SPOTIFY SONGS POPULARITY VISUALIZATION AND ANALYSIS

Simrik Rijal (300340875)

Sisir Ghimire Chettri (300340871)

OVERVIEW

Spotify tracks dataset from Kaggle which was collected from Spotify's Web API (2022)

25 MB file size (CSV)

114,000 rows of data initially

76,962 rows after pre-processing

125 genres

1000s of songs

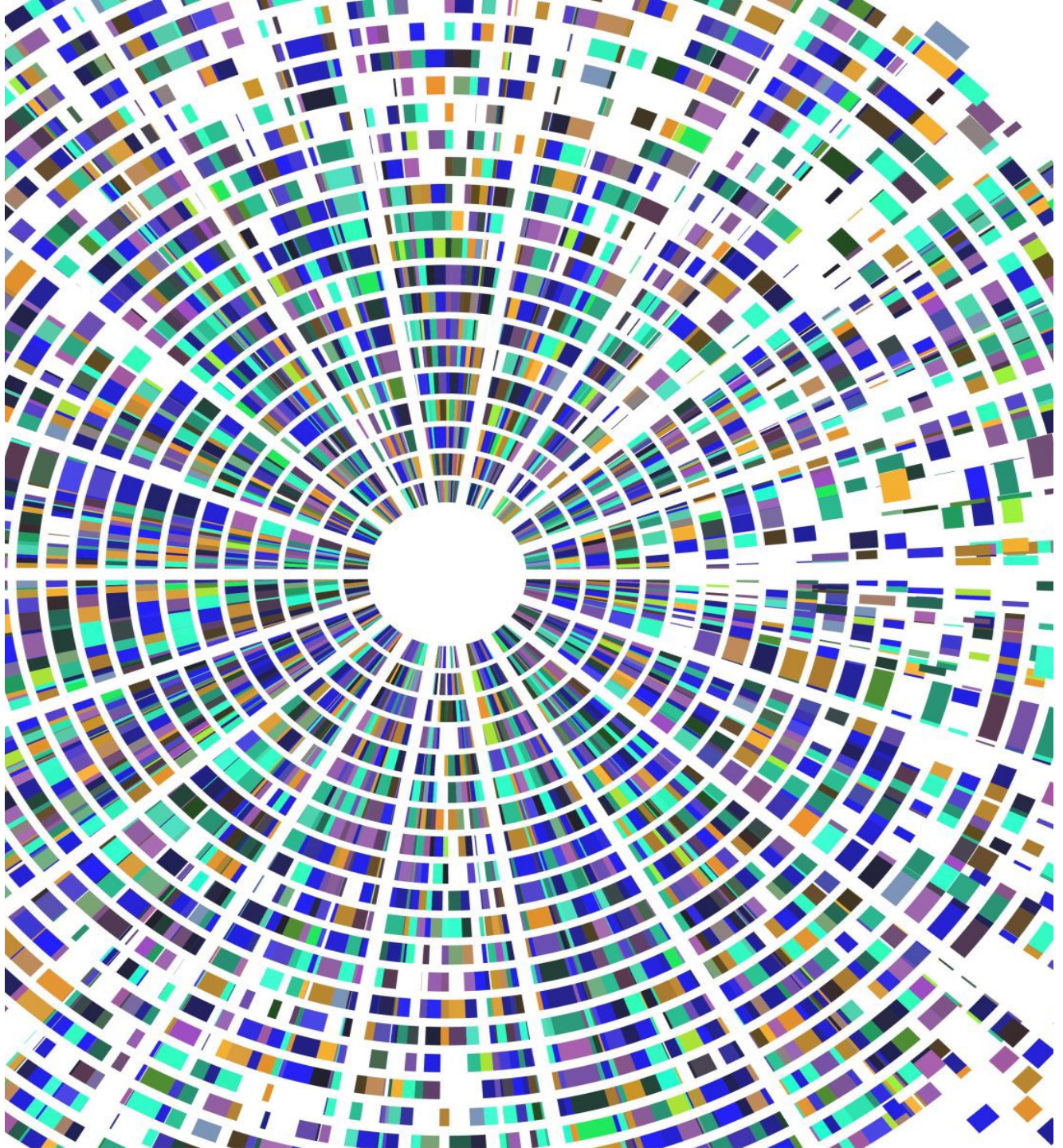
Audio features like danceability, loudness, liveness, energy and valence of each track

Popularity based on artists, genres, tracks and audio features

Popularity (0-100), with 100 being most popular

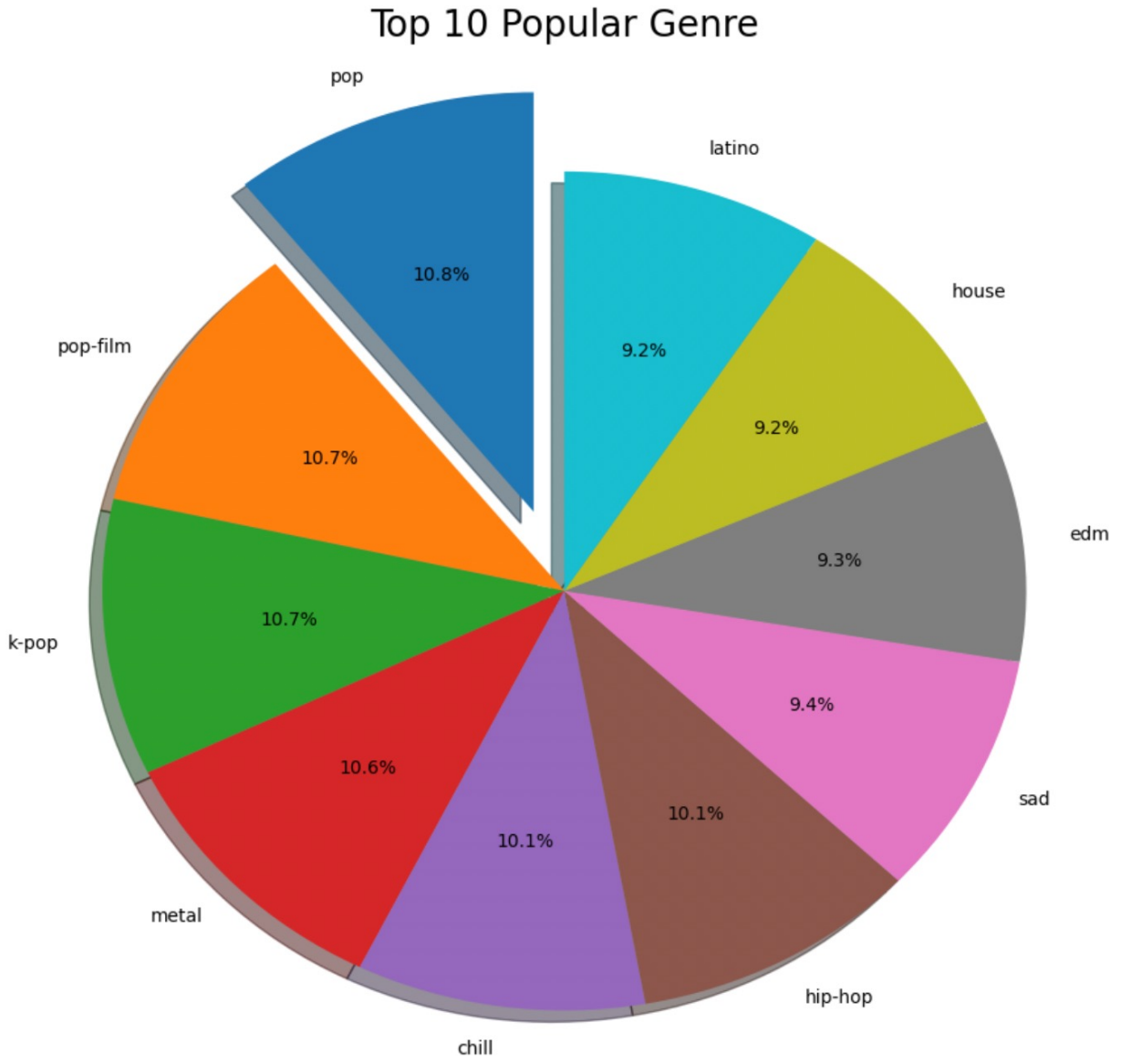
EXPLORATORY DATA ANALYSIS (EDA)

- Matplotlib, Seaborn and Word cloud libraries



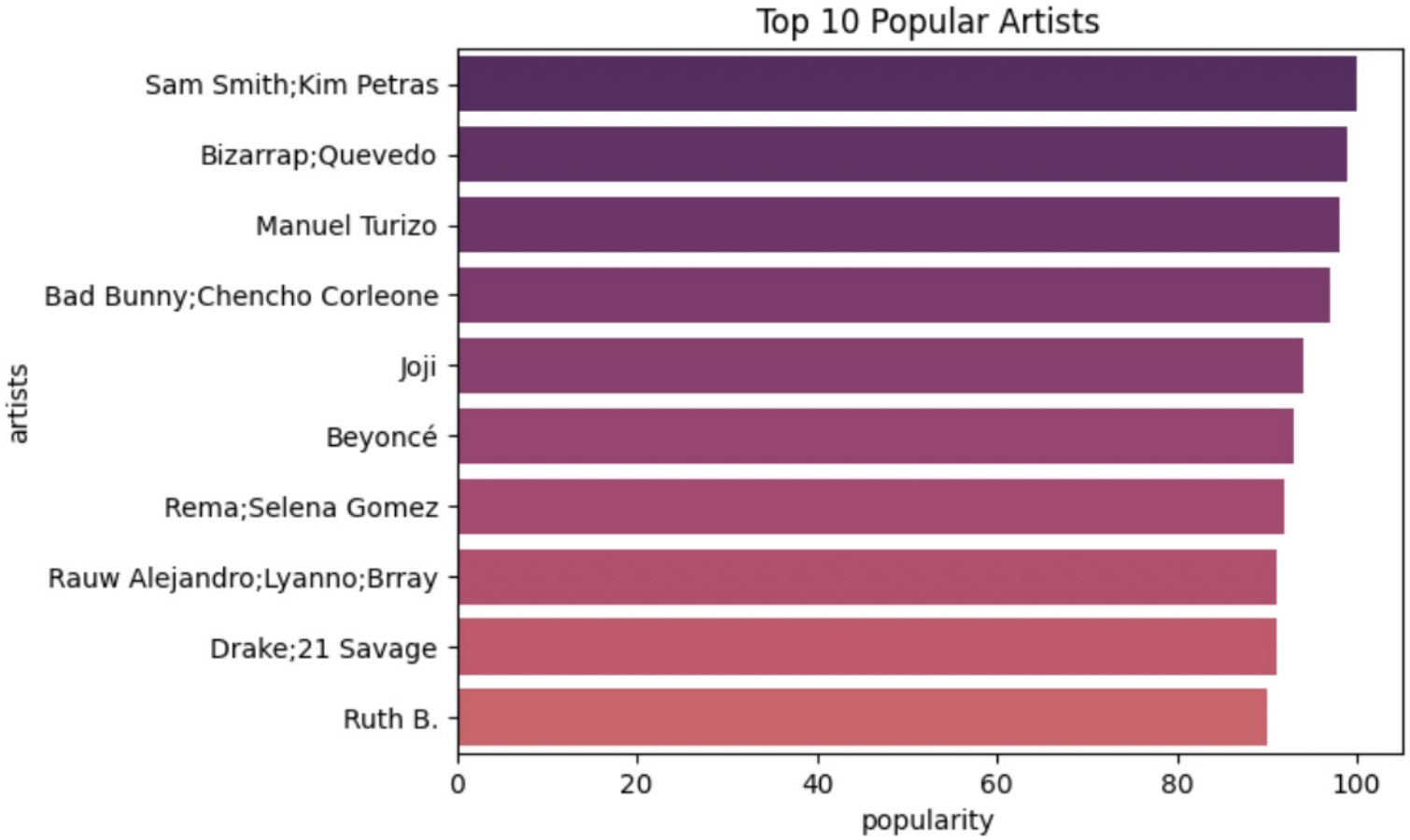
TOP 10 POPULAR GENRES

Genres	Popularity (0-100)
Pop	67.95
Electro	63.22
Dance	62.18
House	61.39
Soul	61.26
Hip-hop	61.21
Edm	60.37
Pop-film	59.21
Metal	59.21
K-pop	58.84



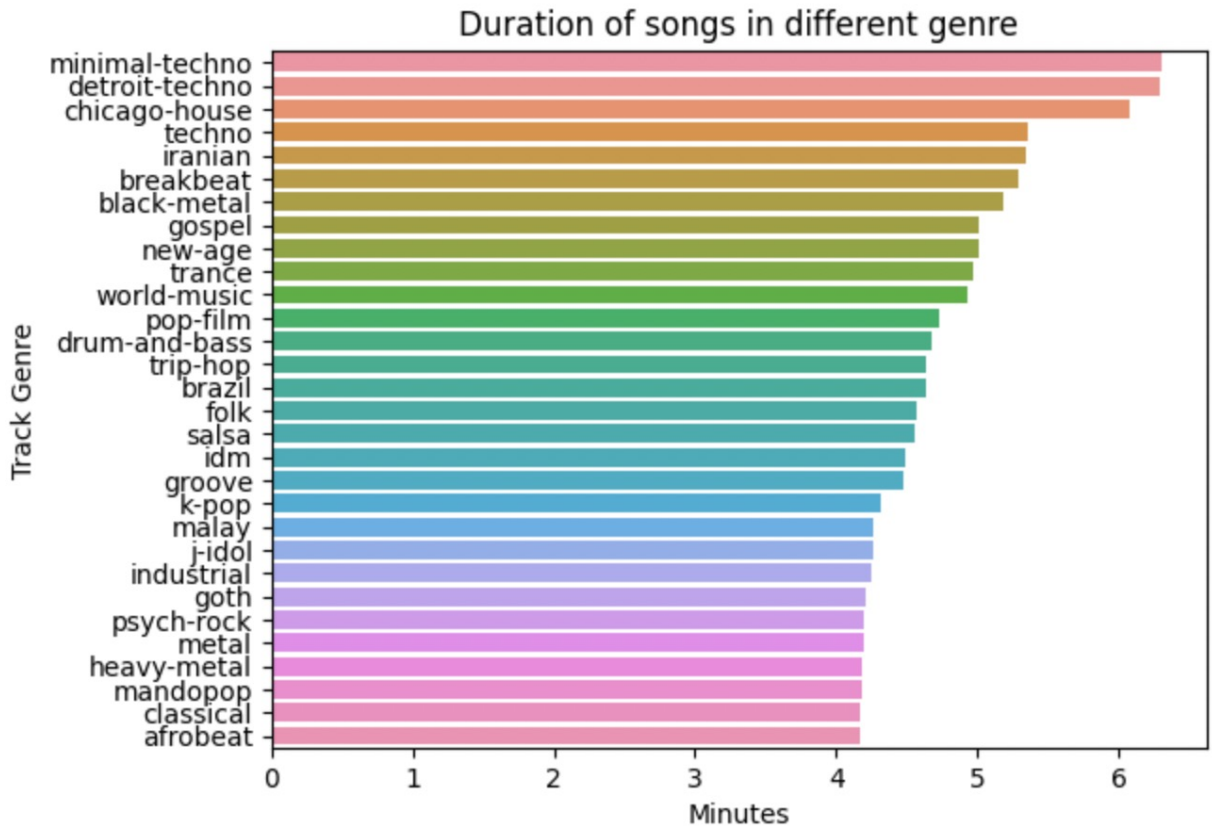
TOP 10 POPULAR ARTISTS

Artists	Popularity (0-100)
Sam Smith	100.0
Bizarrap	99.0
Manuel Turizo	98.0
Bad Bunny	97.0
Joji	94.0
Beyonce	93.0
Selena Gomez	92.0
Rauw Alejandro	91.0
Drake	91.0
Ruth B.	90.0



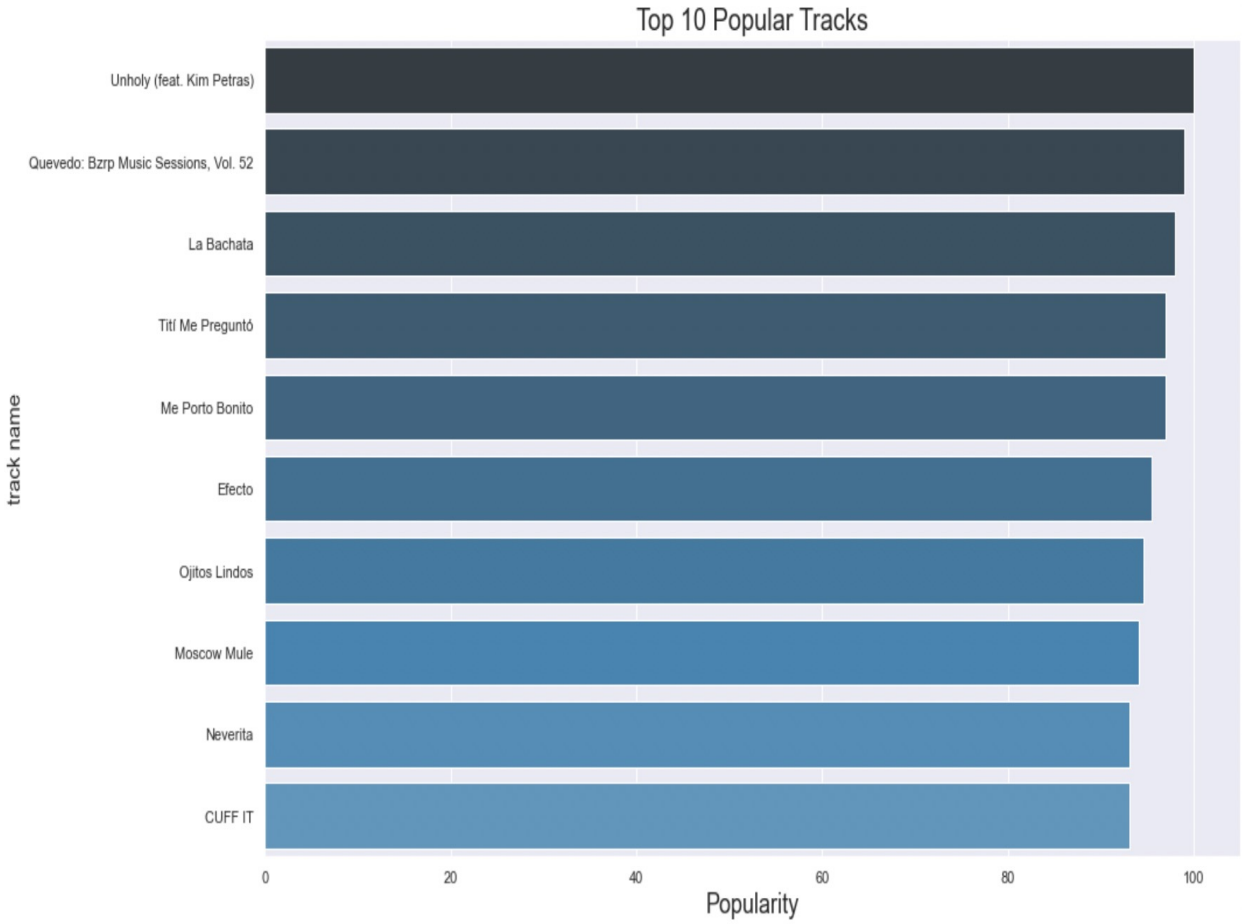
DURATION OF SONG IN DIFFERENT GENRES

Genres	Duration (min)
Minimal-techno	6.30
Detroit-techno	6.29
Chicago-house	6.08
Iranian	6.01
Techno	5.33
Breakbeat	5.28
Black-metal	5.18
New-age	5.02
Gospel	5.01
Trance	4.95



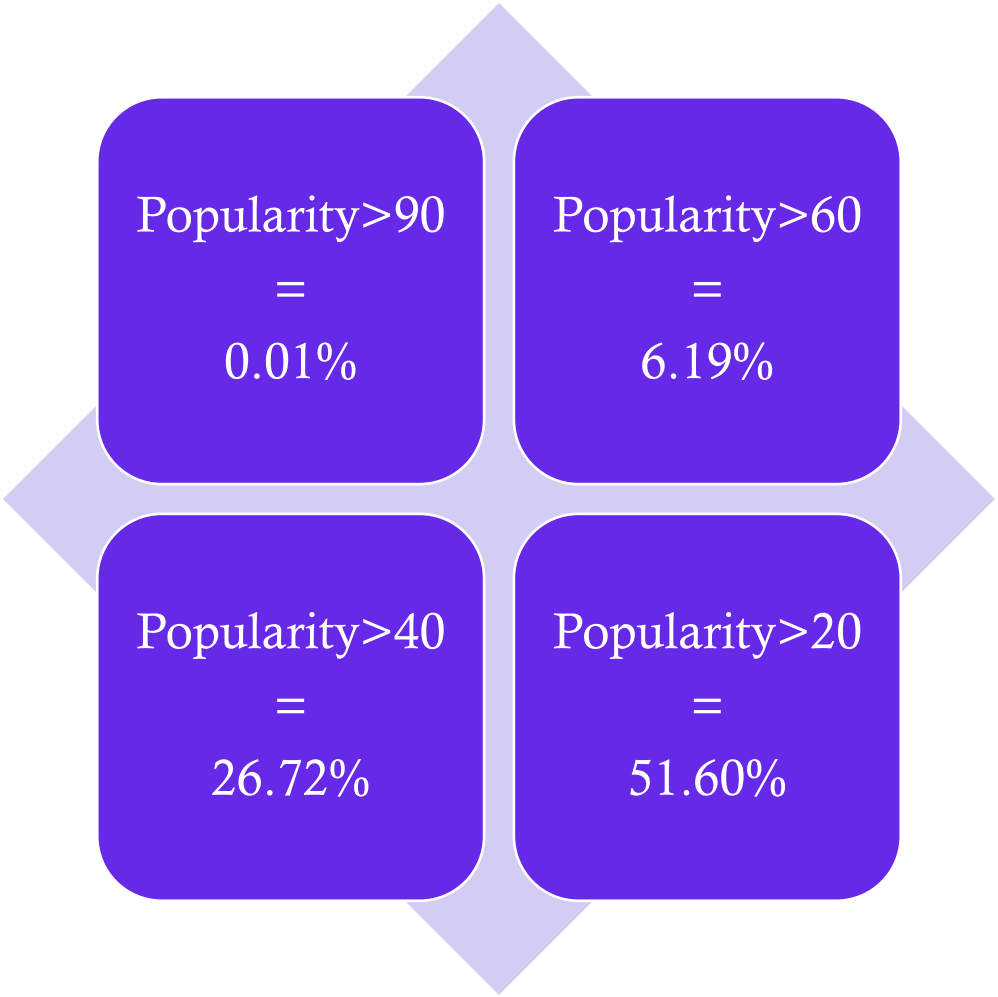
TOP 10 POPULAR TRACKS

Track Name	Popularity (0-100)
Unholy (Sam Smith ft. Kim Petras)	100.0
Quevedo: Bzrp Music Sessions	99.0
La Bachata	98.0
Titi Me Pregunto	97.0
Me Porto Bonito	97.0
Efecto	95.5
Ojitos Lindos	94.5
Moscow Mule	94.0
Neverita	93.0
Cuff It	93.0



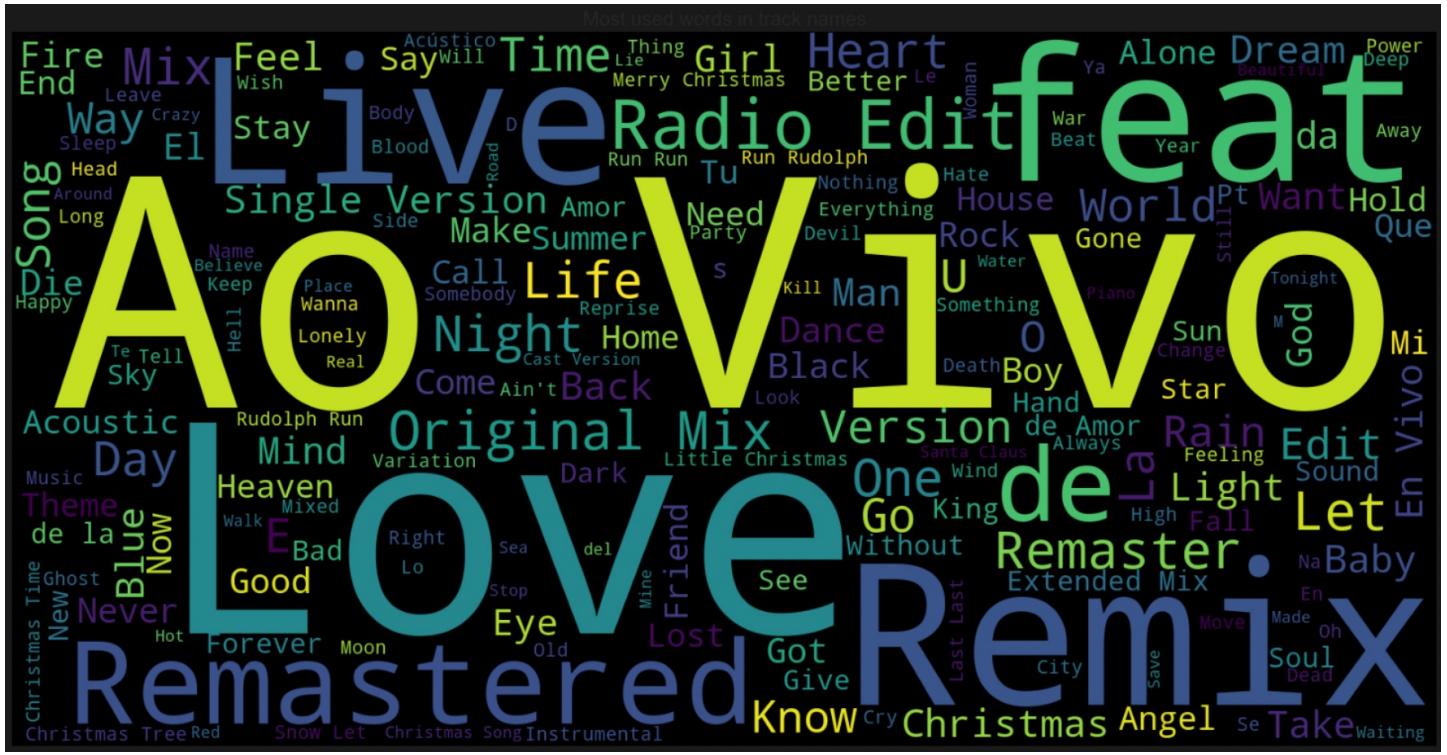
PROBABILITIES OF POPULARITY

- Songs that are most played at current time fall under popular tracks rather than most played songs from all time.
- Difficult to score high popularity based on Spotify's algorithm.



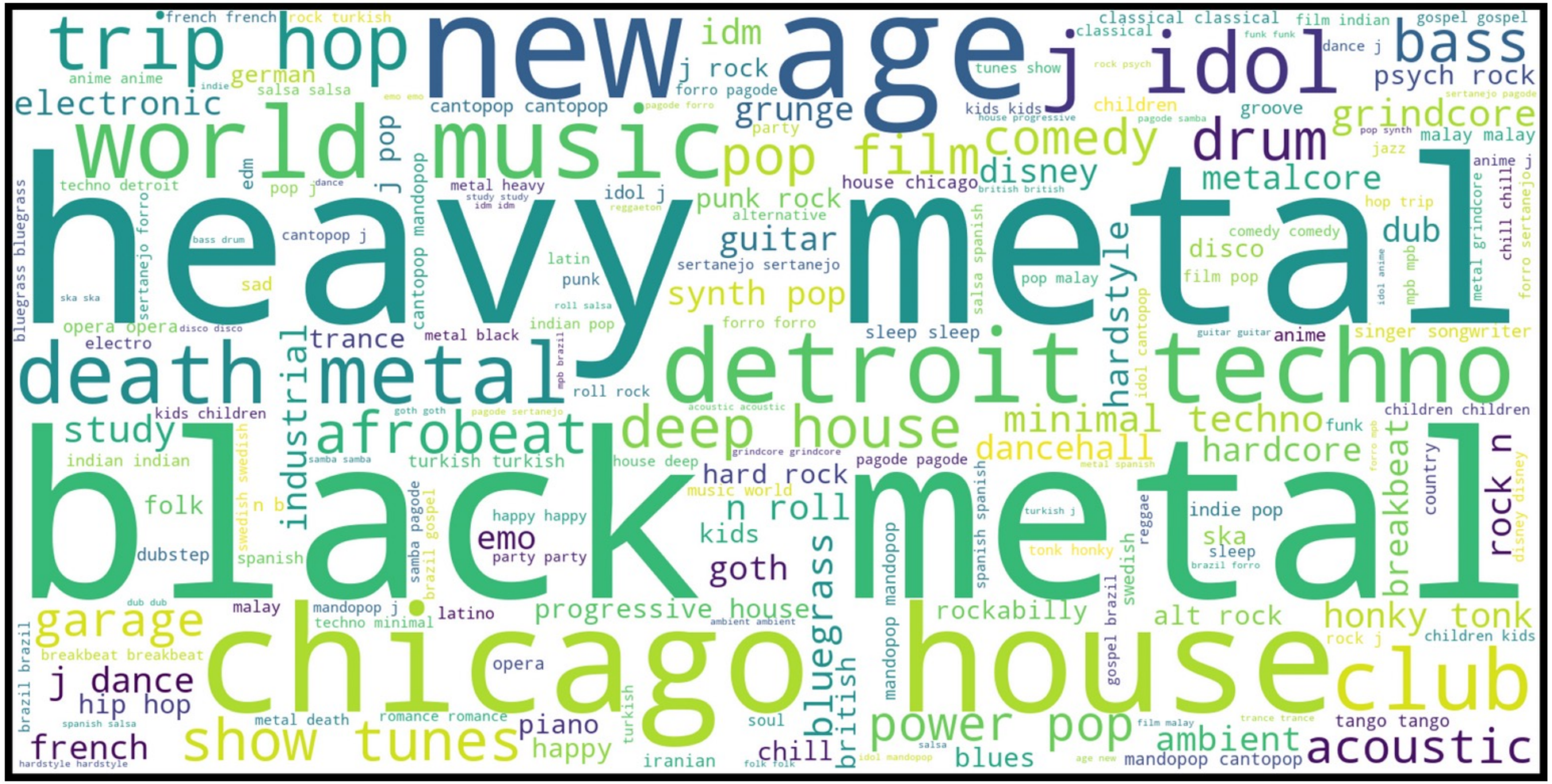
WORD CLOUD OF FREQUENTLY USED WORDS IN TRACK NAMES

1. Love
2. Live
3. Ao Vivo
4. Original Mix
5. Remix
6. Remastered
7. Feat
8. Radio Edit



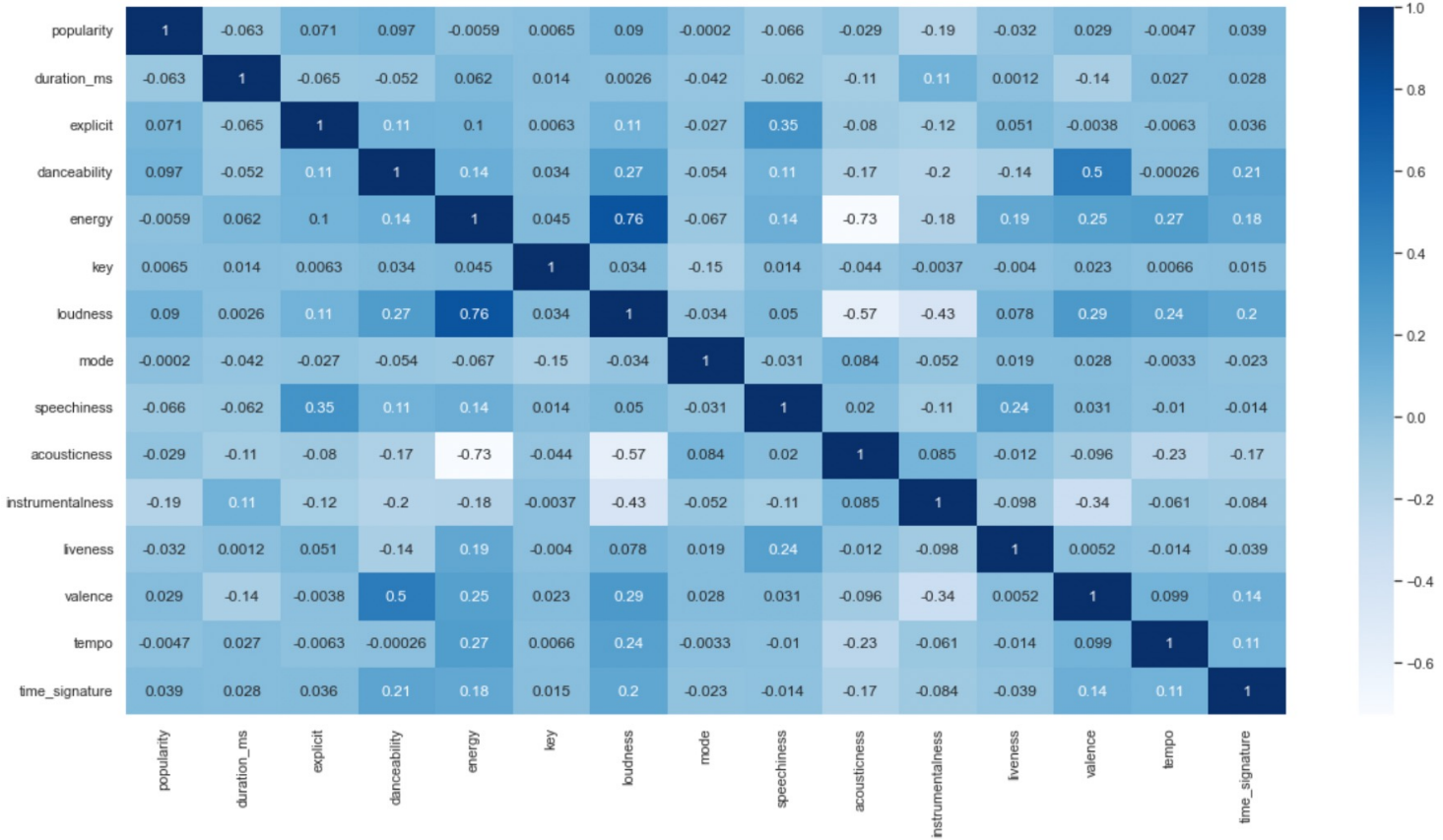
WORD CLOUD OF MOST REPEATED GENRES

1. Heavy metal
2. Black Metal
3. New age
4. World music
5. Death metal
6. Detroit techno
7. Hard rock
8. Trip hop



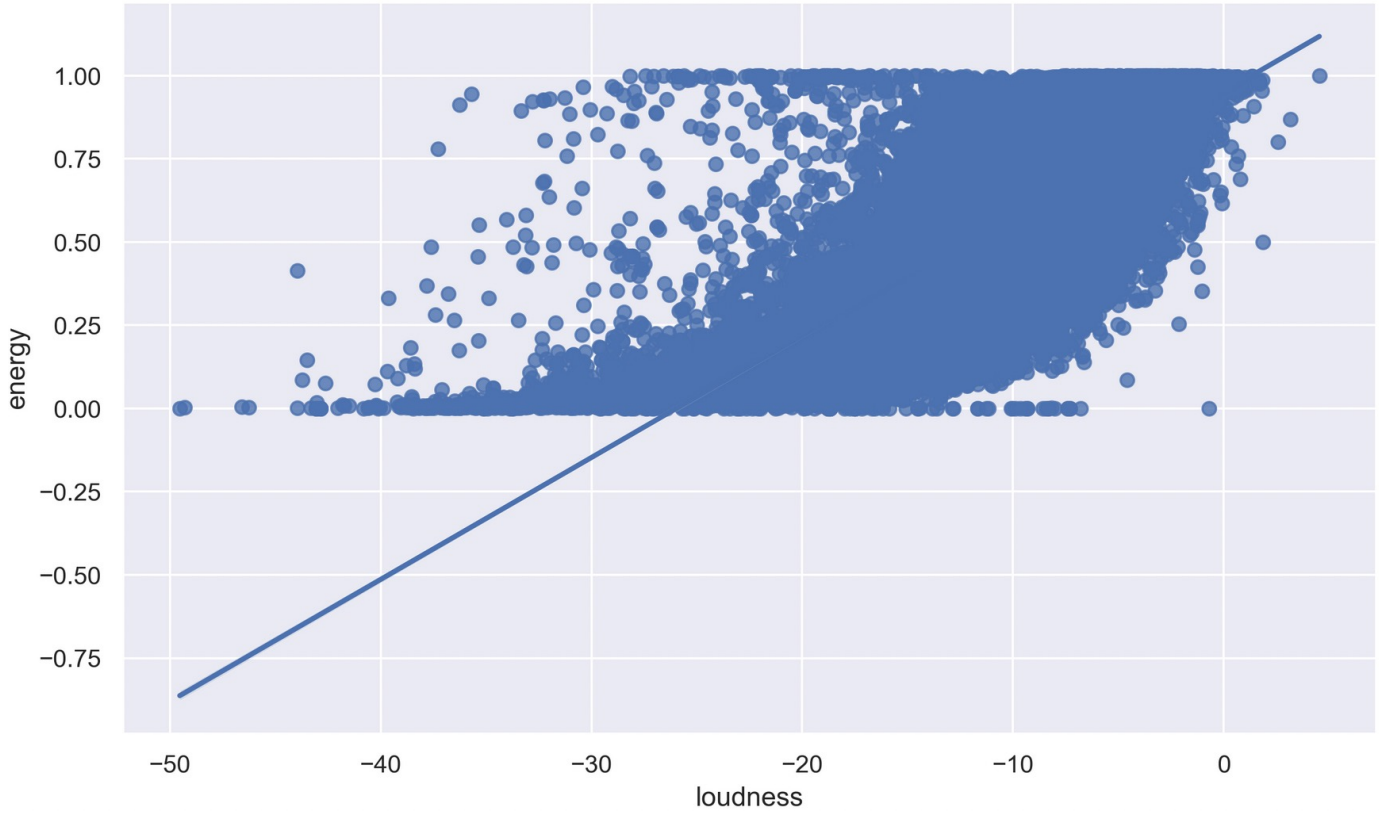
CORRELATION (HEATMAP)

- Energy and loudness are positively correlated with each other because it makes sense for loud music to be energetic. Similarly, valence is also positively correlated with danceability because valence represents songs which are happy and cheerful.
- Whereas, acousticness is negatively correlated with energy and loudness because acoustic songs are quiet and calm.



CORRELATION: LOUDNESS VS ENERGY

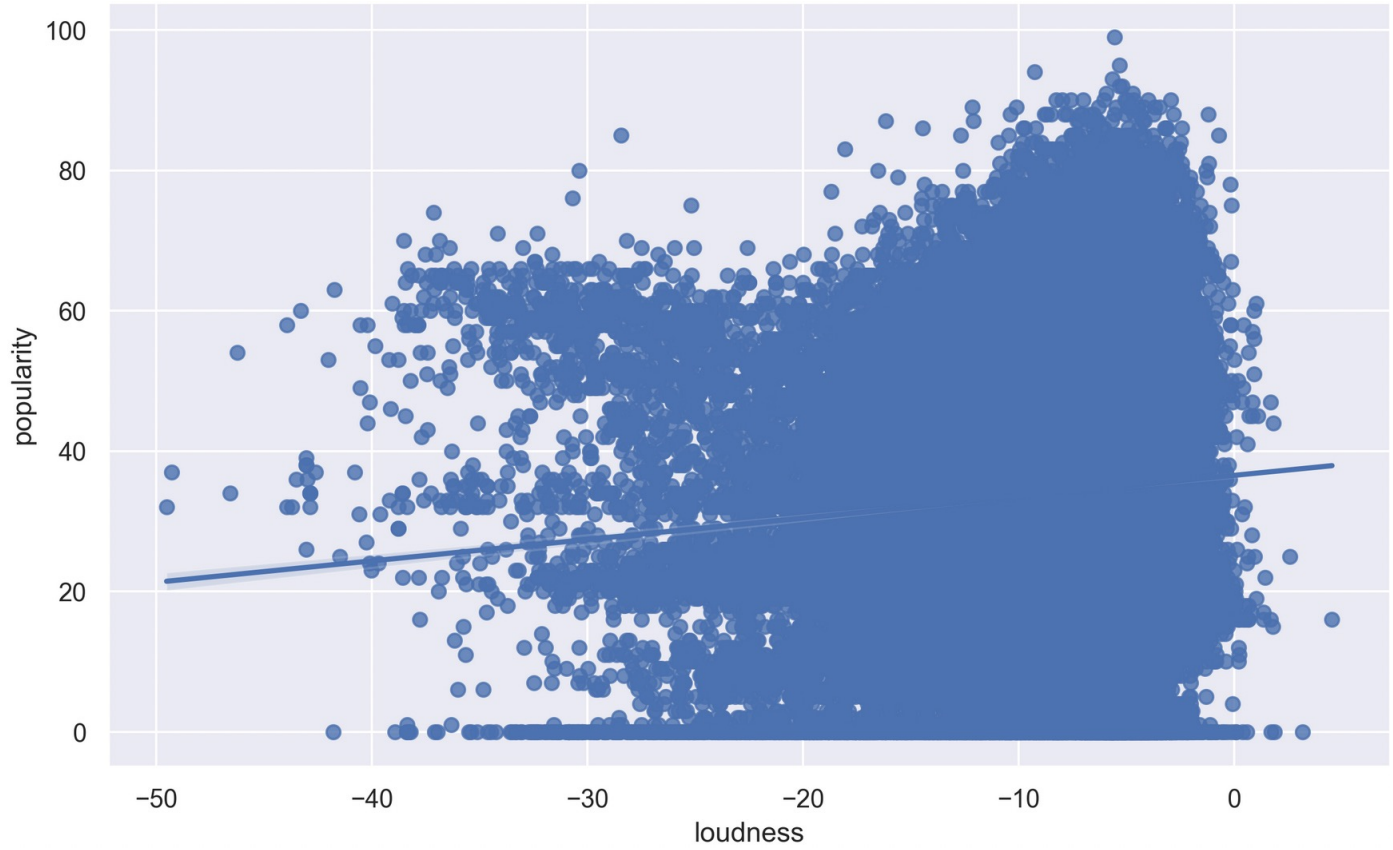
Loudness and energy correlated with each other at almost 45 degree which suggests that there is strong relation between a song being loud and energetic.



CORRELATION: LOUDNESS VS POPULARITY

With over 100,000 songs we can infer that there's slight correlation between the music being and loud and popular.

Most of the loud music are popular compared to the ones that are less loud.



INITIAL CHALLENGES WITH DATASET

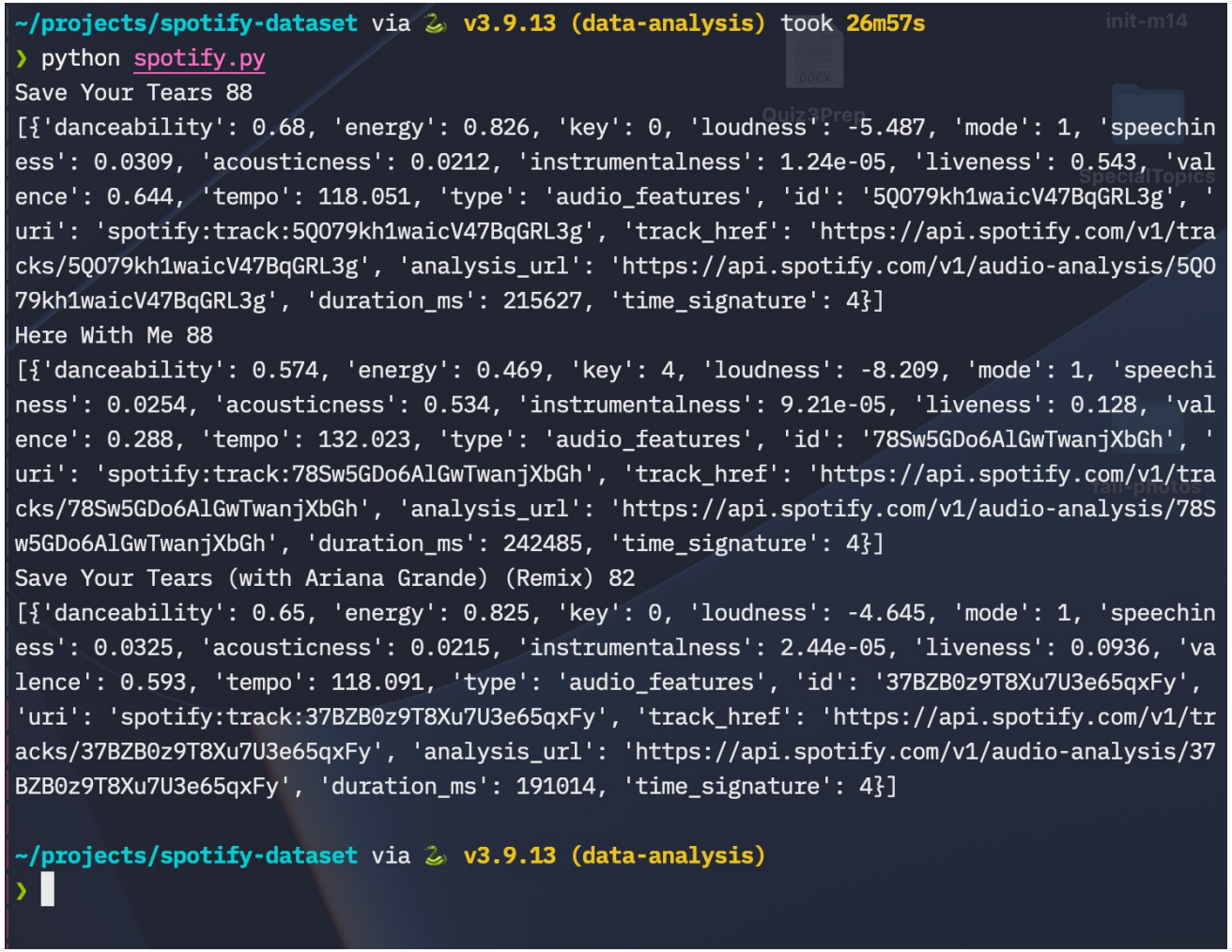
- Domain knowledge
- Work on limited set of features
- Primary data vs Secondary Data

SPOTIFY WEB API

- Get tracks, album, artists
- Python package:
- pip install spotipy

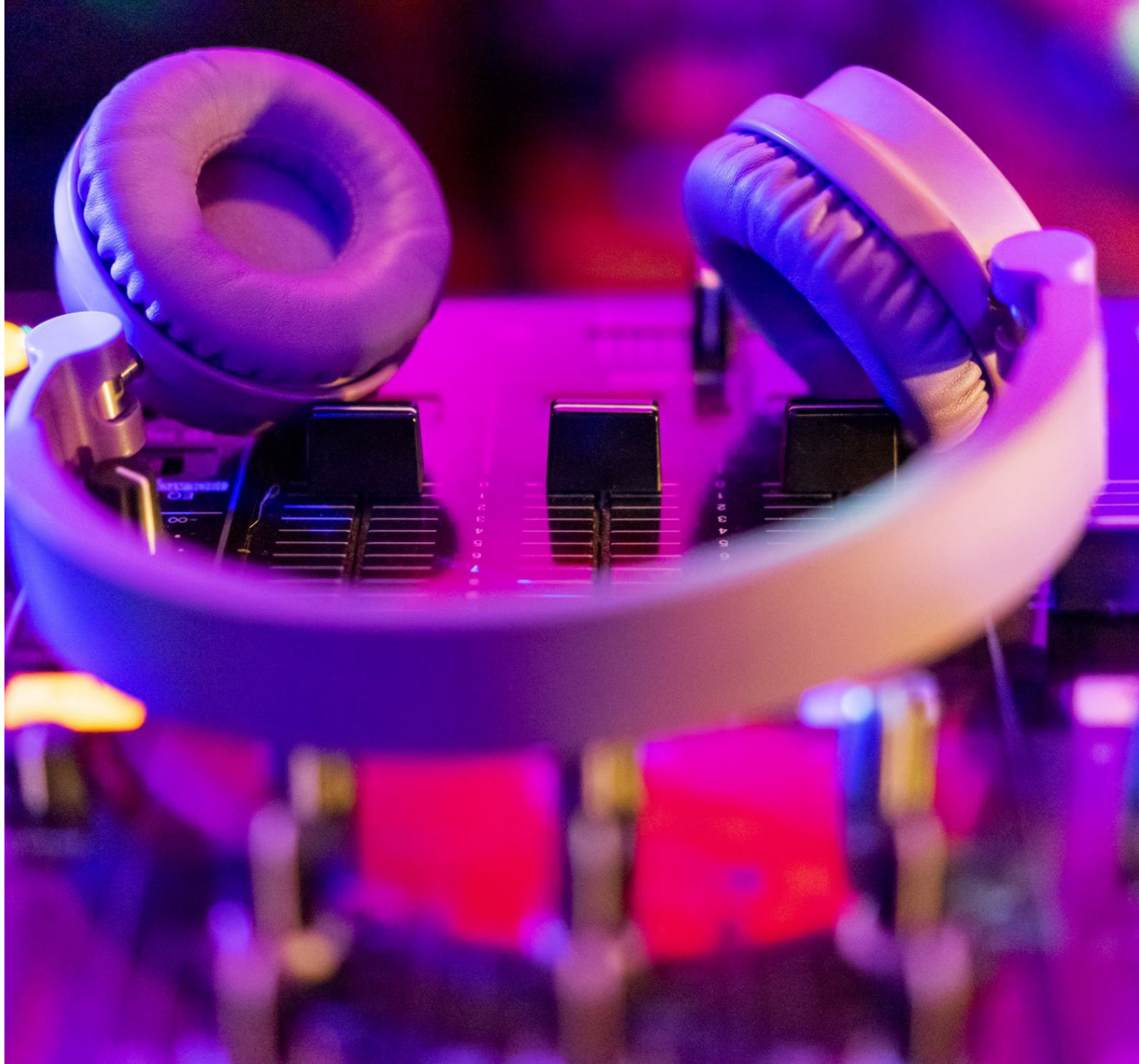
```python
def search_track():
    sp = spotipy.Spotify(client_credentials_manager=SpotifyClientCredentials())
    results = sp.search(q='Save your tears', limit=3)
    for idx, track in enumerate(results['tracks']['items']):
        # print(track)
        print(track['name'], sp.track(track['id'])['popularity'])
        print(sp.audio_features(tracks=track['id']))

def main():
    birdy_uri = 'spotify:artist:2WX2uTcsvV5OnS0inACecP'
    spotify = spotipy.Spotify(client_credentials_manager=SpotifyClientCredentials())

    results = spotify.artist_albums(birdy_uri, album_type='album')
    albums = results['items']
    while results['next']:
        results = spotify.next(results)
        albums.extend(results['items'])

    for album in albums:
        print(album['name'])

if __name__ == "__main__":
    search_track()
    # artist_albumart()
```

```
NORMAL  ⊗ 6 ♀ 2  spotify.py                    utf-8  python  62%  35:52
"spotify.py" 56L, 1819B written
```


~/projects/spotify-dataset via v3.9.13 (data-analysis) took 26m57s
❯ python spotify.py
Save Your Tears 88
[{'danceability': 0.68, 'energy': 0.826, 'key': 0, 'loudness': -5.487, 'mode': 1, 'speechiness': 0.0309, 'acousticness': 0.0212, 'instrumentalness': 1.24e-05, 'liveness': 0.543, 'valence': 0.644, 'tempo': 118.051, 'type': 'audio_features', 'id': '5QO79kh1waicV47BqGRL3g', 'uri': 'spotify:track:5QO79kh1waicV47BqGRL3g', 'track_href': 'https://api.spotify.com/v1/tracks/5QO79kh1waicV47BqGRL3g', 'analysis_url': 'https://api.spotify.com/v1/audio-analysis/5QO79kh1waicV47BqGRL3g', 'duration_ms': 215627, 'time_signature': 4}]
Here With Me 88
[{'danceability': 0.574, 'energy': 0.469, 'key': 4, 'loudness': -8.209, 'mode': 1, 'speechiness': 0.0254, 'acousticness': 0.534, 'instrumentalness': 9.21e-05, 'liveness': 0.128, 'valence': 0.288, 'tempo': 132.023, 'type': 'audio_features', 'id': '78Sw5GDo6AlGwTwanjXbGh', 'uri': 'spotify:track:78Sw5GDo6AlGwTwanjXbGh', 'track_href': 'https://api.spotify.com/v1/tracks/78Sw5GDo6AlGwTwanjXbGh', 'analysis_url': 'https://api.spotify.com/v1/audio-analysis/78Sw5GDo6AlGwTwanjXbGh', 'duration_ms': 242485, 'time_signature': 4}]
Save Your Tears (with Ariana Grande) (Remix) 82
[{'danceability': 0.65, 'energy': 0.825, 'key': 0, 'loudness': -4.645, 'mode': 1, 'speechiness': 0.0325, 'acousticness': 0.0215, 'instrumentalness': 2.44e-05, 'liveness': 0.0936, 'valence': 0.593, 'tempo': 118.091, 'type': 'audio_features', 'id': '37BZB0z9T8Xu7U3e65qxFy', 'uri': 'spotify:track:37BZB0z9T8Xu7U3e65qxFy', 'track_href': 'https://api.spotify.com/v1/tracks/37BZB0z9T8Xu7U3e65qxFy', 'analysis_url': 'https://api.spotify.com/v1/audio-analysis/37BZB0z9T8Xu7U3e65qxFy', 'duration_ms': 191014, 'time_signature': 4}]
~/projects/spotify-dataset via v3.9.13 (data-analysis)
❯

AUDIO FEATURES

- Acousticness
- Speechiness
- Loudness (Decibels)
- Energy
- Danceability
- Valence (Cheerful, happy)



PREDICTION OF THE POPULARITY

(BASED ON THE DATASET)

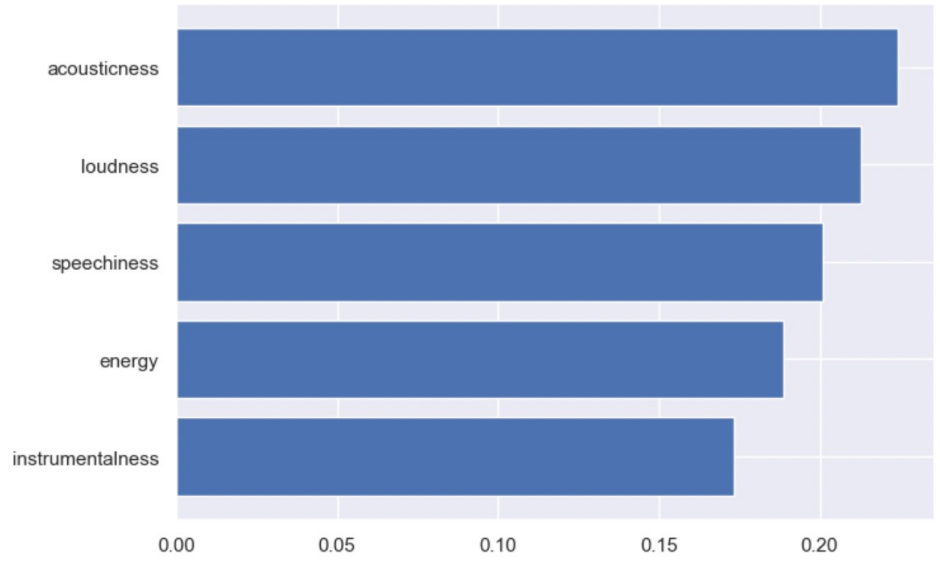
" The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. " – **Spotify WEB API Docs**

FEATURE SELECTION

- We tried different models and came up with the best score of important features at 86.75.
- Acousticness, loudness, speechiness, energy and instrumentalness

```
In [638]: random_forest = RandomForestRegressor()

random_forest.fit(x_train, y_train)
Y_pred_rf = random_forest.predict(x_test)
random_forest.score(x_train,y_train)
acc_random_forest = round(random_forest.score(x_train,y_train) * 100, 2)

print("Important features")
pd.Series(random_forest.feature_importances_,x_train.columns).sort_values(ascending=True).plot.barh(width=0.8)
print('__'*30)
print(acc_random_forest)
```

```
Important features
____________________________________________________________
86.75
```



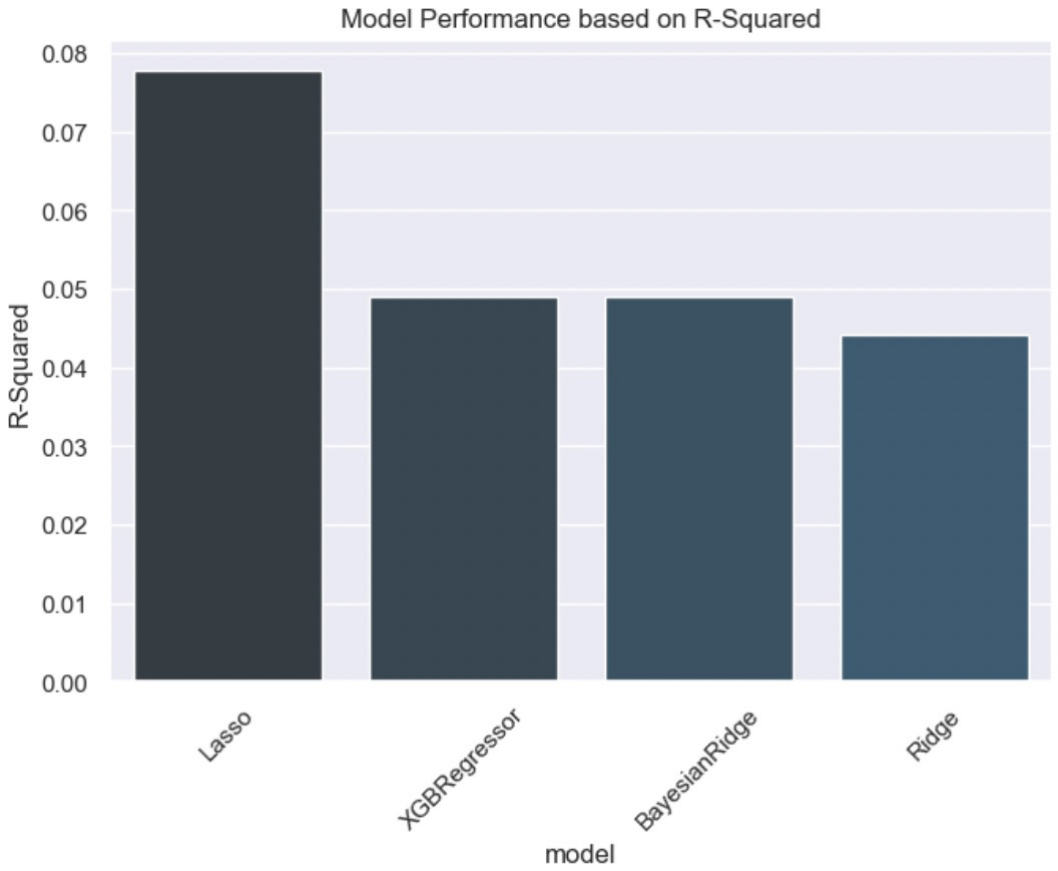
TRAIN AND TEST SETS / MIN MAX SCALER

```
# minmax (values should be between 0 and 1)
from sklearn.preprocessing import MinMaxScaler
mm = MinMaxScaler()
minmax = mm.fit_transform(X)

feature_mm = pd.DataFrame(minmax, index=X.index,
                          columns=X.columns)
feature_mm = feature_mm.reset_index(drop=True)
feature_mm
```

```
# Build the model
x_train_mm, x_test_mm, y_train_mm, y_test_mm = train_test_split(feature_mm, y, test_size=0.2, random_state=42)
```

MODEL OUTCOMES

We can see that even the well performing model, Lasso has relatively very low R-squared which suggests that even though some of the data points are correlated with popularity but can be problematic to precisely predict the value of popularity based on these correlated data points and from above R2 squared.



LEARNING PERFORMANCE METRICS FROM REGRESSION MODELS

- Predicting popularity is not only based on the audio features.
- As mentioned in the spotify web api docs, different measures are carried out to calculate the popularity of a track
- By now we had realized that there was too much variance in the dataset and with Low R-squared values we can expect of imprecise predictions.
- Off we go to classifications now.

R SQUARE FROM DIFFERENT MODELS

	model	mean_squared_error	R-Squared	time
2	Lasso	310.32760	0.07780	18
0	XGBRegressor	320.01504	0.04901	0
4	BayesianRidge	320.01735	0.04900	0
1	Ridge	321.66368	0.04411	0

CLASSIFIER MODELING – BINNING PREDICTION

Classifier Models

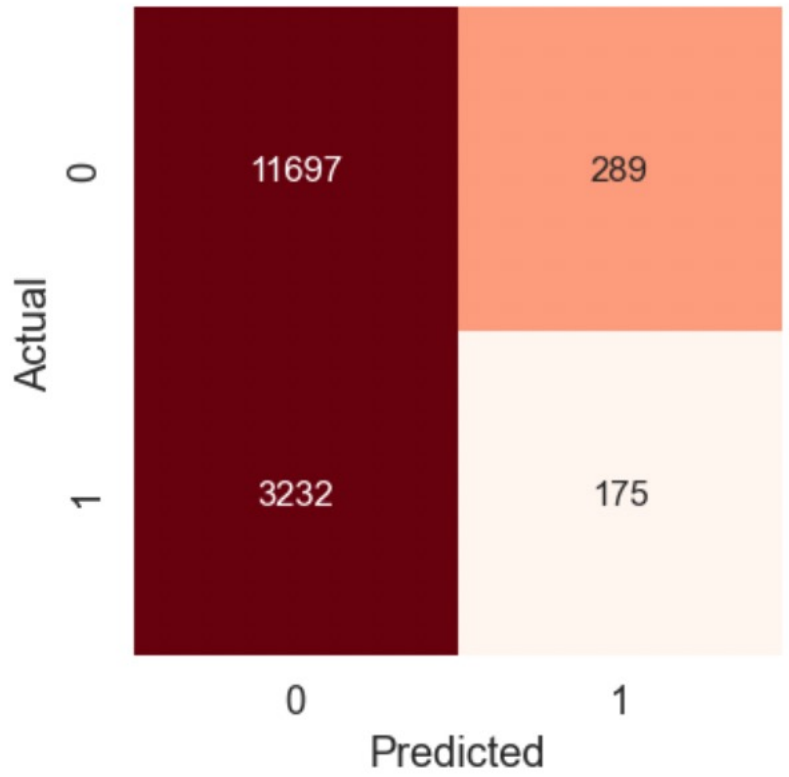
```
from sklearn.linear_model import LogisticRegression #Logistic Regression
from sklearn.naive_bayes import GaussianNB #Naive Bayes
from sklearn.tree import DecisionTreeClassifier #Decision Tree
from sklearn.neighbors import KNeighborsClassifier #KNN
from xgboost import XGBClassifier #XGB
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import AdaBoostClassifier

from sklearn.model_selection import train_test_split

from statistics import mean
from sklearn.metrics import accuracy_score, log_loss
from sklearn.model_selection import KFold, cross_val_score

from sklearn.pipeline import Pipeline

df_final['is_popular'] = df['popularity'].apply(lambda x: 1 if x > 50 else 0)

y = df_final['is_popular']
X = df_final.drop(columns=['popularity', 'explicit', 'key', 'mode', 'time_signature', 'is_popular', 'duration_ms'])

x_train,x_test,y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
X
```

MODEL SCORE FROM DIFFERENT CLASSIFIERS

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
KNeighborsClassifier	0.83	0.71	0.71	0.82	0.45
LabelSpreading	0.80	0.69	0.69	0.80	815.03
LabelPropagation	0.80	0.69	0.69	0.80	710.68
XGBClassifier	0.84	0.68	0.68	0.82	2.12
DecisionTreeClassifier	0.78	0.67	0.67	0.78	0.58
BaggingClassifier	0.83	0.66	0.66	0.81	3.00
LGBMClassifier	0.84	0.66	0.66	0.82	0.48
RandomForestClassifier	0.82	0.60	0.60	0.79	10.11
ExtraTreeClassifier	0.73	0.59	0.59	0.73	0.06
QuadraticDiscriminantAnalysis	0.76	0.57	0.57	0.74	0.08
ExtraTreesClassifier	0.81	0.57	0.57	0.76	3.37
AdaBoostClassifier	0.80	0.53	0.53	0.73	1.86
SGDClassifier	0.33	0.53	0.53	0.33	0.58
LinearSVC	0.44	0.52	0.52	0.48	3.20
NearestCentroid	0.51	0.51	0.51	0.56	0.07
GaussianNB	0.79	0.51	0.51	0.71	0.03
PassiveAggressiveClassifier	0.78	0.51	0.51	0.71	0.09
Perceptron	0.79	0.50	0.50	0.71	0.05
BernoulliNB	0.79	0.50	0.50	0.70	0.04
LinearDiscriminantAnalysis	0.79	0.50	0.50	0.70	0.18
LogisticRegression	0.79	0.50	0.50	0.70	0.81
DummyClassifier	0.79	0.50	0.50	0.70	0.03
CalibratedClassifierCV	0.79	0.50	0.50	0.70	9.68
RidgeClassifier	0.79	0.50	0.50	0.70	0.04
RidgeClassifierCV	0.79	0.50	0.50	0.70	0.35
SVC	0.79	0.50	0.50	0.70	38.17

CONFUSION MATRIX

- True negative and false negative is quite high but we got true negative on the higher side.
- True positive and false positive although there are less occurrences, confusion matrix is showing the false negative on the higher side.



OUT OF SAMPLE PREDICTION

From the analysis, it seems that for a song to be popular it is quite hard and not every feature of song can be easily integrated to make a song popular.

	energy	loudness	speechiness	acousticness	instrumentalness
0	0.83	-5.49	0.03	0.02	0.00
1	0.86	-8.32	0.40	0.93	0.00
2	0.65	-1.89	0.30	0.74	0.83
3	0.75	-6.44	0.07	0.45	0.97

```
Predicted value for popularity :  1 , which means  "yes"
Predicted probability is 0.083
Predicted value for popularity :  0 , which means  "no"
Predicted probability is 0.833
Predicted value for popularity :  0 , which means  "no"
Predicted probability is 0.750
Predicted value for popularity :  0 , which means  "no"
Predicted probability is 0.750
```

LIMITATIONS

This dataset did not have date variable which restricted us from making analysis based on time-series, about how music is evolving with time.

Availability of many audio features made it quite difficult to choose right set of features.

Since this dataset is big and has a lot of rows it was at times very time consuming to run different models.

LEARNING, SUMMARY AND PREDICTION

- The popularity of a song is influenced by the **danceability**, **loudness** and **valence**.
- **Pop** music is most popular nowadays.
- **Minimal-techno** genre has the longest track duration.
- Sam Smith's (ft. Kim Petras) **Unholy** is the most popular track.
- High energy dance songs and songs with duration of approximately 3 minutes are more likely to become popular.
- Although regression models were not with good R2 scores and highly significant in terms of different metrics, but we learned that with every dataset comes a challenge to build better models and improve scores by applying different tuning to models.

Thank you