Spotify: Audio Features Application and Analysis

Ming-Hung Hung - mh57987
Ming-Hao Hsieh - mh57533
Han-Hsuan Lin - hl29622

Abstract

Outline

- Abstract
- Exploratory Data Analysis
- Popularity & Billboard 100 Prediction
- Music Recommendation System
- Future Step
- Reference

Why Spotify?

- Largest music streaming service provider
- Over 365 million monthly active users
- Over 60,000 tracks are now uploaded to Spotify every day
- Complete music data and audio features

Using audio features to build the music recommendation system

Workflow

- Gathering the data from Spotify API and Kaggle
- Raw data analysis
- Data preprocessing
- Modeling
- Recommendation system building
- Application deploying

Exploratory Data Analysis

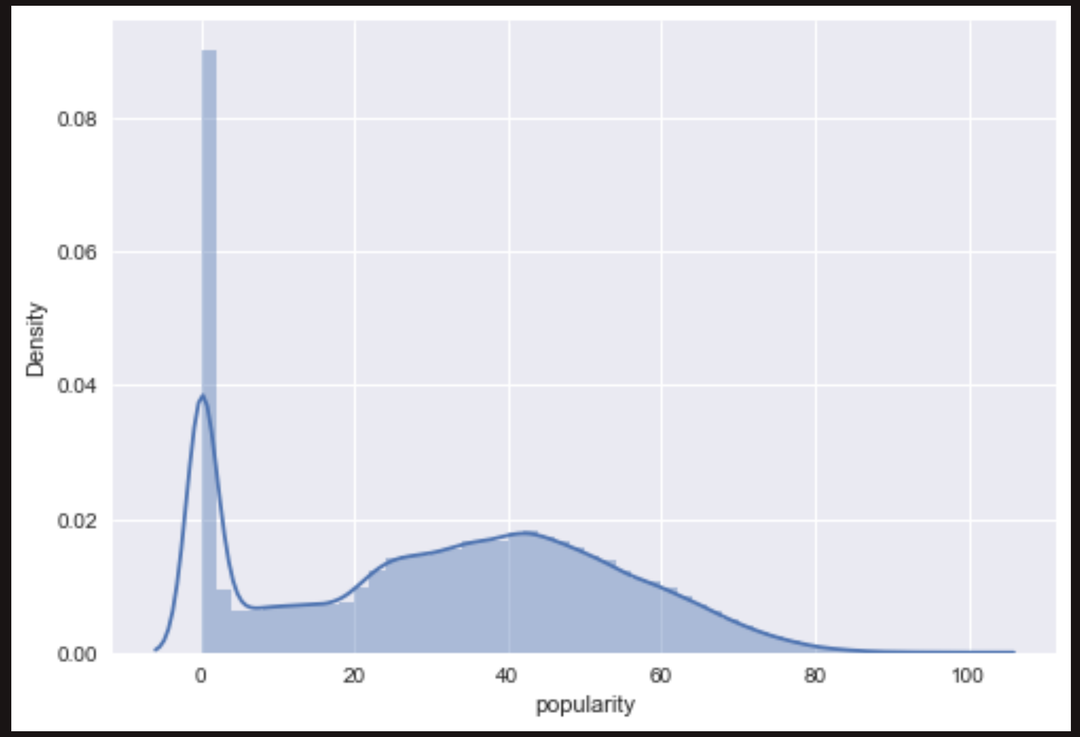
Data Source

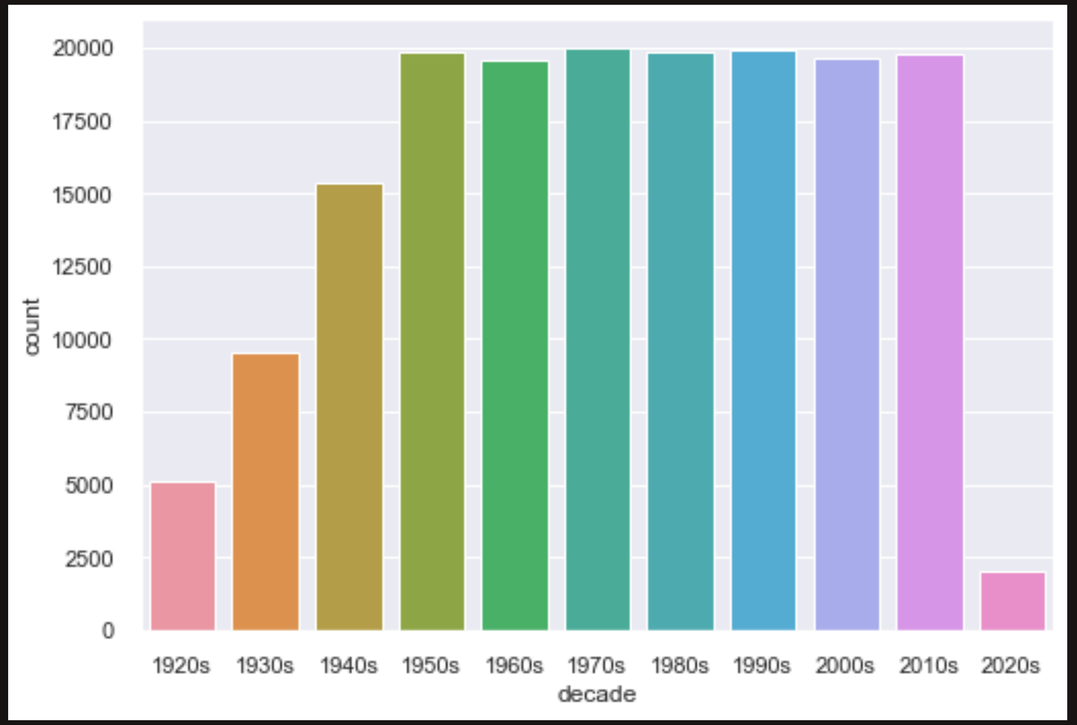
- Kaggle
  - dhruvildave/spotify-charts
  - Historical Data of the Top 200 Chart
- Spotify API
  - Track Data API
    - Track Metadata (e.g., name, artists, duration)
  - Track Analysis API
    - Audio Features (e.g., energy, key, tempo)
    - Generated by Spotify Audio Analyzer

Audio feature selection

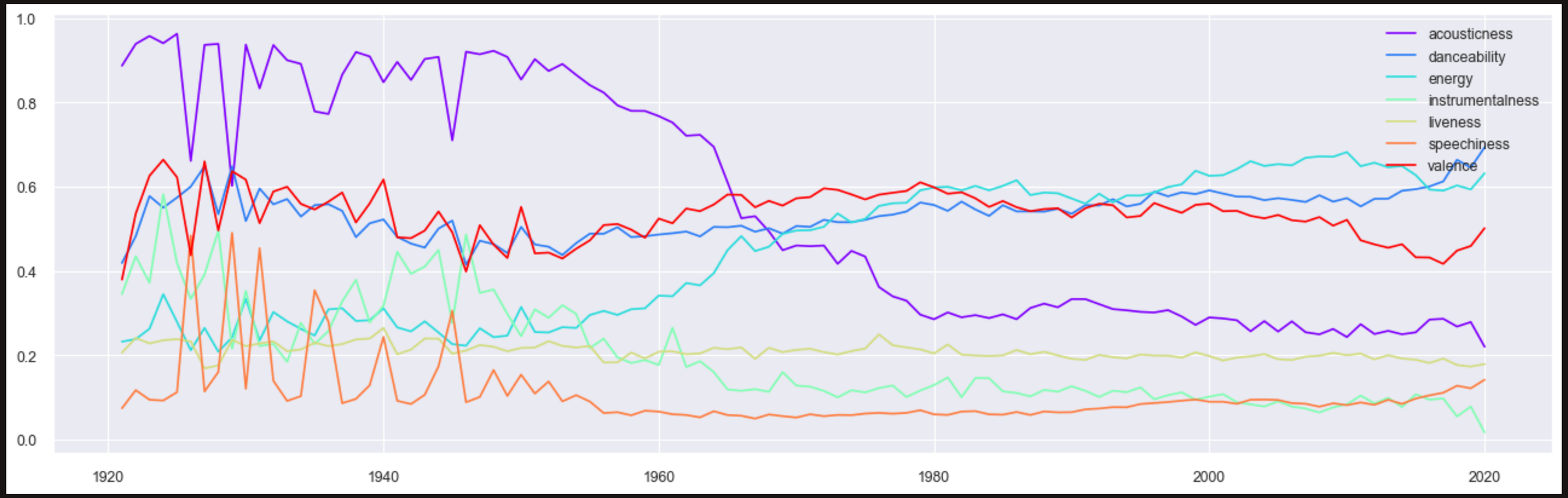
- ***Popularity***
- Year
- Acousticness
- Artists
- Danceability
- Duration_ms
- Engery
- Explicit
- Id
- Instrumentalness
- Key
- Liveness
- Loudness
- Mode
- Name
- Valence
- Release_date
- Speechiness
- Tempo

Charts


0.08
0.06
0.04
0.02
0.00
Density
0
20
40
60
80
100
popularity


20000
17500
15000
12500
10000
7500
5000
2500
0
count
1920s
1930s
1940s
1950s
1960s
1970s
1980s
1990s
2000s
2010s
2020s
decade

Charts


1.0
0.8
0.6
0.4
0.2
0.0
1920
1940
1960
1980
2000
2020
acousticness
danceability
energy
instrumentalness
liveness
speechiness
valence

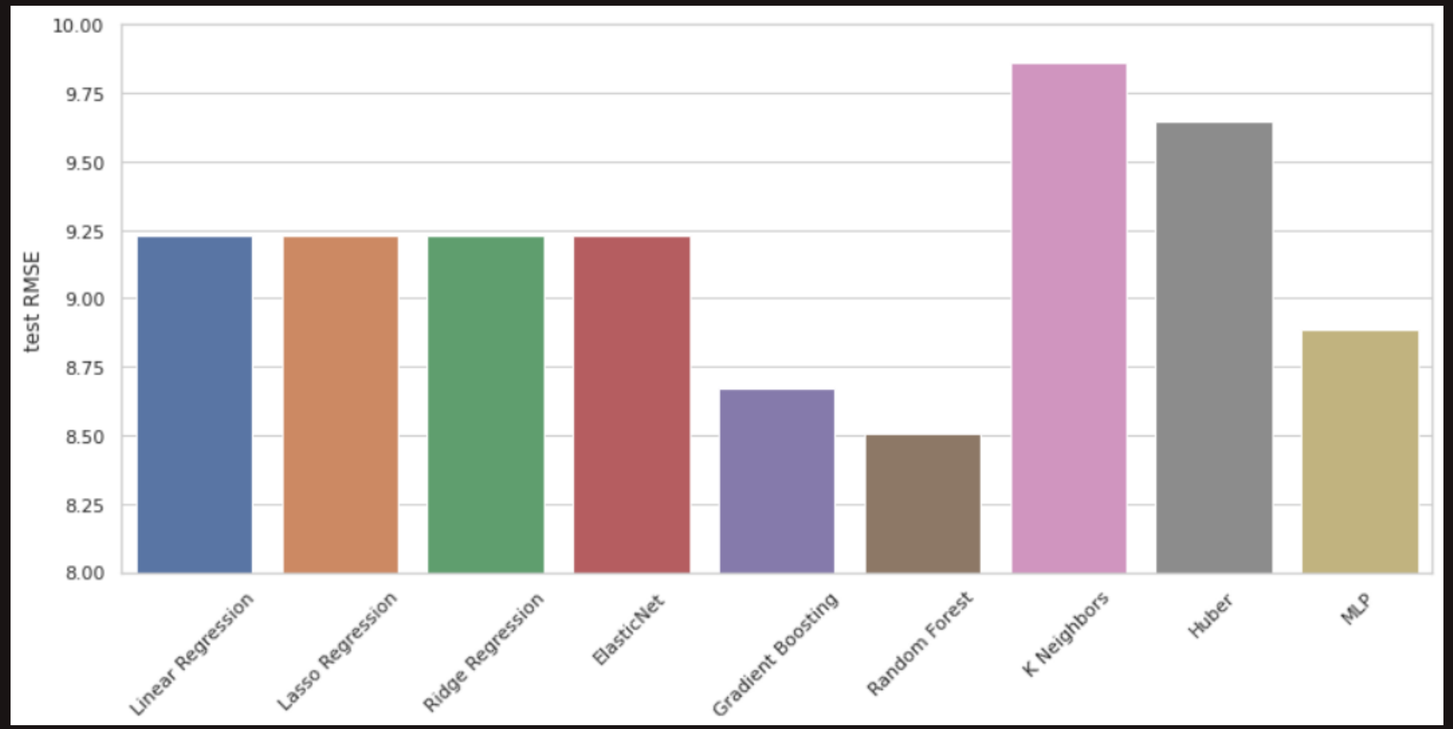
Popularity & Billboard 100 Predction

Data Preprocessing

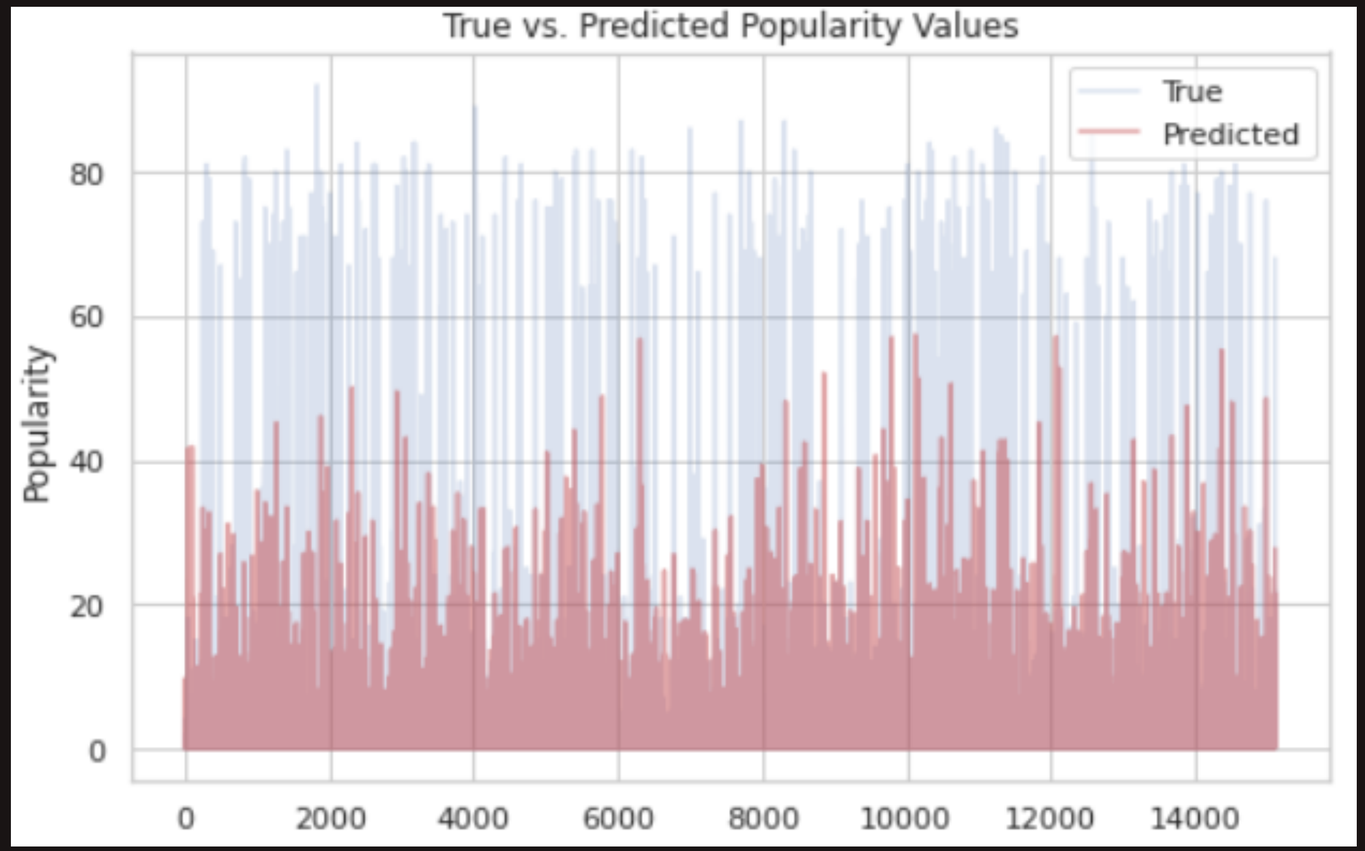
- Feature Selection: drop unnecessary columns (e.g., track_id)
- Apply “Label Encoding” to the feature airtist_name
- Deal with missing value
- Normalization
- Split the training and testing data (test_size = 0.33)

Popularity Prediction (Regression)

Train multiple models and evaluate them by RMSE:



Result Analysis



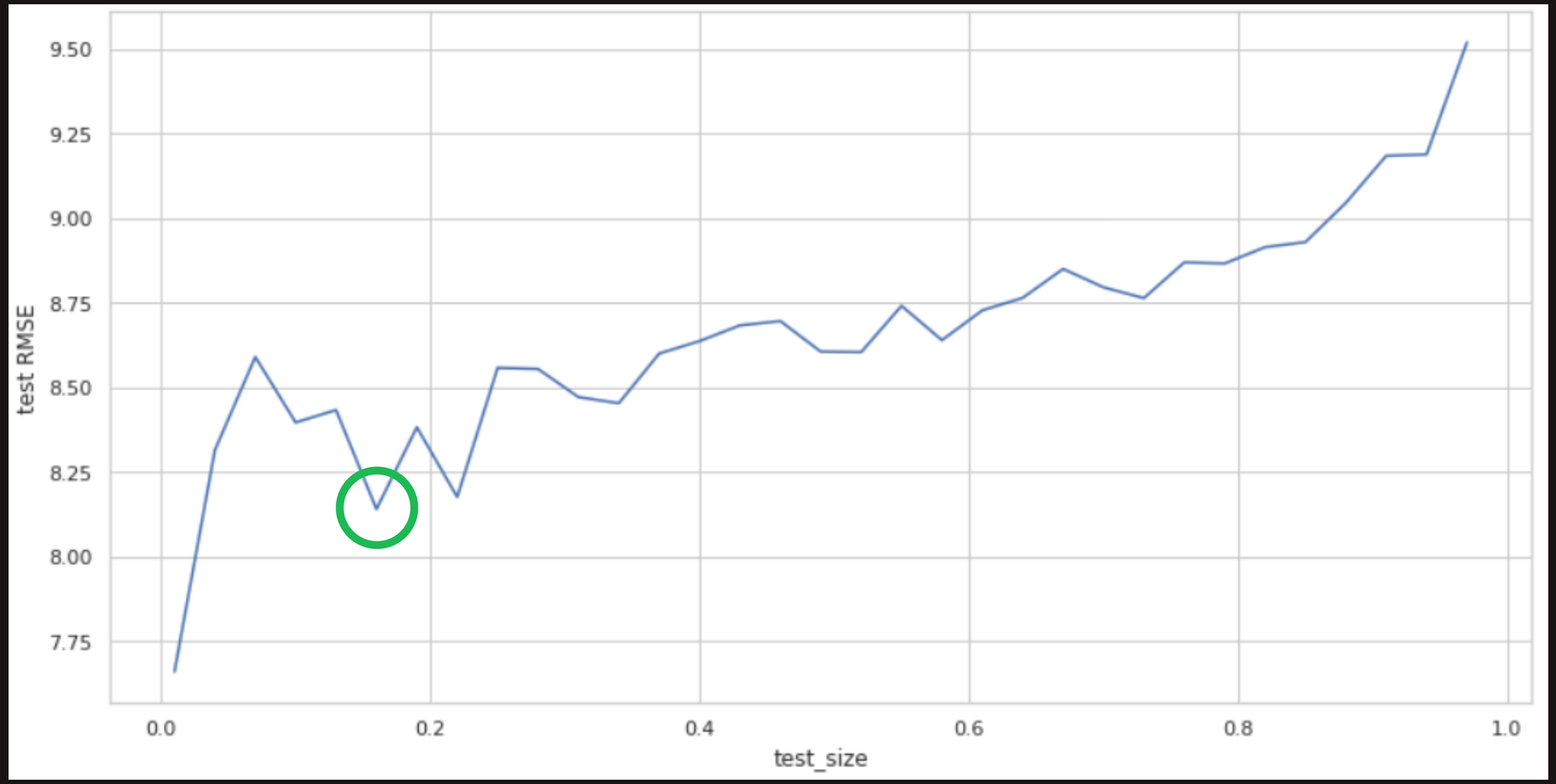
Total data size: 45770

Popularity > 50: 719

Popularity < 10: 43511

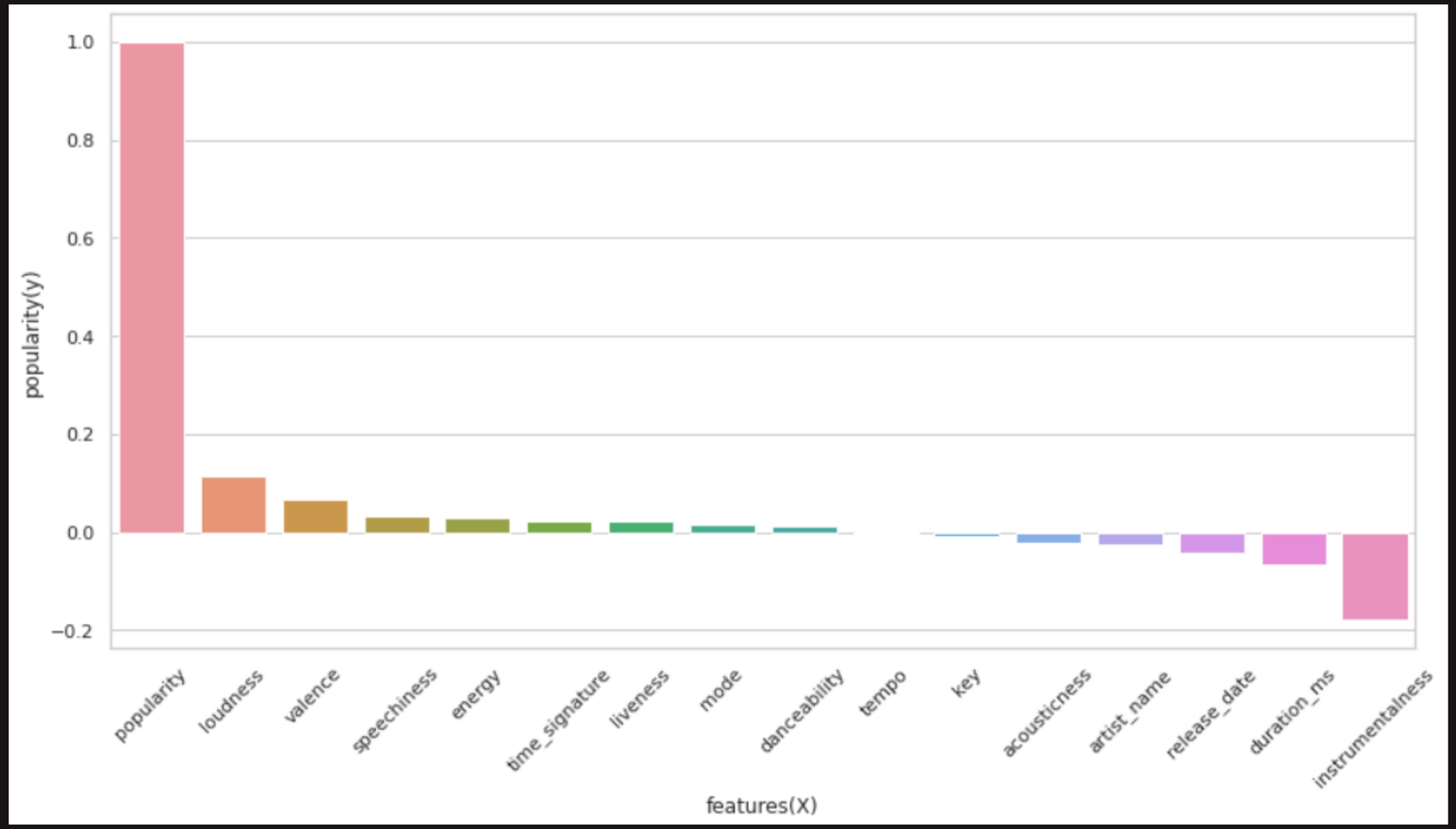
Model Enhancements

test_size tuning



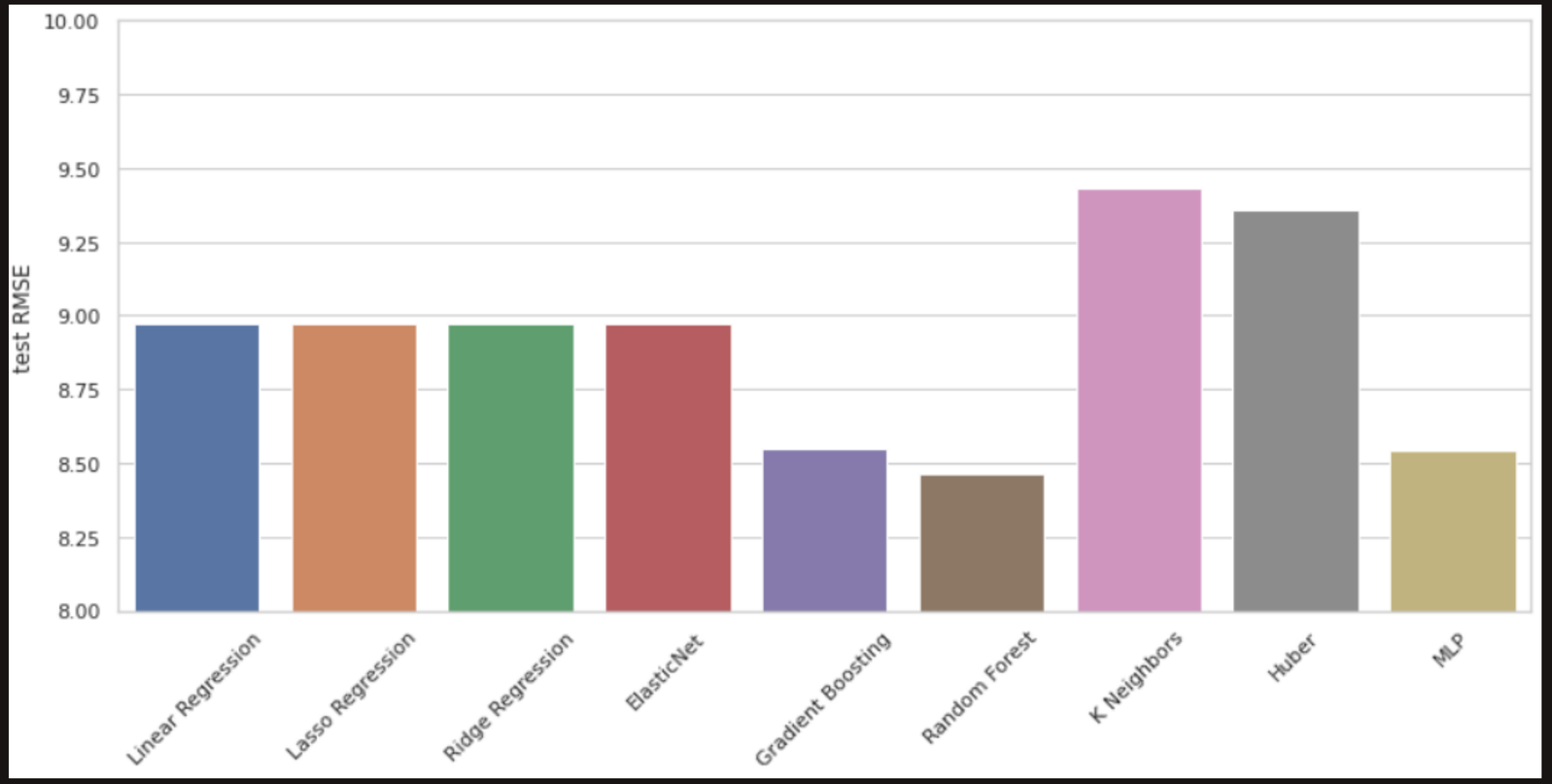
Model Enhancements

Correlation between $X_{(s)}$ & y



Model Enhancements

Result



Enhanced Testing RMSE by %	
Linear Regression	2.785092
Lasso Regression	2.779623
Ridge Regression	2.781009
ElasticNet	2.780521
Gradient Boosting	1.415410
Random Forest	0.245947
K Neighbors	4.360819
Huber	2.976659
MLP	3.885918

Billboard 100 Prediction

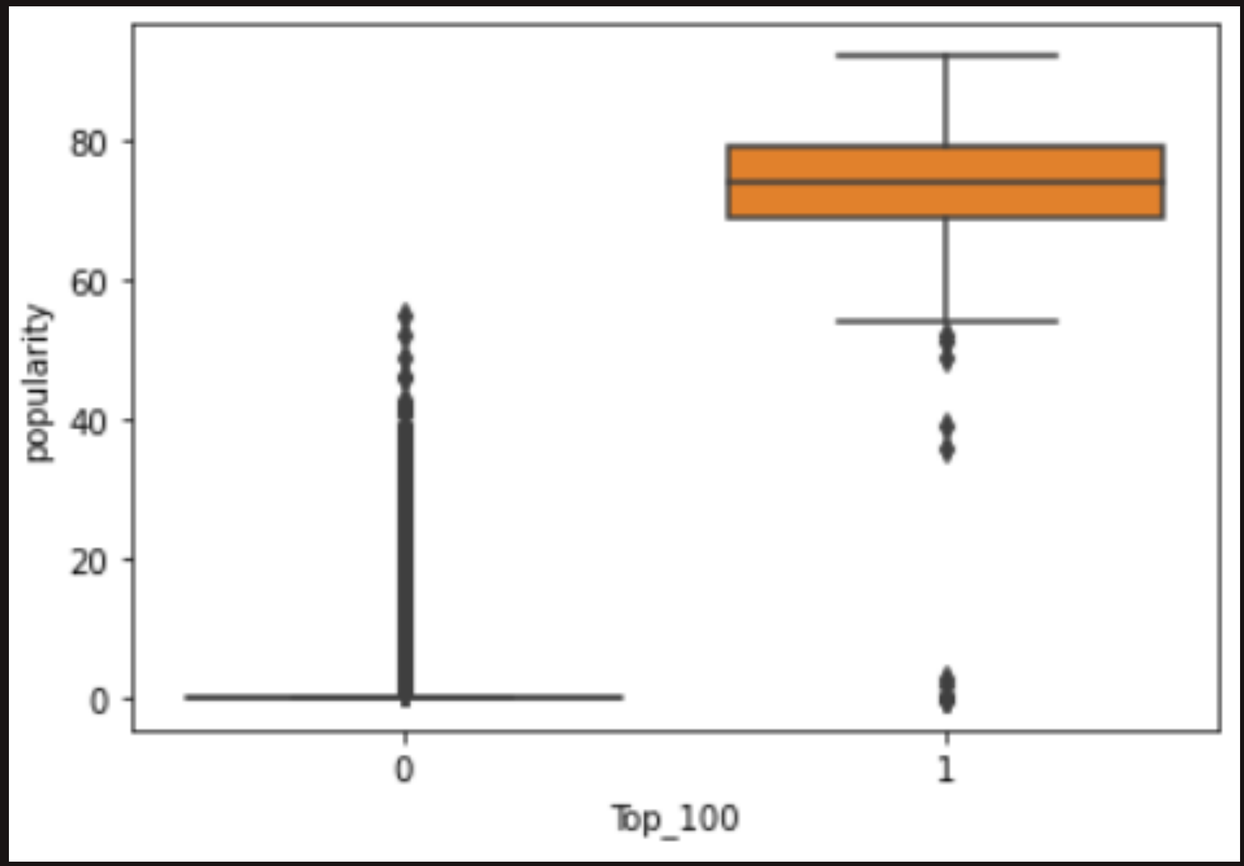
(Classification)

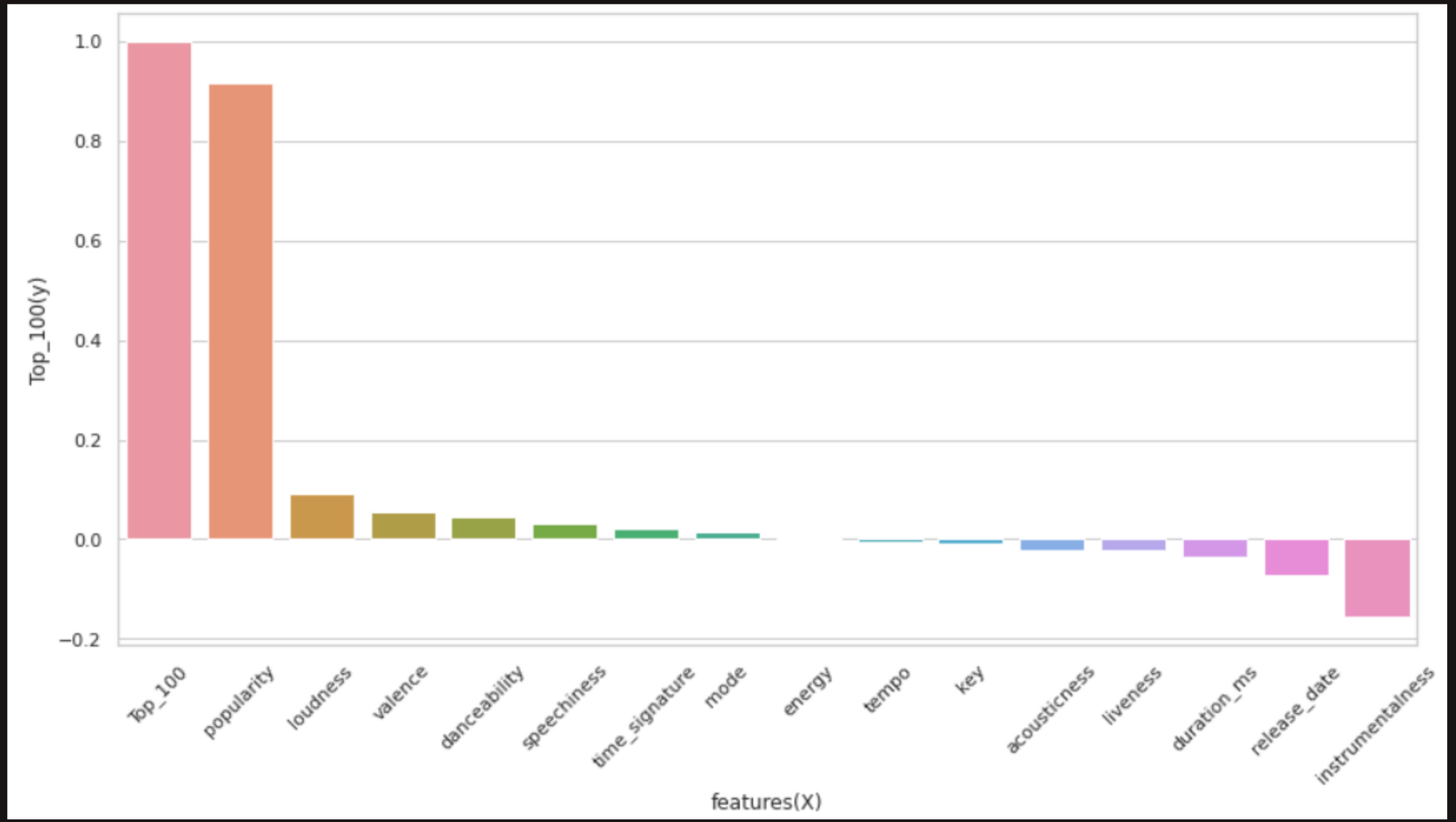
Train data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	6205
1	1.00	0.97	0.98	60
accuracy			1.00	6265
macro avg	1.00	0.98	0.99	6265
weighted avg	1.00	1.00	1.00	6265

Test data:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5588
1	1.00	0.96	0.98	84
accuracy			1.00	5672
macro avg	1.00	0.98	0.99	5672
weighted avg	1.00	1.00	1.00	5672

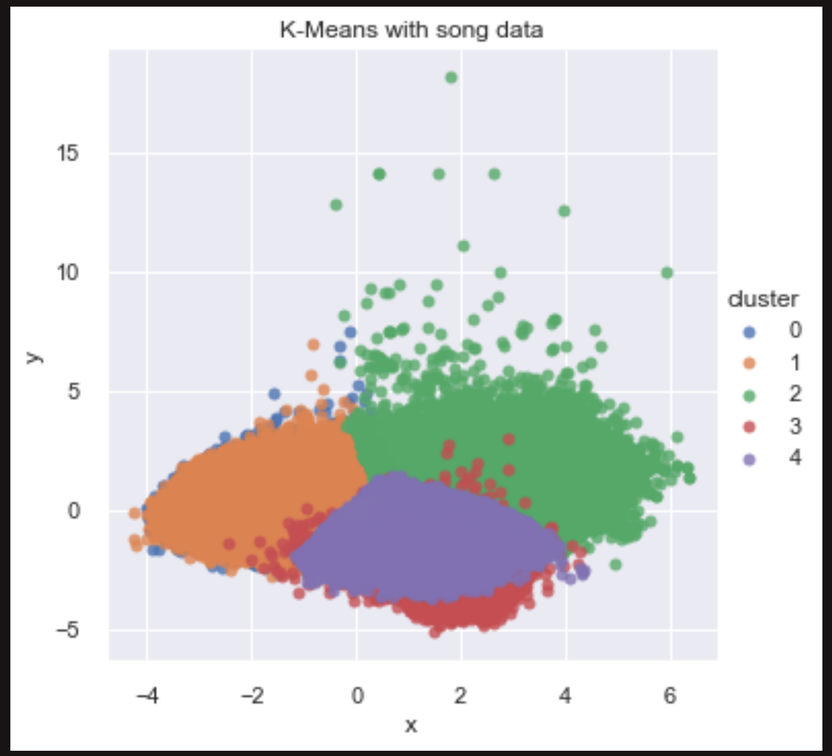


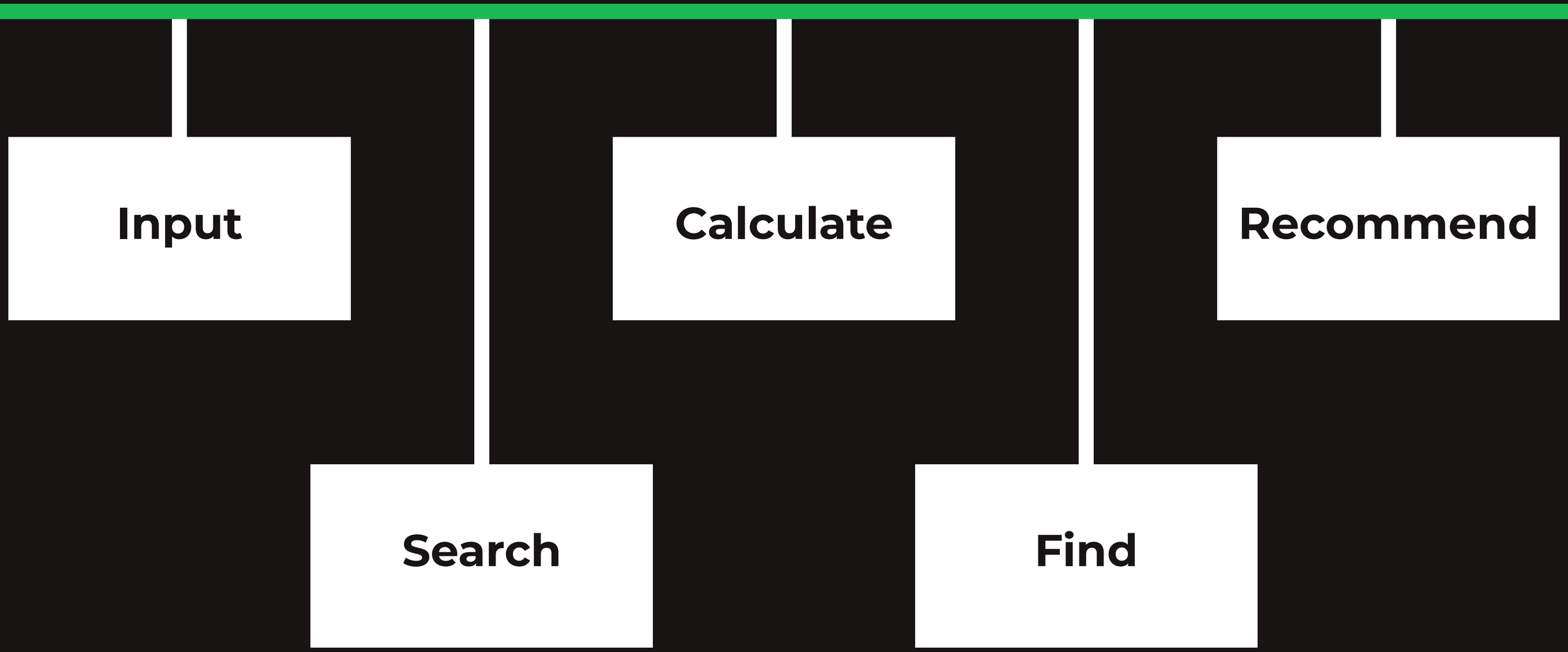

1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Top_100(y)
Top_100
popularity
loudness
valence
danceability
speechiness
time_signature
mode
energy
tempo
key
acousticness
liveness
duration_ms
release_date
instrumentalness
features(X)

Music Recommendation System

Key Idea

- K-Means Clustering - To partition the data into k clusters where each data belongs the cluster with the nearest mean.




Input
Search
Calculate
Find
Recommend

```
result = recommend([{'name':'STAY', 'year':2021}])
print_result(result)
```

✓ 0.2s

```
Top 1 recommended song
Track name: Blinding Lights, year: 2020, artist: ['The Weeknd']
Top 2 recommended song
Track name: Come & Go (with Marshmello), year: 2020, artist: ['Juice WRLD', 'Marshmello']
Top 3 recommended song
Track name: bloody valentine, year: 2020, artist: ['Machine Gun Kelly']
Top 4 recommended song
Track name: Hasta la Raíz, year: 2015, artist: ['Natalia Lafourcade']
Top 5 recommended song
Track name: Thunder, year: 2017, artist: ['Imagine Dragons']
Top 6 recommended song
Track name: They Don't Know About Us, year: 2012, artist: ['One Direction']
Top 7 recommended song
Track name: Inside Out (feat. Griff), year: 2020, artist: ['Zedd', 'Griff']
Top 8 recommended song
Track name: Secrets, year: 2009, artist: ['OneRepublic']
Top 9 recommended song
Track name: Lose Somebody, year: 2020, artist: ['Kygo', 'OneRepublic']
Top 10 recommended song
Track name: obvious, year: 2020, artist: ['Ariana Grande']
```

```python
playlist = [{'name':'Wake Me Up', 'year':2014}, {'name':'The Nights', 'year':2014}, {'name':'I Really Like You', 'year':2015},{'name':'Tik Tok',
'year':2010}]
```

✓ 0.2s Python

```python
result = recommend(playlist)
print_result(result)
```

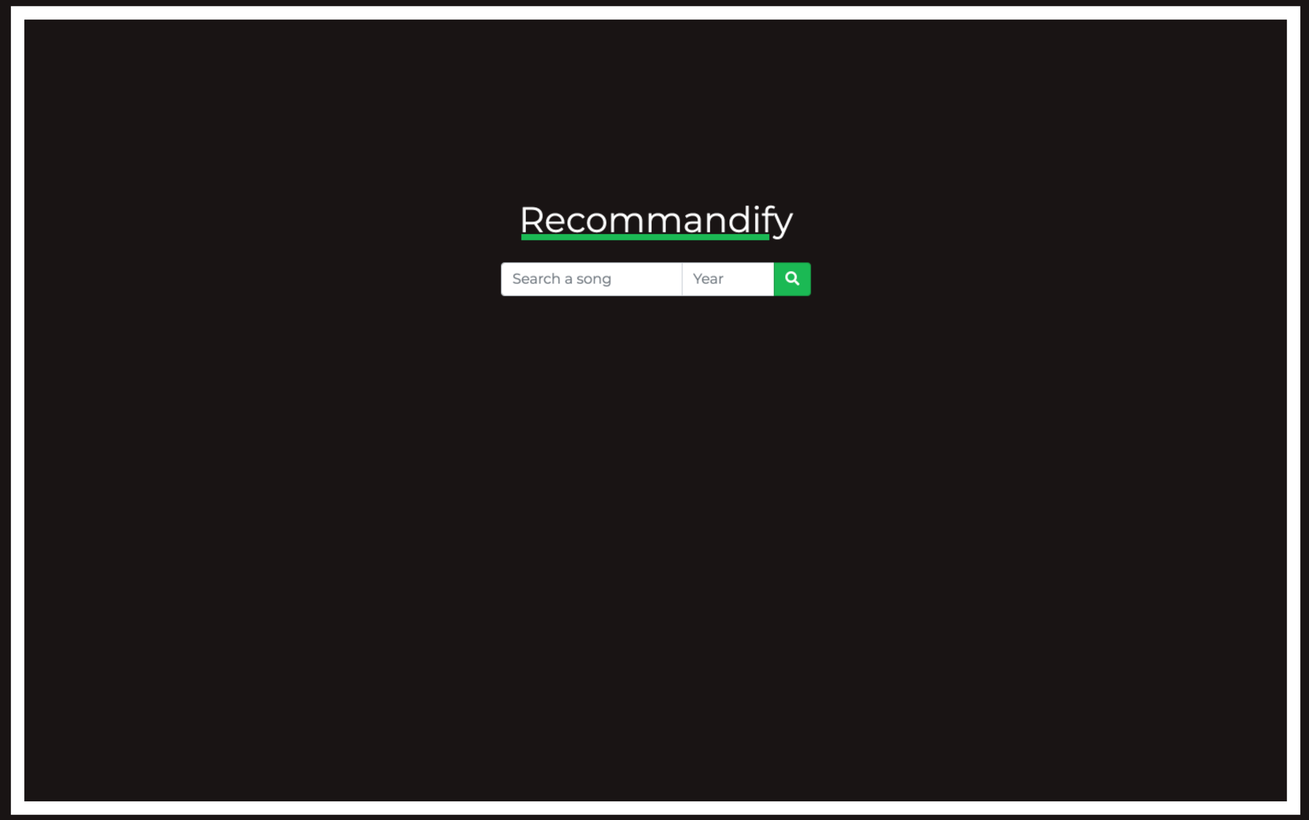
✓ 0.3s Python

```
Top 1 recommended song
Track name: Shut Up and Dance, year: 2014, artist: ['WALK THE MOON']
Top 2 recommended song
Track name: La Da Dee, year: 2019, artist: ['Cody Simpson']
Top 3 recommended song
Track name: Lovin' On You, year: 2019, artist: ['Luke Combs']
Top 4 recommended song
Track name: Lonely, year: 2020, artist: ['Joel Corry']
Top 5 recommended song
Track name: Where Them Girls At (feat. Nicki Minaj & Flo Rida), year: 2011, artist: ['David Guetta', 'Flo Rida', 'Nicki Minaj']
Top 6 recommended song
Track name: eight(Prod.&Feat. SUGA of BTS), year: 2020, artist: ['IU', 'SUGA']
Top 7 recommended song
Track name: Cut To The Feeling, year: 2017, artist: ['Carly Rae Jepsen']
Top 8 recommended song
Track name: Shut Up and Dance, year: 2014, artist: ['WALK THE MOON']
Top 9 recommended song
Track name: Love Me, year: 2009, artist: ['Justin Bieber']
Top 10 recommended song
Track name: Highway Tune, year: 2017, artist: ['Greta Van Fleet']
```

Recommendify

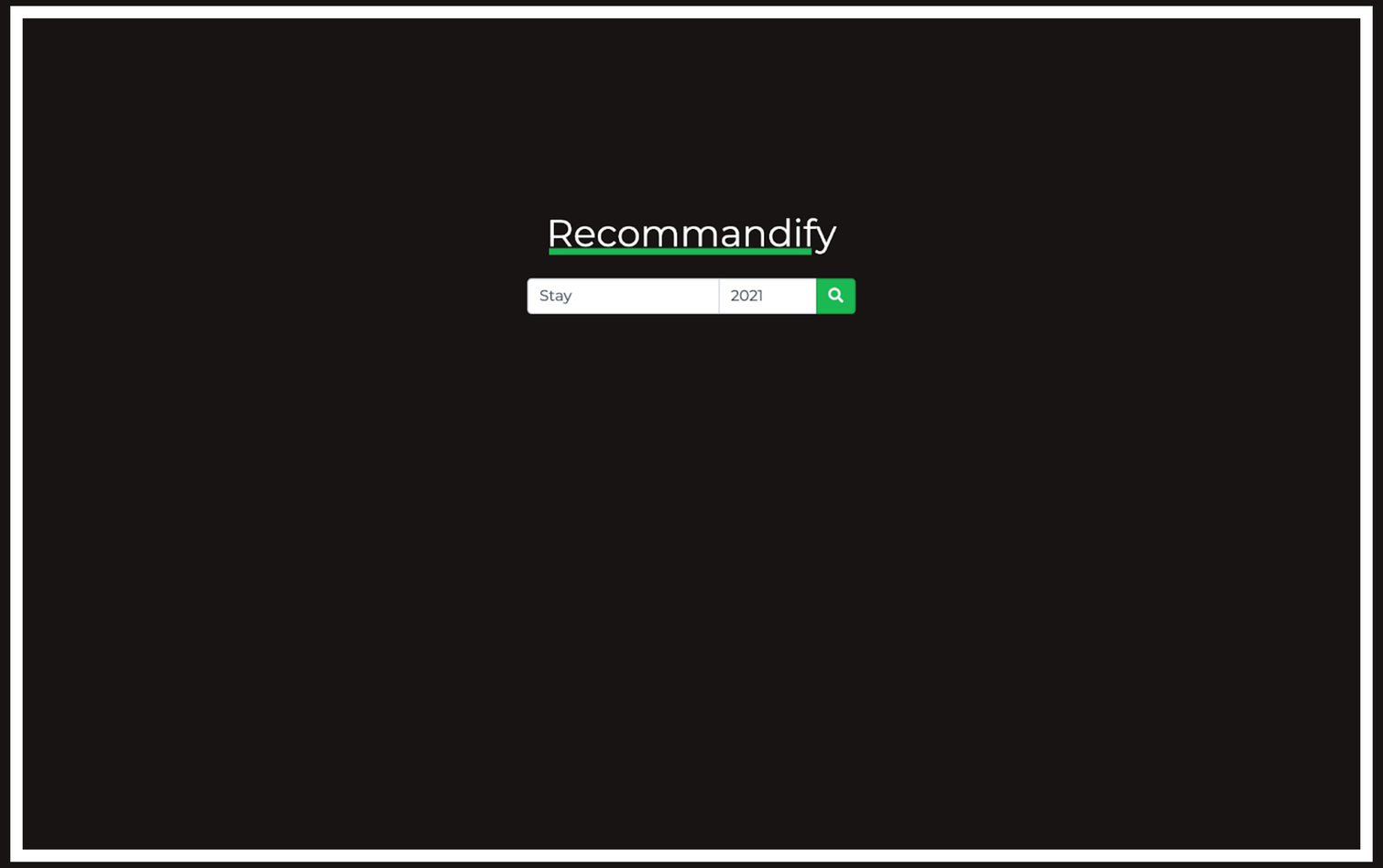
Application in the real world

- Interactive Recommendation System
- Website server
  - JQuery, Bootstrap
  - Node.js as backend
- Model server
  - Python-based
  - Flask as backend
- Deployed on Heroku

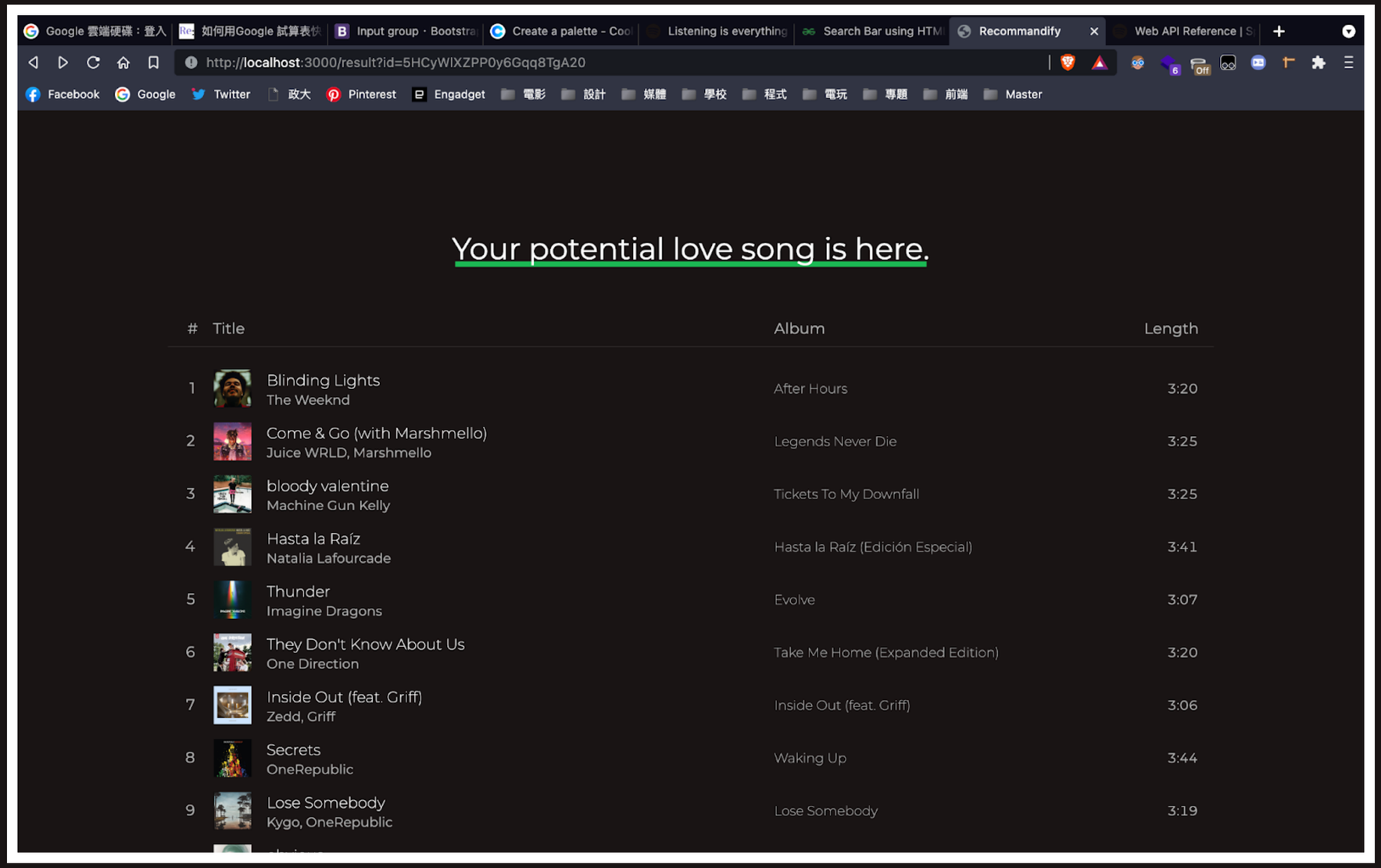


Search

Search with track names and years



Recommendation Result



Future Work

Future Work

- Building Pipeline to update the model periodically
- Export predicted playlist straight to Spotify
- Complete the recommendify and deploy to production environment

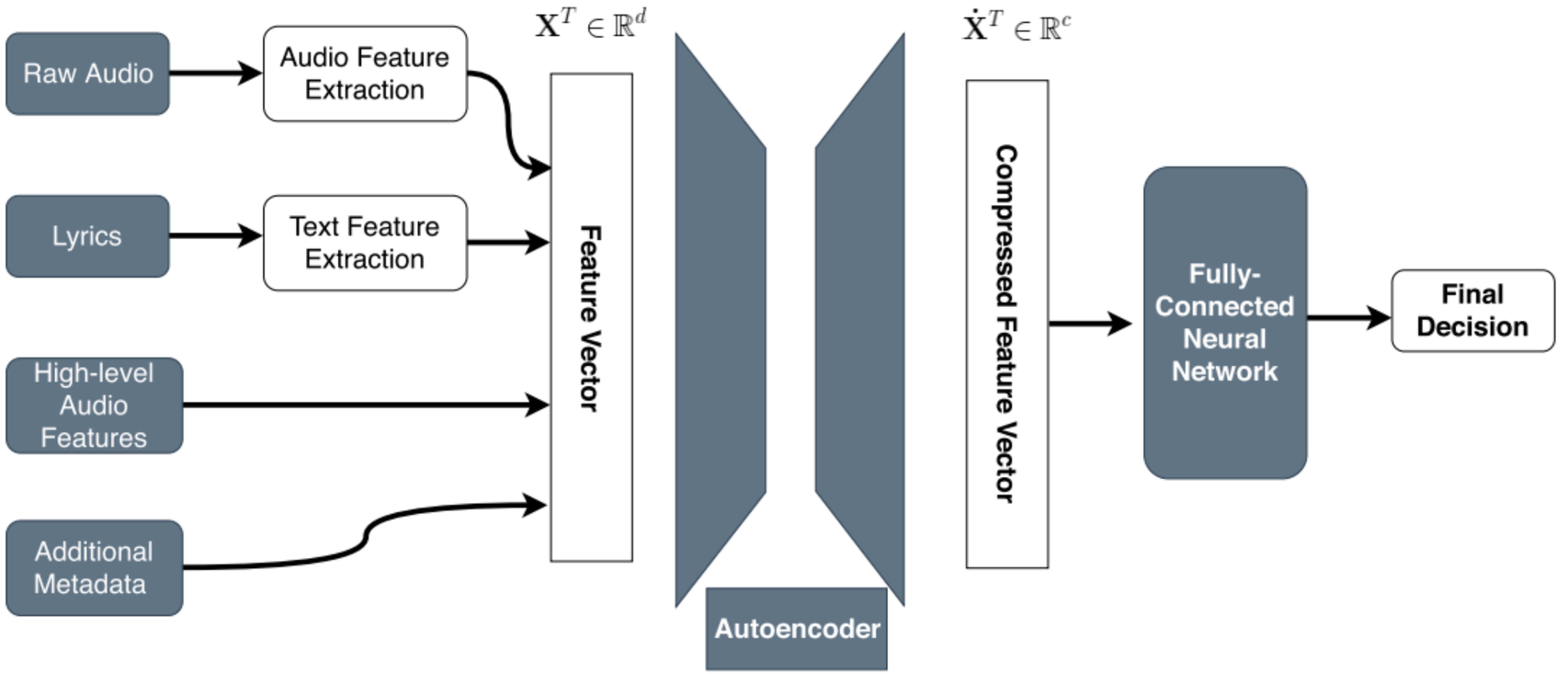


FIGURE 2. A general block schema outlining the principal functionalities and data components that form the proposed music popularity prediction system. After applying a feature extraction procedure for both the raw audio and the lyric components, the system obtains a high-dimensional feature vector and then, two additional steps are followed: a feature compression stage using an Autoencoder and a classifier via a fully-connected Neural Network.

Reference

- https://www.kaggle.com/dhruvildave/billboard-the-hot-100-songs
- https://github.com/tgel0/spotify-data/blob/master/notebooks/SpotifyDataRetrieval.ipynb
- https://spotipy.readthedocs.io/en/2.19.0/
- https://towardsdatascience.com/finding-and-removing-duplicate-rows-in-pandas-dataframe-c6117668631f
- https://developer.spotify.com/documentation/web-api/reference/#/
- https://github.com/AmolMavuduru/SpotifyRecommenderSystem
- https://towardsdatascience.com/how-to-build-an-amazing-music-recommendation-system-4cce2719a572
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9007339

Thank You!

Any Questions?