Predicting billboard hits using Spotify Data

By: Piyusha More

M.Sc..(Data Science and Spatial Analytics)

Agenda

- Introduction
- Objective
- Methodology
- Result
- Conclusion
- References

Introduction

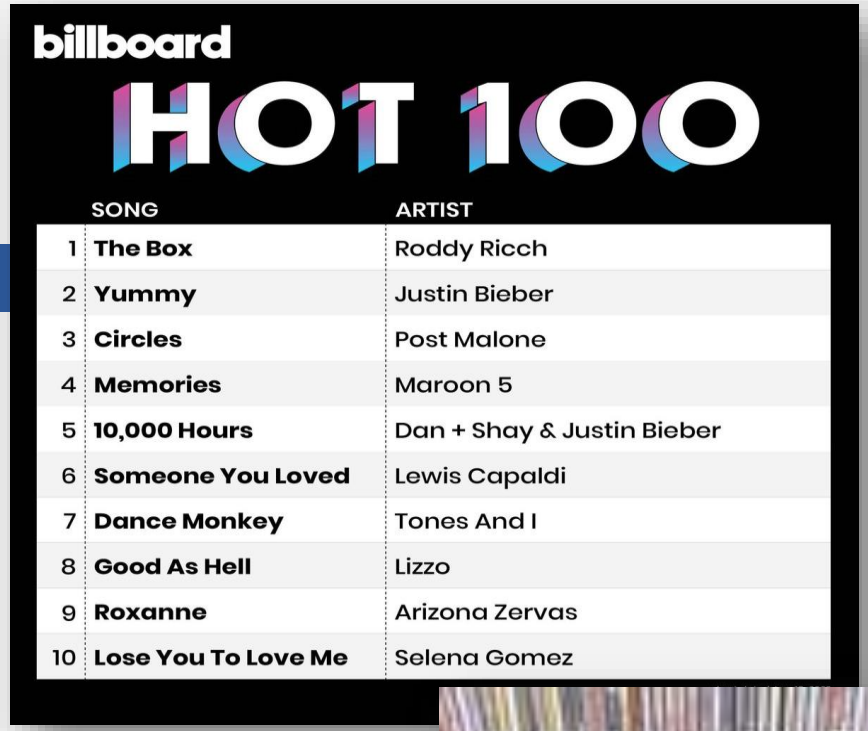
The Billboard Hot 100 Chart remains one of the definitive ways to measure the success of a popular song.

The Million Dataset is a freely available collection of audio features and metadata for contemporary popular music tracks.

Spotify Web API was used for extracting audio features of collected songs from this two sites.

Investigation was done using machine learning techniques to predict which songs will become Billboard Hot 100 Hits.

It was able to able to predict the Billboard success of a song with ~75% accuracy using machine-learning algorithms including Logistic Regression, GDA, SVM, and Decision Trees.

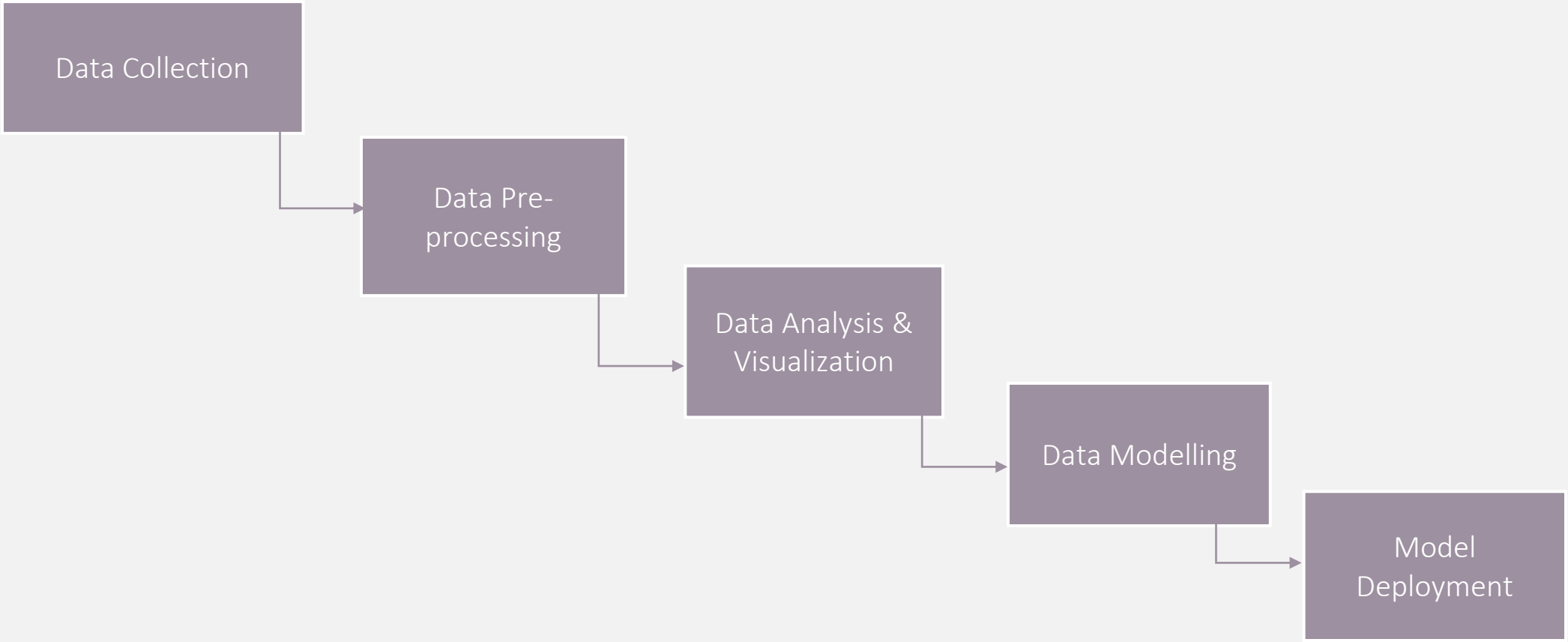


	SONG	ARTIST
1	The Box	Roddy Ricch
2	Yummy	Justin Bieber
3	Circles	Post Malone
4	Memories	Maroon 5
5	10,000 Hours	Dan + Shay & Justin Bieber
6	Someone You Loved	Lewis Capaldi
7	Dance Monkey	Tones And I
8	Good As Hell	Lizzo
9	Roxanne	Arizona Zervas
10	Lose You To Love Me	Selena Gomez

Objective

To predict songs will become Billboard Hot 100 Hits or not using Spotify API data by applying machine learning algorithms

Flowchart


Data Collection
Data Pre-processing
Data Analysis & Visualization
Data Modelling
Model Deployment

Data Collection

- Data of approximately 4000 songs was collected from Billboard.com and the Million Song Dataset

- Spotify Web API was used for extracting audio features of collected songs from given two sites.

Danceability	Valence
Instrumentalness	Energy
Acousticness	Loudness
Speechiness	Tempo
Liveness	Mode

First rows

	df_index	Track	Artist	SpotifyID	danceability	energy	key	mode	speechiness	acousticness	instrumentalness	liveness
0	0	Lucid Dreams	Juice WRLD	285pBltuF7vW8TeWk8hdRR	0.511	0.566	6	0	0.2000	0.34900	0.00000	0.3400
1	1	Better Now	Post Malone	7dt6x5M1jzdTEt8oCbisTK	0.680	0.578	10	1	0.0400	0.33100	0.00000	0.1350
2	2	Drip Too Hard	Lil Baby & Gunna	78QR3Wp35dqAhFEc2qAGjE	0.897	0.662	1	0	0.2920	0.08520	0.00000	0.5340
3	3	Sicko Mode	Travis Scott	2xLMifQCjDGFmkHkpNLD9h	0.834	0.730	8	1	0.2220	0.00513	0.00000	0.1240
4	4	Youngblood	5 Seconds Of Summer	2iUXsYOEPhVqEBwsqP70rE	0.596	0.854	7	0	0.4630	0.01690	0.00000	0.1240
5	5	I Like It	Cardi	58q2HKrzhC3ozto2nDdN4z	0.816	0.726	5	0	0.1290	0.09900	0.00000	0.3720
6	6	In My Feelings	Drake	2G7V7zsVDxg1yRsu7Ew9RJ	0.835	0.626	1	1	0.1250	0.05890	0.00006	0.3960
7	7	Natural	Imagine Dragons	2FY7b99s15jUprqC0M5NCT	0.704	0.611	2	1	0.0409	0.21700	0.00000	0.0812
8	8	Trip	Ella Mai	6CTWathupliDs7U4InHnDA	0.477	0.610	11	0	0.1440	0.22500	0.00000	0.1070
9	9	I Love It	Kanye West & Lil Pump	4S8d14HvHb70ImctNgVzQQ	0.901	0.522	2	1	0.3300	0.01140	0.00000	0.2590

Data Pre-Processing

- Track and artist column consisted few of the null values which were dropped using dropna() function

```
In [6]: billboard_df.info
        billboard_df.isnull().sum()

Out[6]: Track                1
        Artist               2
        SpotifyID            0
        danceability         0
        energy               0
        key                  0
        mode                 0
        speechiness          0
        acousticness         0
        instrumentalness     0
        liveness             0
        valence              0
        tempo                0
        duration_ms          0
        loudness             0
        dtype: int64
```

- The target column “mode” should be having values as 0 or 1 but it was consisting some unknown values

```
In [33]: print(pd.unique(MSD_Billboard_Dataset['mode']))

         [   0    1 -999]
```

Data Pre-Processing

- Spotify Id column in the dataset consisted repeated values which were necessary to remove those

```
In [26]: MSD_Billboard_Dataset.SpotifyID.value_counts()

Out[26]: 23wfXwnsPZYe5A1xXRHb3J    4
         51TVALqY7g3McuAwjJzVxG    3
         3aiKybRCTBazAplseCewQc    3
         5BkHkyO9PFXs1m7vSMnXp4    3
         74irxdVWstNlEQjsvArITq    3
                                  ..
         3mDoAC8R1miOQ6Ld1NkAYH    1
         1DJgRkwljWXGb1sFxfSlOE    1
         4PzovBqgnSHKd8opsP7IVM    1
         76vMKwFtdDDCLcM6zXybjB    1
         1Aw4wEcRorA2Y7wyBgxEmX    1
         Name: SpotifyID, Length: 14745, dtype: int64
```

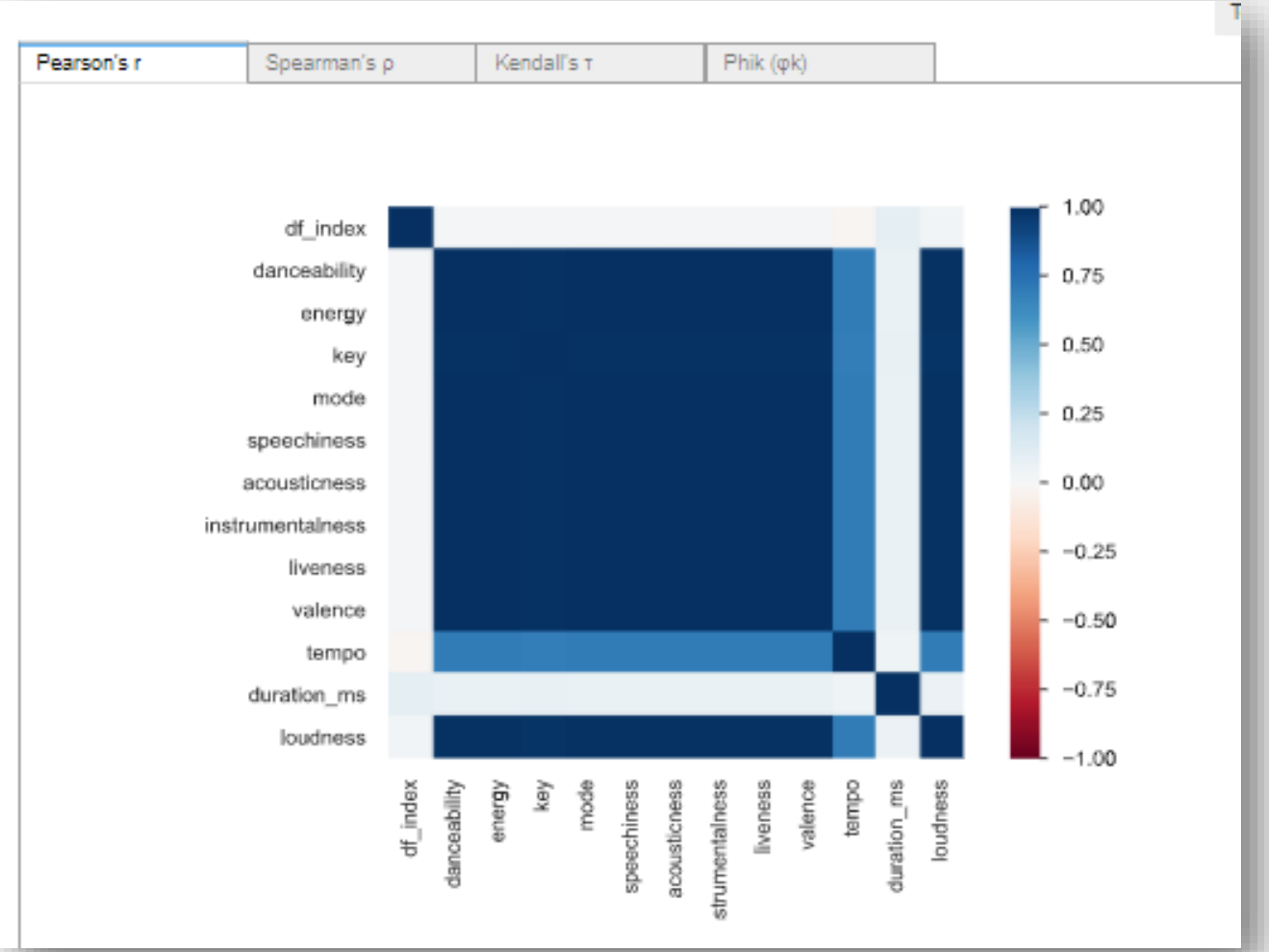
spotify id is unique to each song but it is repeating so remove those repeating songs

Data Analysis & Visuals

From this visual, it is easily interpreted that the variables are correlated to each other.

Why it could not be?

Afterall they are features of song which go together to make a song pretty good to hear.



Data Modeling

- Two algorithms were used for data modelling which are as follows,
- SVM(Support Vector Machine)
- Random Forest

```
In [83]: #X has independent variables
         X = df.iloc[:,:11]
         #y has dependent variable
         y = df.hit
```

Data split was done as 75/25 into training/testing sets and separated the features as dependent and independent variables.
And both the algorithms were applied to it.

```
###split data into train and test data

In [8]: from sklearn.model_selection import train_test_split
        X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

Results

SVM classifier using linear kernel

```
In [23]: ### Implement SVM classifier
         from sklearn.svm import SVC
         SVMclassifier = SVC(kernel= "linear")
         SVMclassifier.fit(X_train, y_train)

Out[23]: SVC(kernel='linear')

In [24]: ### Prediction
         y_pred = SVMclassifier.predict(X_test)

In [27]: ### Check Accuracy
         from sklearn.metrics import accuracy_score
         score=accuracy_score(y_test,y_pred)
         score

Out[27]: 0.6800330715171559
```

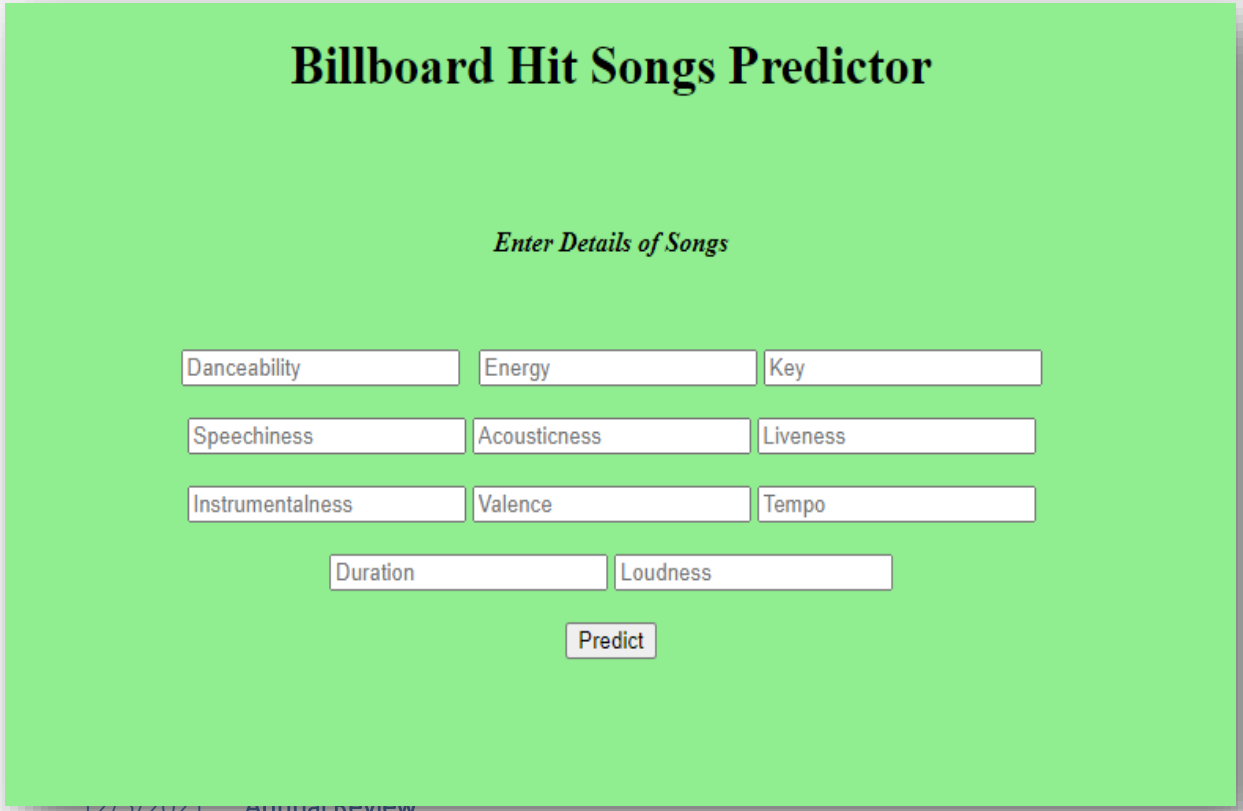
Random forest classifier using 100 estimators

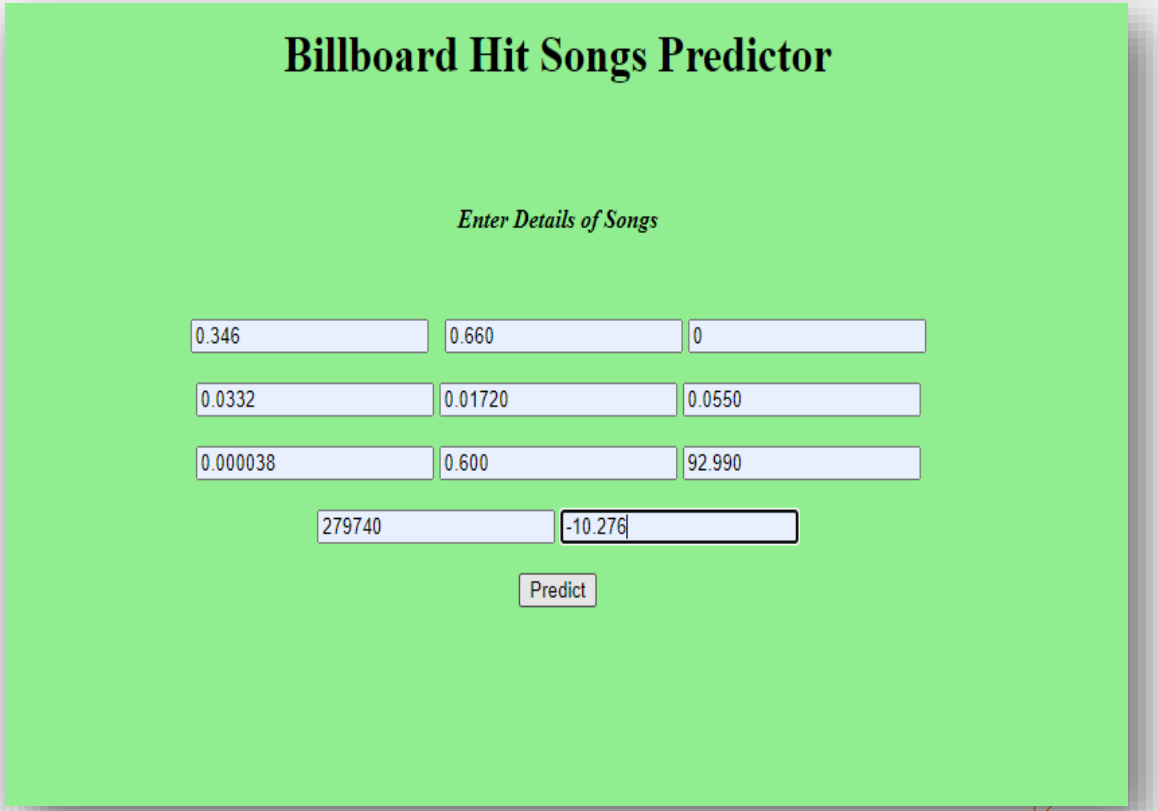
```
In [11]: ### Implement Random Forest classifier
         from sklearn.ensemble import RandomForestClassifier
         classifier = RandomForestClassifier(n_estimators=100)
         classifier.fit(X_train, y_train)

Out[11]: RandomForestClassifier()

In [12]: ### Prediction
         y_pred = classifier.predict(X_test)

In [13]: ### Check Accuracy
         classifier.score(X_test, y_test)

Out[13]: 0.7254383063182269
```

- Looking at the results the model gaining higher accuracy was decided to be deployed

Model Deployment

- Flask is an API in python that allows model developers to build up web applications.
- The model gaining higher accuracy was random forest so flask web-application was created based on it.

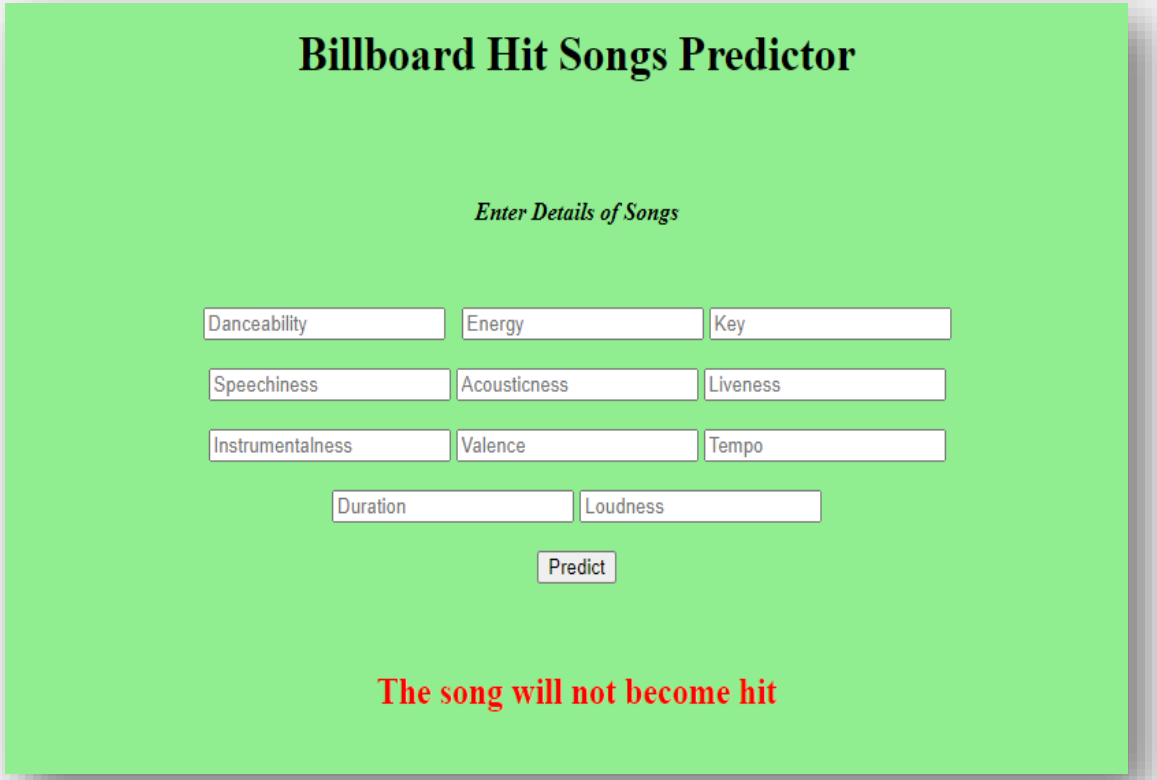




Model Deployment

Continued..

Based on the song features the outcome was as below,



Conclusion

Billboard hits is not a new thing, but I tried a small part of myself to let me understand how I can apply data science techniques in this domain.

How much useful it can be if I try to apply a particular set of algorithms to the dataset.

I originated that random forest is one of the most powerful and widely used algorithms which enables organizations to solve classification and regression problems effectively.

The created models can be deployed on many platforms like Streamlit, Heroku, etc..

References

- H. Dai, “Research on SVM improved algorithm for large data classification,” 2018 IEEE 3rd Int. Conf. Big Data Anal. ICBDA 2018, no. 1, pp. 181–185, 2018, doi: 10.1109/ICBDA.2018.8367673.
- M. Claesen, F. De Smet, J. A. K. Suykens, and B. De Moor, “Fast Prediction with SVM Models Containing RBF Kernels,” 2014, [Online]. Available: http://arxiv.org/abs/1403.0736.
- J. R. Thompson and B. L. Licklider, “Visualizing Urban forestry: Using concept maps to assess student performance in a learning-centered classroom,” J. For., vol. 109, no. 7, pp. 402–408, 2011, doi: 10.1093/jof/109.7.402.
- https://www.billboard.com/charts/hot-100
- http://millionsongdataset.com/pages/getting-dataset/
- https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
- https://builtin.com/data-science/random-forest-algorithm 5. https://www.geeksforgeeks.org

Thank You!