

# Project Report - Super Predictor for Indian Premier League

Team – Python Eyes

Team members – Amogh Singh, Harisha S and Madhan Kumar S

## Introduction

### Overview

The project we have built consists of IPL dashboard with all the important statistics and details about the IPL season through the years 2008 to 2019. The project includes a website which has a home page displaying more statistics about the IPL seasons. The website also consists of a Predictor which takes input from the user and predicts a winner from the entered details.

### Purpose

This project aims to create a dashboard with essential and an important statistic from the IPL which are often overlooked. This project also includes a predictor and a dashboard to allow the user to view the data and use the predictor and run his own predictions.

## Literature Survey

### Existing problem

Currently there are very few places for information access which provides clear and concise data. The lack of functional and simple websites with a clear UI for users and the lack of accurate predictor services for IPL matches is a huge drawback. Analyzing the game records of the players and teams which is one of the most crucial aspects for trainers and investors alike. Traditional forms of analysis rely on individual interpretations of data and their corresponding ill-defined predictions. The usage of Machine Learning models elevates performance analysis to a great extent. This potential is untapped to its full potential currently.

### Proposed Solution

With our project, we aim to create a clear and concise dashboard which is built using all the exhaustive data of IPL from 2008 to 2019. This dashboard is made with an ease of knowledge understanding and understand ability with precise and color coded graphs which are perfect for the data they represent. The predictor and the website are made with the aim of a simple UI without giving up on the quality and usability. These allow for easier access and simpler knowledge access.

## Theoretical Analysis

### Block Diagram

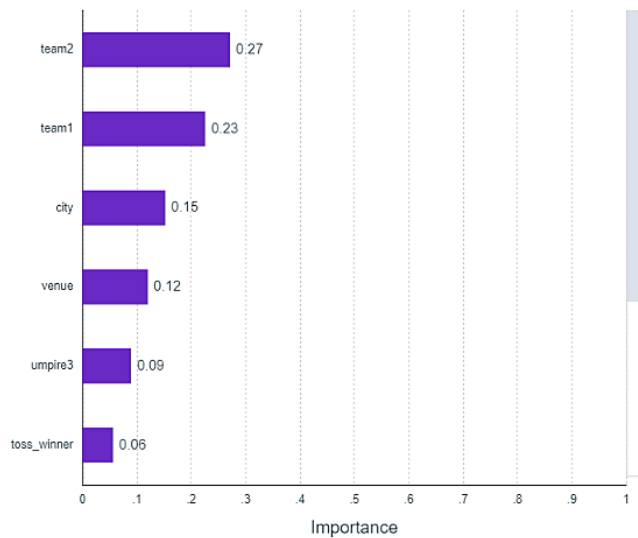
### Hardware/Software Designing

The dashboard was made using the IBM Cognos Analytics service which includes a host of graphs and data representation methods which allows for more powerful and visually pleasing graphs helping us convey more information in a smaller space and time.

The website was made using Flask for the backend with the frontend being made using HTML, CSS and JS. The predictor model was made on Watson Studio using the SPSS Modeler flow and deployed on IBM Cloud for simpler access in the web app. The datasets were uploaded on IBM DB2 for easy storage and accessibility in the web apps for the predictor in the app.

## Experimental Investigations

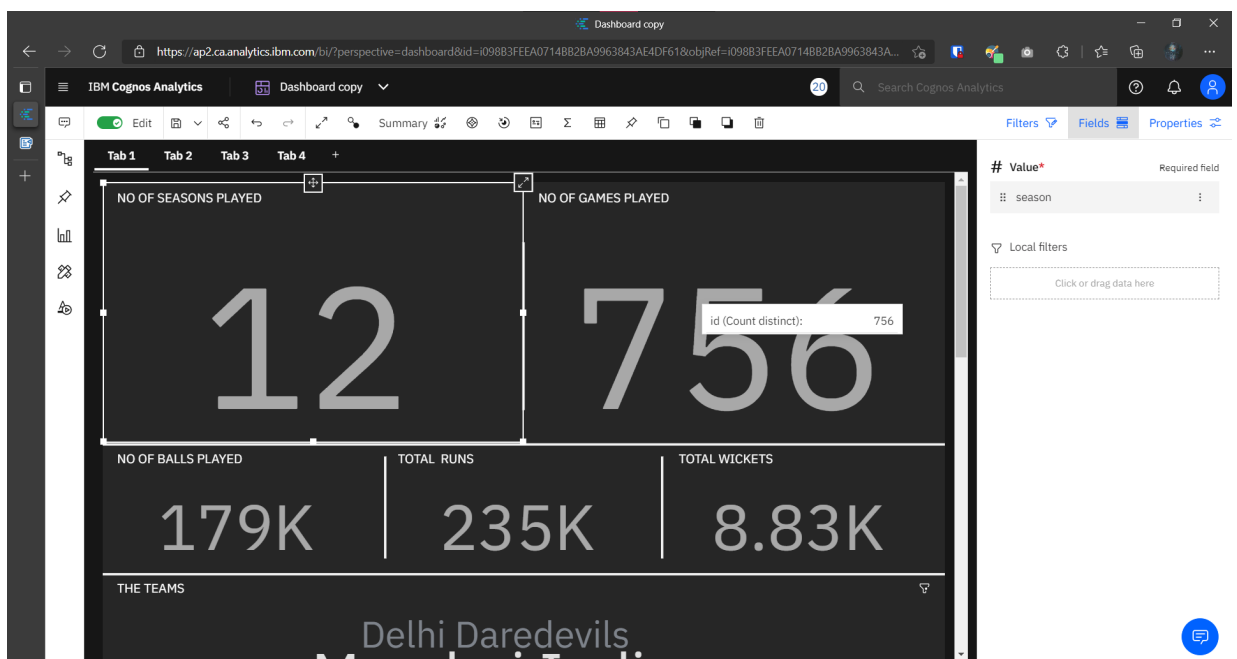
During the making of the dashboard, the data was sorted through and a few mistakes were fixed like mistakes in team names, a few repeated stadium names, etc. These were then used to make a dashboard and a few of the investigations which were found are that winning the toss is not as significant in winning the match as it was perceived to be. The data reveals that winning the toss gives the toss winner a probability of 0.523 of winning which is very close to an equal, 50-50 chance. During the training of the Machine Learning model, the model ranked a few features which were important for a team to win which are,



This gives an insight into the importance of factors. First and second are the opponent team and the team itself. The next most important feature is the city the match is held in, followed by the venue, the third umpire and the toss winner.

## Dashboard:

### 1. Number of seasons

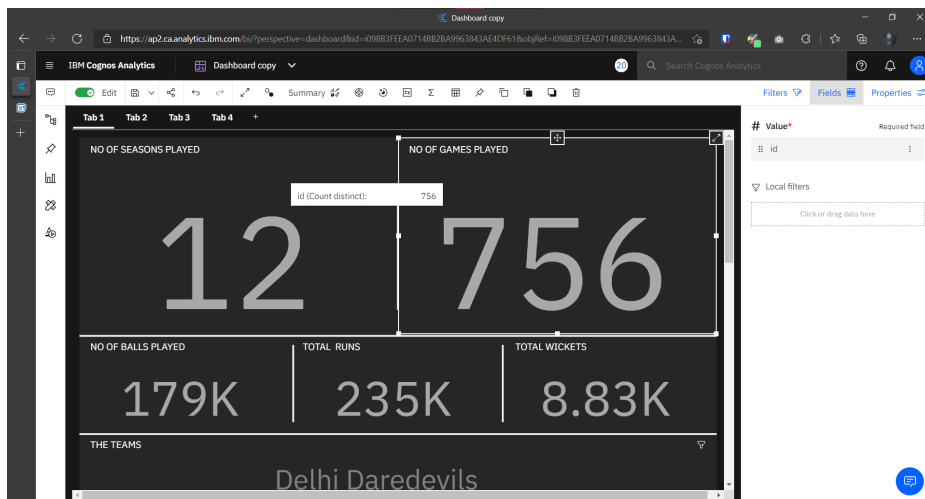


Visualization Used: Summary

Attribute Used: Season from matches.csv (counted distinctly)

Result: 12 seasons

## 2. Number of games played

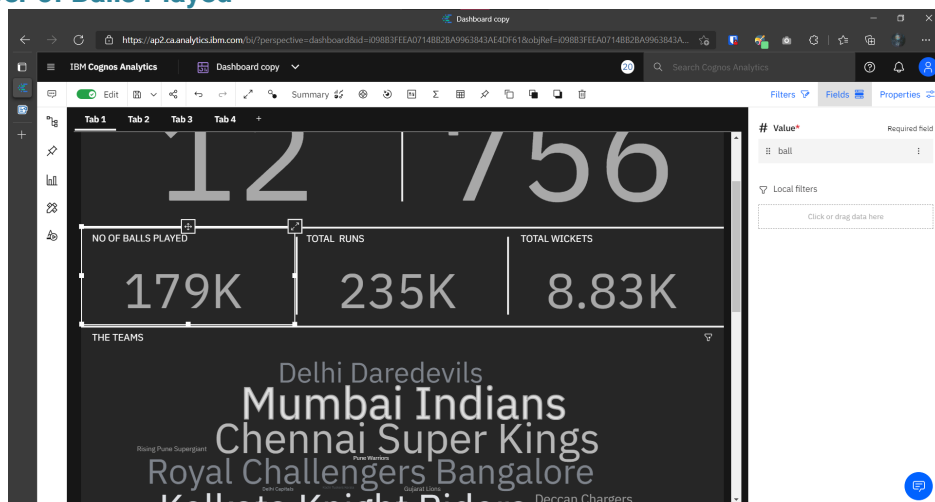


Visualization Used: Summary

Attribute Used: id from matches.csv (counted distinctly)

Result: 756

## 3. Number of Balls Played

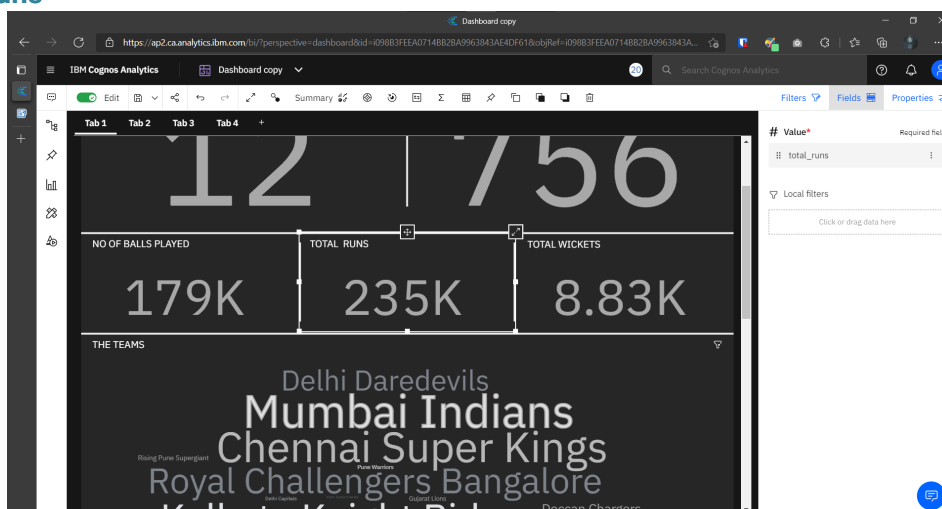


Visualization Used: Summary

Attribute Used: ball from deliveries.csv (count)

Result: 179K

## 4. Total Runs



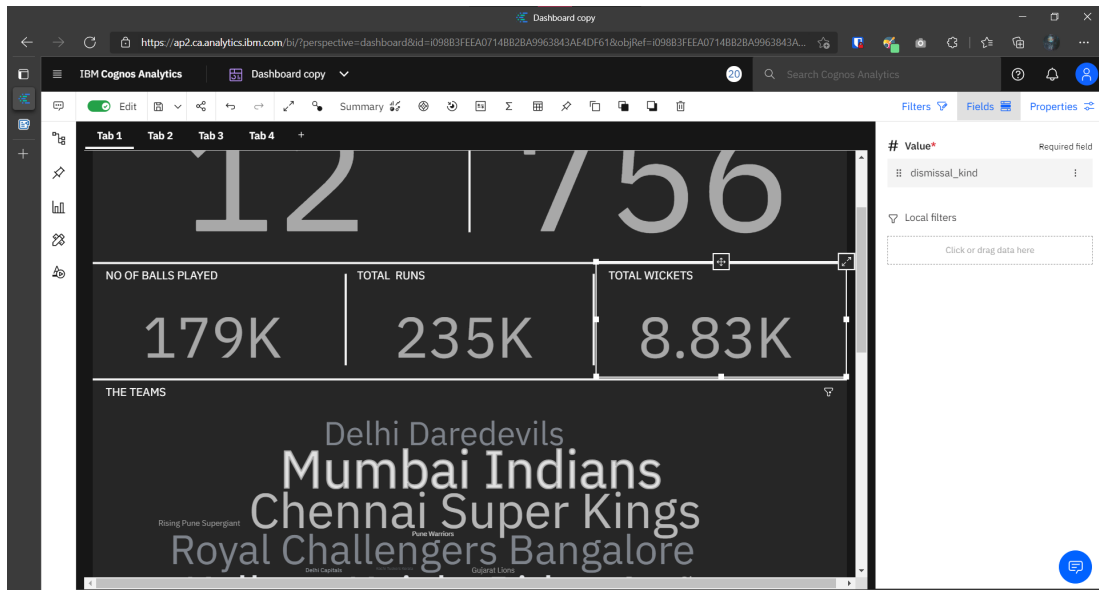
4

Visualization Used: Summary

Attribute Used: total\_runs from deliveries (summed)

Result: 235K

## 5. Total Wickets

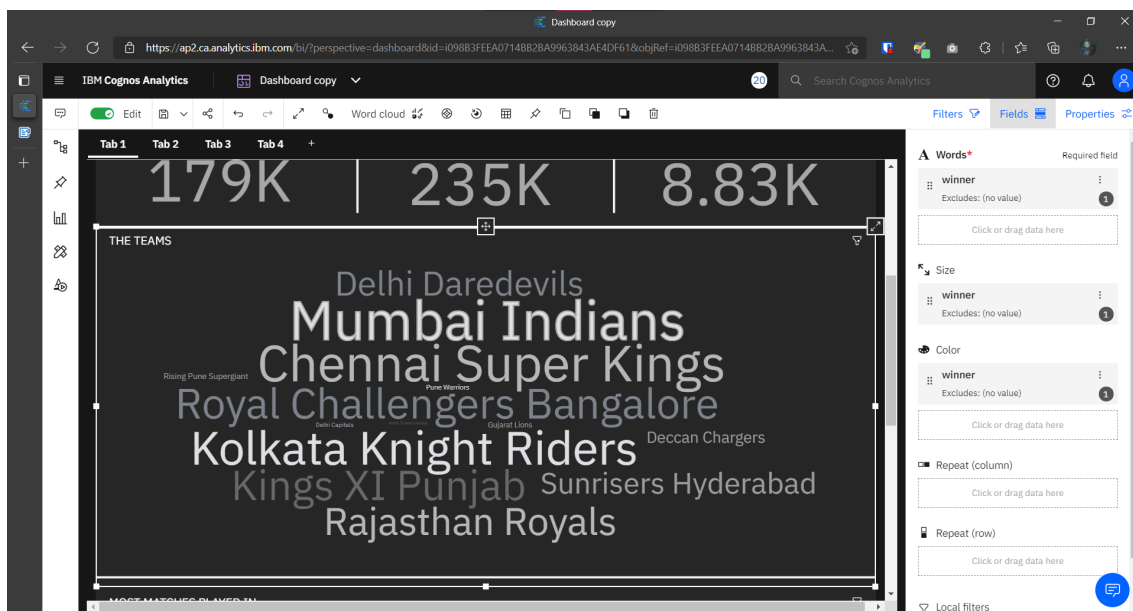


Visualization Used: Summary

Attribute Used: dismissal\_kind from deliveries (count)

Result: 8.83K

## 6. The Teams

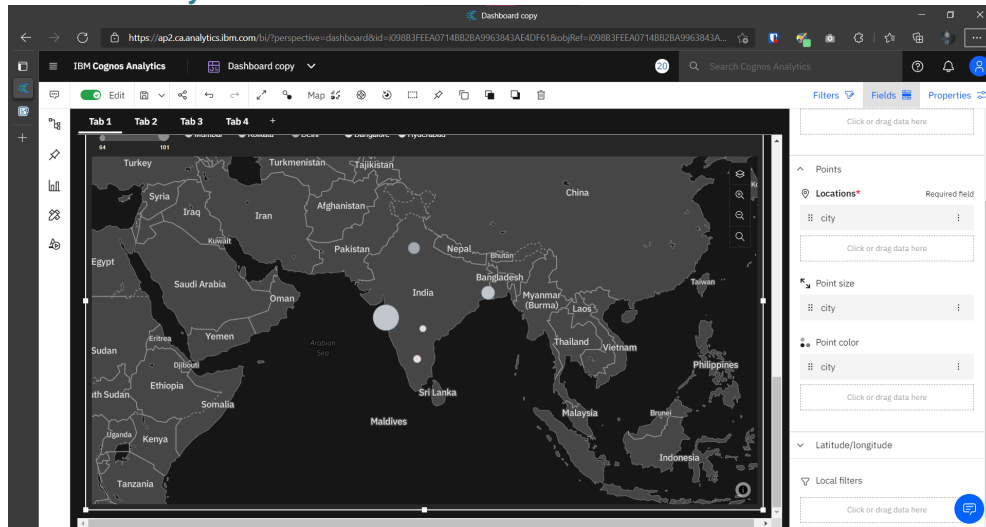


Visualization Used: Word Cloud

Attribute Used: Winner excluding 'no value'

Result: Team Names displayed according to the number of matches won by team in total

## 7. Most Matches Played In

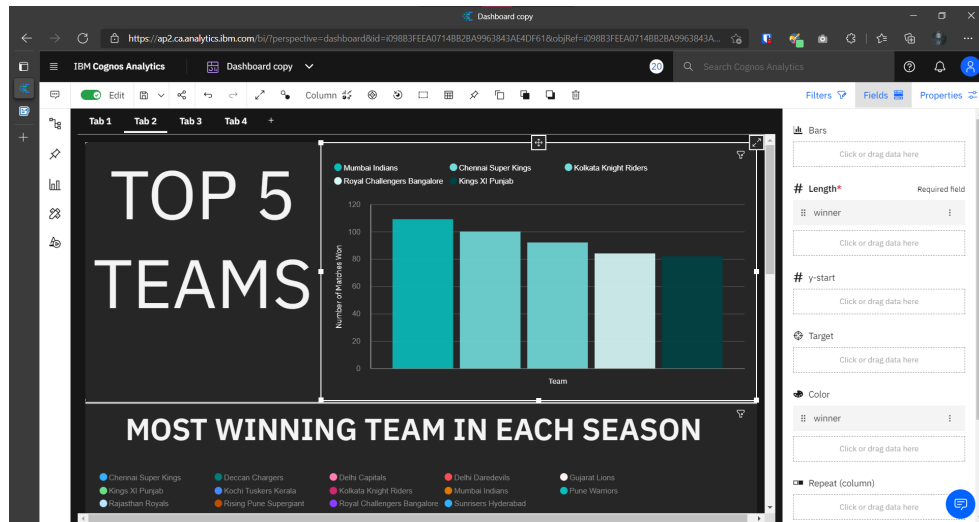


Visualization Used: Map

Attribute Used: city from matches.csv (Top 5)

Result: Mumbai(101) has hosted the most number of IPL matches

## 8. Top 5 Teams

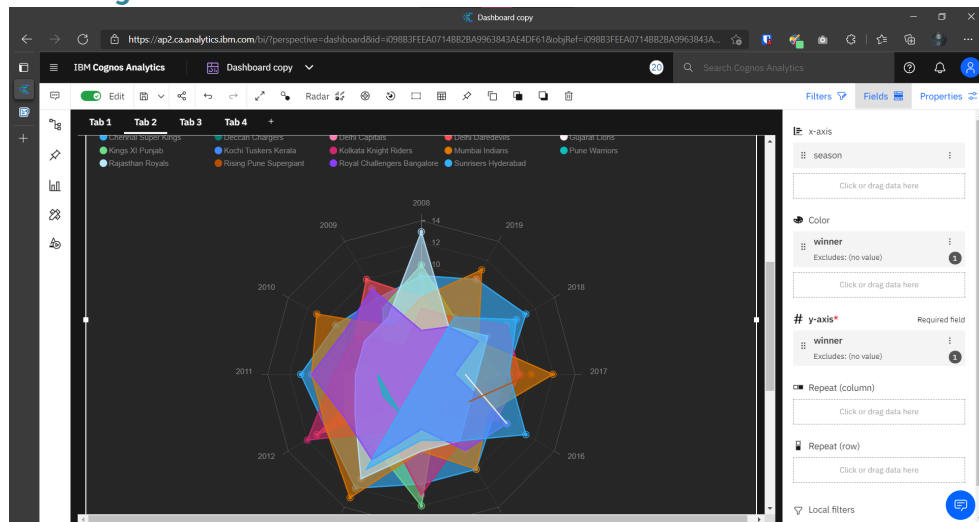


Visualization Used: Column

Attribute Used: Winner from matches.csv (Top 5)

Result: Mumbai Indians (109) have won the most matches

## 9. Most Winning Team in Each Season

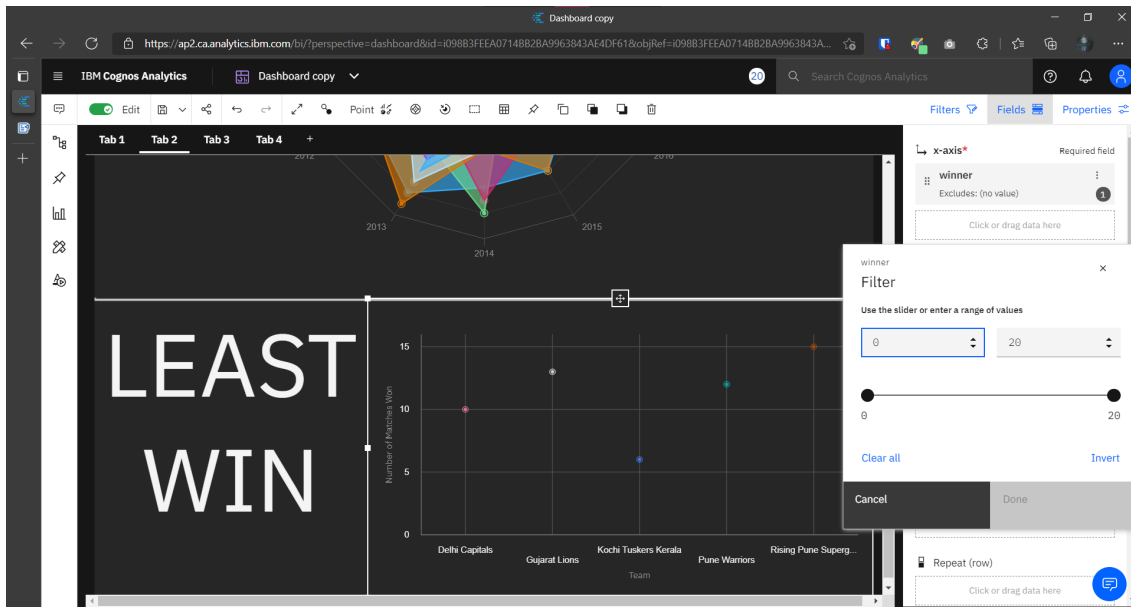


Visualization Used: Radar

Attribute Used: Season (x-axis) and Winner (y-axis)

Result: Number of matches won by every team in each season displayed

## 10. Least Wins

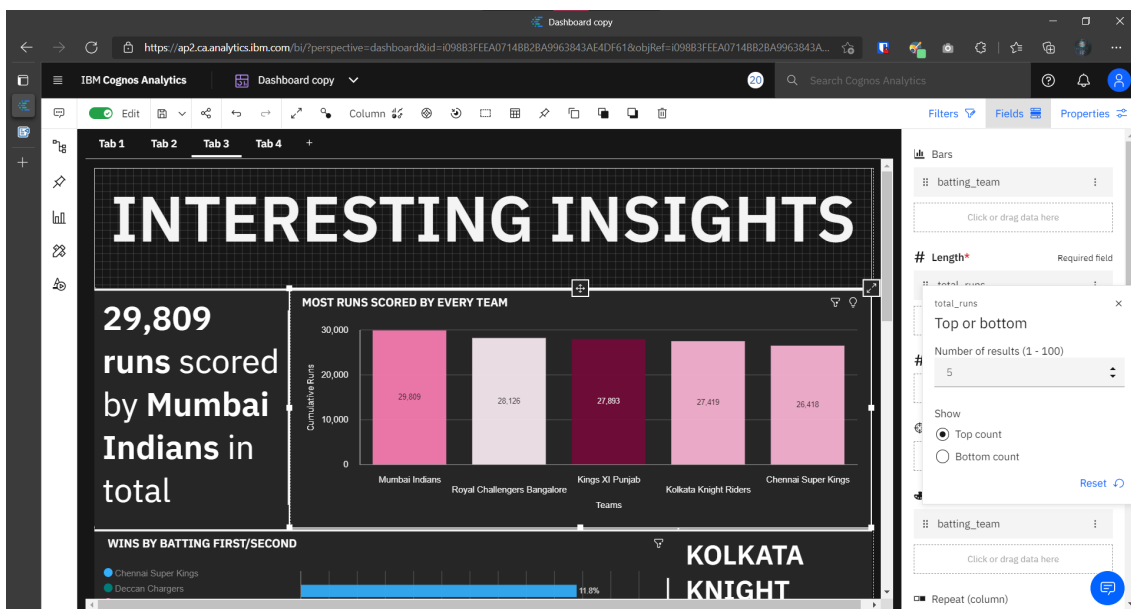


Visualization Used: Point

Attribute Used: Winner excluding 'no value' (Winner count filtered from 0-20)

Result: Kochi Tuskers Kerala have the least wins- 6 Matches won

## 11. Most Scored By Every Team

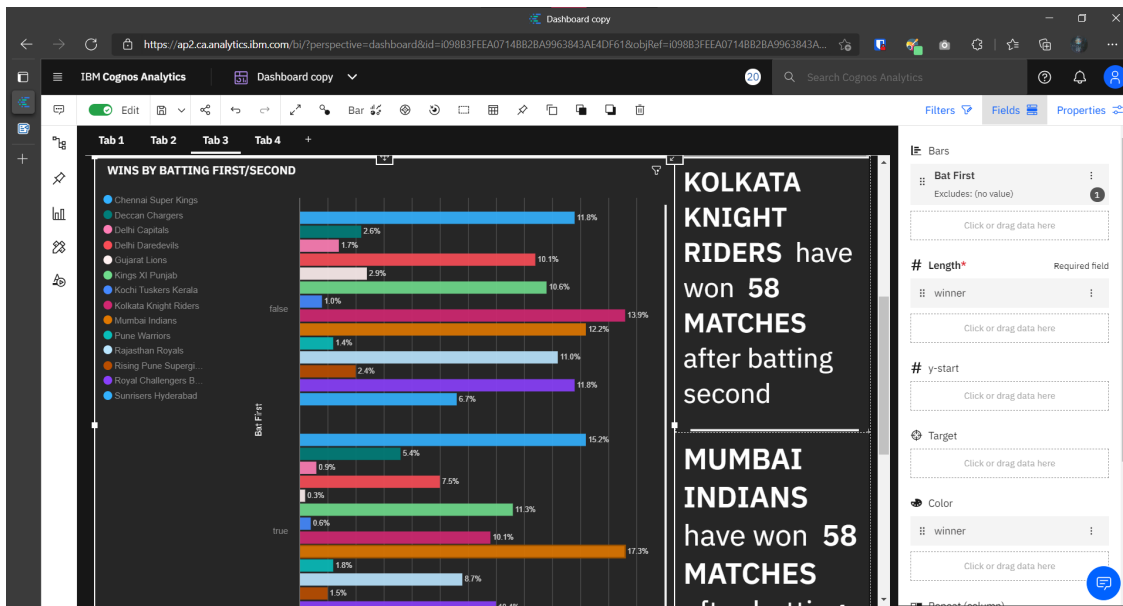


Visualization Used: Column

Attribute Used: Batting team (bars), Total Runs (length) (Top 5 results)

Result: Mumbai Indians have scored a total of 29,809 runs during the entire IPL

## 12. Wins By Batting First/ Second



### Bat First Calculation:

Expression

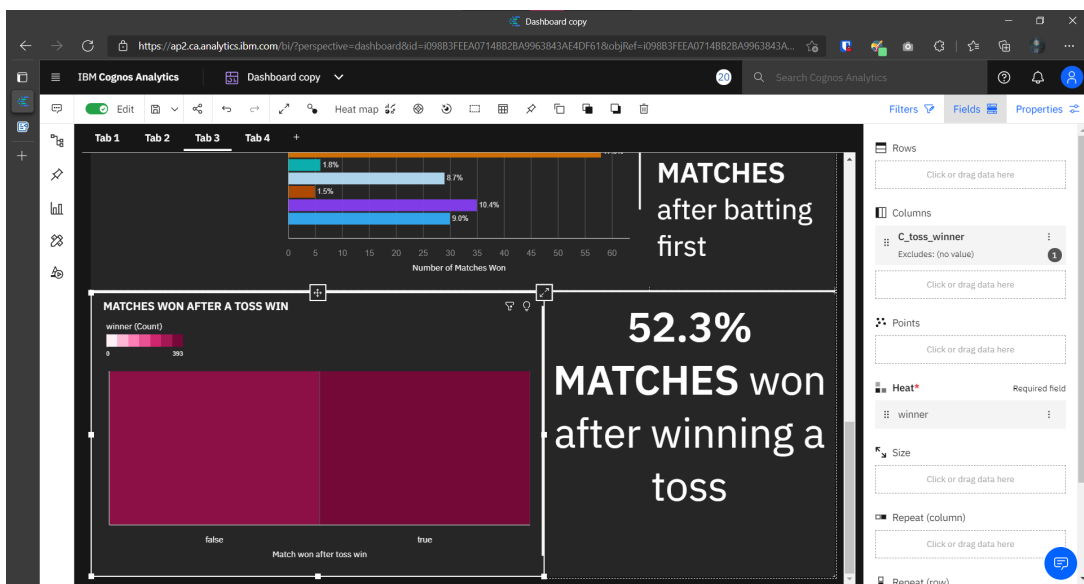
```
1 ((toss_decision like 'bat') and (toss_winner like winner)) or ((toss_decision like 'field') and not(toss_winner like winner))
```

Visualization Used: Bar

Attribute Used: Bat First (calculation), Winner

Result: Mumbai Indians and Kolkata Knight Riders have won maximum matches by batting first and second respectively.

## 13. Matches Won After Toss Win



### C toss\_winner

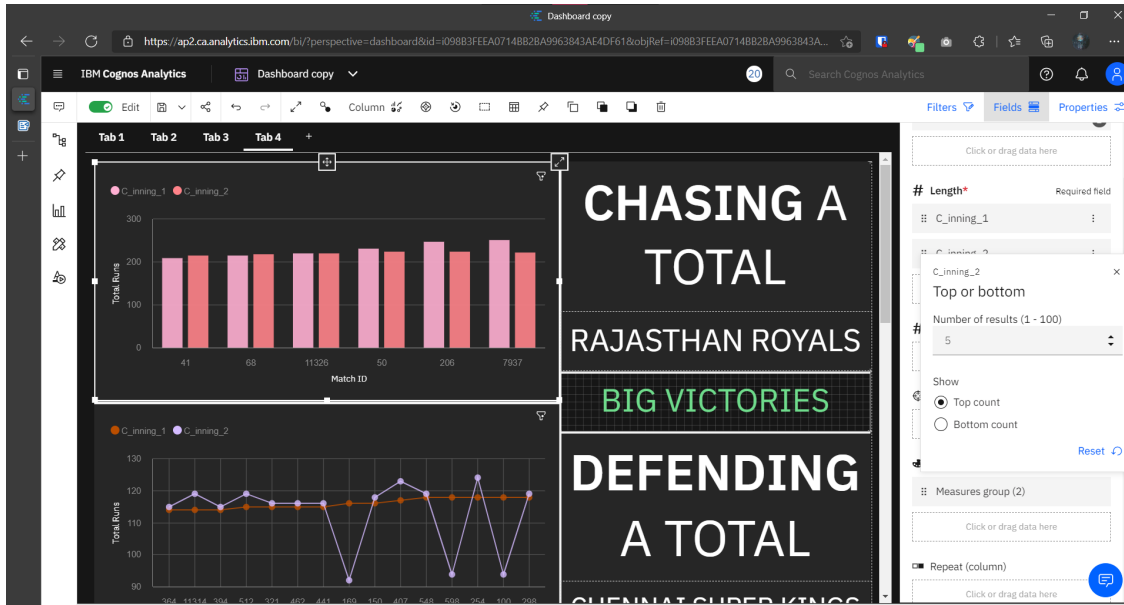
```
matches_csv.toss_winner like matches_csv.winner
```

Visualization Used: Heat Map

Attribute Used: C\_toss\_winner (calculation), Winner

Result: 42.3% matches won after winning a toss

## 14. Biggest Victories – Highest Chase



C\_inning\_1:

```
1 if (inning=1) then (total (total_runs for inning , match_id)) else (0)
```

C\_inning\_2:

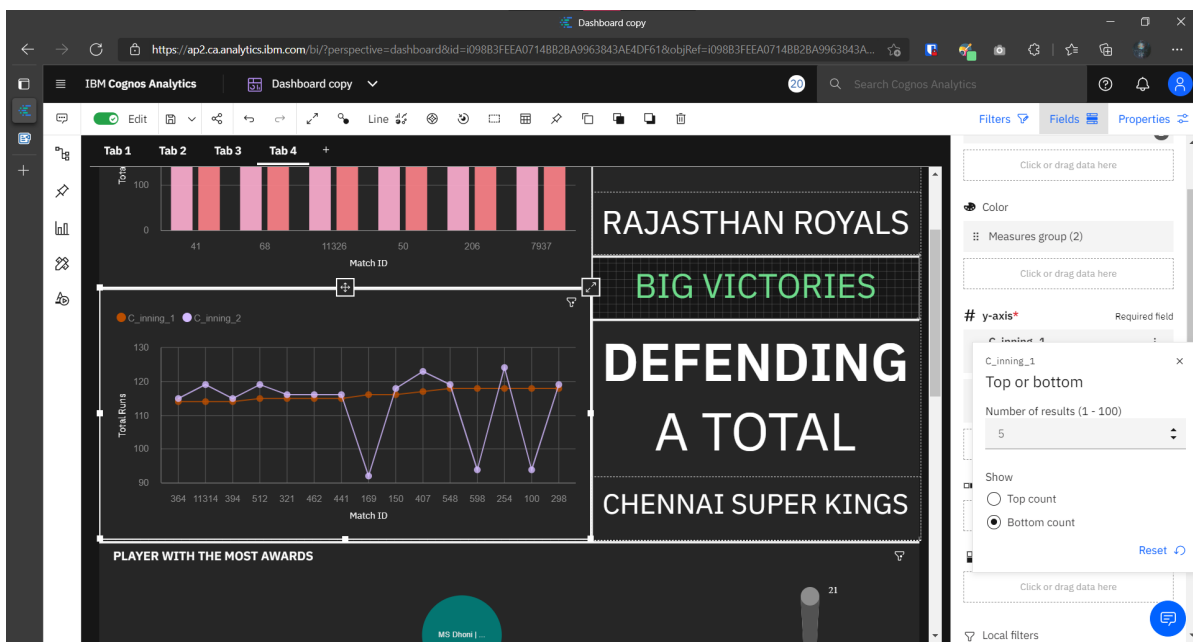
```
if (inning=2) then (total (total_runs for inning , match_id)) else (0)
```

Visualization Used: Column

Attribute Used: C\_inning\_1, C\_inning\_2 (Calculation), Match ID

Result: Match ID 68- Kings XI Punjab (214) Vs Rajasthan Royals (217) is the highest run chase

## 15. Biggest Victories - Lowest Score Defended



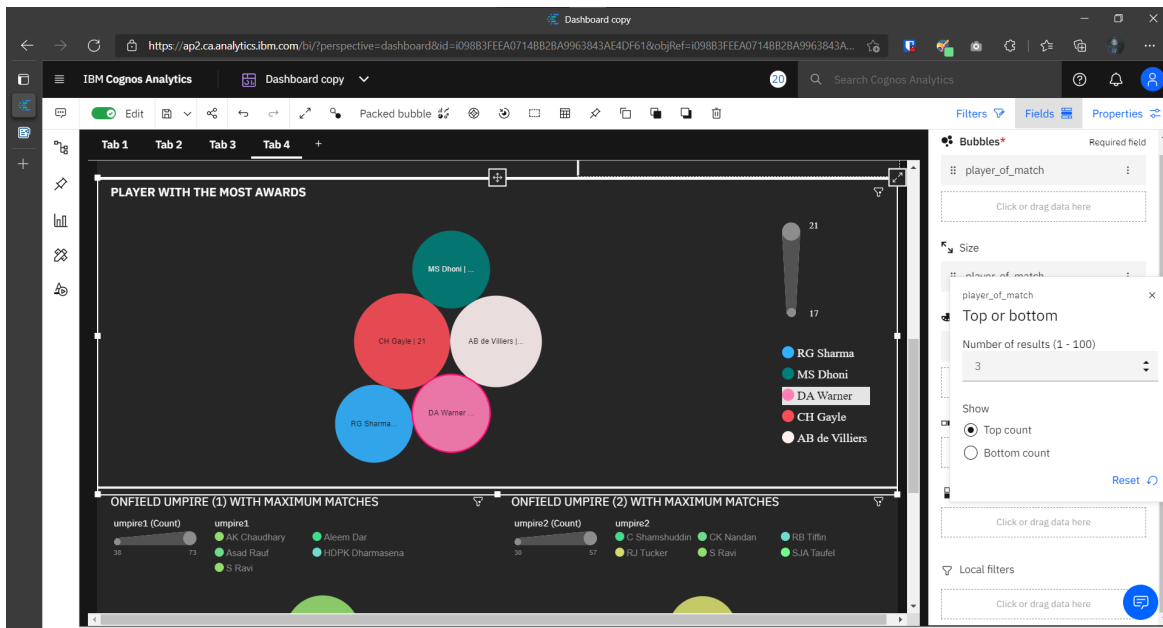
Visualization Used: Column

Attribute Used: C\_inning\_1, C\_inning\_2 (Calculation), Match ID

Result: Match ID 169- Chennai Super Kings (116) vs Kings XI Punjab(92) is the lowest total defend



## 16. Player with Most Awards

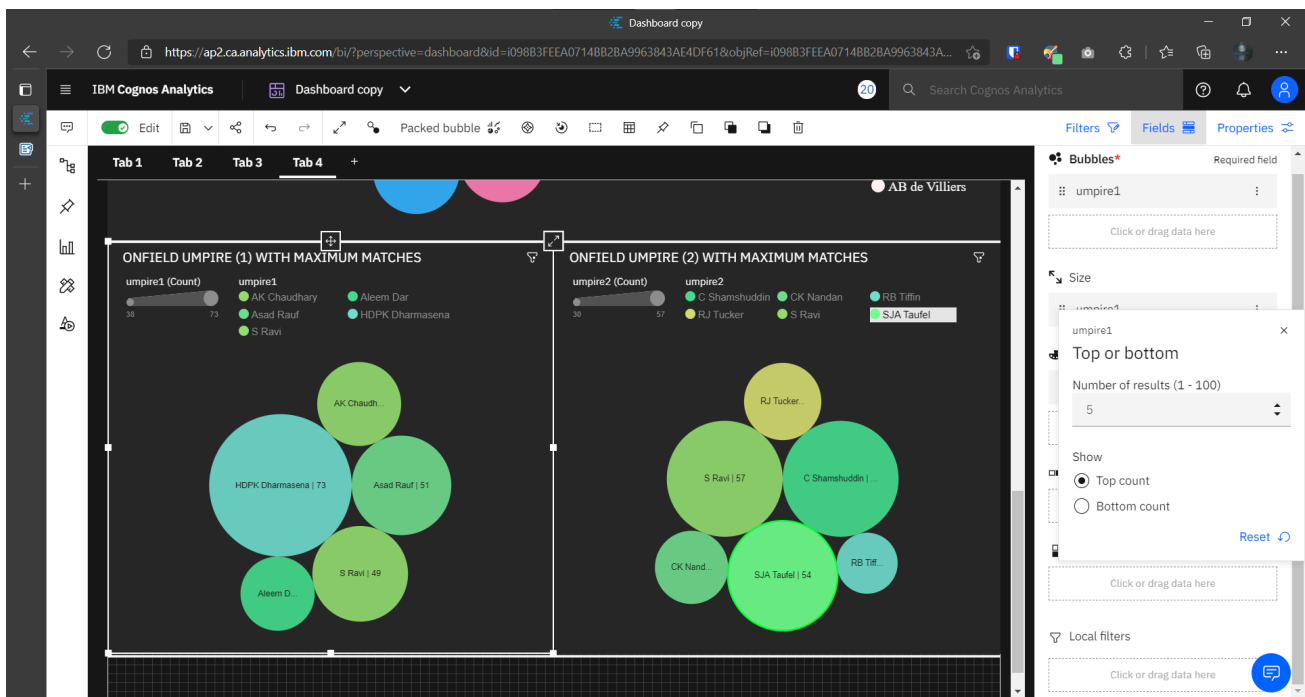


Visualization Used: Packed bubble

Attribute Used: Player of Match from matches.csv (Top 3)

Result: CH Gayle has the maximum Player of the match awards

## 17. On-Field Umpire with Maximum Matches

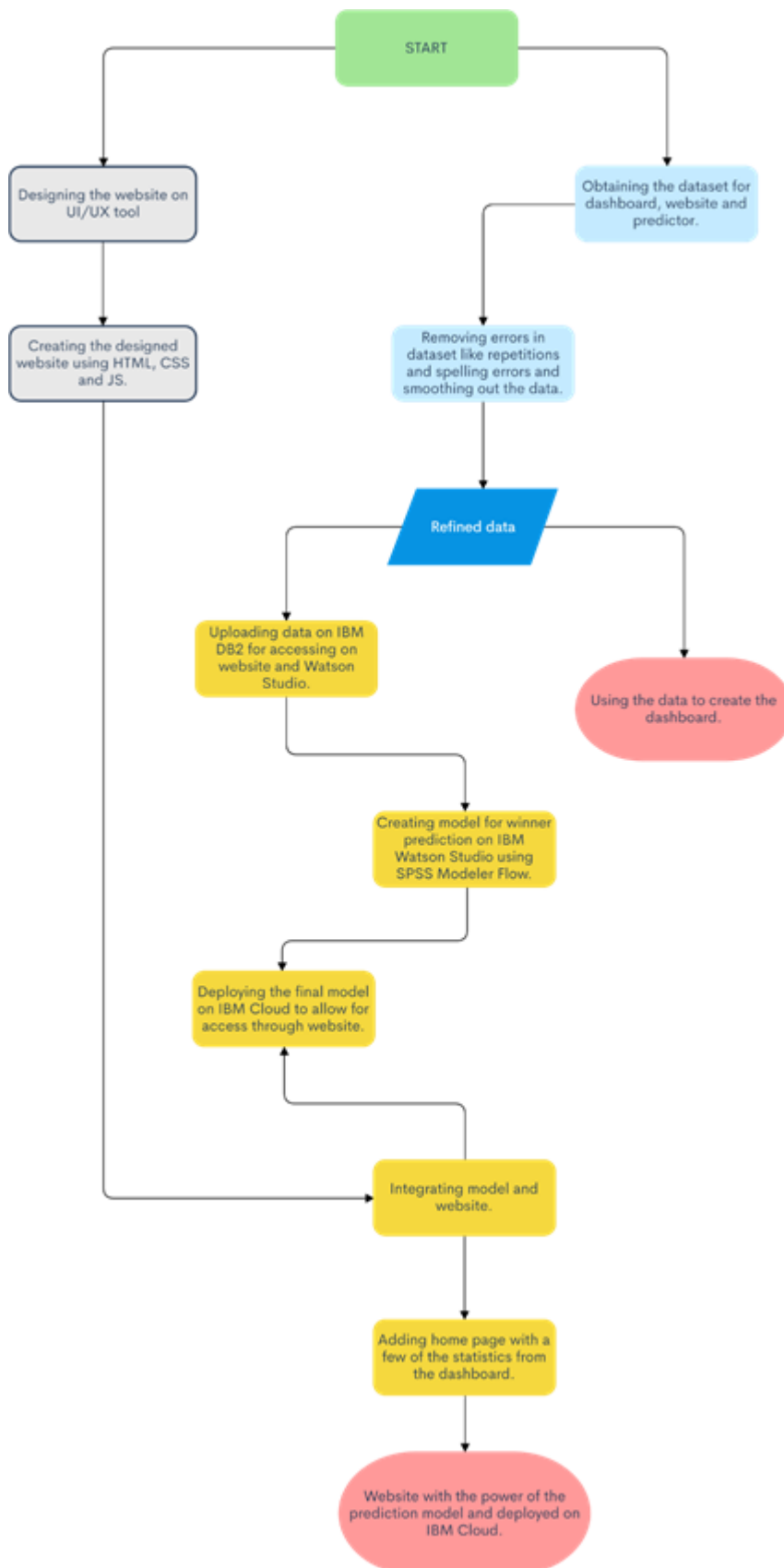


Visualization Used: Packed bubble

Attribute Used: Umpire1, Umpire2 from matches.csv (Top 5)

Result: HDPK Dharmasena (Umpire 1), Ravi S (Umpire 2) and C Shamshuddin (Umpire 2) have maximum matches

## Flowchart



## Result

The Website Home page

## Predictor Page


Win Predictor

Venue	Team 1	Team 2
M. A. Chidambaram Stadium , Chennai ✓	Kolkata Knight Riders ✓	Chennai Super Kings ✓
Include Toss Details?	Toss Winner	Toss Decision
<input checked="" type="radio"/> Yes <input type="radio"/> No	Chennai Super Kings ✓	<input checked="" type="radio"/> Bat <input type="radio"/> Field
Umpire 1	Umpire 2	Umpire 3
A. Jayaprakash ✓	Brj Oxenford ✓	Kumar Dharmasena ✓
<input type="button" value="Submit form"/>		

## Predictor Result

Win Predictor

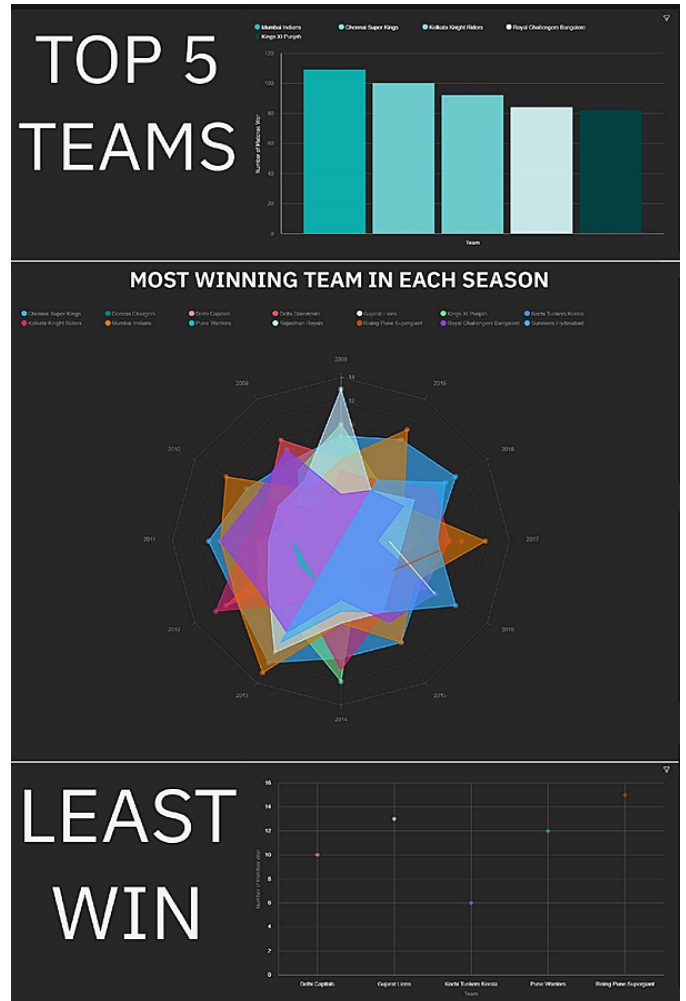
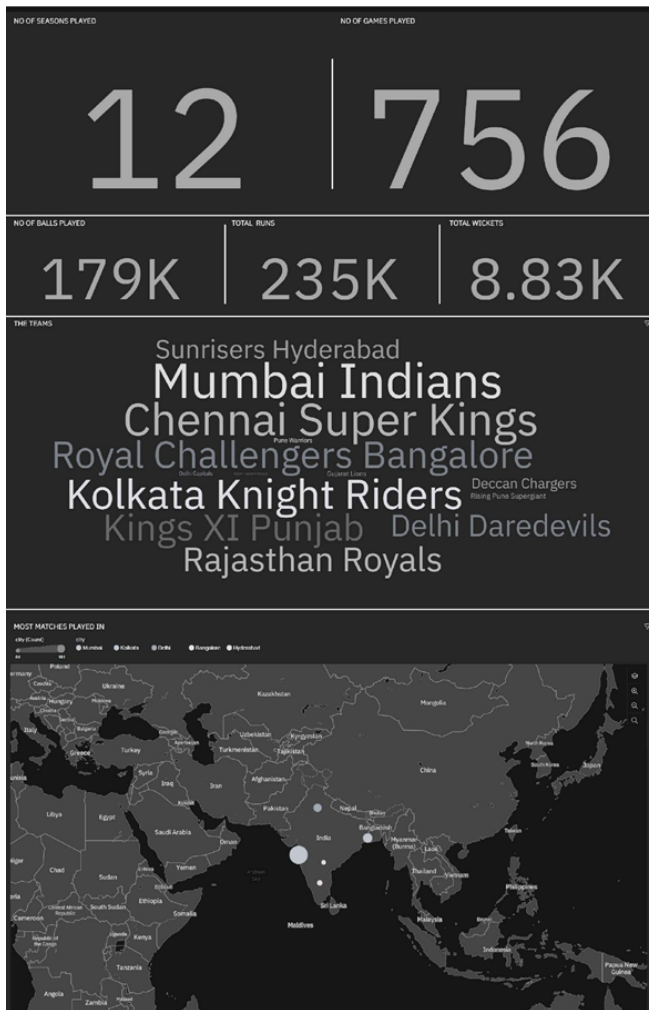
Predicted Winner



Chennai Super Kings

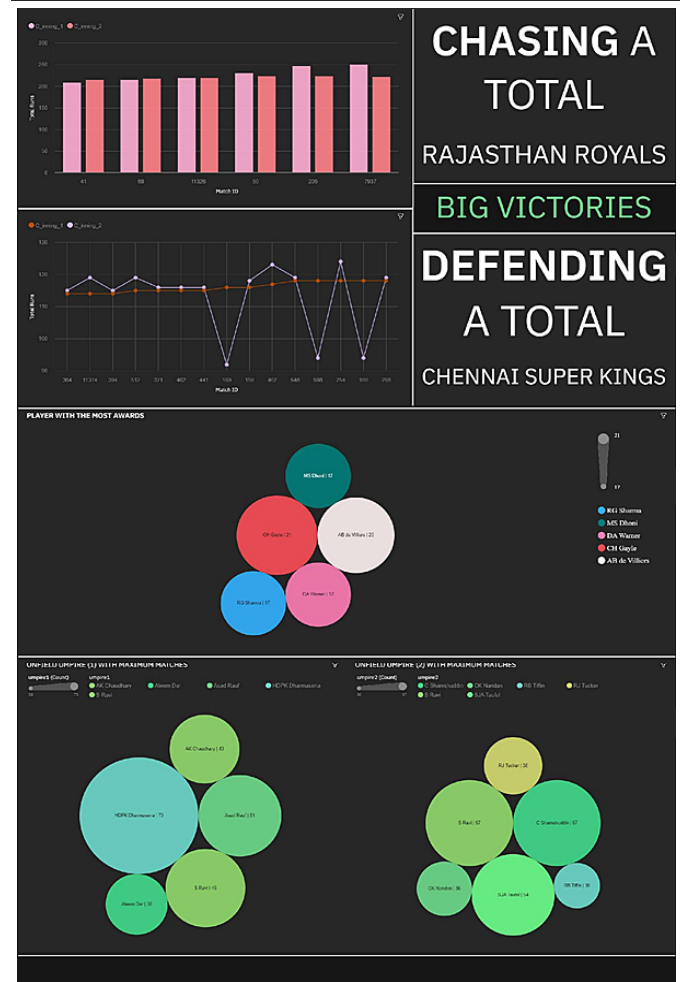
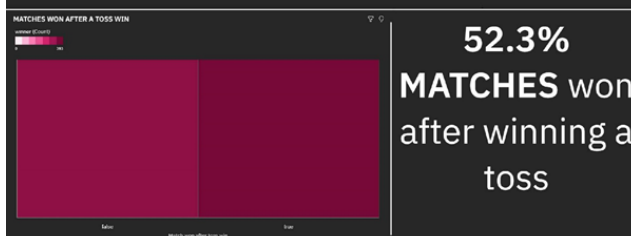
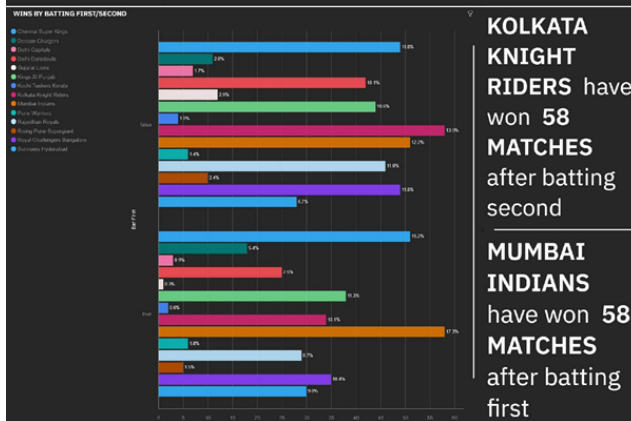
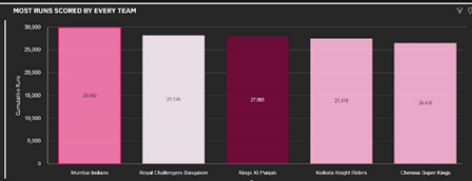
Probability = 0.3545

## Dashboard



## INTERESTING INSIGHTS

**29,809**  
runs scored  
by **Mumbai Indians**  
in total



## Advantages

This project can be used to predict match outcomes and also allows for understanding features in that play an important role in the winning of a team as shown earlier.

This project also gives an insight into the IPL data through the years which allows for drawing conclusions and understanding the teams' progression and their capabilities.

## Disadvantages

The model is not trained on a large amount of data as traditional models are and therefore it is not as accurate. Training the model on a larger dataset would allow for higher accuracy and better prediction.

Another disadvantage is that the parameters that are included are not exhaustive and there can be more parameters that can be included in the model creation and training to allow for more precise predictions.

## Applications

The usage of Machine Analysis helps for understanding the performance of the teams and can also be used to predict future results without the bias which is introduced due to a human intervention. The technology also allows to experiment with various other features and try to understand the result in an attempt to maximize the winning chances.

The machine learning model also helps the investors and sponsors understand the team's performance and their predicted future performance based on the inputs and choose a team to invest in using the predictions.

## Conclusion

To conclude the project brought to surface few important points which are but not limited to :

- Winning the toss is not as significant as was thought to be.
- The umpires are more important as a factor than winning the toss which might hint at the presence of human bias during decisions.
- The city and venue were important factors and this confirms the home team advantage that is usually believed.

## Future Scope

Further enhancements that can be made to the current project include

- Training the model on a larger dataset.
- Including more factors for the model which allows to include more factors for the prediction which would enhance the accuracy and precision of the model in predicting the results of the match.
- Exploring other model types which can allow for greater accuracy and precision.

## Bibliography

Works referenced and used in the project:

- Logos - [IPL Logo PNG Download All Team – Vivo IPL 2021 Best Png \(kreditings.com\)](#)
- Website building references - [GeeksforGeeks](#) , [Stack Overflow](#)
- Background Image - [Dribbble](#)
- Loading animation - [Code My UI](#)

## Appendix

### Sample source code for API call of online hosted prediction model

```

1  import requests
2
3  def call(data):
4      tosswinner=data['tosswinner']
5      tossdec=data['tossdec']
6      if(data['tossch']!='true'):
7          tosswinner=None
8          tossdec=None
9      API_KEY = {{ API_KEY }}
10     token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
11     mltoken = token_response.json()["access_token"]
12     temp=data['venue'][0]
13     temp=list(map(str.strip, temp.split(',')))
14     city=temp[1]
15     venue=temp[0]
16     header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
17
18     payload_scoring = {
19         "input_data": [
20             {
21                 "fields": [
22                     "id",
23                     "season",
24                     "city",
25                     "date",
26                     "team1",
27                     "team2",
28                     "toss_winner",
29                     "toss_decision",
30                     "result",
31                     "dl_applied",
32                     "winner",
33                     "win_by_runs",
34                     "win_by_wickets",
35                     "player_of_match",
36                     "venue",
37                     "umpire1",
38                     "umpire2",
39                     "umpire3",
40                     "winner_id",
41                     "team1_id",
42                     "team2_id",
43                     "toss_winner_id"
44                 ],
45                 "values": [

```

```

46         [
47             1,
48             None,
49             city,
50             None,
51             data['team1'][0],
52             data['team2'][0],
53             data['tosswinner'][0],
54             data['tossdec'][0],
55             None,
56             None,
57             None,
58             None,
59             None,
60             None,
61             venue,
62             data['umpire1'][0],
63             data['umpire2'][0],
64             data['umpire3'][0],
65             None,
66             None,
67             None,
68             None,
69             None
70         ]
71     ]
72 }
73 ]
74 }
75 response_scoring =
requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/3403e0f6-b50b-469a-b82d-9853e5
a5d934/predictions?version=2021-08-28', json=payload_scoring, headers={'Authorization':
'Bearer ' + mltoken})
76 response_scoring=response_scoring.json()
77 winner=response_scoring['predictions'][0]
78 winner=winner['values']
79 win=winner[0]
80 return win

```