TECHNOCOLABS DATA SCIENCE INTERNSHIP

PROJECT REPORT

TITLE: Predicting Billboard Hit Songs Using Spotify Data

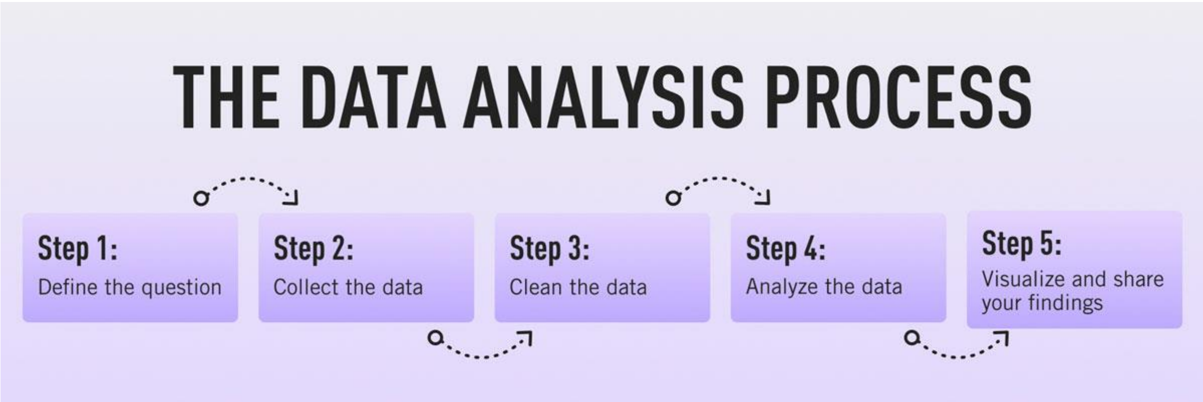
AIM: To find Songs which will make it into Billboard Hit List in future

ABSTRACT:

The Billboard Hot 100 Chart remains one of the definitive ways to measure the success of a popular song. We investigated using machine learning techniques to predict which songs will become Billboard Hot 100 Hits.

What is the data analysis process?

1. Define why you need data analysis.
2. Begin collecting data from sources.
3. Clean through unnecessary data.
4. Begin analysing the data.
5. Interpret the results and apply them.



1. Why we need to do data analysis?

Answer: To fulfil our aim of predicting Billboard Hits using Spotify data.

2. What data needs to be collected to fulfil the above aim?

Answer: We need two sets of data –

I. **Million Song Dataset (MSD):** It contains 1 million songs (western commercial music) released between 1990 to 2019. Refer to - http://millionsongdataset.com/pages/getting-dataset/

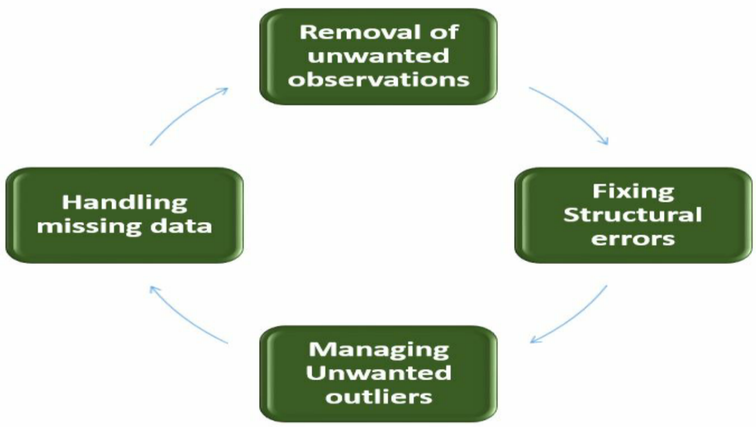
II. **Billboard Dataset:** Using billboard's Hot 100 charts from 1990–2019 and Spotify's API, we want to take a closer look at popular music. Refer to - https://www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019/download

Defining the data:

The data describes each song through certain features, such as, 'danceability', 'energy', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms', 'loudness'
along with these we have the information about “Track”, “Artist”, “song_name”, “spotify_id”, “key”, “mode”, “target”.

3. *Clean through unnecessary data:*

Many fields in the dataset were unusable and some were missing values. Thus, we need to drop such certain fields. We also have to merge the two datasets using a unique field. We realised such unique field could be “Spotify_id”, it is a unique id allotted to each song by Spotify. The field “mode” is an attribute of each song possessing two values 0 and 1. Mode value 0 signifies that the song uses a minor key in its production while value 1 signifies that the song uses major key in its production. There were some mode values = -999 (to be accurate : 10 unknown mode values) representing unknown mode. Therefore, while cleaning the data we need to drop such unknown mode values.



RESULTS:

In [54]:
```python
## Box plot of numerical features
fig = plt.figure(figsize=(30,20))
for i in range(len(num_features.columns)):
    fig.add_subplot(4,5,i+1)
    sns.boxplot(y = num_features.iloc[:,i])
plt.tight_layout()
plt.show()
```

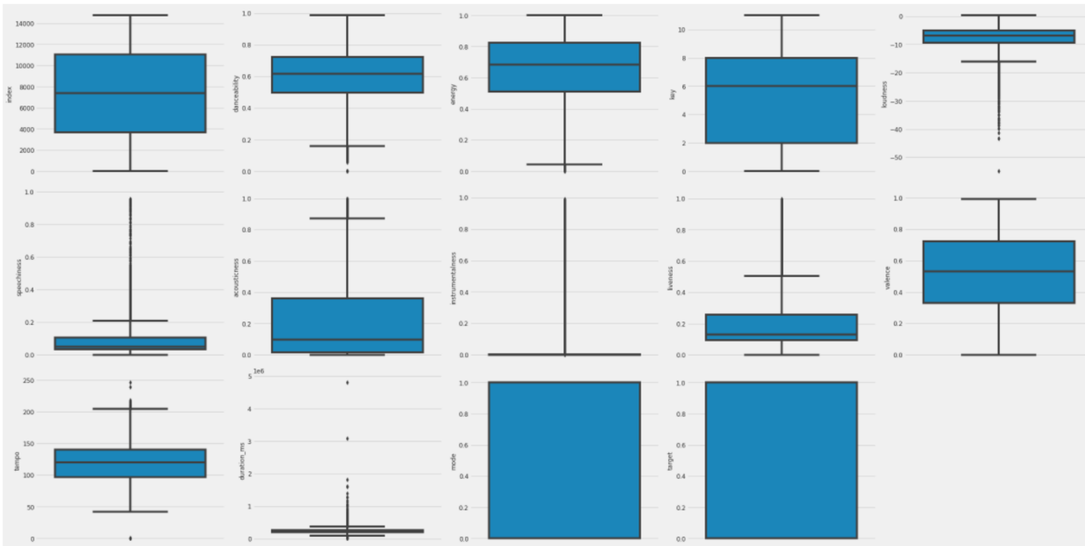


Fig:boxplot

TECHNOCOLABS DATA SCIENCE INTERNSHIP

```
In [61]: ## Dist plot of numerical features
fig = plt.figure(figsize=(30,20))
for i in range(len(num_features.columns)):
    fig.add_subplot(4,5,i+1)
    sns.distplot( num_features.iloc[:,i])
plt.tight_layout()
plt.show()
```

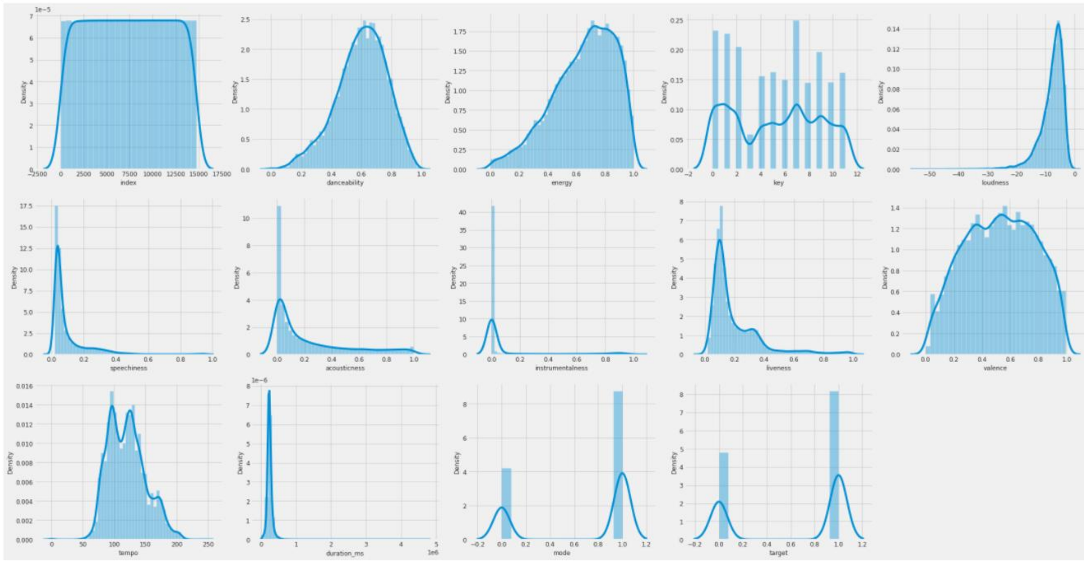


Fig:distplot

```
In [63]: #loudness and energy are highly corelated , so we will drop one of them later
dataplot = sns.heatmap(df.corr(), cmap="YlGnBu", annot=True)
plt.show()
```

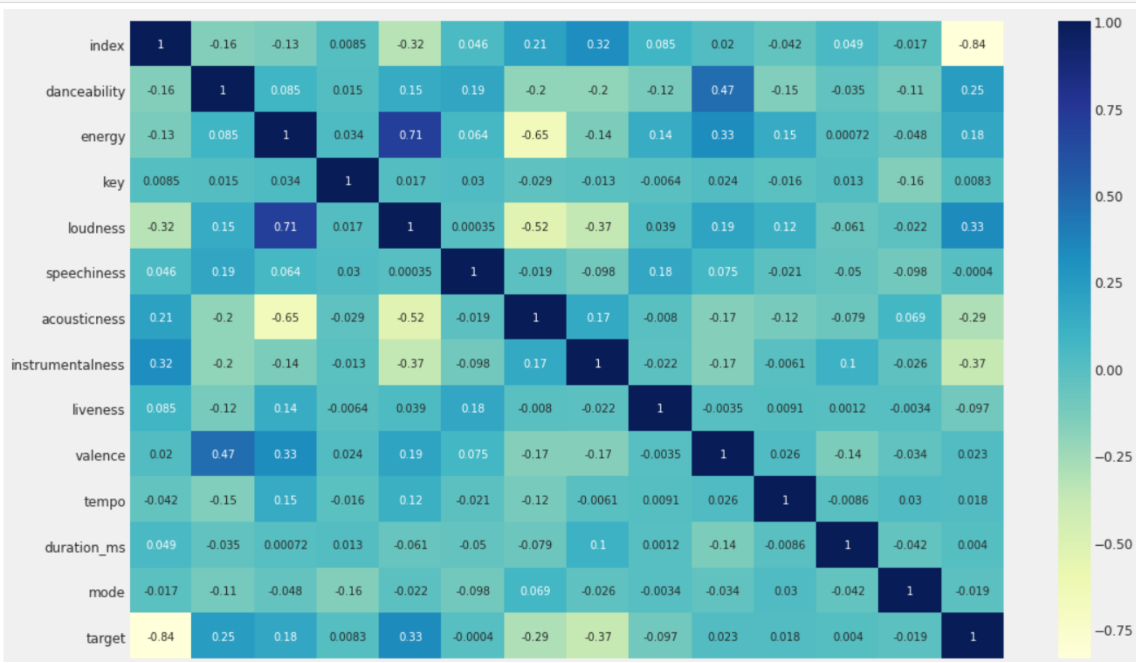
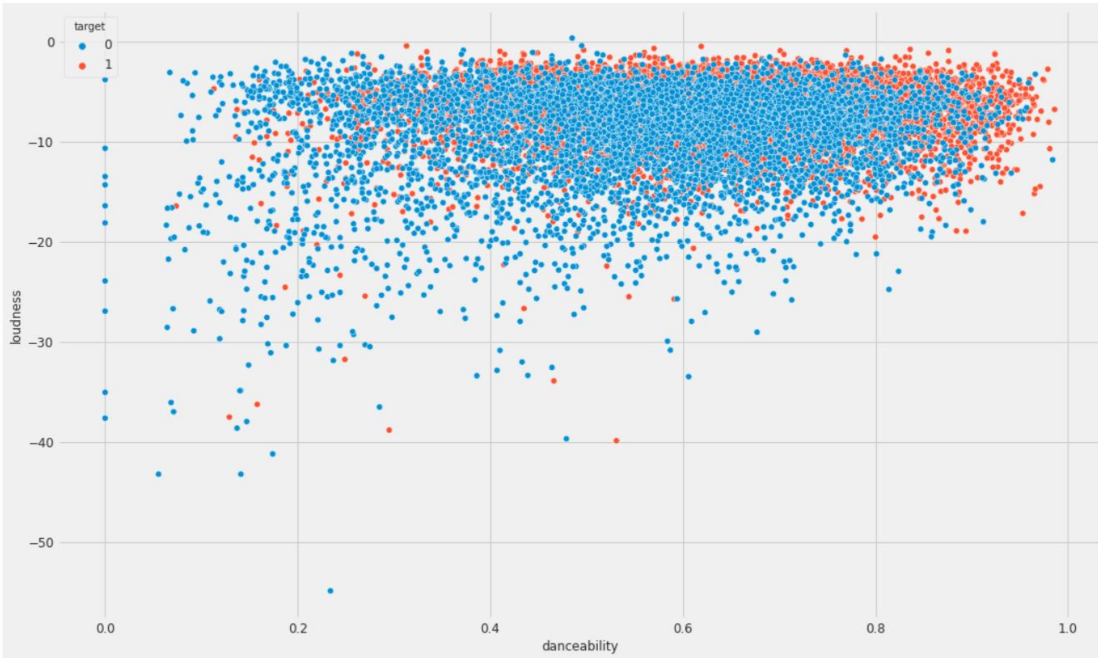
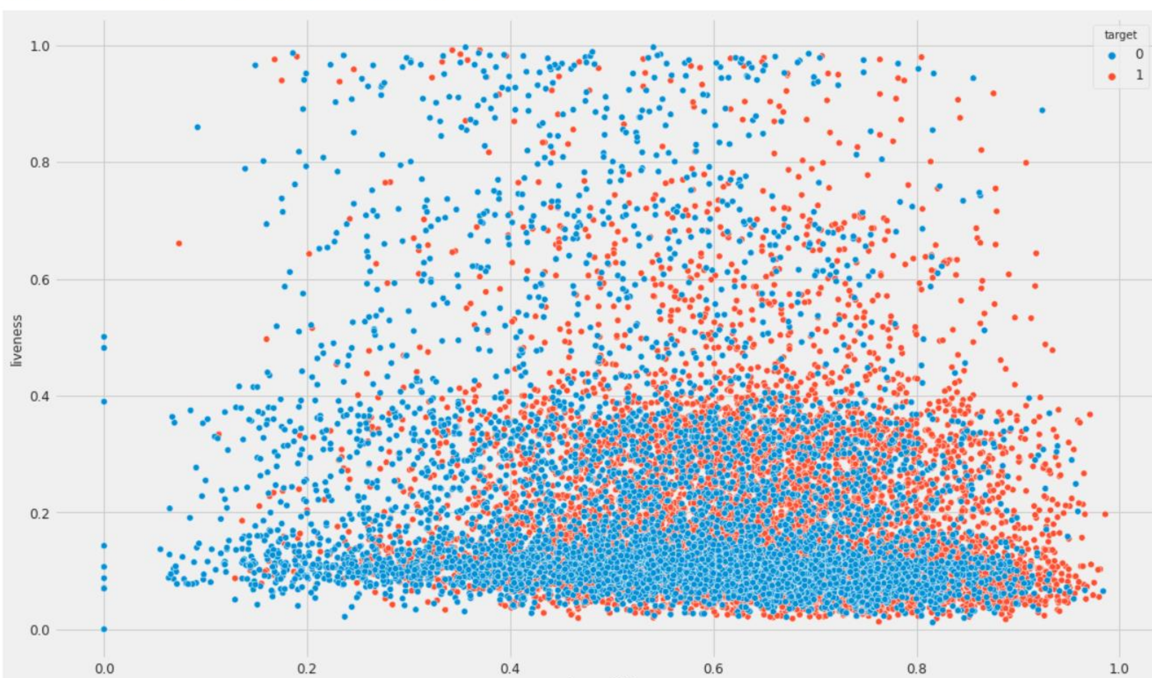


Fig:Heatmap

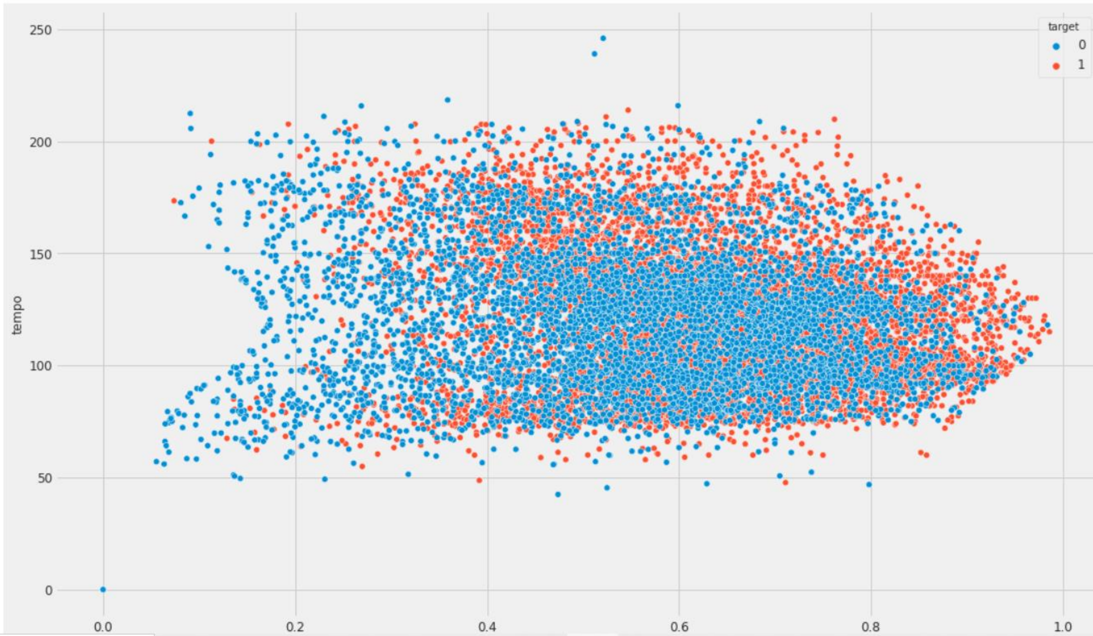
```
In [66]: sns.scatterplot(x='danceability',y='loudness',hue='target',data=df)
Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1a2efaf190>
```



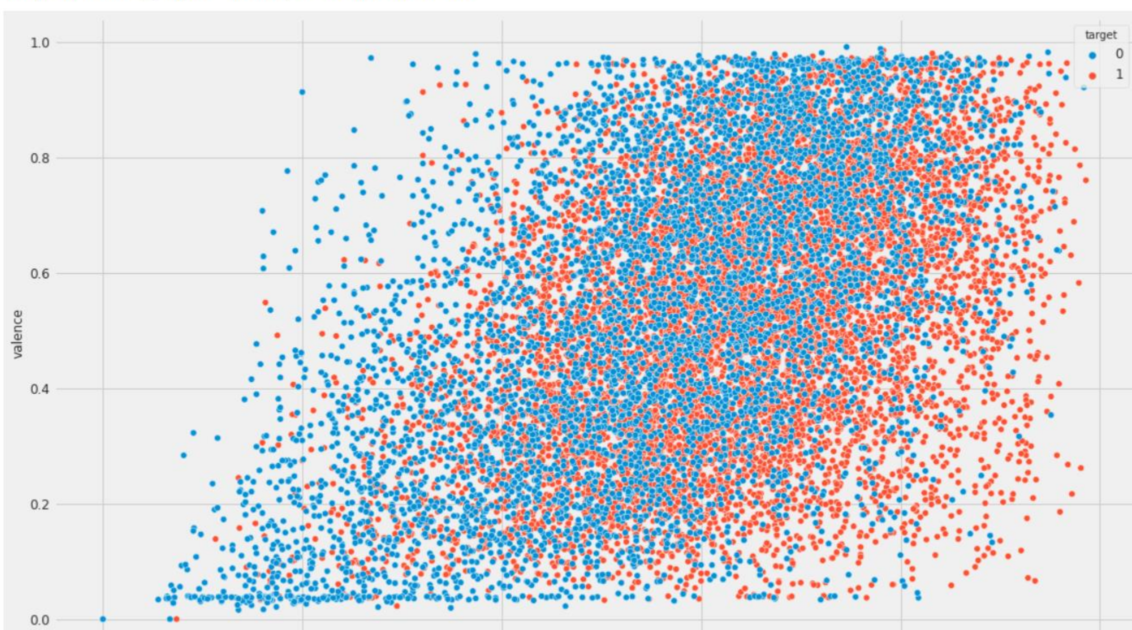

```
In [67]: sns.scatterplot(x='danceability',y='liveness',hue='target',data=df)
Out[67]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1a2f4a2990>
```



```
In [68]: sns.scatterplot(x='danceability',y='tempo',hue='target',data=df)
Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1a35fefdd0>
```




```
In [69]: sns.scatterplot(x='danceability',y='valence',hue='target',data=df)
Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1a2f4f1450>
```



```
In [70]: sns.scatterplot(x='danceability',y='energy',hue='target',data=df)
Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1a2eff5f90>
```

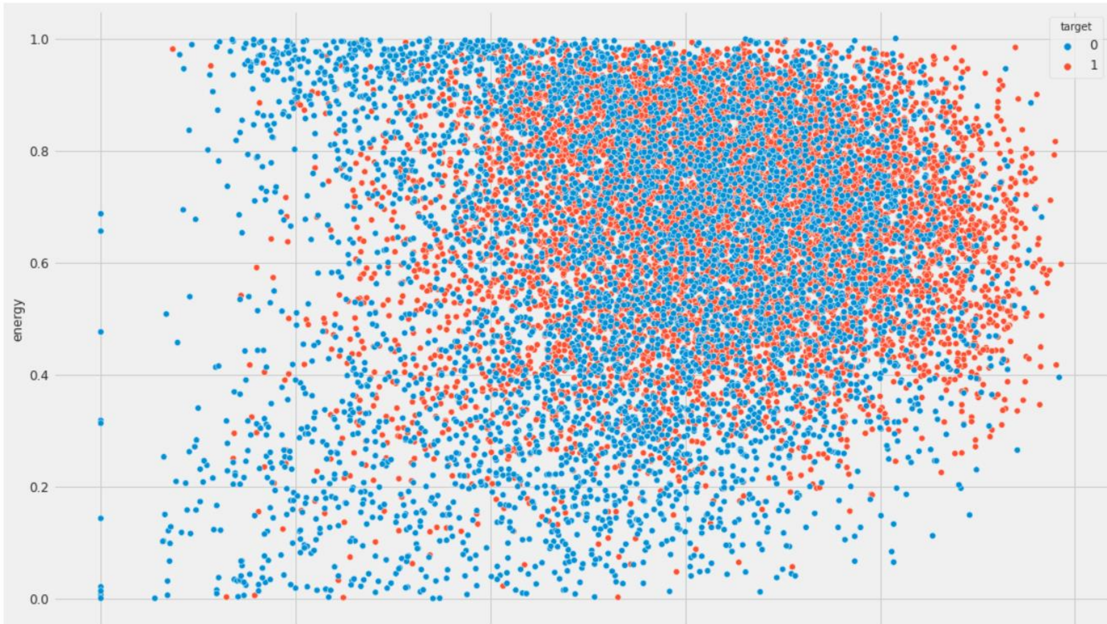


Fig:Scatterplots

	vif	Features
0	1.59	danceability
1	3.32	energy
2	1.03	key
3	2.51	loudness
4	1.12	speechiness
5	1.94	acousticness
6	1.27	instrumentalness
7	1.10	liveness
8	1.58	valence
9	1.07	tempo
10	1.05	duration_ms
11	1.05	mode

Fig:VIF

```
y_pred = log_reg.predict(x_test)
```

```
accuracy = accuracy_score(y_test,y_pred)
accuracy
```

```
0.7676438653637351
```

Fig:Logistic Regression Accuracy

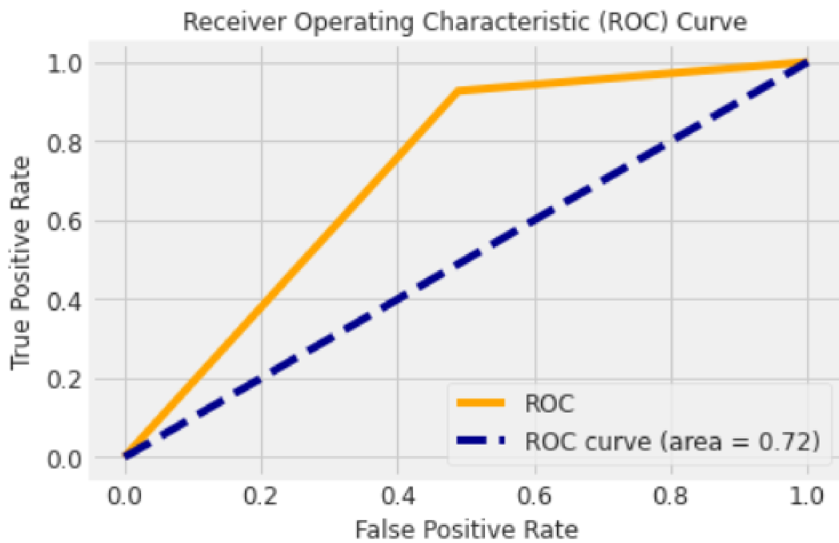


Fig:ROC CURVE

```
In [107]: Ldaparam_grid = {
              'solver': ['svd', 'lsqr', 'eigen'],
              'shrinkage': list(arange(0,1,0.01)),
          }
          Lda_search = GridSearchCV(lda, param_grid=Ldaparam_grid,n_jobs=-1)

          # fitting the model for grid search
          Lda_search.fit(x_train , y_train)
          Lda_search.best_params_
          # summarize
          print('Mean Accuracy: %.3f' % Lda_search.best_score_)
          print('Config: %s' % Lda_search.best_params_)

          Mean Accuracy: 0.766
          Config: {'shrinkage': 0.06, 'solver': 'lsqr'}
```

Fig:LDA ACCURACY

```
In [115]: qda = QuadraticDiscriminantAnalysis(reg_param=0.01,store_covariance=True,tol=0.0001)
          qda.fit(x_train,y_train)

Out[115]: QuadraticDiscriminantAnalysis(reg_param=0.01, store_covariance=True)

In [116]: qda.score(x_test,y_test)

Out[116]: 0.7717155266015201
```

Fig:QDA ACCURACY

DEPLOYMENT & APP INTERFACE:

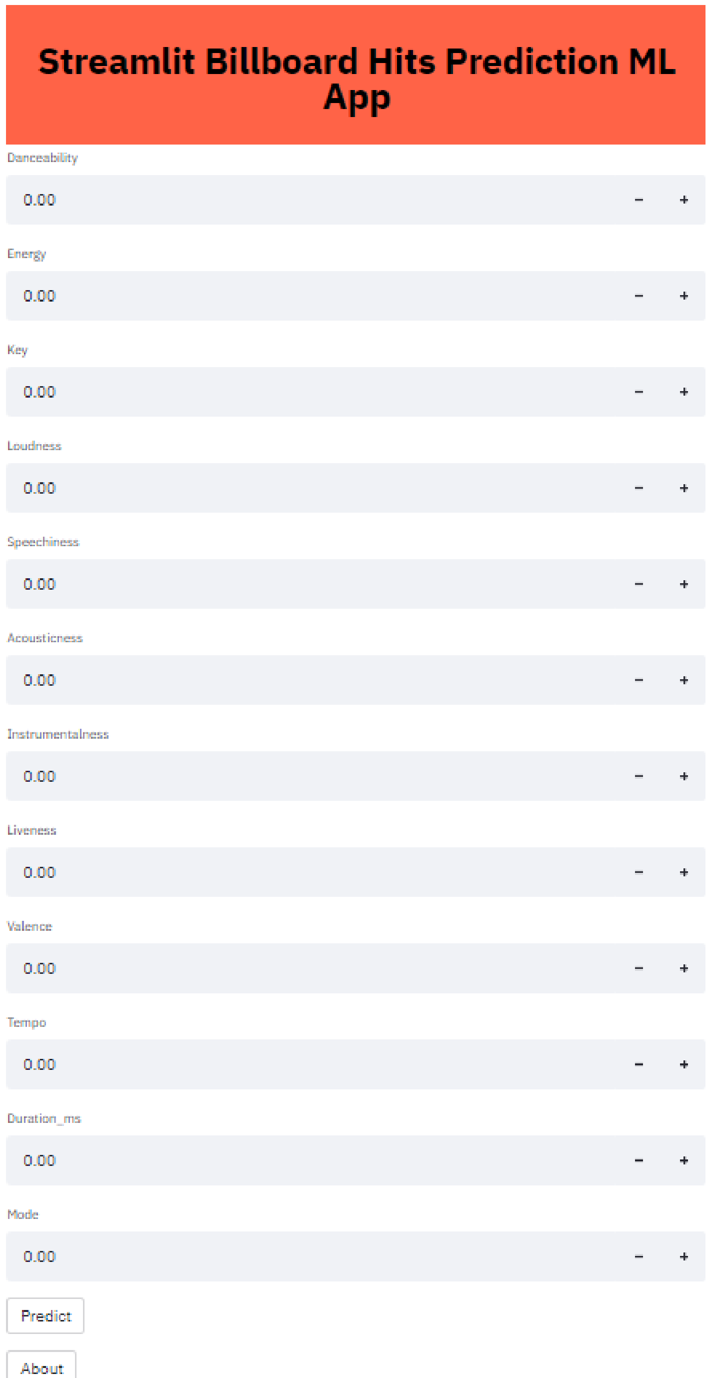


Fig:STREAMLIT APP

Fig:FLASK APP INTERFACE

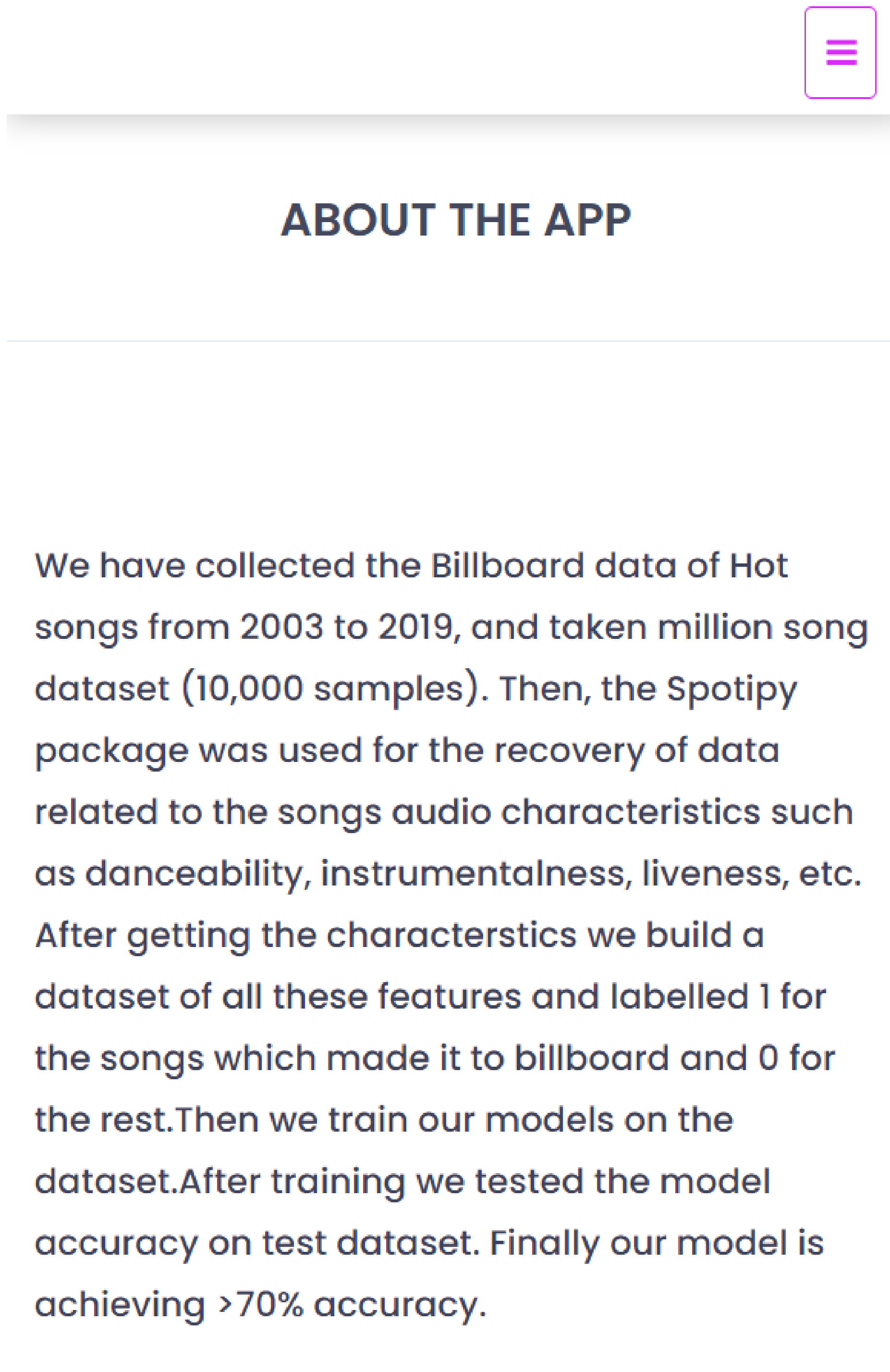


Fig:FLASK-APP ABOUT PAGE

TECHNOCOLABS DATA SCIENCE INTERNSHIP

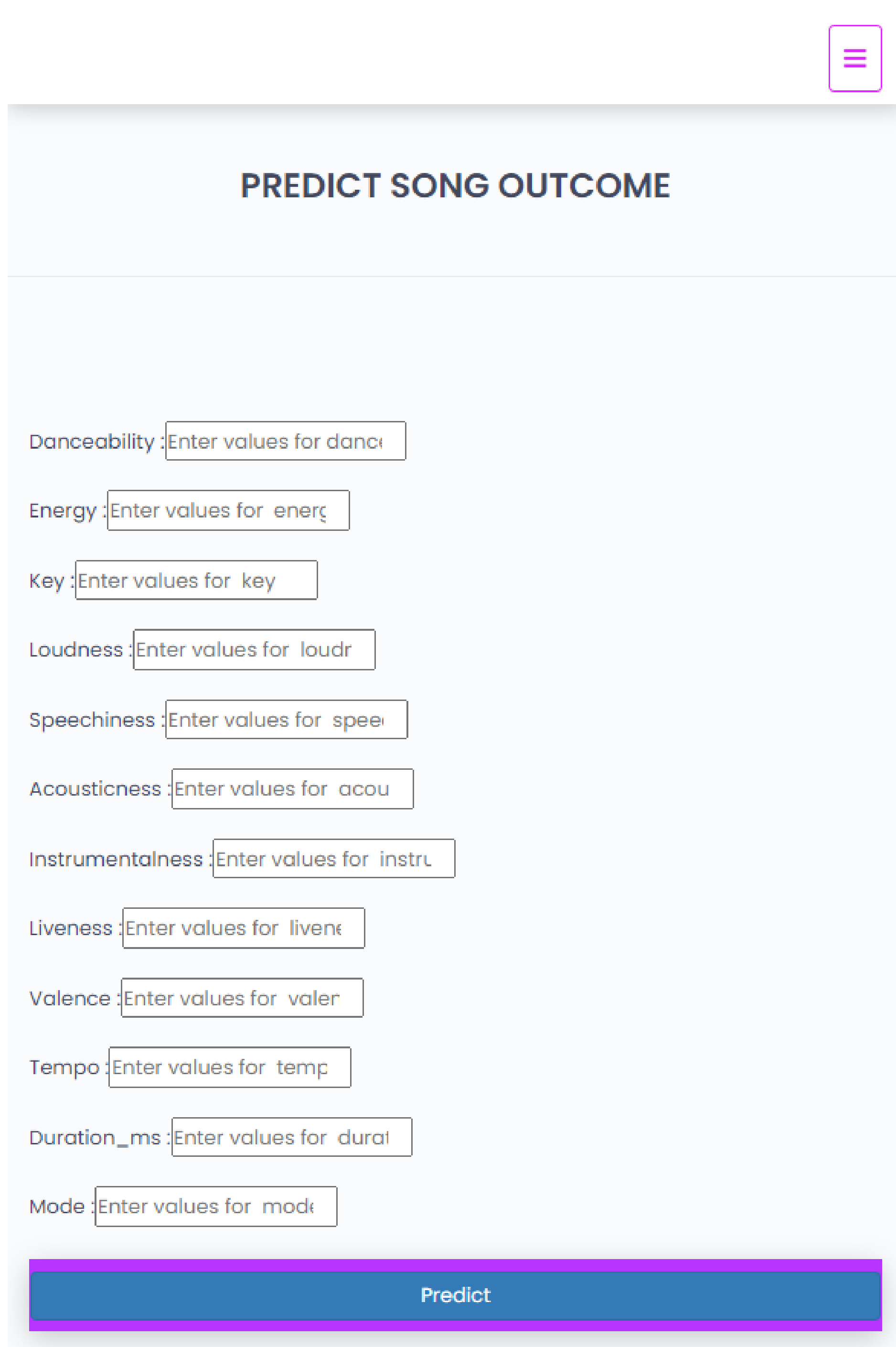


Fig:PREDICT PAGE