Spotify Analysis

Group 20
Abhinav Krishnan / ak10776
SungJoon (Jason) Moon / sjm643
12/20/2023

Author Contributions.
Abhinav Krishnan: Null Processing, Question 1, Question 4, Question 5, Question 6 (PCA), Question 7, Question 8, Question 10, Extra Credit
SungJoon (Jason) Moon: Duplicates Processing, Question 1, Question 2, Question 3, Question 4, Question 6 (Clustering), Question 7, Question 9
ChatGPT: Assisted with Visualizations for Q6, Q8, data preprocessing for Q10,

Data Preprocessing.
Seed: We set the seed number to 15547187, SungJoon (Jason) Moon's N number
Spotify 52k: We checked for null values and found no null values. We checked and processed the duplicates for each question uniquely. When duplicates were present and relevant, we assumed that they add bias and also have a risk of leakage to the test set.
Data Scaling: We scaled the features using Scikit-Learn StandardScaler to z-score for appropriate models.

Question 1.

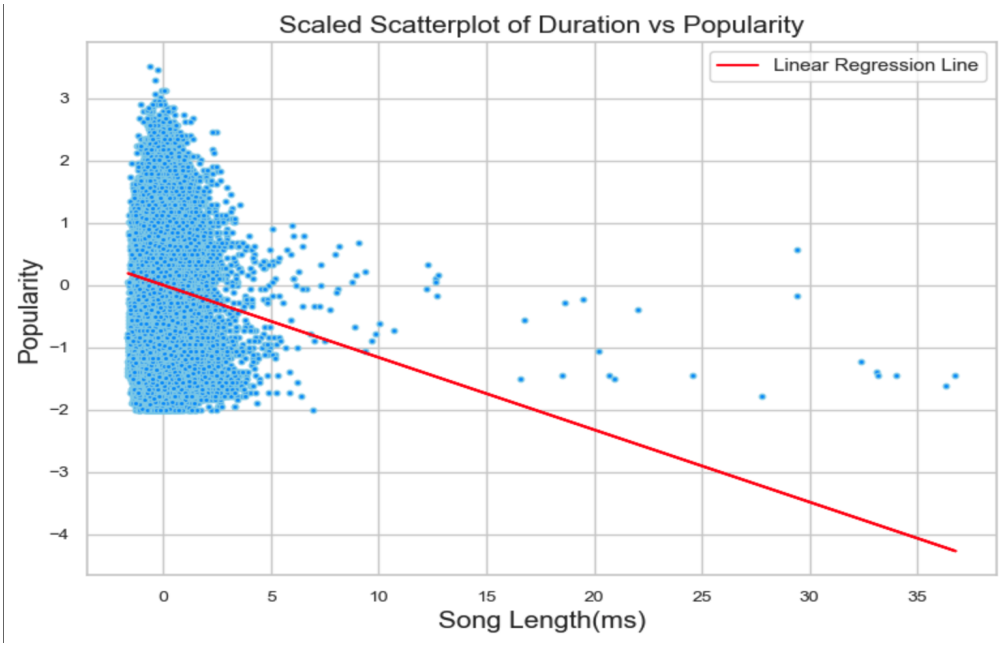
D: We identified 15,119 duplicates on the track_name, artists, and duration. We calculated the mean popularity of the duplicates and kept only one duplicate song. We calculated Pearson's Correlation Coefficient and ran linear regression to check COD. We also calculated the same metric after dropping popularity values 0.

Y: We dropped the duplicates based on our assumption in data preprocessing. We calculated the mean popularity as it was reasonable to balance the difference in popularity of the same songs. We calculated Person's Correlation Coefficient to check the linear relationship between the variables. We also ran linear regression to calculate COD and check if the linear relationship explains the variance.

F: Pearson Correlation Coefficient = -0.096715 / COD = 0.009354

Without popularity 0 songs: Pearson Correlation Coefficient = -0.116137 / COD = 0.013488

Fig 1. Scatter plot of Duration vs Popularity



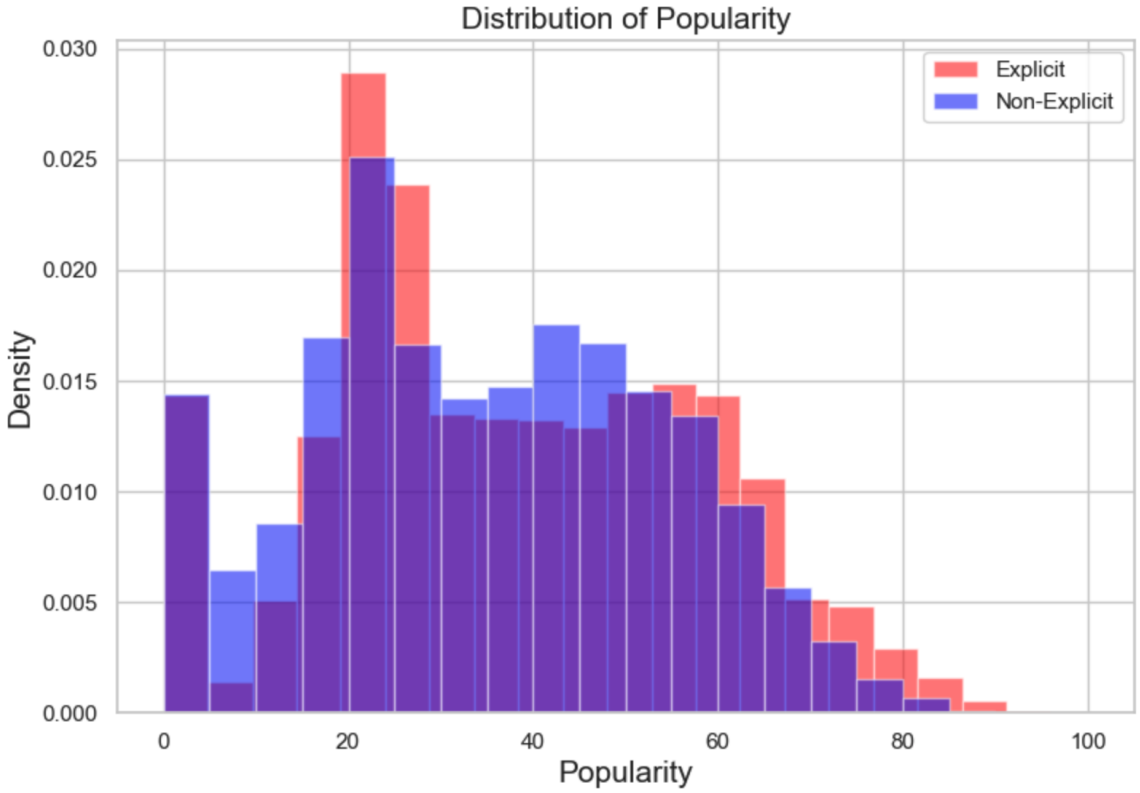
A: There is a weak negative relationship between song duration and popularity, as shown by the correlation coefficients. This relationship gets stronger when we drop 0 popularity songs as they are dominant in the popularity distribution. However, as the COD values are low, the linear regression model cannot explain the variance of popularity well.

Question 2.

D: We identified 16,200 duplicates on the track_name, artists, and explicit. We calculated the mean popularity of the duplicates and kept only one duplicate song. We ran one-sided Welch's t-test and Mann-Whiteney U test to check if explicit songs are more popular than non-explicit songs. We set alpha = 0.05

Y: We dropped the duplicates based on our assumption in data preprocessing. We ran two null hypothesis significance tests because we did not have information about how popularity was measured by Spotify, so we needed to check for both the median and mean, and could not assume homogeneity in variance.

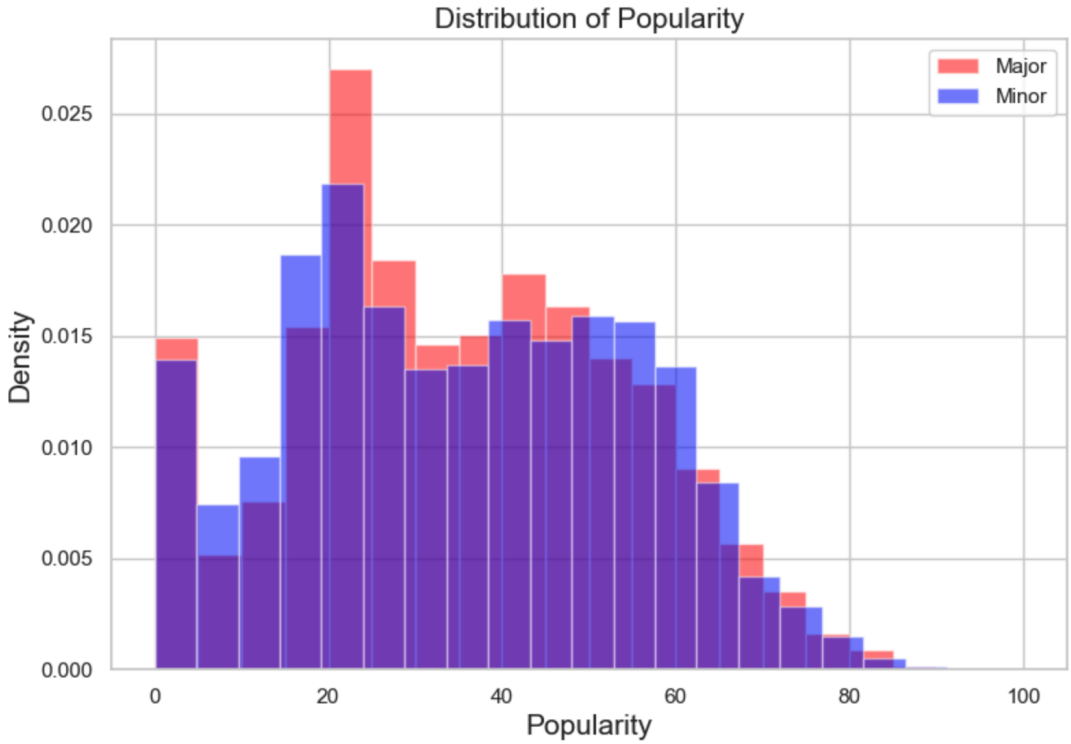
F: Fig 2. Distribution of Popularity



Welch's t = 9.698721339950838, p-value = 2.24250877381239e-22
Mann Whitney U = 94258995.0, p-value = 3.808712921292051e-20
A: Both tests result in p-value < alpha = 0.05. We can reject the null hypothesis that explicit songs and non-explicit songs have the same popularity mean/median. Therefore we conclude that explicit songs are more popular than non-explicit songs.

Question 3.
D: We identified 16,115 duplicates on the track_name, artists, and mode. We calculated the mean popularity of the duplicates and kept only one duplicate song. We ran one-sided Welch's t-test and Mann-Whiteney U test to check if songs in the major key are more popular than songs in the minor key. We set alpha = 0.05
Y: We dropped the duplicates based on our assumption in data preprocessing. We ran two null hypothesis significance tests because we did not have information about how popularity was measured by Spotify, so we needed to check for both the median and mean, and could not assume homogeneity in variance.

F: Fig 3. Distribution of Popularity



Welch's t = -3.1547075166654617, p-value = 0.9991959711263277
Mann Whitney U = 202757061.0, p-value = 0.9948959679471046
A: Both tests result in p-value > alpha = 0.05, so we cannot reject the null hypothesis that songs in major key and songs in minor key have the same mean/median popularity. We cannot conclude that songs in the major key are more popular than songs in the minor key.

Question 4.
D:We identified 14,525 duplicates on the track_name, artists, and all the song features. We calculated the mean popularity of the duplicates and kept only one duplicate song. We ran simple linear regression using each of the predictors to predict popularity. We split data to 80/20 train and test sets. We selected the model with the highest COD and the lowest RMSE.
Y: We dropped the duplicates based on our assumption in data preprocessing. We used simple linear regression to get a simple prediction model. We split the train/test to avoid overfitting. We selected a model with highest COD for explaining the variance of outcome well and the lowest RMSE to get the lowest prediction error.
F : Table 1. COD and RMSE for Each Predictor for Popularity

Predictor	COD	RMSE	Predictor	COD	RMSE
Duration	0.007313	0.989392	Acousticness	0.000154	0.992954
Danceability	0.004795	0.990646	Instrumentalness	0.044124	0.970874
Energy	0.005976	0.990058	Liveness	0.004235	0.990925
Loudness	0.006175	0.989959	Valence	-0.000008	0.993034
Speechiness	0.006476	0.989809	Tempo	0.000734	0.992665

A: The best predictor of popularity is instrumentalness with COD = 0.044124 and RMSE = 0.970874, explaining the variance in popularity the best and having the lowest prediction error.

Question 5.

D: We used the same train/test set as Question 4 and used all predictors to run multiple linear regression. We calculated the COD and RMSE. We also ran Ridge regression models with GridSearchCV to find the best alpha.

Y: We used the same train/test set as Question 4 to make a 1:1 comparison between Question 4 and 5. We calculated the COD and RMSE to assess the explained variance by the model and prediction errors. We ran a ridge with GridSearchCV to find the optimal hyperparameter for the regularized model.

F: Table 2. Multiple Regression Model and Ridge Regression Outcome

Model	COD	RMSE
Multiple Linear Regression	0.089485	0.947558
Ridge Regression (Alpha=10)	0.089486	0.947558

A: The multiple linear regression model outperforms the simple linear regression models in Question 4. The COD increased by 103% from the simple linear regression model and RMSE decreased about 0.023. From the result, we believe that the combination of song features has more explanatory power and predictive power over single features. Additionally, the ridge regression has almost no difference from the multiple regression model.

Question 6.

D: We standardized the features and ran PCA without defining the number of components. We chose Principal Component 1 to 7 and ran DBSCAN to cluster the data and compared it with track_genre.

Y: We ran PCA without defining the number of components to analyze explained variance of all principal components. We used Principal Component 1 to 7 because that explains 90% of the variance. We ran DBSCAN to identify the appropriate number of clusters.

F: Fig 4. Explained Variance Scree Plot

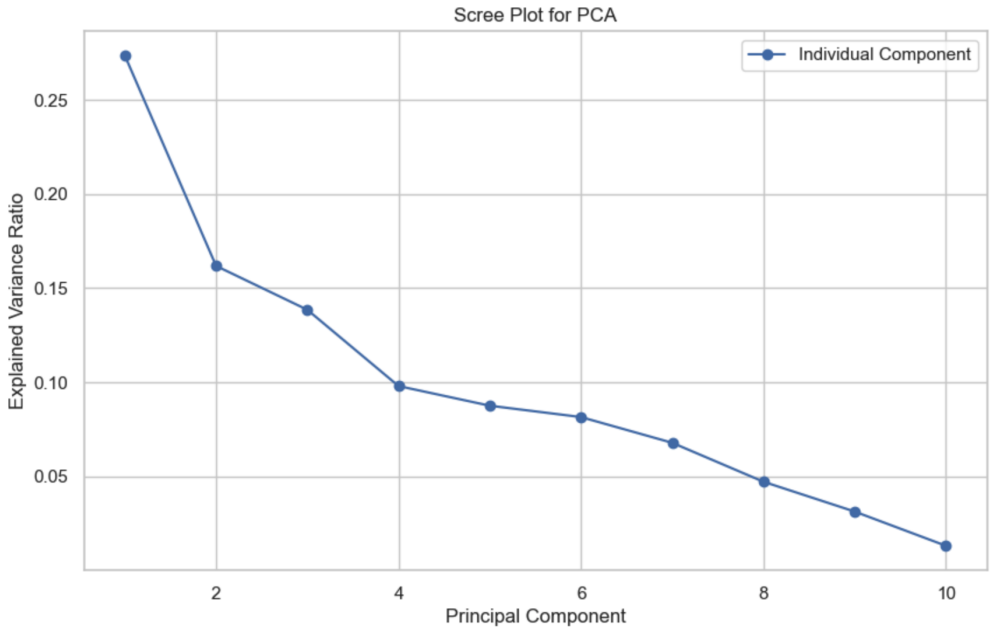


Table 3. Principal Component Analysis

PC	1	2	3	4	5	6	7	8
Eigenvalue	2.733934	1.617391	1.384605	0.979607	0.875226	0.814846	0.678282	0.471581
Explained Variance	0.273388	0.161736	0.138458	0.097959	0.087521	0.081483	0.067827	0.047157
Cumulative Explained	0.273388	0.435124	0.573582	0.671541	0.759062	0.840545	0.908372	0.955529

Table 4. Clusters Identified from DBSCAN(epsilon = 0.75, min_samples = 8) and Corresponding Genre

Cluster	Genre
-1	'acoustic' 'afrobeat' 'alt-rock' 'alternative' 'ambient' 'anime' 'black-metal' 'bluegrass' 'blues' 'brazil' 'breakbeat' 'british' 'cantopop' 'chicago-house' 'children' 'chill' 'classical' 'club' 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house' 'detroit-techno' 'disco' 'disney' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro' 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german' 'gospel' 'goth' 'grindcore' 'groove' 'grunge' 'guitar' 'happy' 'hard-rock' 'hardcore' 'hardstyle' 'heavy-metal' 'hip-hop'
0	'acoustic' 'afrobeat' 'alt-rock' 'alternative' 'ambient' 'anime' 'black-metal' 'bluegrass' 'blues' 'brazil' 'breakbeat' 'british' 'cantopop' 'chicago-house' 'children' 'chill' 'classical' 'club' 'comedy' 'country' 'dance' 'dancehall' 'death-metal' 'deep-house' 'detroit-techno' 'disco' 'disney' 'drum-and-bass' 'dub' 'dubstep' 'edm' 'electro' 'electronic' 'emo' 'folk' 'forro' 'french' 'funk' 'garage' 'german' 'gospel' 'goth' 'grindcore' 'groove' 'grunge' 'guitar' 'happy' 'hard-rock' 'hardcore' 'hardstyle' 'heavy-metal' 'hip-hop'
1	'acoustic' 'anime' 'blues' 'british' 'classical' 'country' 'electronic' 'emo' 'forro'
2	'alt-rock' 'alternative'
3	'ambient' 'classical'
4	'anime' 'dub' 'dubstep' 'gospel'
5	'alt-rock' 'bluegrass' 'blues' 'groove' 'grunge'
6	'blues'
7	'black-metal' 'blues' 'garage' 'grindcore' 'happy' 'hard-rock' 'heavy-metal'
8	'brazil' 'gospel'
9	'ambient' 'brazil' 'gospel'
10	'brazil' 'gospel' 'grunge'
11	'anime' 'brazil' 'breakbeat' 'gospel' 'guitar' 'hard-rock' 'hardcore' 'heavy-metal'
12	'brazil' 'funk' 'hip-hop'
13	'british' 'chicago-house'
14	'bluegrass' 'cantopop' 'chill' 'disco'
15	'cantopop' 'country'
16	'classical' 'german'
17	'classical' 'german'
18	'comedy'
19	'comedy'
20	'comedy'

21	'country'
22	'dance' 'hip-hop'
23	'black-metal' 'death-metal' 'french' 'grindcore' 'groove'
24	'chicago-house' 'deep-house' 'detroit-techno' 'disco' 'electronic' 'garage'
25	'breakbeat' 'detroit-techno'
26	'alt-rock' 'brazil' 'country' 'forro' 'hard-rock'
27	'acoustic' 'brazil' 'chill' 'forro' 'french'
28	'forro' 'hard-rock'
29	'bluegrass' 'brazil' 'comedy' 'german' 'gospel'
30	'black-metal' 'breakbeat' 'club' 'death-metal' 'dubstep' 'grindcore'
31	'alt-rock' 'alternative' 'brazil' 'emo' 'groove'
32	'acoustic' 'children' 'chill' 'guitar'
33	'breakbeat' 'deep-house' 'happy' 'hard-rock' 'hardcore'
34	'breakbeat' 'drum-and-bass' 'happy' 'hardstyle'
35	'edm' 'german' 'hard-rock'
36	'black-metal' 'goth' 'hard-rock' 'hardstyle'
37	'death-metal' 'drum-and-bass' 'goth' 'hardstyle' 'heavy-metal'

A: From the 39 clusters we identified using DBSCAN, we believe some are very reasonable clusters to represent track_genre. We observe clusters of blues and comedy, and we see a cluster that has hip-hop and dance together, and a cluster that has alt-rock and alternative together.

Question 7.

D: We identified 14,730 duplicates in mode and valence and dropped the duplicates. We used logistic regression, support vector machine, and XGBoost to classify whether a song is a major key or a minor key (Mode) using valence as predictor. We split the data to 80/20 train and test set to train the models and test the classification performance.

Y: We dropped the duplicates based on our assumption in data preprocessing. Since Mode is a binary categorical variable, we used classification models to predict the key of the song. We split the data to train and test sets to calculate classification performance metrics of the model on unseen data and avoid overfitting.

F:Table 5. Classification Report

	Logistic Regression	**SVM**	**XGBoost**
Accuracy	0.49	0.49	0.51
Recall	0.47	0.39	0.50
Precision	0.62	0.65	0.63
Specificity	0.53	0.66	0.52
AUROC	0.51	0.52	0.51

Fig 5. Logistic Confusion Matrix

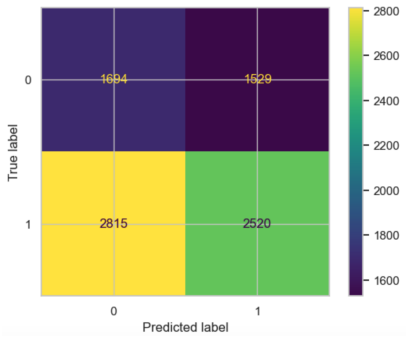


Fig 6. SVM Confusion Matrix

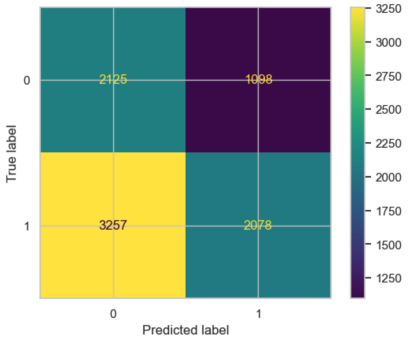


Fig 7. XGBoost Confusion Matrix

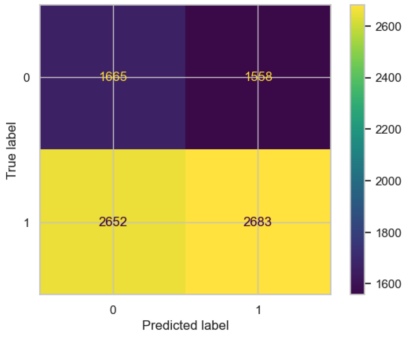
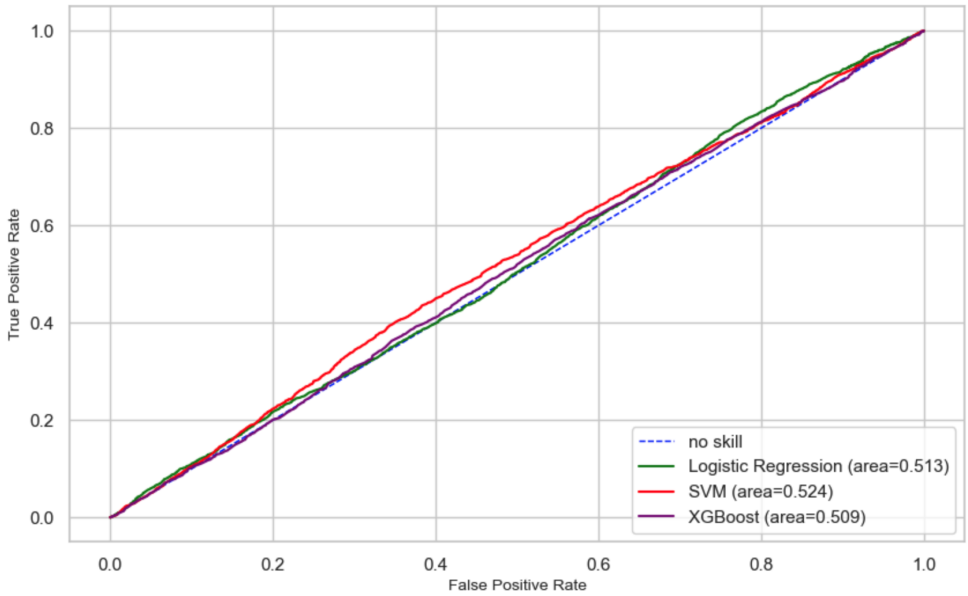


Fig 8. ROC Curves



A: Comparing logistic regression model and SVM, SVM predicts minor key better while logistic regression predicts major key better. SVM has higher AUROC, so we conclude that SVM is a slightly better model. Additionally, we observed that XGBoost has the highest accuracy and recall, but also the lowest AUROC. We conclude that SVM is the best at predicting minor key and XGBoost is the best at predicting major key, but none of them are really good models to predict the key.

Question 8.

D: We used the original features to predict track_genre. We identified 6241 duplicates across all features including track_genre and dropped duplicates. We used PyTorch to build a fully connected feedforward network. We split the data to 80/20 train and test set to train the models and test the multilabel classification performance. We used cross entropy loss function in our model.

Y: We used the original features over PCA transformed because We dropped the duplicates based on our assumption in data preprocessing. We used a fully connected feedforward network as the question suggested. We split the data to train and test set to train the model and test the performance using the unseen test set to avoid overfitting. We used cross entropy loss as this is a multilabel problem.

F: Fig 9. Loss over epoch

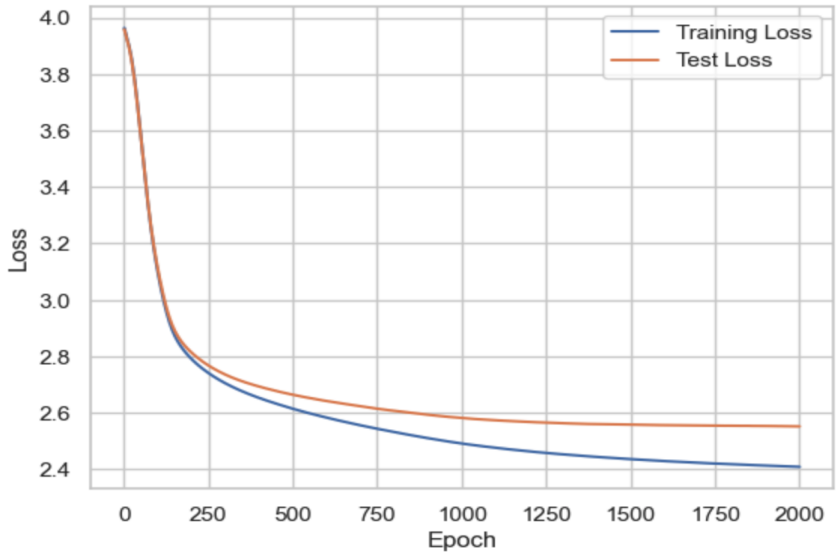


Table 6. Classification Report: Overall Accuracy = 0.3024

Genre	acoustic	afrobeat	alt-rock	alternative	ambient	anime	black-metal
Precision	0.27	0.28	0.13	0.00	0.34	0.22	0.41
Recall	0.27	0.23	0.04	0.00	0.48	0.22	0.46
Genre	bluegrass	blues	brazil	breakbeat	british	cantopop	chicago-house
Precision	0.32	0.09	0.16	0.25	0.23	0.24	0.41
Recall	0.47	0.02	0.07	0.24	0.05	0.44	0.55
Genre	children	chill	classical	club	comedy	country	dance
Precision	0.45	0.24	0.39	0.29	0.89	0.15	0.09
Recall	0.46	0.30	0.47	0.22	0.81	0.10	0.05
Genre	dancehall	death-metal	deep-house	detroit-techno	disco	disney	drum-and-bass
Precision	0.22	0.31	0.23	0.47	0.26	0.31	0.50
Recall	0.45	0.35	0.45	0.57	0.37	0.31	0.55
Genre	dub	dubstep	edm	electro	electronic	emo	folk
Precision	0.09	0.17	0.13	0.13	0.12	0.15	0.14
Recall	0.04	0.27	0.13	0.07	0.04	0.13	0.11
Genre	forro	french	funk	garage	german	gospel	goth
Precision	0.40	0.19	0.47	0.21	0.37	0.31	0.14
Recall	0.69	0.10	0.33	0.22	0.08	0.52	0.03
Genre	grindcore	groove	grunge	guitar	happy	hard-rock	hardcore
Precision	0.67	0.08	0.17	0.33	0.32	0.17	0.28
Recall	0.76	0.03	0.25	0.43	0.34	0.15	0.26
Genre	hardstyle	heavy-metal	hip-hop				
Precision	0.40	0.24	0.20				
Recall	0.58	0.35	0.19				

A: Based on the accuracy, our model doesn't perform the best. However, from the classification report, we observe that some genres are predicted very well, and some poorly. For example, the model has both high precision and high recall for grindcore, while the model cannot predict alternative at all.

Question 9.

D: We computed the average user rating for each song ignoring the null values in ratings. We calculated Pearson's Correlation Coefficient for Spotify's popularity measure for the rated songs and the average user ratings. We then selected the top ten unique songs by the mean user ratings.

Y: We computed average user ratings ignoring the null values to get the raw mean ratings. Pearson's Correlation Coefficient gives us the linear relationship between the popularity and mean user ratings. We used mean user ratings to select top ten as popularity based recommendation is the average utility.

F: Pearson Correlation Coefficient = 0.569391

Table 7. Top Hits of 5000 Target Songs

Album	Song	Artists	Popularity	Mean User Rating
By the Way (Deluxe Edition)	Can't Stop	Red Hot Chili Peppers	82	3.744554
Rise And Fall, Rage And Grace	You're Gonna Go Far, Kid	The Offspring	81	3.743202
I Love You.	Sweater Weather	The Neighbourhood	93	3.729651
TALKING IS HARD	Shut Up and Dance	WALK THE MOON	83	3.729124
New Gold (feat. Tame Impala and Bootie Brown)	New Gold (feat. Tame Impala and Bootie Brown)	Gorillaz; Tame Impala; Bootie Brown	82	3.727451
Meteora	Numb	Linkin Park	83	3.685801
Nevermind (Remastered)	Smells Like Teen Spirit	Nirvana	83	3.677518
Americana	The Kids Aren't Alright	The Offspring	81	3.672234
Toxicity	Chop Suey!	System Of A Down	83	3.661677
Toxicity	Toxicity	System Of A Down	81	3.641434

A: The correlation suggests that there is a positive and somewhat strong relationship between Spotify's Popularity metrics and the average user ratings. We also observe that the top 10 songs from our popularity-based model have also very high popularity ratings. However, when we select the top 10 hits using Spotify's popularity, we have a different recommendation.

Question 10.

D: We used the ratings data and flattened it to get a list of ratings per song per user. We dropped all nulls and performed user-user collaborative filtering. We split the data to 80/20 train and test set. We create the similarity matrix and use the top k similar users to generate predictions. We calculated the RMSE score on the test set. We created a personal mixtape for all the users and visualized the mixtape of one user.

Y: We flattened our ratings data to make it compatible with our recommendation package. We dropped all nulls because the dataset was intractably large for our model. We split into train and test to test the performance using the unseen test set to avoid overfitting. We used RMSE to evaluate the model performance because it was a simpler metric to interpret.

F: Table 8. Personalized Mixtape for User 3

Album	Song	Artists	Popularity	Mean User Rating
Fallen	Bring Me To Life	Evanescence	82	3.744554
I Love You.	Sweater Weather	The Neighbourhood	93	3.743202
TALKING IS HARD	Shut Up and Dance	WALK THE MOON	83	3.729651
I'm In Love With You	I'm In Love With You	The 1975	78	3.729124
Californication (Deluxe Edition)	Californication	Red Hot Chili Peppers	82	3.727451
Rise And Fall, Rage And Grace	You're Gonna Go Far, Kid	The Offspring	81	3.685801
Out Of Time (25th Anniversary Edition)	Losing My Religion	R.E.M.	84	3.677518
By the Way (Deluxe Edition)	Can't Stop	Red Hot Chili Peppers	82	3.672234
Little Dark Age	Little Dark Age	MGMT	83	3.661677
Nevermind (Remastered)	Smells Like Teen Spirit	Nirvana	84	3.641434

A: We see that our model gives more granular recommendations - as can be seen by contrasting the above table with the popularity based model in Q9. The recommendations have some overlap with the popularity based recommendations, but have some different songs, which we interpret as a result of personalized recommendation. Our model has an RMSE score of 1.0111 on the test set.

Extra Credit.

D: We identified 14,558 duplicates in tempo, valence, danceability, acousticness and explicit and dropped the duplicates. We used logistic regression and XGBoost to classify whether a song is explicit or not using valence as predictor. We split the data to 80/20 train and test set to train the models and test the classification performance.

Y: We dropped the duplicates based on our assumption in data preprocessing. Since Explicit is a binary categorical variable, we used classification models to predict the key of the song. We split the data to train and test sets to calculate classification performance metrics of the model on unseen data and avoid overfitting.

F: Table 9. Classification Report

	Logistic Regression	XGBoost
Accuracy	0.57	0.74
Recall	0.56	0.53
Precision	0.14	0.22
Specificity	0.57	0.77
AUROC	0.61	0.70

Fig 10. Logistic Confusion Matrix

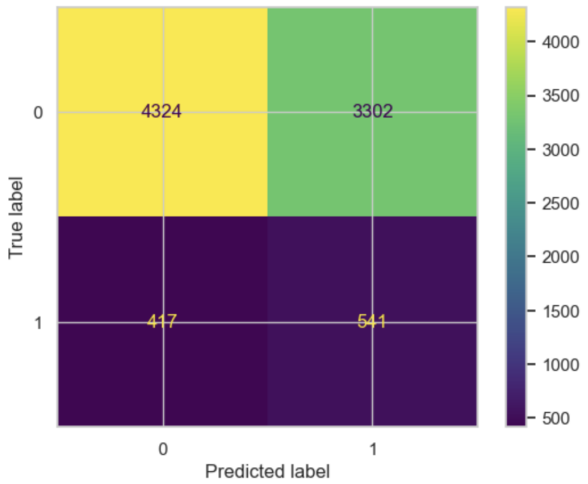


Fig 11. XGBoost Confusion Matrix

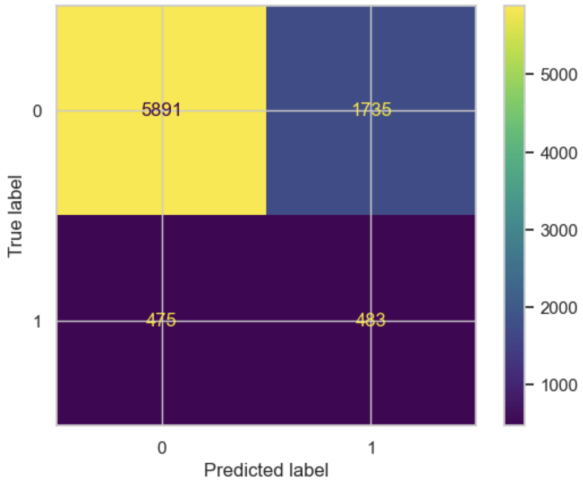
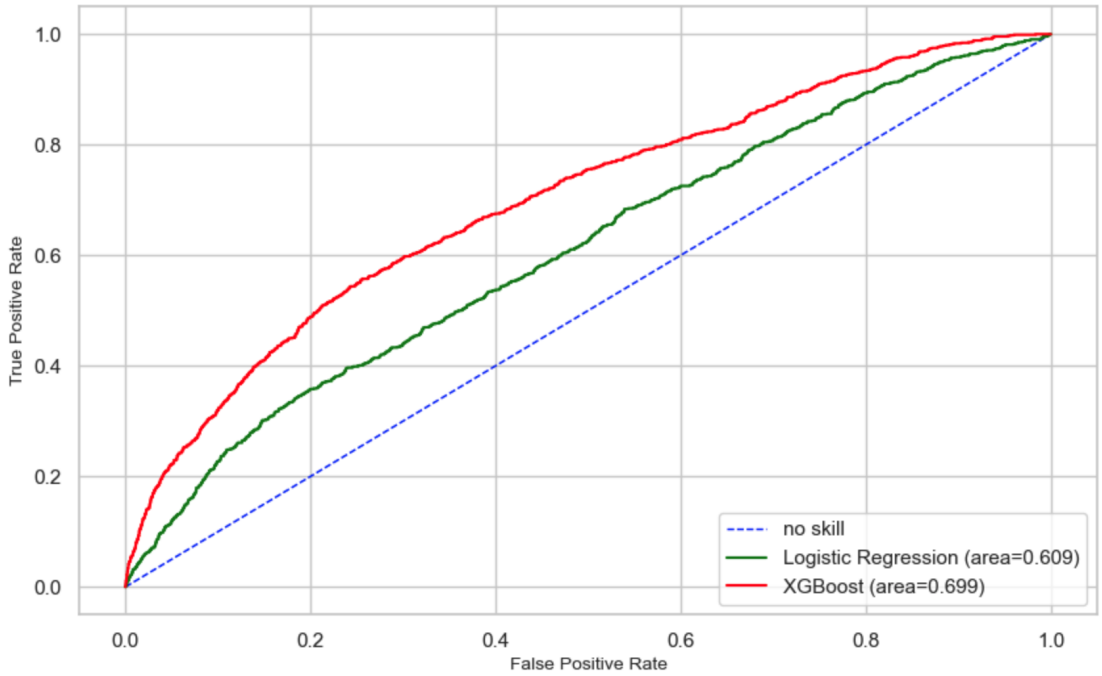


Fig 12. Roc Curves



A: We see that explicitness of a song can be predicted reasonably well using tempo, valence, danceability and acousticness of a song. Both Logistic Regression and XGBoost predict explicit = false much better. XGBoost performs better with higher accuracy, precision, specificity and F1 score. We conclude that XGBoost is the better classification model for predicting explicitness and can be fine-tuned further to improve the prediction performance.