Win Prediction: Data Analytics Project Overview

INTRODUCTION

IPL (Indian Premier League) is a highly popular professional Twenty20 cricket league in India. With the increasing popularity of cricket and the IPL, there has been a growing demand for accurate match predictions. Machine learning can be used to develop an IPL match prediction system that can analyze the past performance of teams, players, weather conditions, and other relevant factors to predict the outcome of a match.

OBJECTIVES

The objective of an IPL match win prediction project is to develop a predictive model that accurately forecasts the outcome of Indian Premier League (IPL) matches. This involves analyzing historical match data, player statistics, team performance metrics, and other relevant factors to identify patterns and trends that can help predict the likelihood of a team winning a match. The ultimate goal is to create a reliable model that can assist cricket enthusiasts, sports analysts, and betting enthusiasts in making informed decisions about match outcomes.

WHY WE CHOOSE THIS TOPIC

- **Entertainment**: People enjoy the excitement and thrill of predicting the outcome of sports events, including IPL matches. It adds an extra layer of excitement to watching the games.
- **Competition**: Prediction contests, whether among friends, colleagues, or online communities, create a sense of competition and engagement. It's fun to compete with others to see who can make the most accurate predictions.

ATTRIBUTE SELECTION

The main attributes like-

- Batting Team
- Bowling Team
- Host City (Stadium)
- Target
- Score
- Overs completed
- Wickets Out

WORKFLOW

- The first step in building an IPL match prediction system using machine learning is to collect and preprocess the data. The data can include historical match data, team and player statistics, weather conditions, pitch conditions, and other relevant information.
- After the model is trained, it can be used to predict the outcome of a match by inserting the relevant data for the upcoming match. The model can provide a probability of the outcome, such as the probability of a team winning or losing the match.

DATASET COLLECTING

↓

DATA CLEANING

↓

ATTRIBUTE SELECTION

↓

DATA MINNING

↓

ANALYSIS RESULT

DATA SET

- Here we collected data from Kaggle which contains the details regarding the winner and match stats.
- It contains the details like teams played, winner, venue of the match, won by how many wickets and runs, toss decision, whether DLS applied or not, names of the umpires etc.

```
match = pd.read_csv('matches.csv')
delivery = pd.read_csv('deliveries.csv')
```

```
match.head()
```

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_of_m
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuvraj S
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	SPD S
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	10	CA
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6	GJ Ma
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	0	KM Ja

IMPLEMENTATION

- The first step is importing all required libraries.

```
import pandas as pd
import numpy as np
import seaborn
import matplotlib.pyplot as plt
import sklearn
```

- Let’s import the dataset. We’ve already discussed the dataset.

```
match = pd.read_csv('matches.csv')
delivery = pd.read_csv('deliveries.csv')
```

IMPLEMENTATION

- first 5 rows data.head()

```
match.head()
```

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets	player_of_m
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0	Yuvraj S
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7	SPD S
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	10	CA
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6	GJ Ma
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	0	KM Ja

```
delivery.head()
```

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_over	...	bye_runs	legbye_runs	noball_runs	penalty_runs
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	0	...	0	0	0	0

IMPLEMENTATION

- We don’t need all the features or columns in order to create the model. so we are dropping some of the features that don’t affect our result.

team1	team2	toss_winner	toss_decision	result	dl_applied	winner	win_by_runs	win_by_wickets
Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	0
Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	7
Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	10
Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	6
Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	0
...	...	...	...	...	...	...	...	...
Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0	Mumbai Indians	0	9
Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	normal	0	Mumbai Indians	0	6

IMPLEMENTATION

- Now let’s look into the total teams listed in this dataset

```
match_df['team1'].unique()
```

```
array(['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',
       'Rising Pune Supergiant', 'Royal Challengers Bangalore',
       'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',
       'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',
       'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiants',
       'Delhi Capitals'], dtype=object)
```

- Here you can see the name Delhi Daredevils and Delhi Capitals; Delhi Daredevils is old name of the Delhi Capitals and Similarly Deccan Chargers. So we are changing the old name to the newer one.

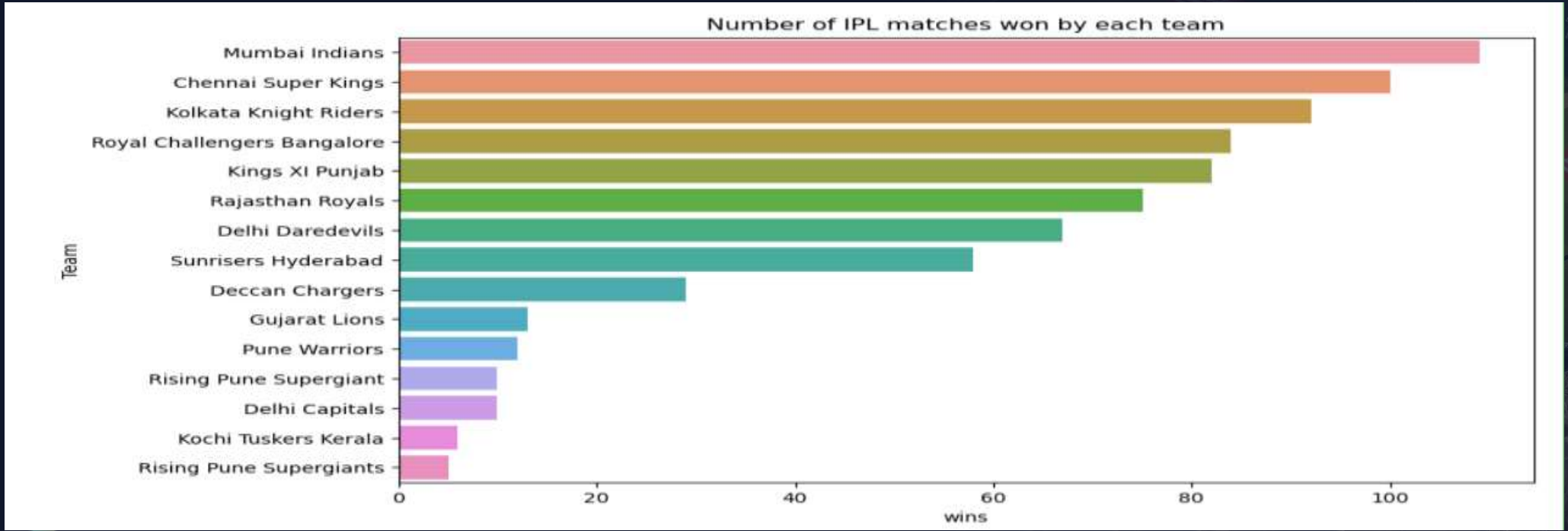
```
match_df['team1'] = match_df['team1'].str.replace('Delhi Daredevils','Delhi Capitals')
match_df['team2'] = match_df['team2'].str.replace('Delhi Daredevils','Delhi Capitals')

match_df['team1'] = match_df['team1'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
match_df['team2'] = match_df['team2'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
```

EXPLORATORY DATA ANALYSIS

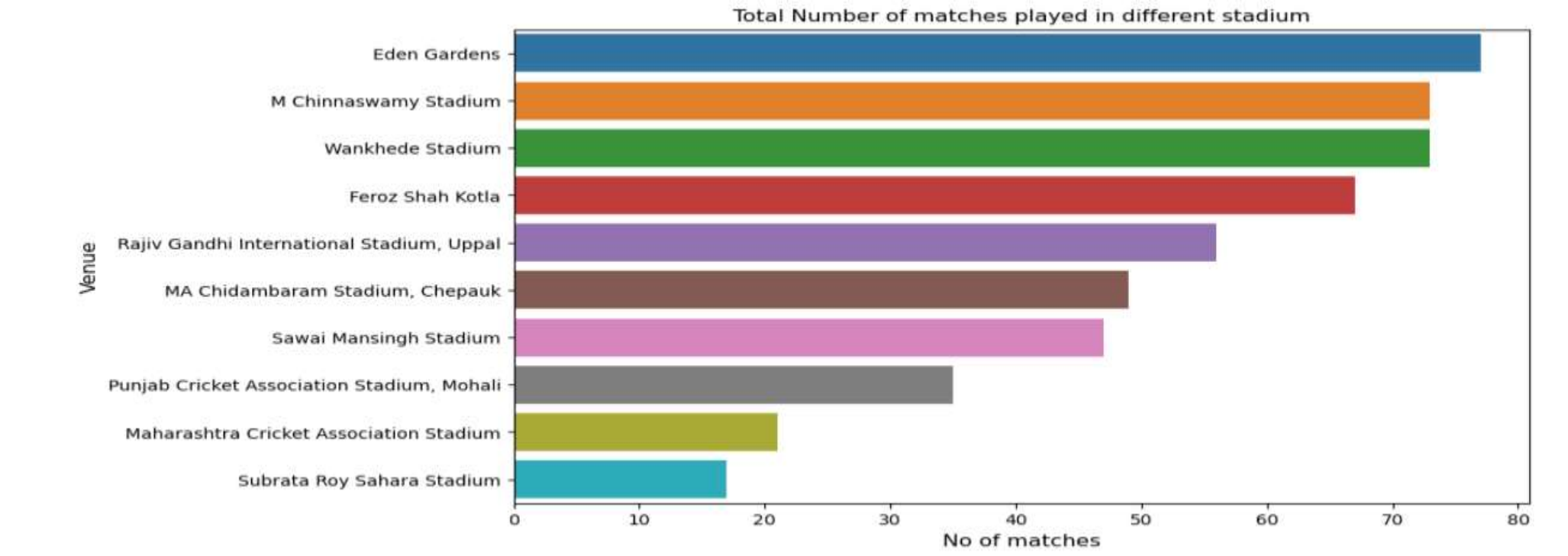
VISUALIZATIONS

- Number of IPL matches won by each team
  Countplot



EXPLORATORY DATA ANALYSIS
VISUALIZATIONS

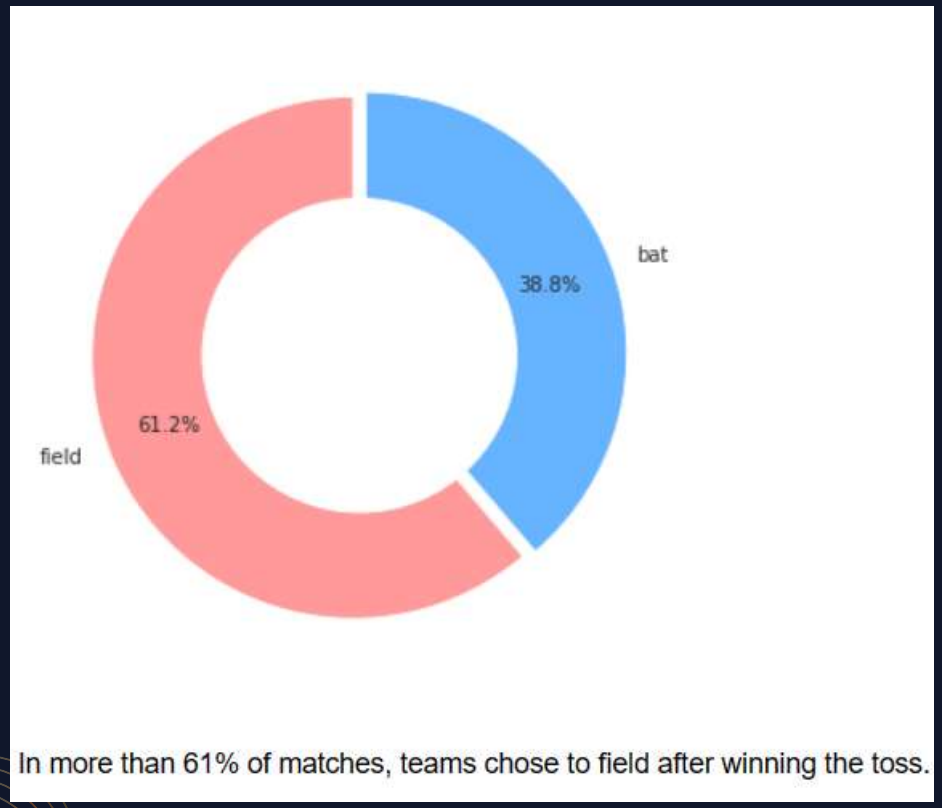
- Total number of matches played in a different stadium



EXPLORATORY DATA ANALYSIS

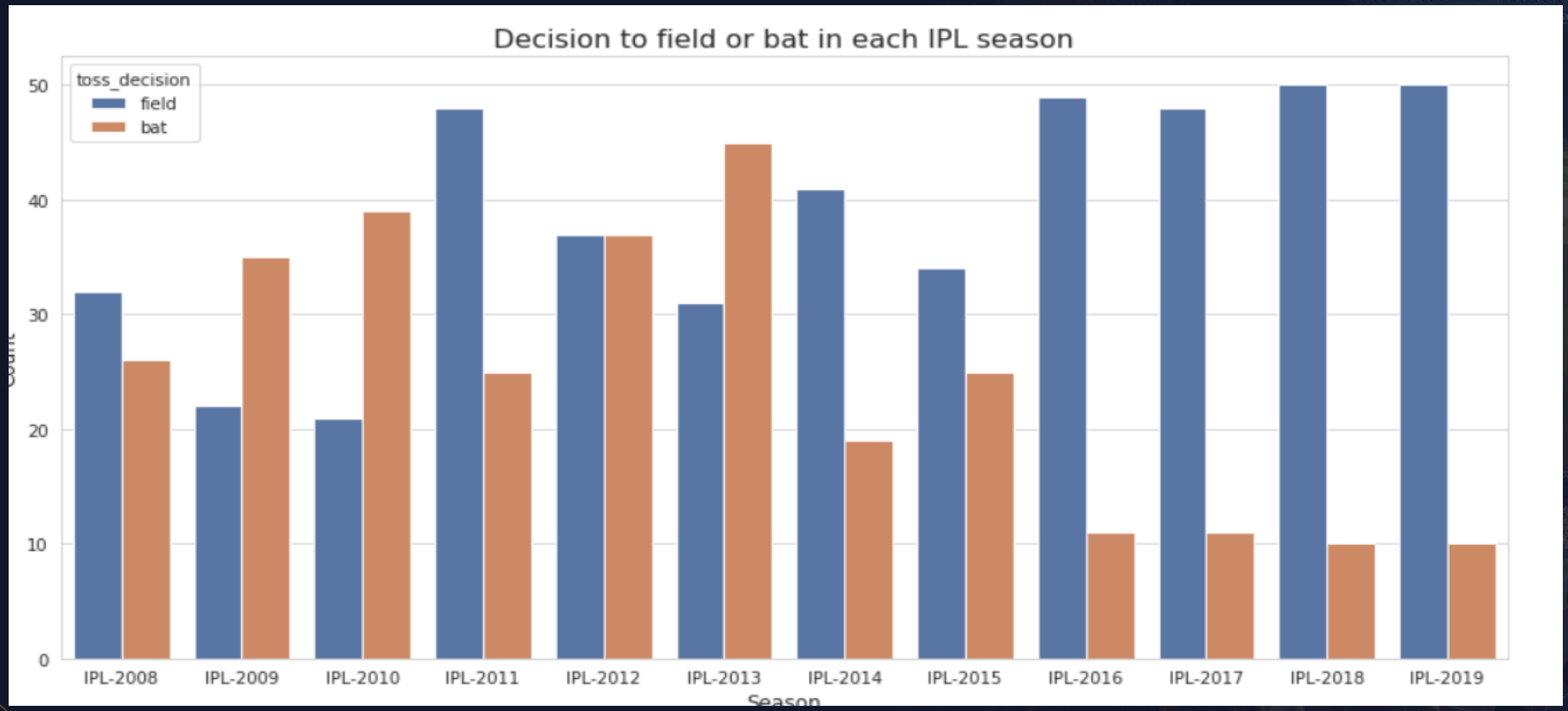
VISUALIZATIONS

- The decision was taken by the toss winning team
- Pie Chart



EXPLORATORY DATA ANALYSIS
VISUALIZATIONS

- Toss decision to field or bat in each IPL season



MODEL CREATION AND EVALUTION

- Now let's convert our data into a training set in order to create the model and test set for evaluating the created model.

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=1)
```

- The next and most important step in the model creation step. So we are using –
  - Random Forest Classification
  - Logistic Regression
  - Pipeline

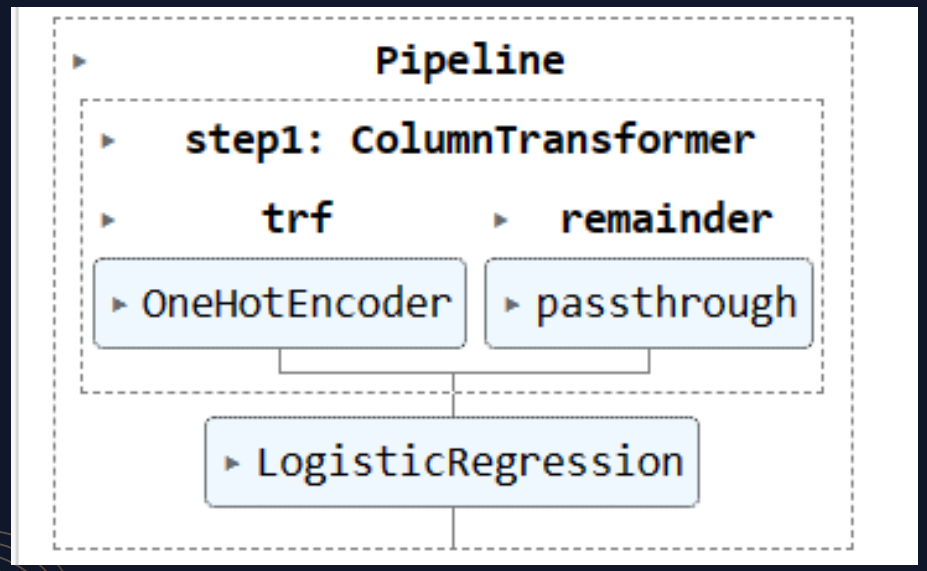
```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
```

```
pipe = Pipeline(steps=[
    ('step1',trf),
    ('step2',LogisticRegression(solver='liblinear'))
])
```

MODEL CREATION AND EVALUTION

- Training the random forest classifier model

```
pipe.fit(X_train,y_train)
```



MODEL CREATION AND EVALUTION

- Predicting the model with x_test values and saving it as y_pred

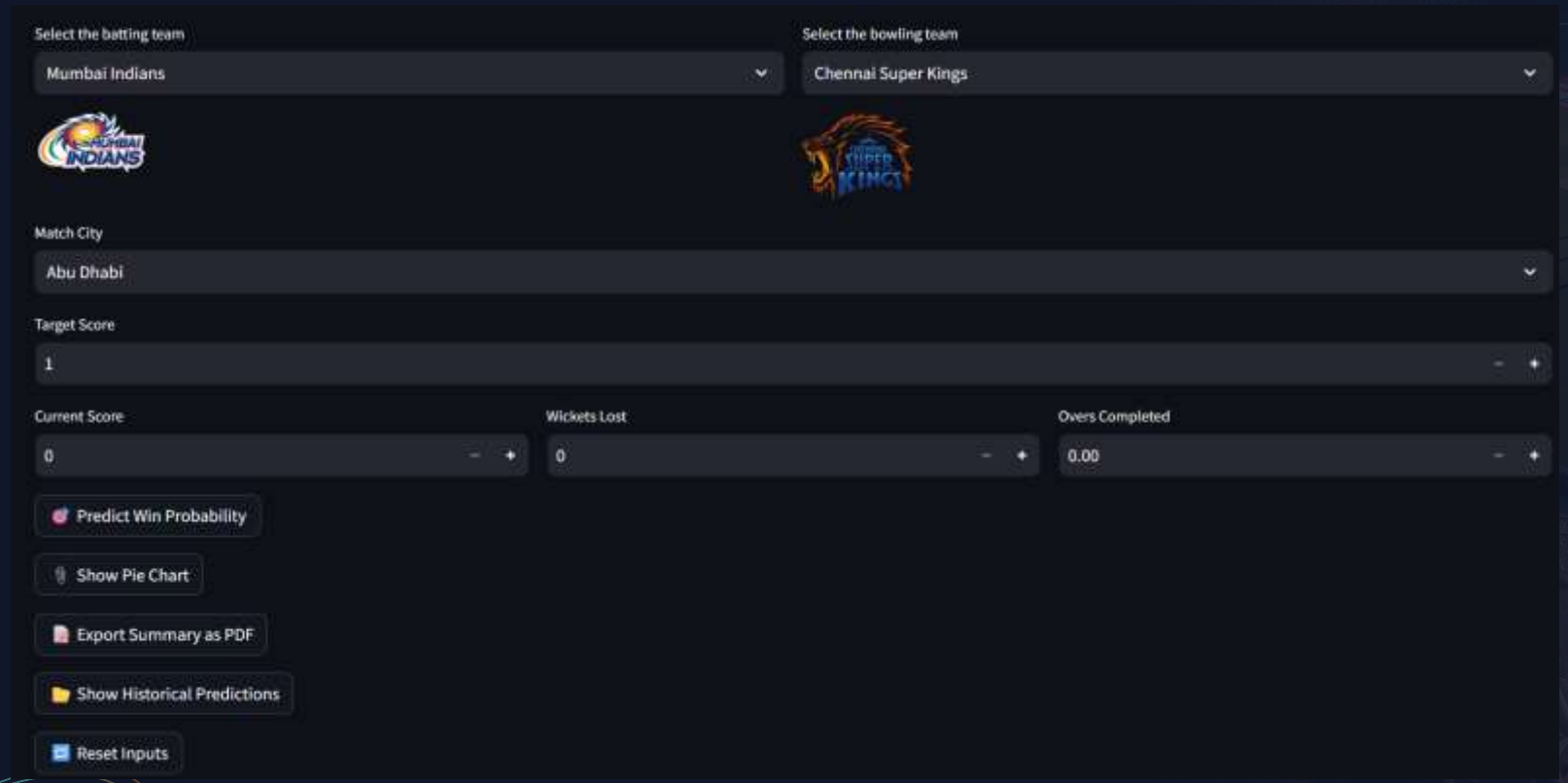
```
y_pred = pipe.predict(X_test)
```

- Using the accuracy score of sklearn, we are evaluating the predicted result and accuracy of the model

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

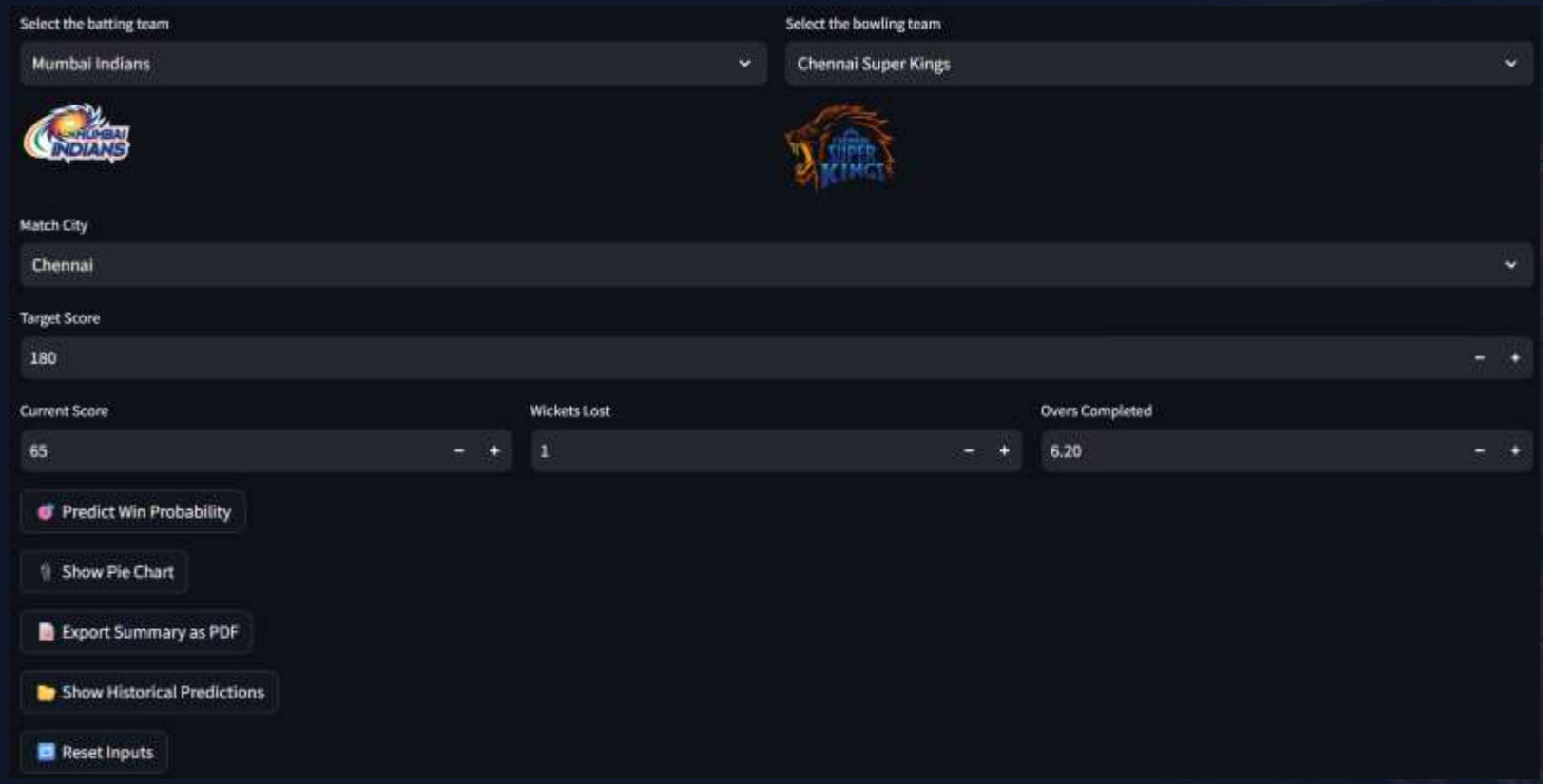
FINAL OUTPUT OF PROJECT

GUI OF PREDICTOR



FINAL OUTPUT OF PROJECT

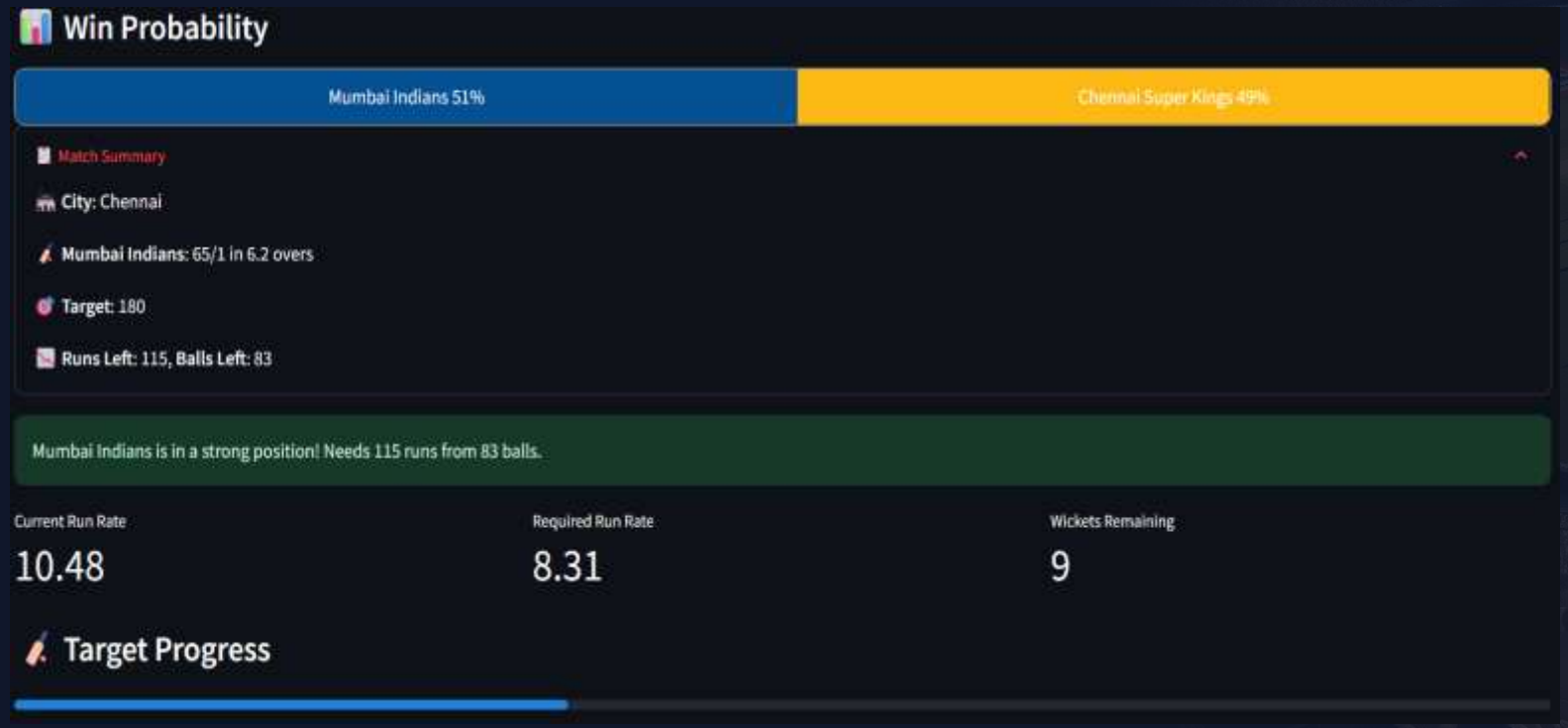
Filling Data on Match Conditions



FINAL OUTPUT OF PROJECT

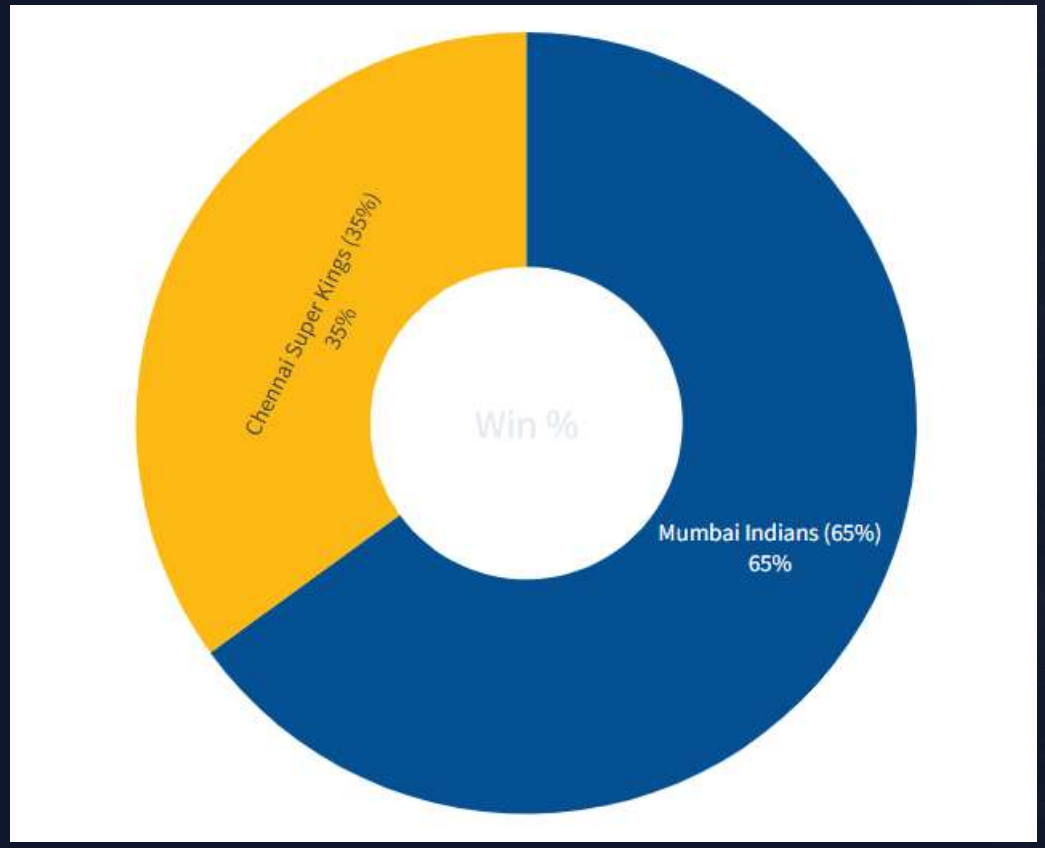
After filling in all the data, click on:

Predict Win Probability to calculate the chance of winning.



FINAL OUTPUT OF PROJECT

Show Pie Chart to visualize the prediction.



FINAL OUTPUT OF PROJECT

Show Historical Predictions

	bat	bowl	city	bat%	bowl%
0	Mumbai Indians	Chennai Super Kings	Chennai	65	35
1	Mumbai Indians	Chennai Super Kings	Chennai	77	23
2	Mumbai Indians	Chennai Super Kings	Chennai	13	87
3	Mumbai Indians	Chennai Super Kings	Chennai	97	3
4	Mumbai Indians	Chennai Super Kings	Mumbai	99	1
5	Mumbai Indians	Chennai Super Kings	Mumbai	100	0

This feature retrieves and displays past win probability predictions between the selected teams under similar match conditions

FINAL OUTPUT OF PROJECT

Export Summary as PDF

Clicking this button allows you to download a PDF report containing the current prediction details. The report includes the selected batting and bowling teams, the match location, the target and current scores, and the win probabilities. If a pie chart has been generated, it will also be included in the PDF. This feature is especially useful for saving the analysis or sharing it with others.

Reset Inputs

Clicking this button clears all input fields and resets them to their default values. It allows you to start fresh with new match data without having to manually delete previous entries. This is especially helpful when you want to run multiple predictions quickly and efficiently.

Try it live: https://ipl-win-pridiction.onrender.com

CONCLUSION

You have got an insight into how to analyze a given raw data and convert that into useful features by removing unwanted features, that is, performing exploratory data analysis.

- IPL Dataset analysis : This analysis includes checking for null values , describing the dataset’s features columns, and analyzing each feature.
- Visualization of data: Done a graphical representation of the dataset in order to get an understanding of the teams, matches and the toss decision of the IPL Matches.
- Performed preprocessing techniques on the IPL Data: Performed several feature engineering techniques in order to make the dataset suitable for making the model.
- Model creation and Evaluation: This was the main part of the project, and we used the RandomForestClassifier model. Later we tested this model with the test set for evalution.

THANKYOU!!