Spotify Data Analysis

DS862 Final Project
2020 Fall
Di Wang
Yu Han

Introduction

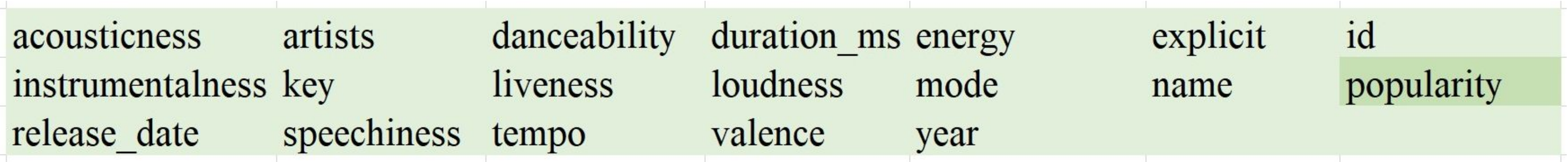
Spotify is one of the largest audio streaming and media services providers in the world.

Dataset: Record of Spotify songs between 1921 and 2020.

Tasks:
(1) Popularity prediction
(2) Genres cluster analysis
(3) Recommendation system

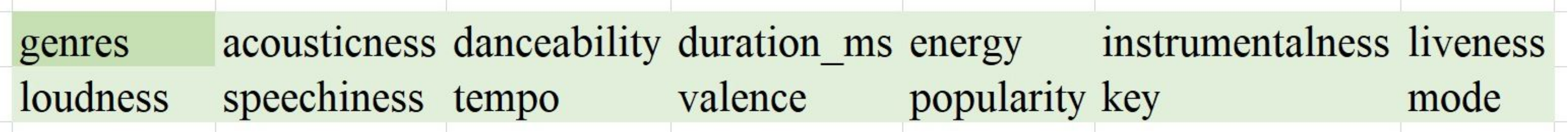
About the data

Data.csv



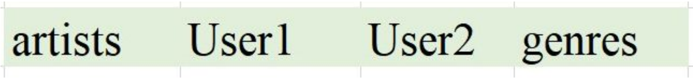
acousticness	artists	danceability	duration_ms	energy	explicit	id
instrumentalness	key	liveness	loudness	mode	name	popularity
release_date	speechiness	tempo	valence	year		

Data_by_genres.csv



genres	acousticness	danceability	duration_ms	energy	instrumentalness	liveness
loudness	speechiness	tempo	valence	popularity	key	mode

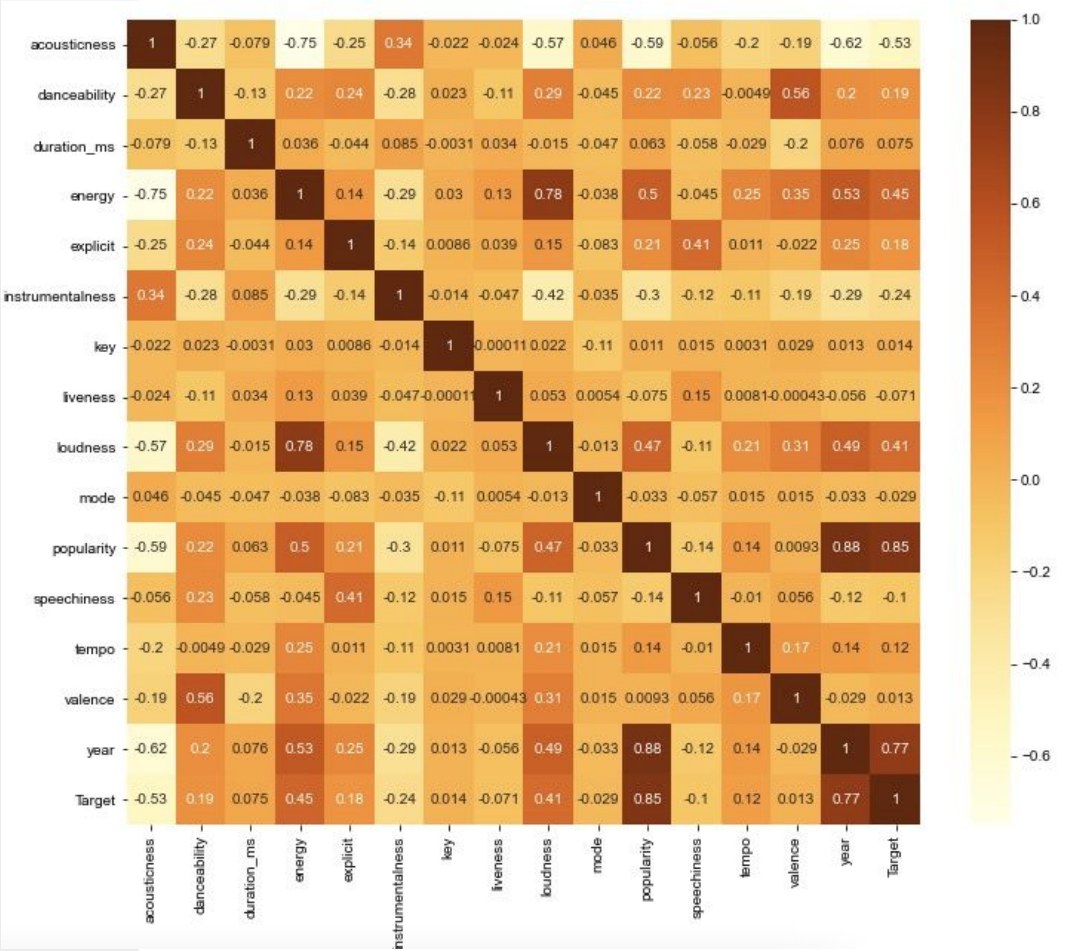
SpotifyRating.csv



artists	User1	User2	genres

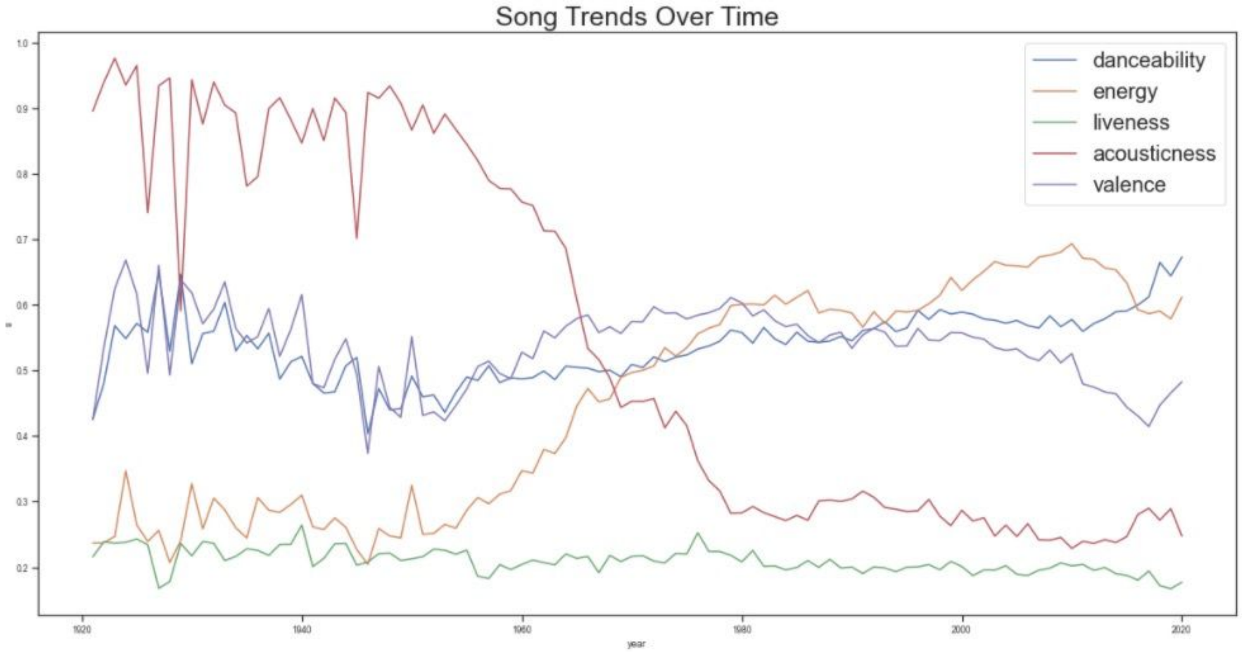
Part I: Feature Exploring and Visualization

Do the different features correlate with each other？

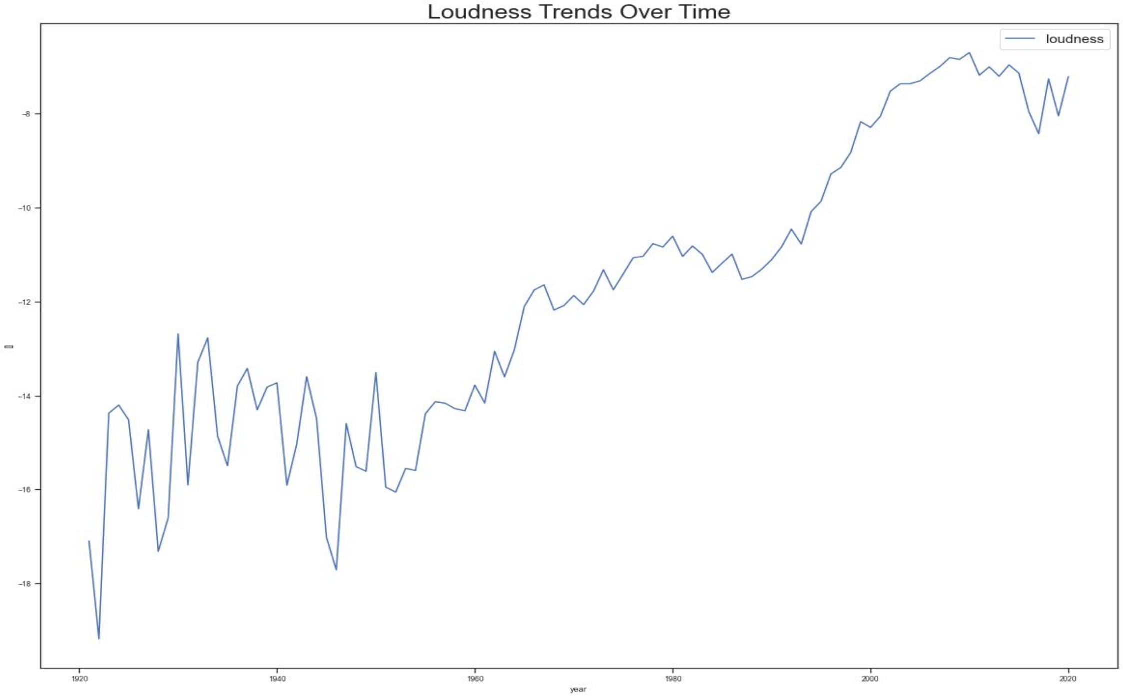


	acousticness	danceability	duration_ms	energy	explicit	instrumentalness	key	liveness	loudness	mode	popularity	speechiness	tempo	valence	year	Target
acousticness	1	-0.27	-0.079	-0.75	-0.25	0.34	-0.022	-0.024	-0.57	0.046	-0.59	-0.056	-0.2	-0.19	-0.62	-0.53
danceability	-0.27	1	-0.13	0.22	0.24	-0.28	0.023	-0.11	0.29	-0.045	0.22	0.23	-0.0049	0.56	0.2	0.19
duration_ms	-0.079	-0.13	1	0.036	-0.044	0.085	-0.0031	0.034	-0.015	-0.047	0.063	-0.058	-0.029	-0.2	0.076	0.075
energy	-0.75	0.22	0.036	1	0.14	-0.29	0.03	0.13	0.78	-0.038	0.5	-0.045	0.25	0.35	0.53	0.45
explicit	-0.25	0.24	-0.044	0.14	1	-0.14	0.0086	0.039	0.15	-0.083	0.21	0.41	0.011	-0.022	0.25	0.18
instrumentalness	0.34	-0.28	0.085	-0.29	-0.14	1	-0.014	-0.047	-0.42	-0.035	-0.3	-0.12	-0.11	-0.19	-0.29	-0.24
key	-0.022	0.023	-0.0031	0.03	0.0086	-0.014	1	-0.00011	0.022	-0.11	0.011	0.015	0.0031	0.029	0.013	0.014
liveness	-0.024	-0.11	0.034	0.13	0.039	-0.047	-0.00011	1	0.053	0.0054	-0.075	0.15	0.0081	-0.00043	-0.056	-0.071
loudness	-0.57	0.29	-0.015	0.78	0.15	-0.42	0.022	0.053	1	-0.013	0.47	-0.11	0.21	0.31	0.49	0.41
mode	0.046	-0.045	-0.047	-0.038	-0.083	-0.035	-0.11	0.0054	-0.013	1	-0.033	-0.057	0.015	0.015	-0.033	-0.029
popularity	-0.59	0.22	0.063	0.5	0.21	-0.3	0.011	-0.075	0.47	-0.033	1	-0.14	0.14	0.0093	0.88	0.85
speechiness	-0.056	0.23	-0.058	-0.045	0.41	-0.12	0.015	0.15	-0.11	-0.057	-0.14	1	-0.01	0.056	-0.12	-0.1
tempo	-0.2	-0.0049	-0.029	0.25	0.011	-0.11	0.0031	0.0081	0.21	0.015	0.14	-0.01	1	0.17	0.14	0.12
valence	-0.19	0.56	-0.2	0.35	-0.022	-0.19	0.029	-0.00043	0.31	0.015	0.0093	0.056	0.17	1	-0.029	0.013
year	-0.62	0.2	0.076	0.53	0.25	-0.29	0.013	-0.056	0.49	-0.033	0.88	-0.12	0.14	-0.029	1	0.77
Target	-0.53	0.19	0.075	0.45	0.18	-0.24	0.014	-0.071	0.41	-0.029	0.85	-0.1	0.12	0.013	0.77	1

Music features trends

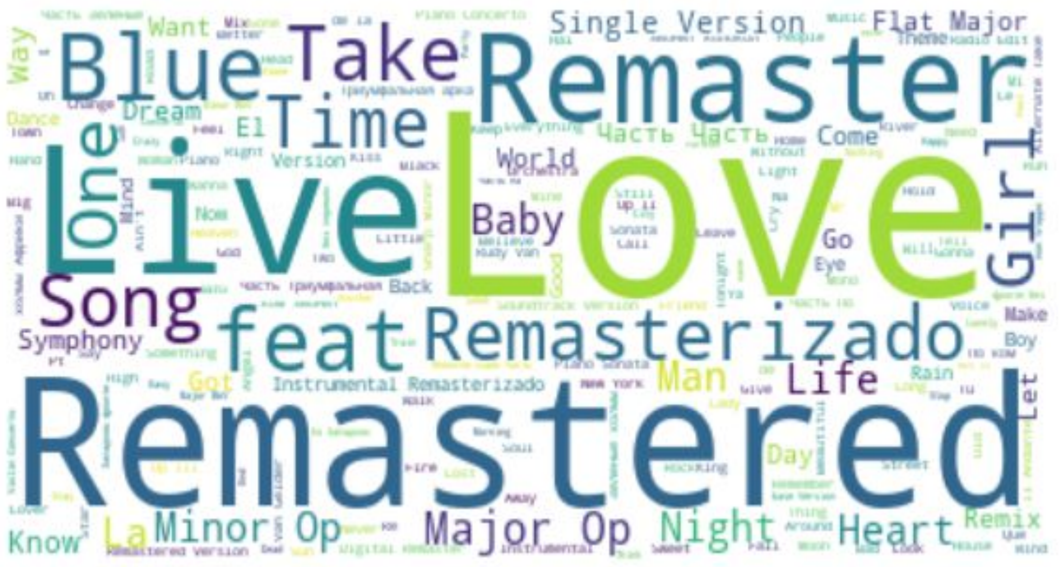


Music features trends

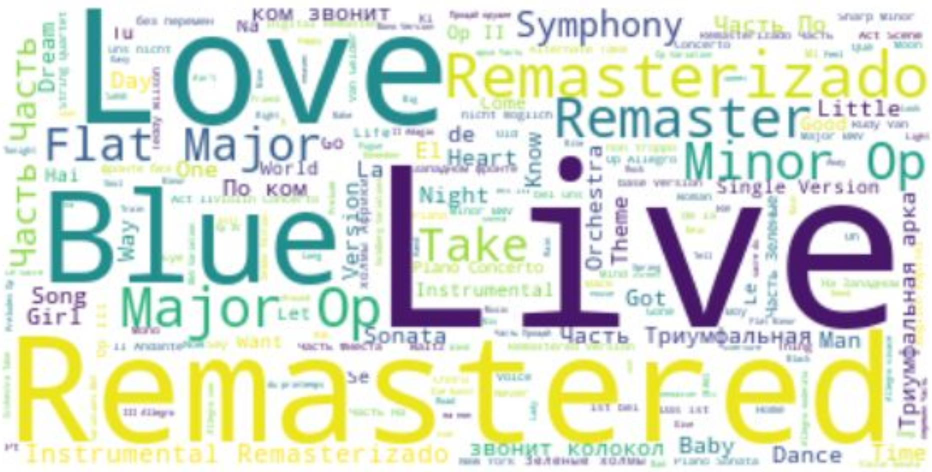


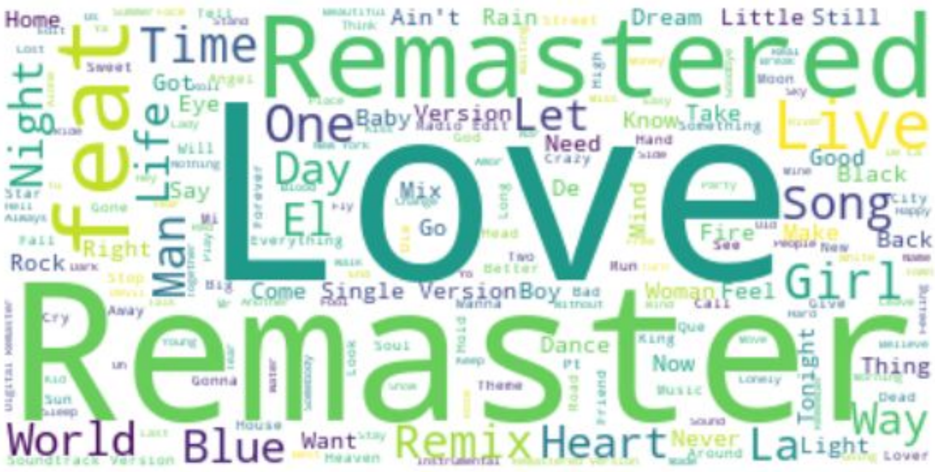
Text Mining

What kind of keywords are most common to use in the song names?



Keywords with low / high popularity

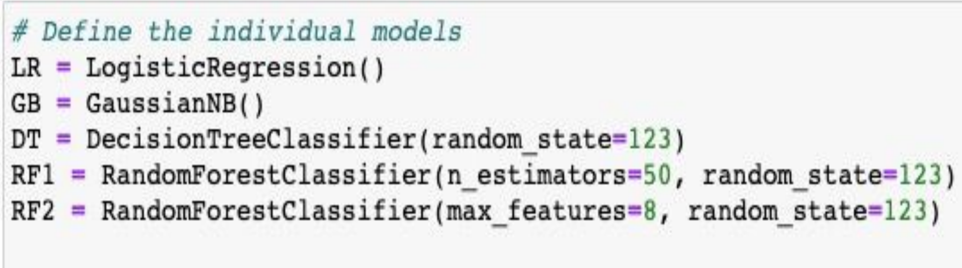




PART II: Classification Models

Predict popularity with different features.

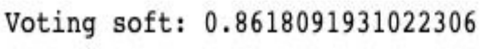
Logistic Regression, Naive Bayes, Decision Tree and the Random Forest as the classifiers.



```
# Define the individual models
LR = LogisticRegression()
GB = GaussianNB()
DT = DecisionTreeClassifier(random_state=123)
RF1 = RandomForestClassifier(n_estimators=50, random_state=123)
RF2 = RandomForestClassifier(max_features=8, random_state=123)
```

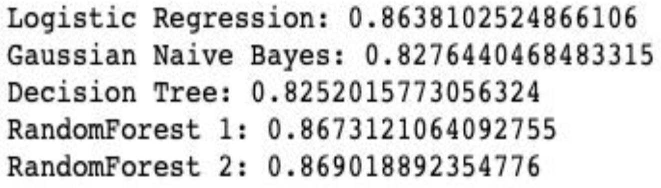
Soft Voting & Individual Models

Soft Voting:



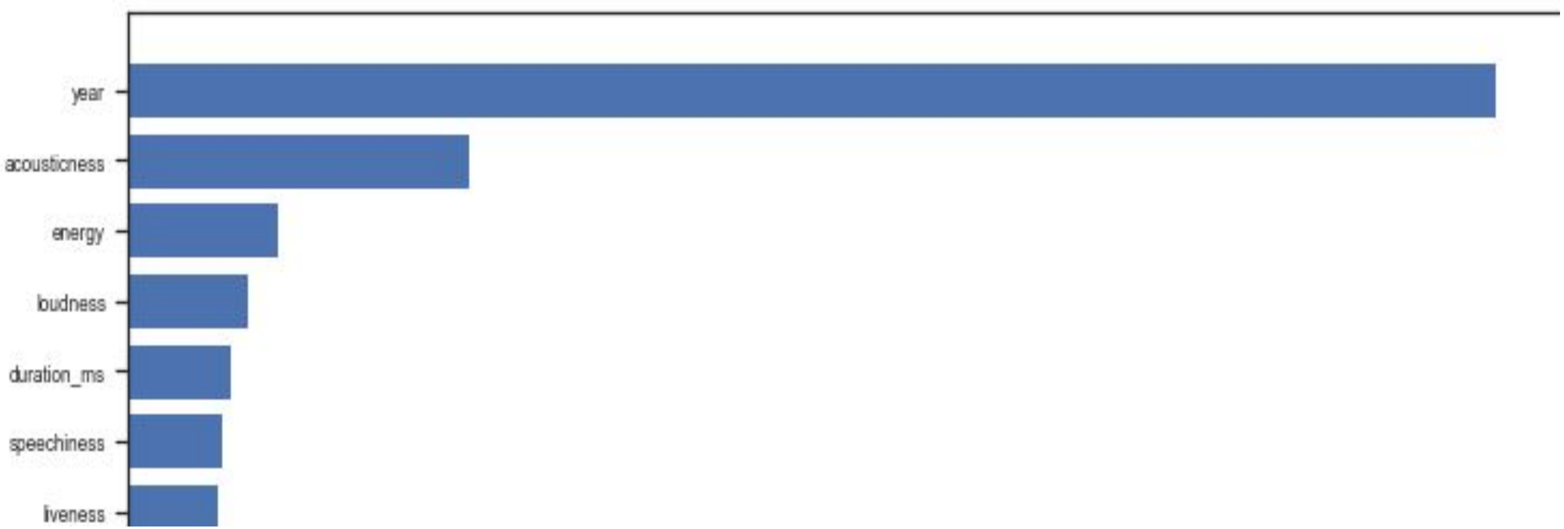
```
Voting soft: 0.8618091931022306
```

Individual Models:

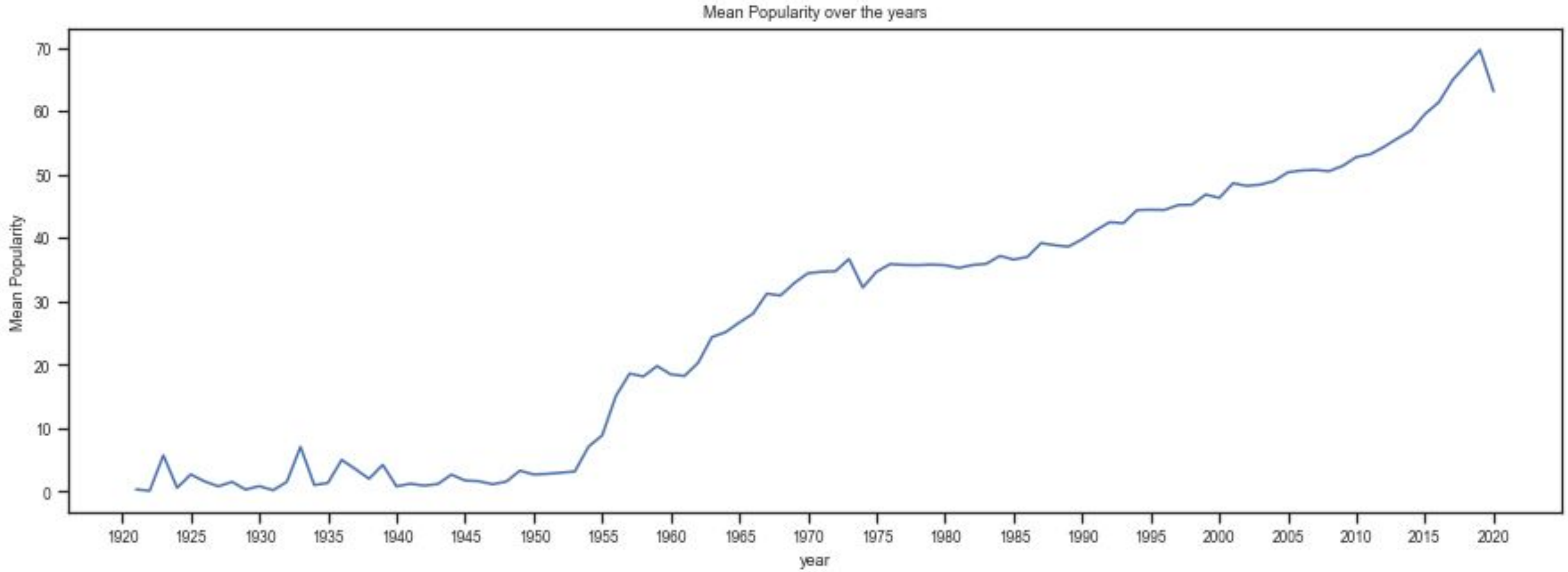


```
Logistic Regression: 0.8638102524866106
Gaussian Naive Bayes: 0.8276440468483315
Decision Tree: 0.8252015773056324
RandomForest 1: 0.8673121064092755
RandomForest 2: 0.869018892354776
```

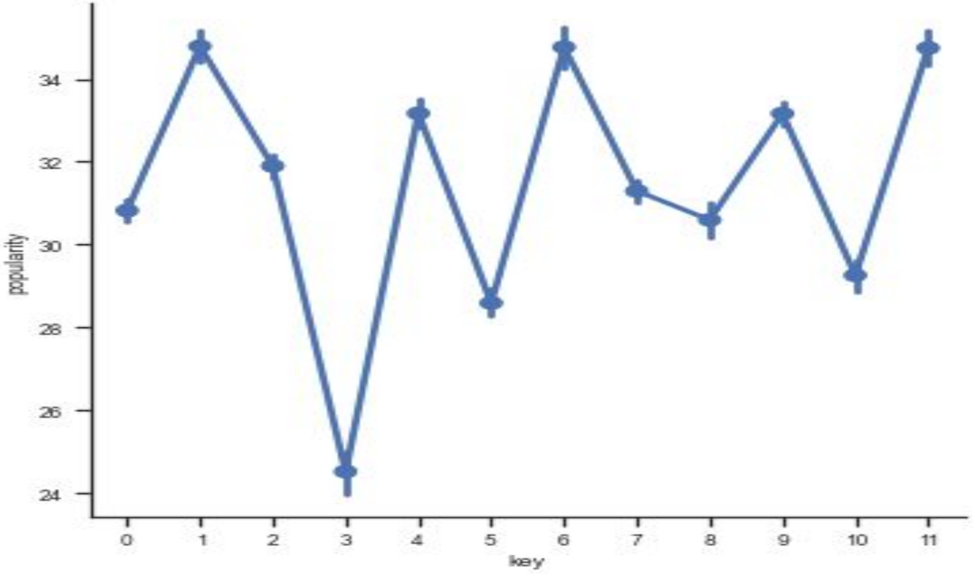
Feature Importance

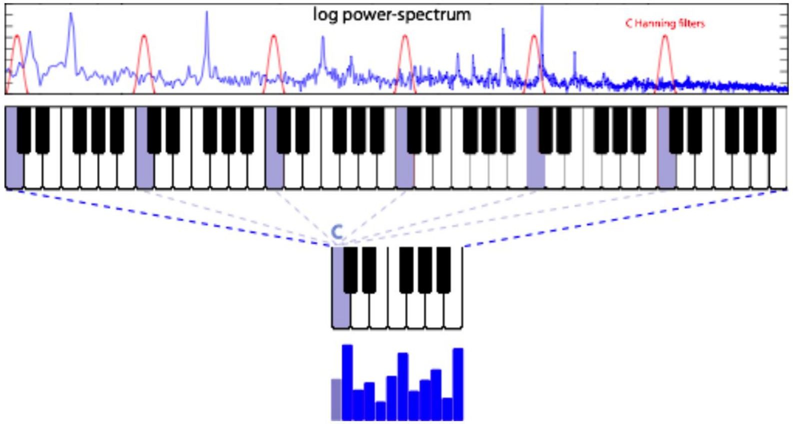


Year & Popularity



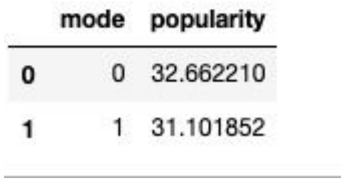
Key & Popularity



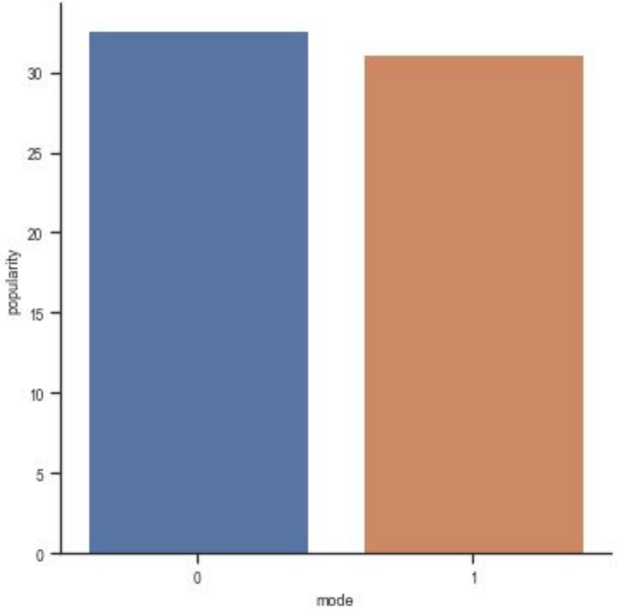


Mode & Popularity

Songs that start with a major (1) chord progression are slightly less popular than the songs that start with a non-major chord (0)



	mode	popularity
0	0	32.662210
1	1	31.101852



Length of songs & Popularity

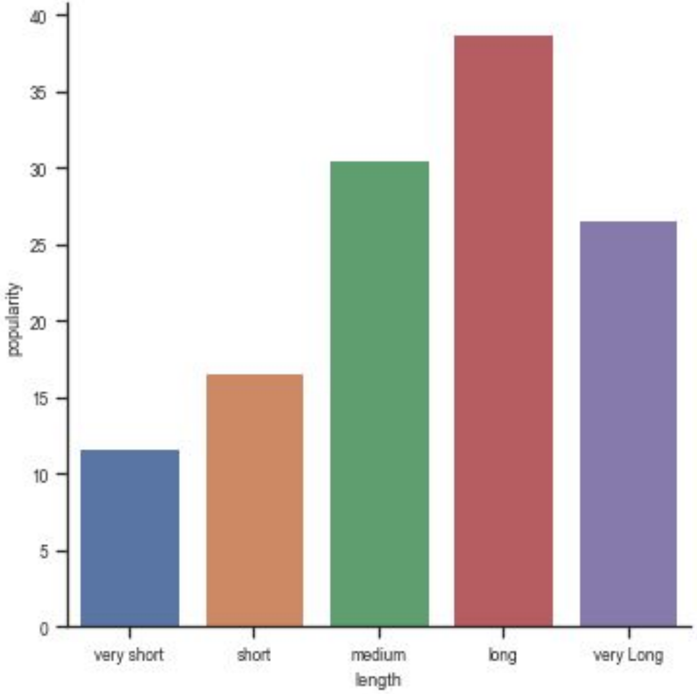
0–0:59 = very short

1:00-1:59 = short

2:00–3:59 = medium

4:00–5:59 = long

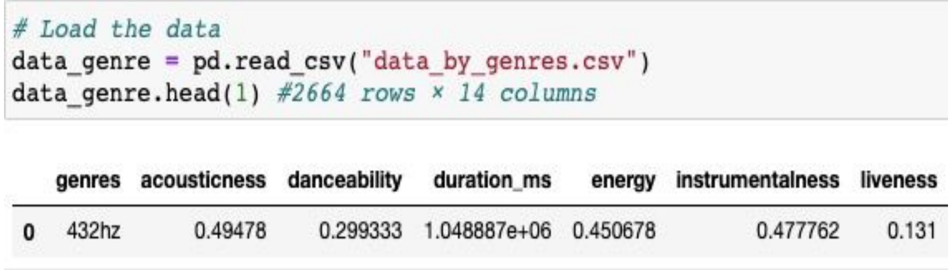
>=6:00 = very long



PART III: Clustering

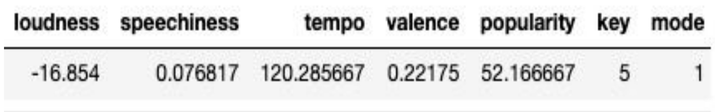
How could we cluster the genres?

Present the data:



```python
# Load the data
data_genre = pd.read_csv("data_by_genres.csv")
data_genre.head(1) #2664 rows × 14 columns
```

	genres	acousticness	danceability	duration_ms	energy	instrumentalness	liveness
0	432hz	0.49478	0.299333	1.048887e+06	0.450678	0.477762	0.131

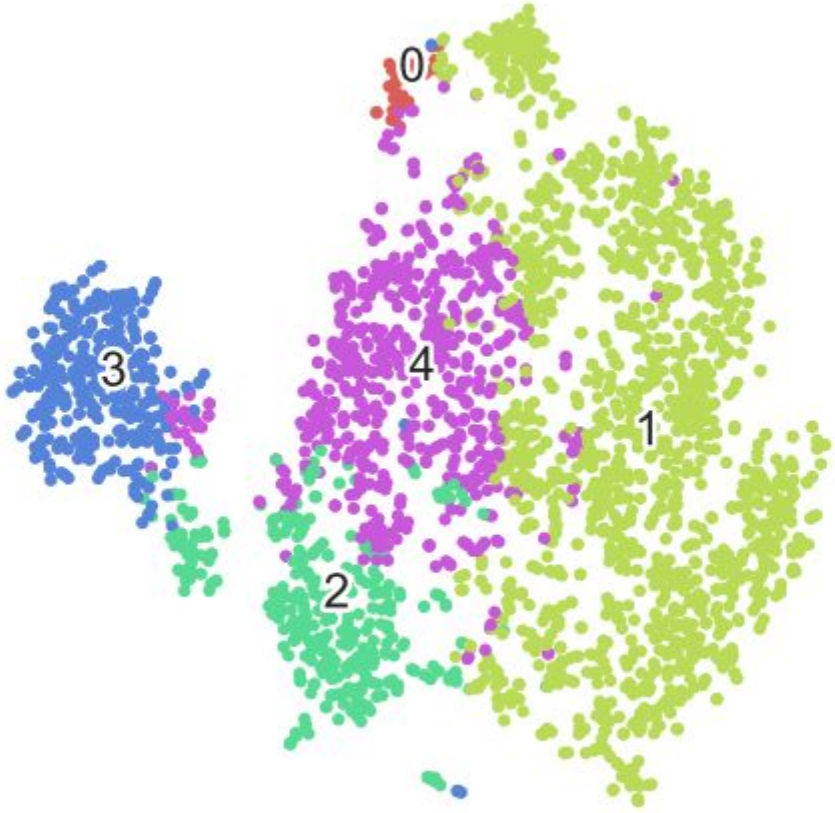


loudness	speechiness	tempo	valence	popularity	key	mode
-16.854	0.076817	120.285667	0.22175	52.166667	5	1

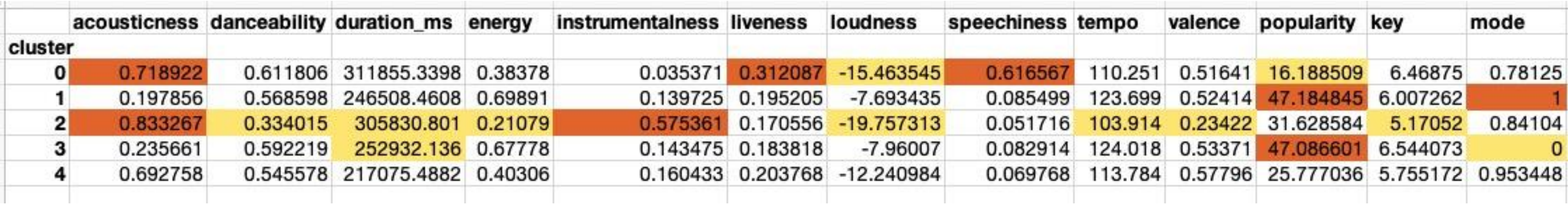
Best K

Test if 5 clusters is a good choice for our data

t-Distributed Stochastic Neighbor Embedding to generate the cluster plot.



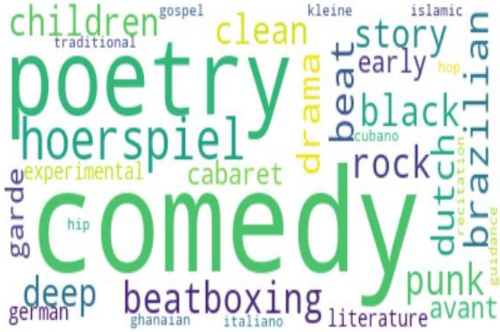
Performance of Different Clusters

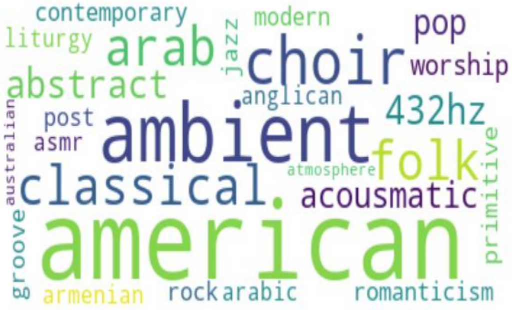


	acousticness	danceability	duration_ms	energy	instrumentalness	liveness	loudness	speechiness	tempo	valence	popularity	key	mode
cluster													
0	0.718922	0.611806	311855.3398	0.38378	0.035371	0.312087	-15.463545	0.616567	110.251	0.51641	16.188509	6.46875	0.78125
1	0.197856	0.568598	246508.4608	0.69891	0.139725	0.195205	-7.693435	0.085499	123.699	0.52414	47.184845	6.007262	1
2	0.833267	0.334015	305830.801	0.21079	0.575361	0.170556	-19.757313	0.051716	103.914	0.23422	31.628584	5.17052	0.84104
3	0.235661	0.592219	252932.136	0.67778	0.143475	0.183818	-7.96007	0.082914	124.018	0.53371	47.086601	6.544073	0
4	0.692758	0.545578	217075.4882	0.40306	0.160433	0.203768	-12.240984	0.069768	113.784	0.57796	25.777036	5.755172	0.953448

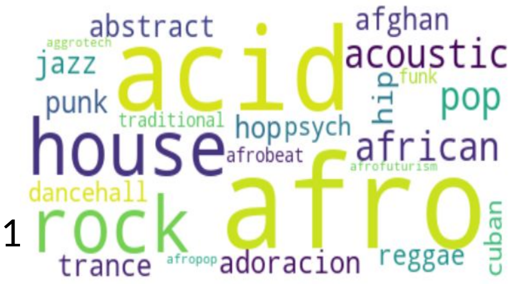
Which genres appear in different groups?

0

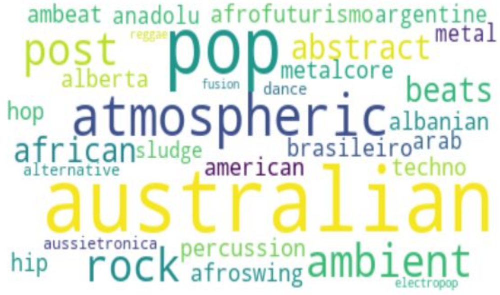




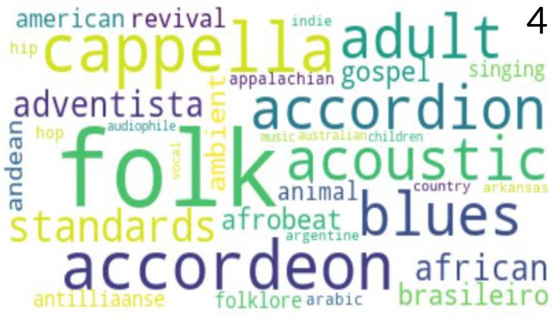
2



1



3

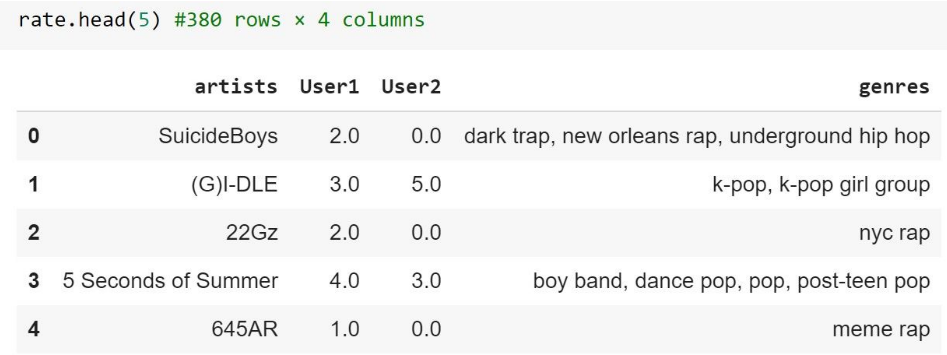


4

PART IV: Recommendation Engine

Which artist to recommend?

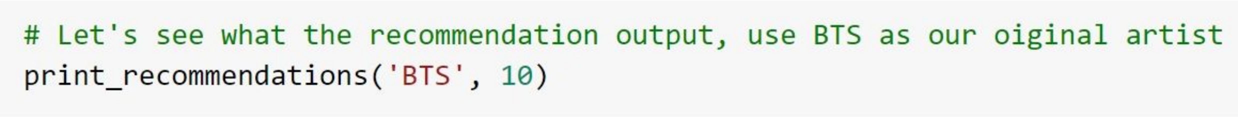
Present the data:



```
rate.head(5) #380 rows × 4 columns
```

	artists	User1	User2	genres
0	SuicideBoys	2.0	0.0	dark trap, new orleans rap, underground hip hop
1	(G)I-DLE	3.0	5.0	k-pop, k-pop girl group
2	22Gz	2.0	0.0	nyc rap
3	5 Seconds of Summer	4.0	3.0	boy band, dance pop, pop, post-teen pop
4	645AR	1.0	0.0	meme rap

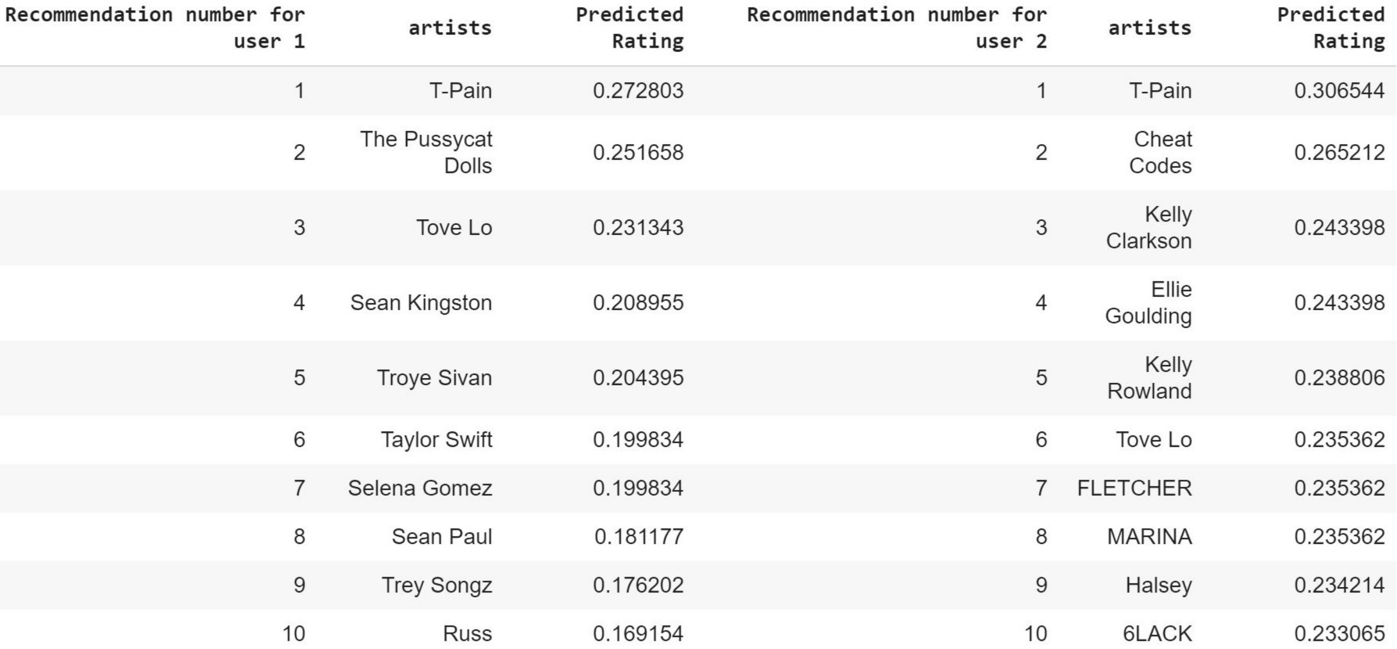
Use the item attribution to provide recommendation



```
# Let's see what the recommendation output, use BTS as our oiginal artist
print_recommendations('BTS', 10)
```

```
Your original artist is ['BTS']
My number  1  recommendation artist is  ['GOT7']
My number  2  recommendation artist is  ['Monsta X']
My number  3  recommendation artist is  ['NCT 127']
My number  4  recommendation artist is  ['NCT DREAM']
My number  5  recommendation artist is  ['TOMORROW X TOGETHER']
My number  6  recommendation artist is  ['CHUNG HA']
My number  7  recommendation artist is  ['BAEKHYUN']
My number  8  recommendation artist is  ['TWICE']
My number  9  recommendation artist is  ['ITZY']
My number  10  recommendation artist is  ['IZ*ONE']
```

Use the user rating profile to provide recommendation



Recommendation number for user 1	artists	Predicted Rating	Recommendation number for user 2	artists	Predicted Rating
1	T-Pain	0.272803	1	T-Pain	0.306544
2	The Pussycat Dolls	0.251658	2	Cheat Codes	0.265212
3	Tove Lo	0.231343	3	Kelly Clarkson	0.243398
4	Sean Kingston	0.208955	4	Ellie Goulding	0.243398
5	Troye Sivan	0.204395	5	Kelly Rowland	0.238806
6	Taylor Swift	0.199834	6	Tove Lo	0.235362
7	Selena Gomez	0.199834	7	FLETCHER	0.235362
8	Sean Paul	0.181177	8	MARINA	0.235362
9	Trey Songz	0.176202	9	Halsey	0.234214
10	Russ	0.169154	10	6LACK	0.233065

Conclusion

In conclusion, we performed data exploration and built the prediction model on the Spotify dataset. We also created the cluster model and recommendation system that performs relatively well as demonstrated above.

In the future, we can use this dataset to answer more questions such as, "What's the average length of songs for different artists?" or "Analyze the data of user's favorite artist".

References

- https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/
- https://medium.com/swlh/analyzing-spotify-data-with-pandas-96be8769fa57
- https://www.datacamp.com/community/tutorials/introduction-t-sne
- https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks/tasks

Q & A

Thank you