.860338	1.853596	100.00
6	instrumentalness	21.487269	-9.732396	1.00
7	liveness	21.837327	-5.307113	10.00
8	valence	21.718829	-2.925948	100.00
9	tempo	21.995632	-0.001070	1000.00

We can see from the table that the RMSE level is relatively high, which means all 10 features are not capable of predicting popularity independently, but the best feature here is “instrumentalness” considering the model has the least RMSE.

5) Using a similar approach as in question 4, we build a model using all 10 features as predictors, and we also compare the results from the model without regularization.
Without regularization:

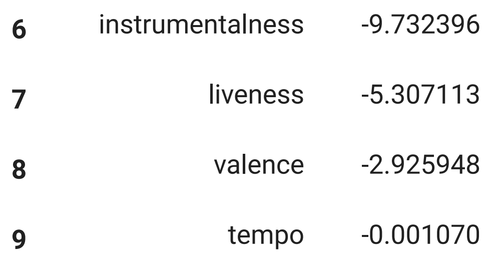
```
RMSE: 21.32831582514701
```

	feature	coef
0	duration	-0.000010
1	danceability	4.430581
2	energy	-4.771806
3	loudness	0.273277
4	speechiness	-8.180358
5	acousticness	1.853596
6	instrumentalness	-9.732396
7	liveness	-5.307113
8	valence	-2.925948
9	tempo	-0.001070

With regularization:

```
RMSE: 21.328165402192973, best hyper: 10
```

	feature	coef
0	duration	-0.000010
1	danceability	4.430581
2	energy	-4.771806
3	loudness	0.273277
4	speechiness	-8.180358
5	acousticness	1.853596



6	instrumentalness	-9.732396
7	liveness	-5.307113
8	valence	-2.925948
9	tempo	-0.001070

This model predicts popularity better than the previous linear regression model since RMSE decreases, and the regularized performs better than the one without regularization. However, the improvements are not very impactful. The feature “instrumentalness” has the largest coefficient, meaning that it predicts y the most, which aligns with question 4.

6) We first applied PCA on the extracted 10 features and plotted the relationship between the number of components and their cumulative explained variance (Figure 6.1). Via the plotted figure, we applied the elbow method to decide the most suitable number of components to be 4. The proportion of explained variance of the first 4 components is 67.15%. Then, based on the 4 components we gained, we decided the number of clusters for the k-mean algorithm. We applied both the elbow method (Figure 6.2) and the silhouette method (Figure 6.2) to determine the best number of clusters. In the elbow method, we plot the wcss value with the number of clusters and found the elbow point to be 2. In silhouette method, we also calculated the highest score at 2, which is 0.311. Therefore, we decided the clusters to be 2. Then we compared the clusters with the genre labels with cross-tabulation. However, as the number of clusters is only 2, some of the genre labels cannot distribute evenly among the 2 clusters while a few can have a good distribution (part of the distribution see Table 6.1).

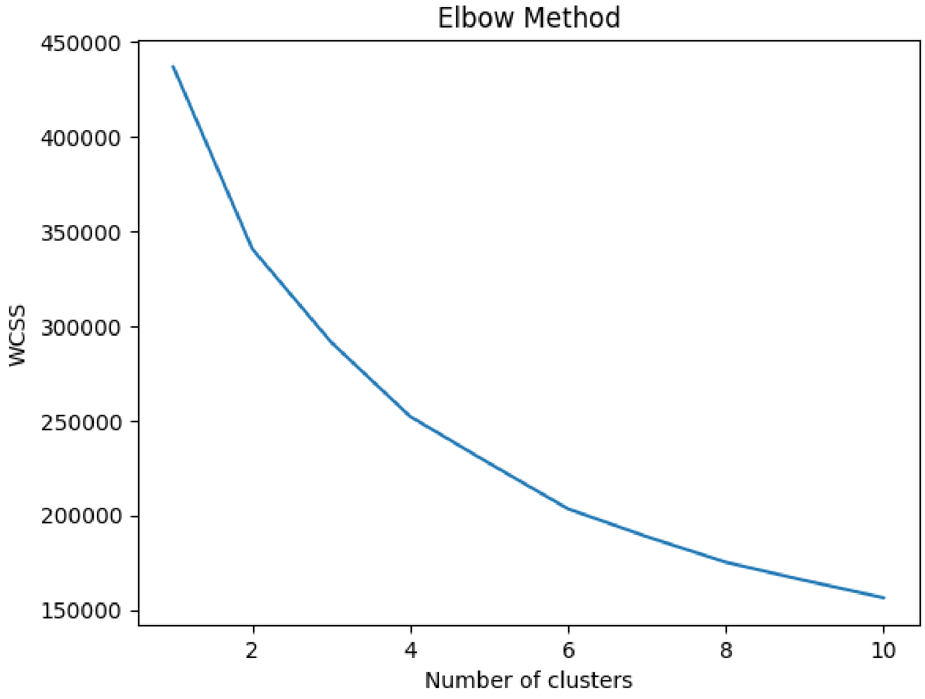


Figure 6.1 Elbow Method for Clusters

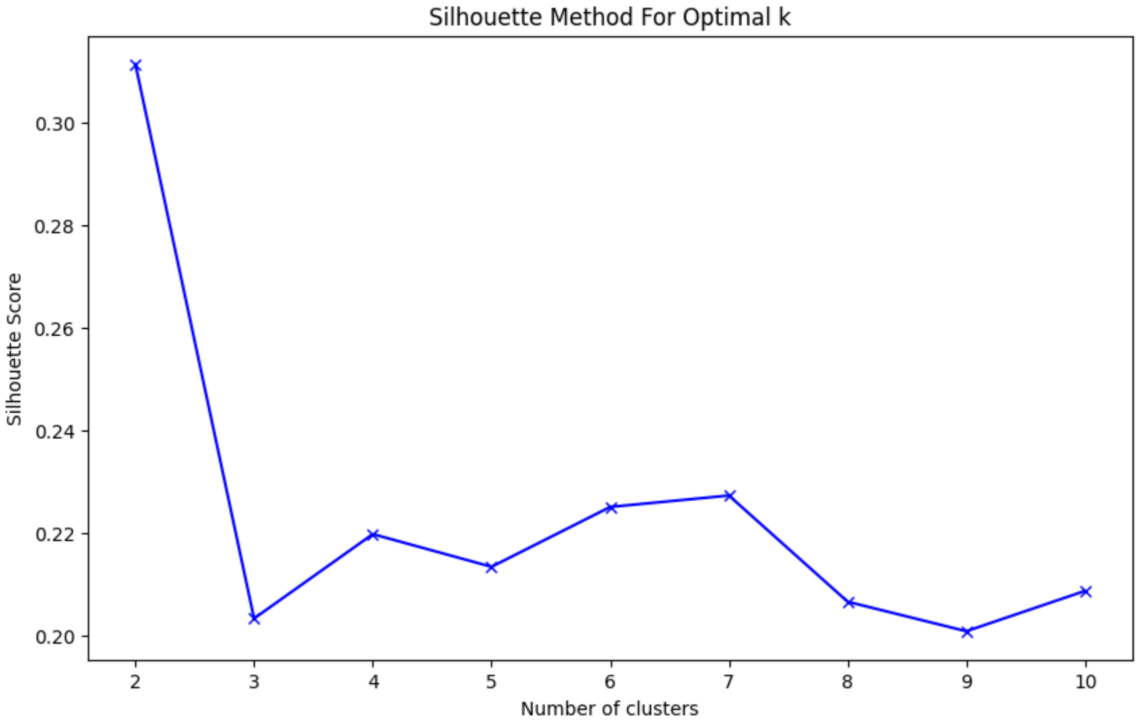


Figure 6.2 Sihouette Method for Clusters

Cluster	acoustic	afrobeat	alt-rock	alternative	ambient	anime	black-metal	bluegrass	blues	brazil	breakbeat	british	cantopop	chicago-house
0	425	889	938	920	104	742	955	598	736	830	993	548	462	968
1	575	111	62	80	896	258	45	402	264	170	7	452	538	32

Table 6.1 Part of the distribution of Genres in Clusters.

7) We use both logistic regression and SVM to fit the data where we set valence to X and mode to y. We can predict whether a song is in a major or minor key from valence using logistic regression or a support vector machine, but the results are not good with AUC ~ 0.50 for both models (Figure 7.1&7.2).

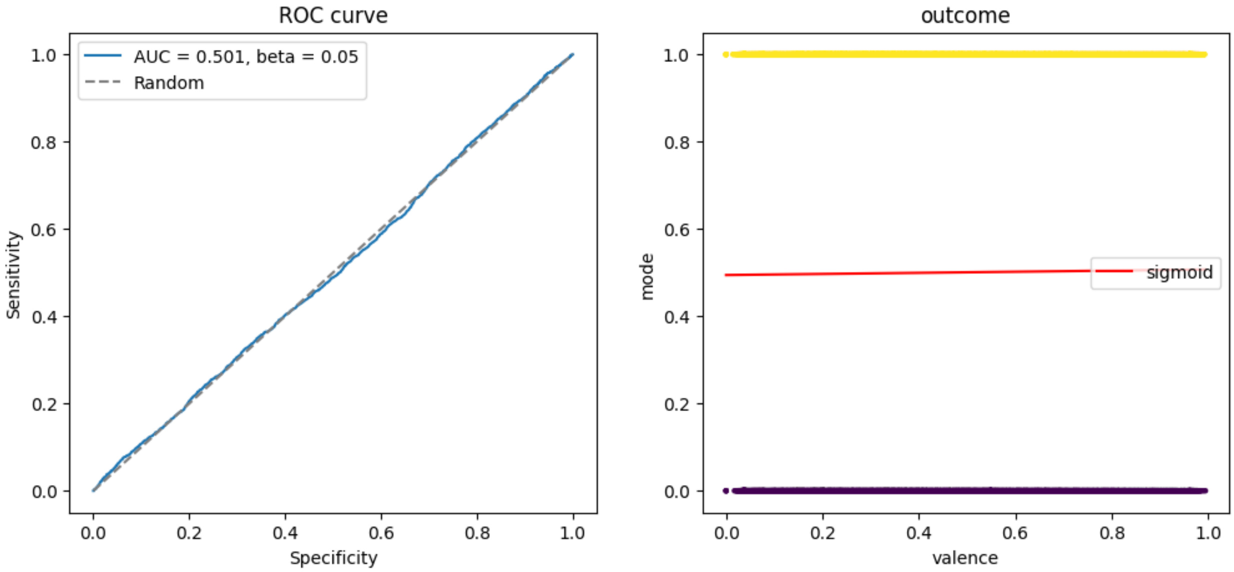


Figure 7.1 Result of Logistic Regression

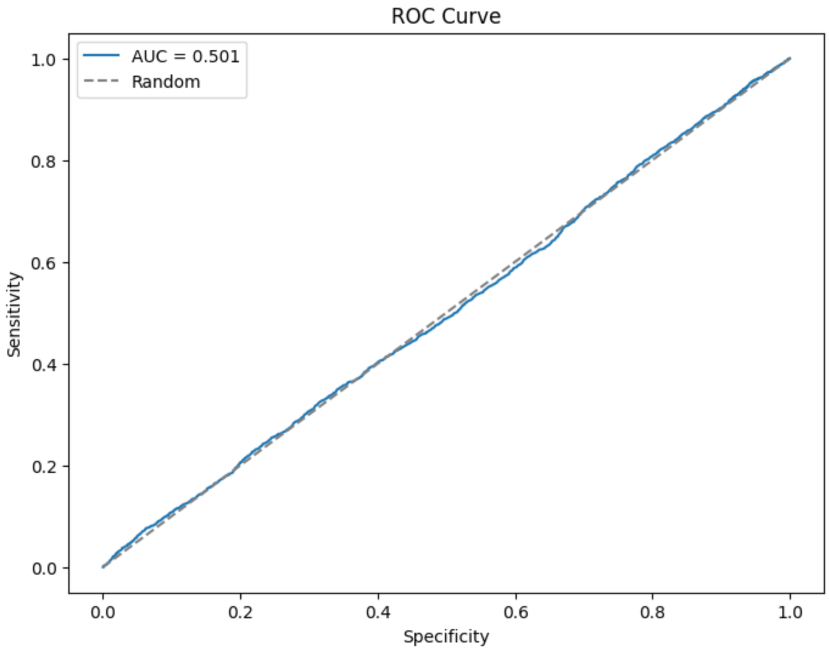
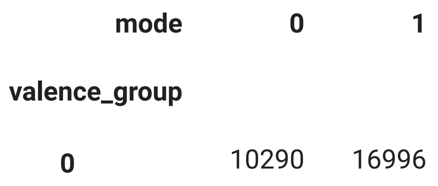


Figure 7.2 Result of SVM

After performing a Chi-square test, we find that the p-value is ~ 0.99, so there is no significant difference between the modes of the songs with high and low valence.



mode	0	1
valence_group		
0	10290	16996

1	9319	15395

Table 7.1 Contingency Table

One better model in our trials is the decision tree, where we can get an AUC ~ 0.56 > 0.50. However, due to the nature of the data, the performance is still poor (Figure 7.3).

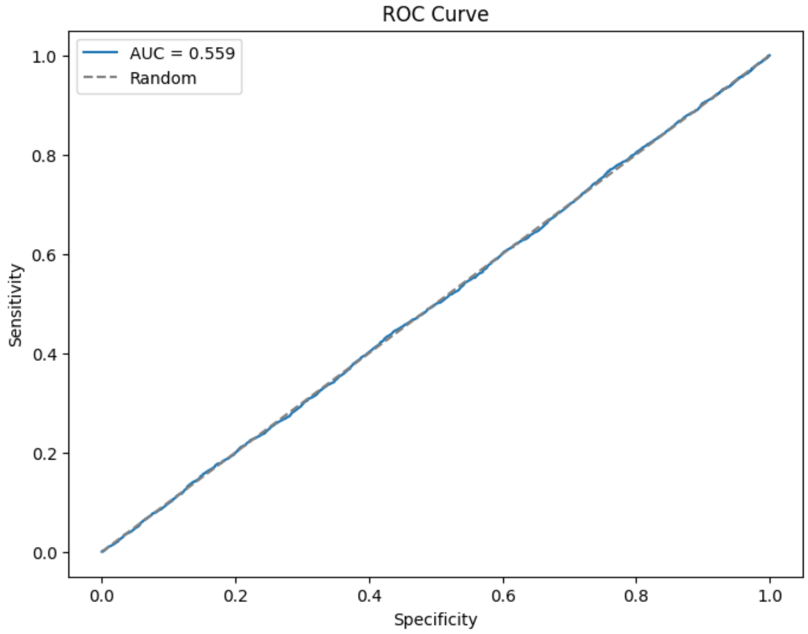


Figure 7.3 Result of Decision Tree

8) We applied a neural network with the original extracted 10 features to output the prediction to the 52 labels. Firstly we split the dataset into train and test sets with the test proportion = 0.2. Then, we applied label encoding to the 52 genre labels. After the preprocessing, we converted the split train, and test datasets into tensor format and trained with our network. The network is composed of one MLP layer and activation functions of ReLu. As it is a classification problem, we chose the loss to be cross-entropy loss. We trained the network for 50 epochs, the final prediction accuracy was around 30.25%. We also calculated the AUC score for each class, which is 0.89. The loss tendency is described in Figure 8.1.

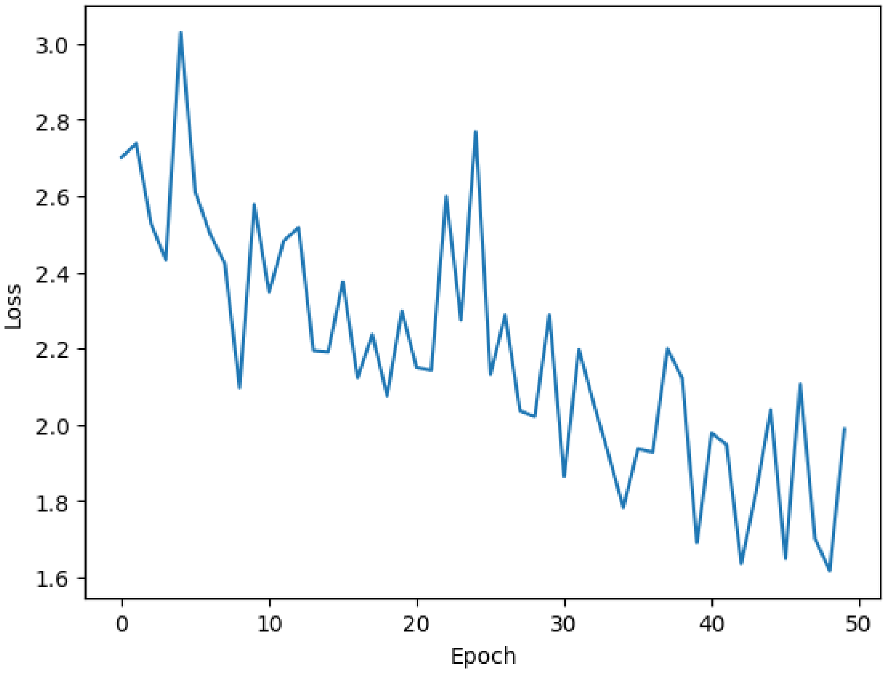


Figure 8.1 Loss Tendency of Net

9) We first test the Spearman correlation between the average ratings of the songs and their popularity. We get a p-value of ~ 0.0 which is less than alpha. Thus, we can claim that there is a correlation between popularity and average rating.

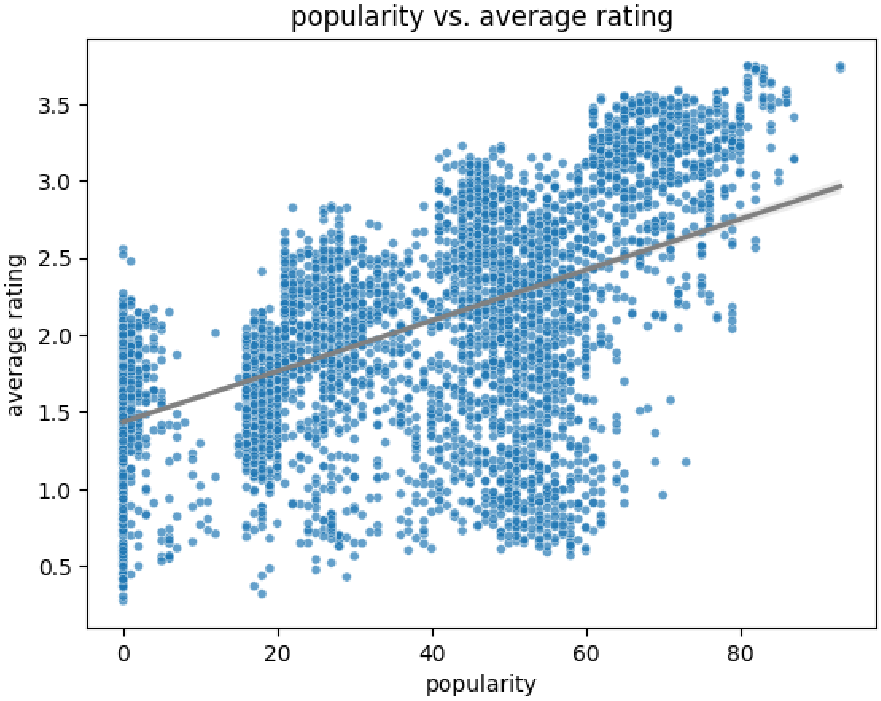
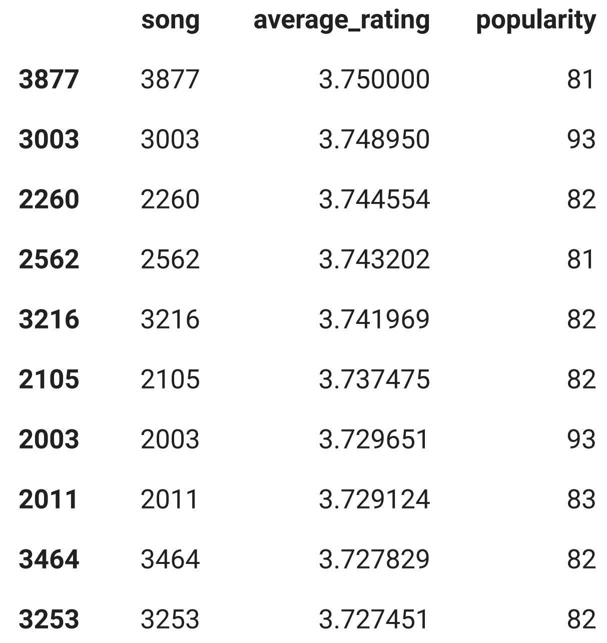


Figure 9.1 Correlation between average rating and popularity

To quantify the popularity we count the number of valid ratings each song has in the ratings dataset. We sort the songs according to their popularity and extract the top 10 songs. Hence we get the recommendation deck of the 10 most popular songs.



	song	average_rating	popularity
3877	3877	3.750000	81
3003	3003	3.748950	93
2260	2260	3.744554	82
2562	2562	3.743202	81
3216	3216	3.741969	82
2105	2105	3.737475	82
2003	2003	3.729651	93
2011	2011	3.729124	83
3464	3464	3.727829	82
3253	3253	3.727451	82

10) First, we do a train-test split on 10000 users, with the percentage of test set to be 0.2. Using train data to fit the model of item-wise cosine similarity, and using test data to evaluate the top 10 songs from the recommendation model with the actual top 10 rated songs from the test set based on mean average precision. Finally, we got the MAP result of our recommendation model to be 0.3. Also, we compared the MAP result of the actual top 10 rated songs with the top 10 songs from the popularity model in (9) and found that the MAP was 0.0, which seems to show that our personal mixtape recommendations have better performance than the popularity model.