Indian Premier League Analysis

Group 4

Urvashi Dube

Sudhish Subramaniam

dube.u@northeastern.edu

subramaniam.su@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Urvashi Dube

Signature of Student 2:

Submission Date: April 22, 2023

Project presentation: April 18

Table of Contents (Index)

1 Introduction

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India, featuring franchise teams representing different cities. The league has become one of the most popular and lucrative cricket leagues in the world, attracting top international players and generating significant revenue through sponsorships, broadcasting rights, and merchandise sales.

Our project will help sports broadcasting networks to increase their viewership and engagement during the upcoming Indian Premier League season. Leveraging the provided IPL database our project will help them to develop new and exciting content for their viewers, such as pre-game and post-game analysis, player interviews, and statistical breakdowns. The network will be able to identify key storylines and narratives that will resonate with their audience, such as underdog teams or breakout players. They will also identify popular match locations and stadiums, as well as trends in viewer engagement, such as the most popular social media platforms and hashtags. The goal is to provide viewers with compelling content that will keep them engaged throughout the season and drive-up viewership and ad revenue for the network. This will surely help broadcasting networks and will also help viewers to enjoy interesting and captivating content.

2 Literature review

The Indian Premier League (IPL) is a highly popular and competitive cricket tournament. In recent years, there has been a growing interest in using data analytics and machine learning techniques to analyze and predict the outcomes of IPL matches. This review presents a summary of several studies that have been conducted in this area.

One study by Barot et al. (2020) analyzed IPL data from previous seasons to identify key factors that contribute to team performance and success. The authors used data visualization and machine learning techniques to predict the outcomes of matches and achieved an accuracy rate of 70% [1].Viswanadha et al. (2015) developed a dynamic winner prediction model for Twenty20 cricket, including the IPL. Their model is based on the relative strengths of the competing teams and includes several features such as team composition and player performance. The authors achieved a prediction accuracy of 66% on IPL data [2]. Bandulasiri (2019) used data from One Day International (ODI) cricket matches to develop a model for predicting the winner of IPL matches. The author used a range of statistical and machine learning techniques and achieved an accuracy rate of 85% [3]. In another study, Priyanka et al. (2021) used data mining algorithms to predict the outcomes of IPL matches. The authors analyzed several features such as the playing conditions and team composition to achieve an accuracy rate of 80% [4]. Agrawal et al. (2019) used machine learning techniques to predict the results of IPL T20 matches. The authors used several features such as team composition and player statistics to achieve an accuracy rate of 66% [5]. Kaluarachchi and Varde (2014) developed a tool called CricAI, which uses classification-based techniques to predict the outcome of ODI cricket matches. The authors achieved an accuracy rate of 70% on IPL data [6]. Finally, Jhanwar and Pudi (2013) used a team composition-based approach to predict the outcomes of ODI cricket matches, including the IPL. The authors achieved an accuracy rate of 73% on IPL data [7].

In conclusion, these studies demonstrate the potential of data analytics and machine learning techniques in analyzing and predicting the outcomes of IPL matches. The use of features such as team composition, player performance, and playing conditions can significantly improve prediction accuracy. Further research in this area may lead to the development of more accurate and reliable prediction models for the IPL and other cricket tournaments.

3 Objective

The project involves a range of analytical tasks related to the Indian Premier League (IPL) cricket tournament. These tasks include determining the number of players in the IPL by country, calculating the number of matches played in each stadium and city, and analyzing how toss decisions have changed over the seasons. The project also involves identifying the Man of the Series, Purple Cap, and Orange Cap winners for each season, calculating the total runs scored in all matches, and creating a cross-tabulation of overs and types of dismissals.

Additionally, the project aims to determine the total number of match wins for each team, identify the top 10 Man of the Match award winners, and determine the top 10 highest run scorers. Analysis of the distribution of batsmen's total runs and total innings, creating a scatter plot of total runs and innings, and analyzing the frequency distribution of different types of dismissals are also part of the project.

Moreover, the project involves determining the frequency of different bowlers, identifying the top 10 players who got out the maximum number of times, and determining the top 10 wicket takers. The variation in total extras in a match over time will also be analysed. Finally, the project aims to develop a robust model to predict the winning team and the win margin in a match. By completing these tasks, the project aims to provide insights and analysis on the IPL tournament and its players, teams, and matches.

4 Overview

The dataset consists of 21 Excel files with a total of 85 columns, containing information about the Indian Premier League (IPL) seasons from 2008 to 2016. The dataset contains data about 577 matches, and describes 469 players who participated in the IPL during this time period.

The files contain a wide range of information about the matches, including details such as the venue, city, toss winner, toss decision, and the outcome of the match. In addition, the dataset contains information about the players, such as their name, age, batting and bowling styles, and the team they played for. This dataset can be used to perform a wide range of analysis on IPL matches and players, including identifying trends in performance over time, comparing performance between teams and players, and predicting outcomes of future matches based on historical data. A detailed analysis of number of columns and rows in the data are shown in table 1.

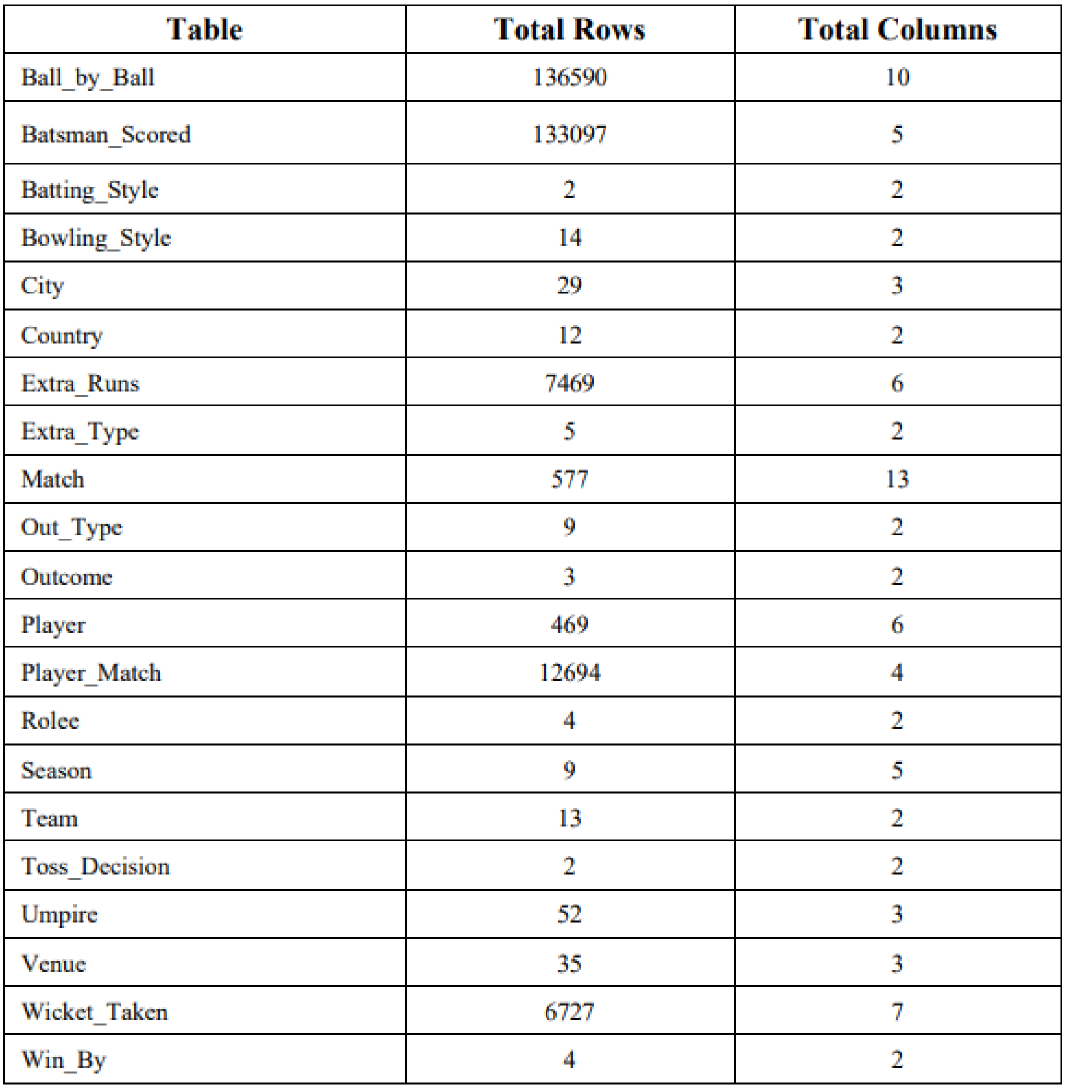


Table	Total Rows	Total Columns
Ball_by_Ball	136590	10
Batsman_Scored	133097	5
Batting_Style	2	2
Bowling_Style	14	2
City	29	3
Country	12	2
Extra_Runs	7469	6
Extra_Type	5	2
Match	577	13
Out_Type	9	2
Outcome	3	2
Player	469	6
Player_Match	12694	4
Rolee	4	2
Season	9	5
Team	13	2
Toss_Decision	2	2
Umpire	52	3
Venue	35	3
Wicket_Taken	6727	7
Win_By	4	2

Table 1. Number of rows and columns in the dataset

5 Relational ERD Model

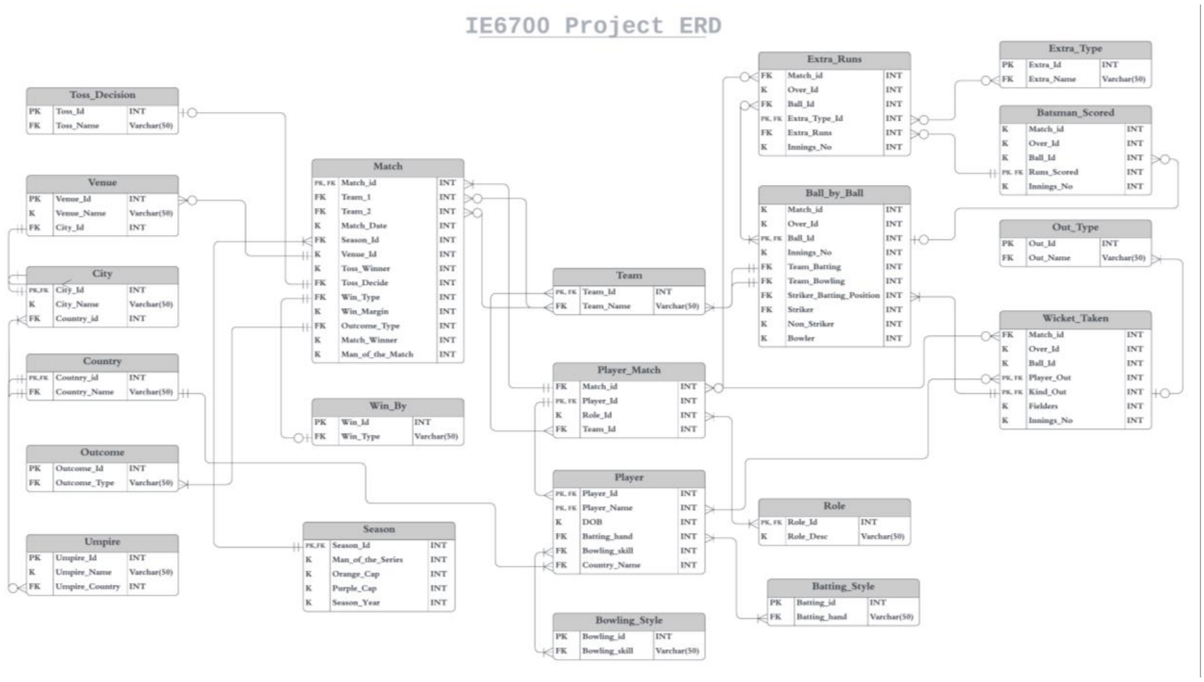


Fig 1 Relational ERD Model

An Entity-Relationship Diagram (ERD) is a graphical representation of entities and their relationships to each other. It is commonly used in database design to illustrate how entities relate to each other and how they can be organized in a database.

In the context of the IPL dataset, an ERD model can be created to visually represent the relationships between the various tables in the database. The ERD model can be used to identify the relationships between different entities in the database and to ensure that the database is structured in a way that is efficient, easy to use, and consistent. It can also be used to identify any potential issues or inconsistencies in the data, such as missing data or duplicate entries. Overall, an ERD model provides a valuable tool for designing and managing databases, particularly in complex datasets such as the IPL dataset.

Some relationships from the current ERD model are:

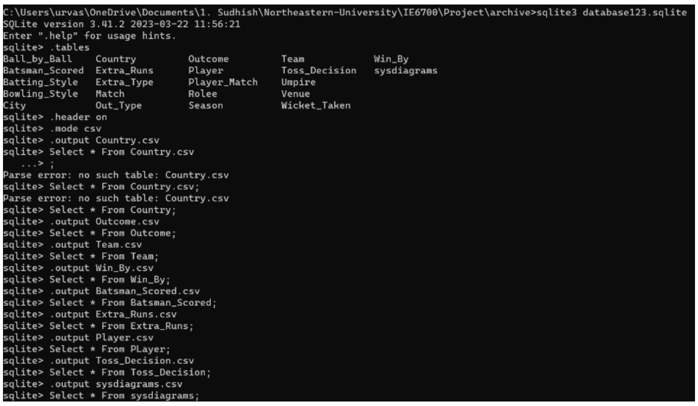
- Match has two teams, Team1 and Team2, which are associated with the Team entity.

- Match has one winner, which is associated with the Team entity.
- Match has one venue, which is associated with the Venue entity.
- Match has one toss winner, which is associated with the Team entity.
- Match has one toss decision, which is associated with the Toss_Decision entity.
- Match has one outcome type, which is associated with the Outcome entity.
- Player can have multiple roles in a match, which is associated with the Role entity.
- Ball_by_Ball is associated with the Match entity through the Match_ID attribute.
- Ball_by_Ball is associated with the Player entity through the Striker_ID, Non_Striker_ID, Bowler_ID, and Fielder_ID attributes.
- Ball_by_Ball is associated with the Extra_Type entity through the Extra_Type attribute.
- Ball_by_Ball is associated with the Out_Type entity through the Dismissal_Type attribute.

One-one, many to one and many to one relationship between database tables can be seen in the ERD model.

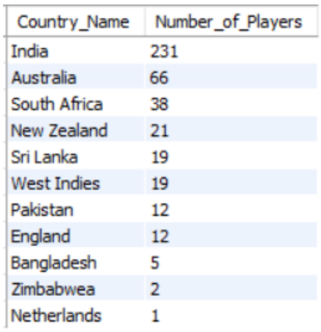
6 SQL Analysis and Normalization

The input SQLite Database files are converted into CSV files for further processing as shown in Fig. 2.



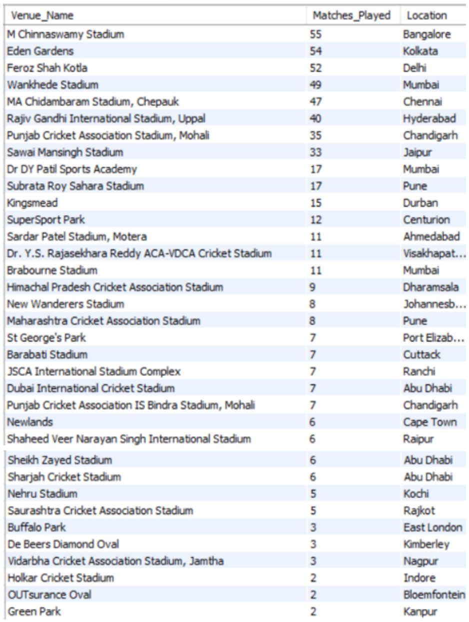
```
C:\Users\urvas\OneDrive\Documents\1. Sudhish\Northeastern-University\IE6700\Project\archive>sqlite3 database123.sqlite
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .tables
Ball_by_Ball    Country         Outcome         Team            Win_By
Batsman_Scored  Extra_Runs      Player          Toss_Decision   sysdiagrams
Batting_Style   Extra_Type      Player_Match    Umpire
Bowling_Style   Match           Rolee           Venue
City            Out_Type        Season          Wicket_Taken
sqlite> .header on
sqlite> .mode csv
sqlite> .output Country.csv
sqlite> Select * From Country.csv
   ...> ;
Parse error: no such table: Country.csv
sqlite> Select * From Country.csv;
Parse error: no such table: Country.csv
sqlite> Select * From Country;
sqlite> .output Outcome.csv
sqlite> Select * From Outcome;
sqlite> .output Team.csv
sqlite> Select * From Team;
sqlite> .output Win_By.csv
sqlite> Select * From Win_By;
sqlite> .output Batsman_Scored.csv
sqlite> Select * From Batsman_Scored;
sqlite> .output Extra_Runs.csv
sqlite> Select * From Extra_Runs;
sqlite> .output Player.csv
sqlite> Select * From PLayer;
sqlite> .output Toss_Decision.csv
sqlite> Select * From Toss_Decision;
sqlite> .output sysdiagrams.csv
sqlite> Select * From sysdiagrams;
```

Fig. 2 Conversion of SQLite Files into CSV Files



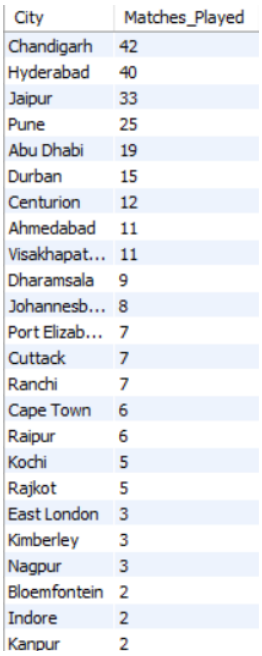
Country_Name	Number_of_Players
India	231
Australia	66
South Africa	38
New Zealand	21
Sri Lanka	19
West Indies	19
Pakistan	12
England	12
Bangladesh	5
Zimbabwea	2
Netherlands	1

Table 2 No of players in the IPL country wise



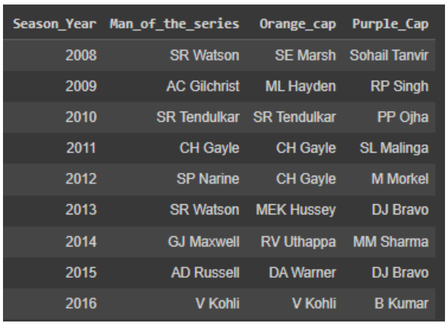
Venue_Name	Matches_Played	Location
M Chinnaswamy Stadium	55	Bangalore
Eden Gardens	54	Kolkata
Feroz Shah Kotla	52	Delhi
Wankhede Stadium	49	Mumbai
MA Chidambaram Stadium, Chepauk	47	Chennai
Rajiv Gandhi International Stadium, Uppal	40	Hyderabad
Punjab Cricket Association Stadium, Mohali	35	Chandigarh
Sawai Mansingh Stadium	33	Jaipur
Dr DY Patil Sports Academy	17	Mumbai
Subrata Roy Sahara Stadium	17	Pune
Kingsmead	15	Durban
SuperSport Park	12	Centurion
Sardar Patel Stadium, Motera	11	Ahmedabad
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium	11	Visakhapat...
Brabourne Stadium	11	Mumbai
Himachal Pradesh Cricket Association Stadium	9	Dharamsala
New Wanderers Stadium	8	Johannesb...
Maharashtra Cricket Association Stadium	8	Pune
St George's Park	7	Port Elizab...
Barabati Stadium	7	Cuttack
JSCA International Stadium Complex	7	Ranchi
Dubai International Cricket Stadium	7	Abu Dhabi
Punjab Cricket Association IS Bindra Stadium, Mohali	7	Chandigarh
Newlands	6	Cape Town
Shaheed Veer Narayan Singh International Stadium	6	Raipur
Sheikh Zayed Stadium	6	Abu Dhabi
Sharjah Cricket Stadium	6	Abu Dhabi
Nehru Stadium	5	Kochi
Saurashtra Cricket Association Stadium	5	Rajkot
Buffalo Park	3	East London
De Beers Diamond Oval	3	Kimberley
Vidarbha Cricket Association Stadium, Jamtha	3	Nagpur
Holkar Cricket Stadium	2	Indore
OUTsurance Oval	2	Bloemfontein
Green Park	2	Kanpur

Table 3 Number of matches played in each stadium



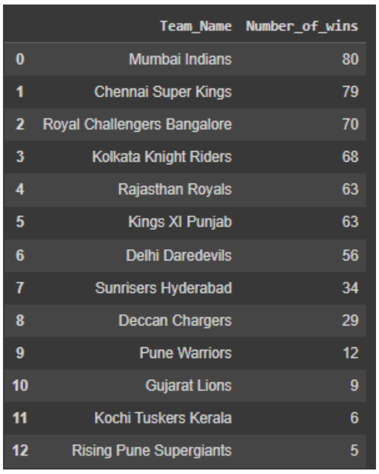
City	Matches_Played
Chandigarh	42
Hyderabad	40
Jaipur	33
Pune	25
Abu Dhabi	19
Durban	15
Centurion	12
Ahmedabad	11
Visakhapat...	11
Dharamsala	9
Johannesb...	8
Port Elizab...	7
Cuttack	7
Ranchi	7
Cape Town	6
Raipur	6
Kochi	5
Rajkot	5
East London	3
Kimberley	3
Nagpur	3
Bloemfontein	2
Indore	2
Kanpur	2

Table 4 Number of matches played in each city



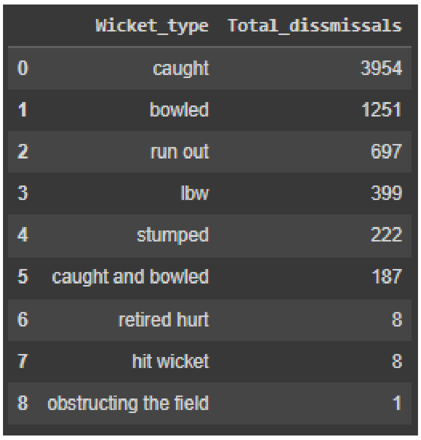
Season_Year	Man_of_the_series	Orange_cap	Purple_Cap
2008	SR Watson	SE Marsh	Sohail Tanvir
2009	AC Gilchrist	ML Hayden	RP Singh
2010	SR Tendulkar	SR Tendulkar	PP Ojha
2011	CH Gayle	CH Gayle	SL Malinga
2012	SP Narine	CH Gayle	M Morkel
2013	SR Watson	MEK Hussey	DJ Bravo
2014	GJ Maxwell	RV Uthappa	MM Sharma
2015	AD Russell	DA Warner	DJ Bravo
2016	V Kohli	V Kohli	B Kumar

Table 5 Man of the Series, Purple Cap, Orange Cap winners in every season



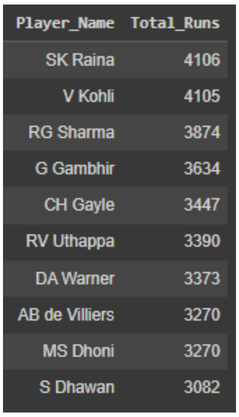
	Team_Name	Number_of_wins
0	Mumbai Indians	80
1	Chennai Super Kings	79
2	Royal Challengers Bangalore	70
3	Kolkata Knight Riders	68
4	Rajasthan Royals	63
5	Kings XI Punjab	63
6	Delhi Daredevils	56
7	Sunrisers Hyderabad	34
8	Deccan Chargers	29
9	Pune Warriors	12
10	Gujarat Lions	9
11	Kochi Tuskers Kerala	6
12	Rising Pune Supergiants	5

Table 5 Total matches won by each team



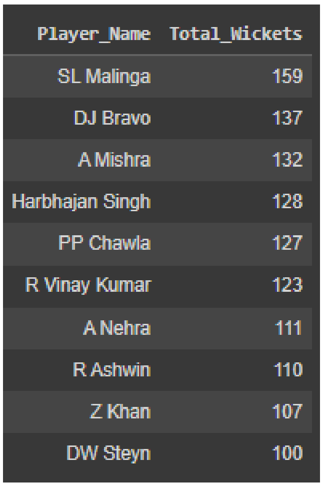
	Wicket_type	Total_dissmissals
0	caught	3954
1	bowled	1251
2	run out	697
3	lbw	399
4	stumped	222
5	caught and bowled	187
6	retired hurt	8
7	hit wicket	8
8	obstructing the field	1

Table 6 Wicket Types



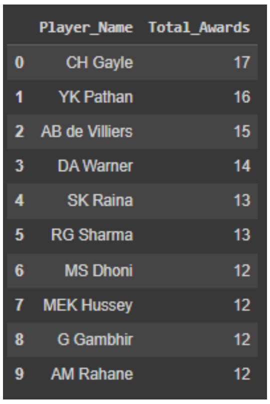
Player_Name	Total_Runs
SK Raina	4106
V Kohli	4105
RG Sharma	3874
G Gambhir	3634
CH Gayle	3447
RV Uthappa	3390
DA Warner	3373
AB de Villiers	3270
MS Dhoni	3270
S Dhawan	3082

Table 7 Top 10 Run Scorers



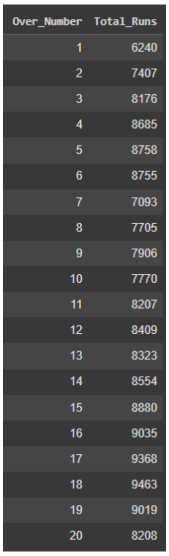
Player_Name	Total_Wickets
SL Malinga	159
DJ Bravo	137
A Mishra	132
Harbhajan Singh	128
PP Chawla	127
R Vinay Kumar	123
A Nehra	111
R Ashwin	110
Z Khan	107
DW Steyn	100

Table 8 Top 10 Wicket Takers



	Player_Name	Total_Awards
0	CH Gayle	17
1	YK Pathan	16
2	AB de Villiers	15
3	DA Warner	14
4	SK Raina	13
5	RG Sharma	13
6	MS Dhoni	12
7	MEK Hussey	12
8	G Gambhir	12
9	AM Rahane	12

Table 9 Players who won the most man of the match awards



Over_Number	Total_Runs
1	6240
2	7407
3	8176
4	8685
5	8758
6	8755
7	7093
8	7705
9	7906
10	7770
11	8207
12	8409
13	8323
14	8554
15	8880
16	9035
17	9368
18	9463
19	9019
20	8208

Table 10 Total runs scored, and total wickets taken in each over

- SELECT: This command would be used to select specific columns from the IPL dataset, such as player name, team name, runs scored, wickets taken, etc.
- JOIN: This command would be used to combine data from different tables in the IPL database, such as player information, team information, match details, etc. For example, to analyze the performance of a specific player, data from the player information and match details tables could be joined.
- GROUP BY: This command would be used to group the IPL data based on specific criteria, such as team name, player name, or match location. For example, to determine which team had the highest total runs scored in a season, the data could be grouped by team name and the runs scored column could be summed up.
- ORDER BY: This command would be used to sort the IPL data based on a specific column, such as runs scored, wickets taken, or match date. For example, to find the top 10 highest run scorers in a season, the data could be ordered by the runs scored column in descending order.
- LIMIT: This command would be used to limit the number of rows returned by a query, such as to display only the top 10 results of a query.
- LEFT JOIN: This command would be used to include all records from one table, even if there are no matching records in the other table. For example, to analyze the performance of all teams in a season, even if they didn't win any matches, a left join between the team information and match details tables could be used.
- COUNT: This command would be used to count the number of records that meet a specific condition, such as the number of matches won by a specific team or the number of wickets taken by a specific player.
- FROM: This command would be used to specify the table or tables to retrieve data from, such as the player information or match details tables.

- WHERE: This command would be used to filter the IPL data based on specific conditions, such as to only include records for a specific player or matches played at a specific location.

7 Exploratory Data Analysis

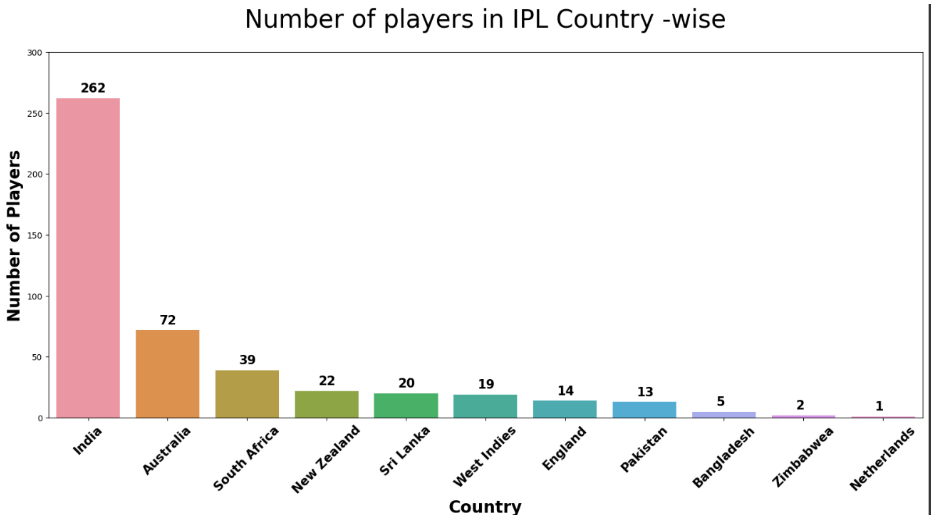


Fig 2 Number of Player in IPL Country Wise

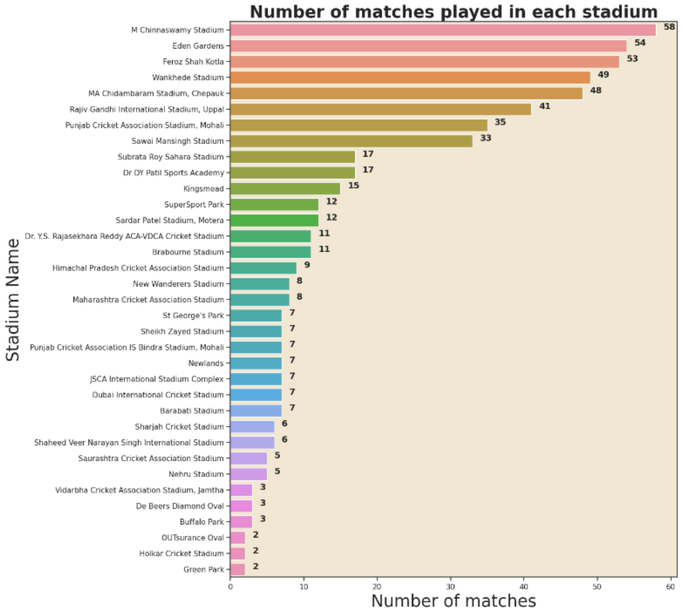


Fig 3 Number of Matches Played in each stadium

It is evident from Fig. 2 and 3 that There are 262 players from India. The majority of the foreign players are from Australia and South Africa with 72 and 39 layers respectively and there is a single player from Netherlands. From Fig 3. The highest number of matches are played in M Chinnaswamy Stadium. follwed by Eden Gardens.

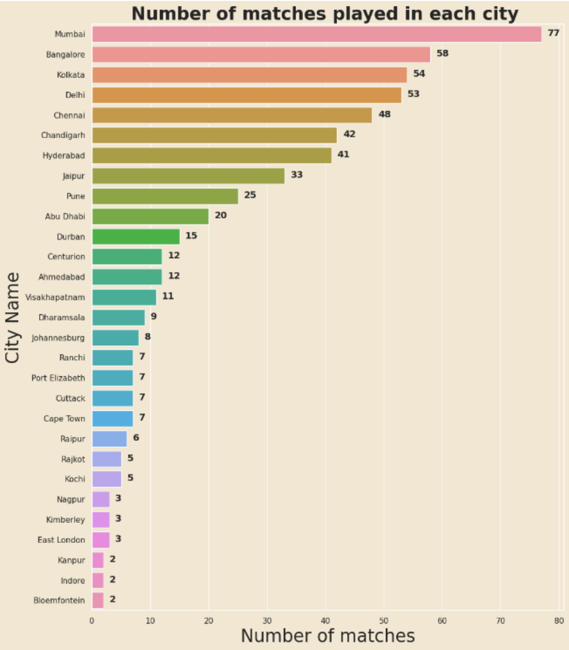


Fig 4 Number of matches played in each city

The highest number of matches are played in Mumbai followed by Bangalore and Kolkata according to Fig 4.

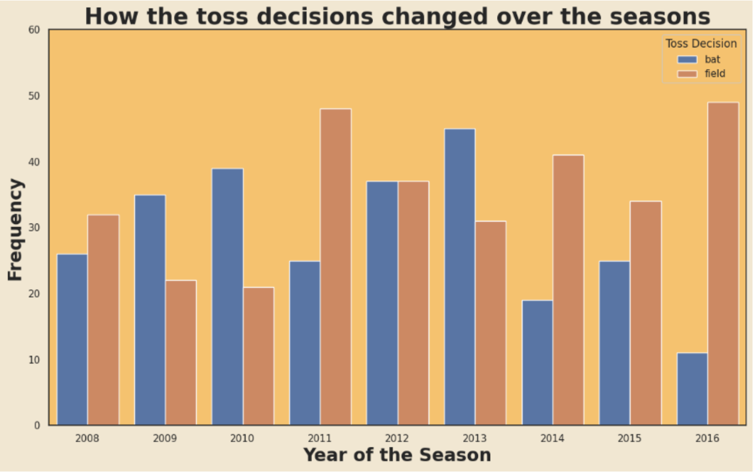
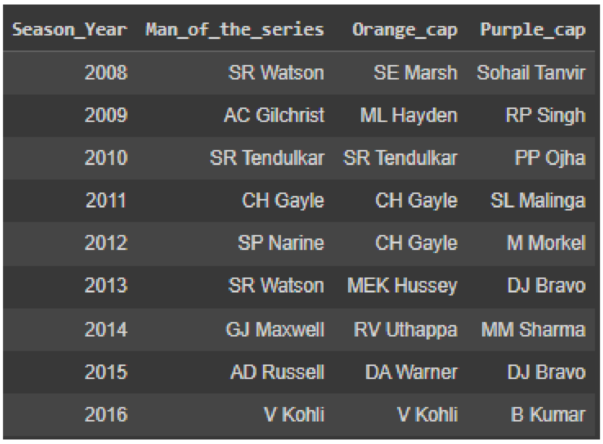


Fig 5 How the toss decisions changes over the seasons

We can observe from Fig 5 that the teams who won the toss preferred to bat first in the initial seasons but from the last 3 season teams are choosing to field first and in the 2016 season field first was the predominant choice.



Season_Year	Man_of_the_series	Orange_cap	Purple_cap
2008	SR Watson	SE Marsh	Sohail Tanvir
2009	AC Gilchrist	ML Hayden	RP Singh
2010	SR Tendulkar	SR Tendulkar	PP Ojha
2011	CH Gayle	CH Gayle	SL Malinga
2012	SP Narine	CH Gayle	M Morkel
2013	SR Watson	MEK Hussey	DJ Bravo
2014	GJ Maxwell	RV Uthappa	MM Sharma
2015	AD Russell	DA Warner	DJ Bravo
2016	V Kohli	V Kohli	B Kumar

Table 11 How the toss decisions changed over the seasons

Table 11 shows that The Orange Cap is presented to the leading run scorer and The Purple Cap is presented to the leading wicket-taker in the IPL. CH Gayle has won 2 Orange Caps and DJ Bravo has won 2 Purple Caps.

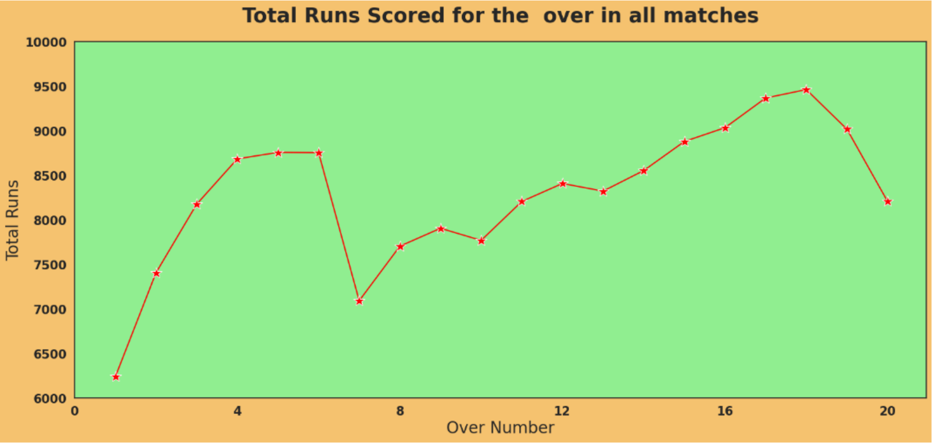


Fig 6 Total Runs Scored for the over in all matches

From fig 6 we can infer:

- The least amount of runs are made in the 1st over and the highest in the 18th over.

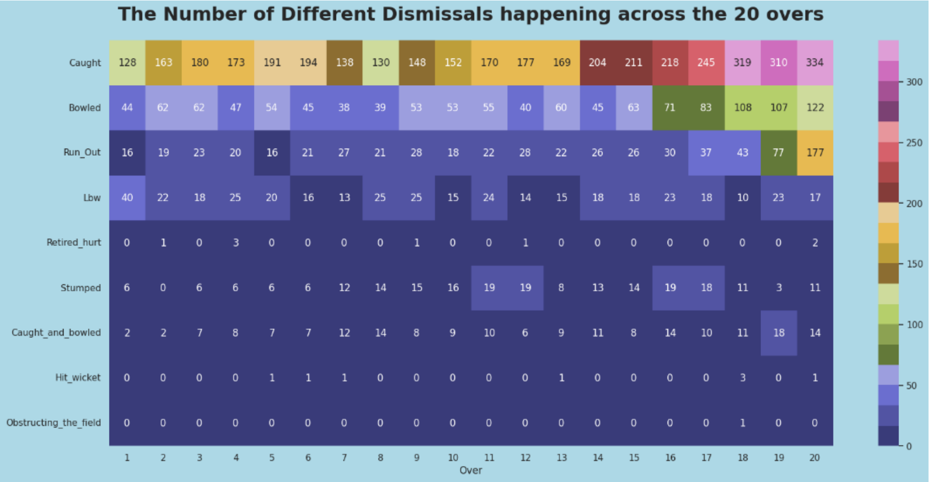
- The runs scored per over are increasing from 1st over to 6th over from then it is having a sharp drop and rising steadily and picking up momentum from 16th over onwards.

- The first six overs of an innings will be a mandatory powerplay, with only two fielders allowed outside the 30-yard circle.

- Beginning with the seventh over, no more than five fielders will be allowed outside the 30-yard circle.

- So the batsman are able to score more runs in the first 6 overs.

- The last 5 overs are commonly referred to as death overs. In these overs batting teams will be trying to score runs quicker than normal.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Caught	128	163	180	173	191	194	138	130	148	152	170	177	169	204	211	218	245	319	310	334
Bowled	44	62	62	47	54	45	38	39	53	53	55	40	60	45	63	71	83	108	107	122
Run_Out	16	19	23	20	16	21	27	21	28	18	22	28	22	26	26	30	37	43	77	177
Lbw	40	22	18	25	20	16	13	25	25	15	24	14	15	18	18	23	18	10	23	17
Retired_hurt	0	1	0	3	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	2
Stumped	6	0	6	6	6	6	12	14	15	16	19	19	8	13	14	19	18	11	3	11
Caught_and_bowled	2	2	7	8	7	7	12	14	8	9	10	6	9	11	8	14	10	11	18	14
Hit_wicket	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	3	0	1
Obstructing_the_field	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fig 7 The number of different dismissals happening across the 20 years

Figure 7 shows the number of dismissals with the darkest cell showing the maximum number of dismissals. We can conclude from this graph that:

- True to its name, death overs the last 5 overs are having most wickets. As batsman try to score more run as the innings is ending they are risking to score more runs and getting out.

- The most interesting is the run out dismissal in the last 2 overs. It may be because as tailenders of the batting team trying to rotate strike to the batsman who can bat well and getting out in the process.

- The most common way of getting out are caught, bowled, run out.

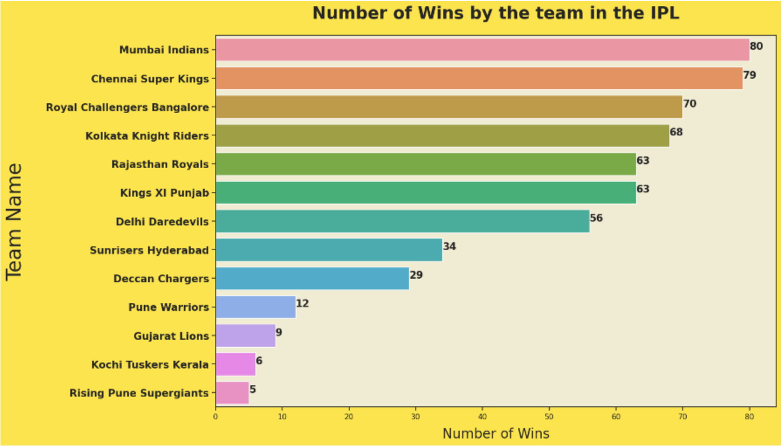


Fig 8 Number of wins by the team in IPL

From Fig 8 we can infer that:

- Mumbai Indians won the highest number of matches in the all seasons from 2008 to 2016. Rising Pune Supergiants won the least number of matches.

- Pune Warriors ,Gujarat Lions, Kochi Tuskers Kerala, Rising Pune Supergiants have won less number of matches.

- These franchises got added in the later seasons of the IPl and some were dissolved too. So they could not play more number of matches.

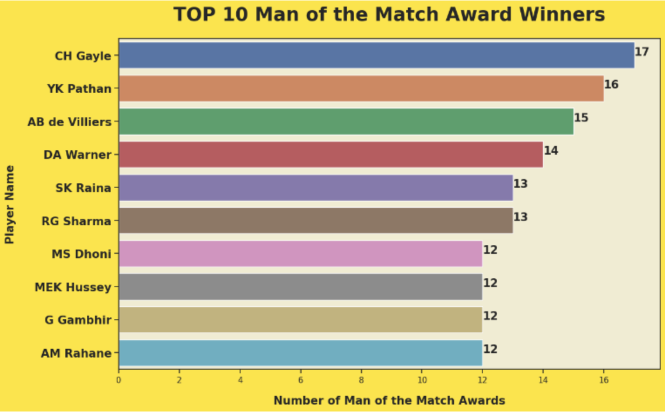


Fig. 9 Top 10 Man of the match award winners

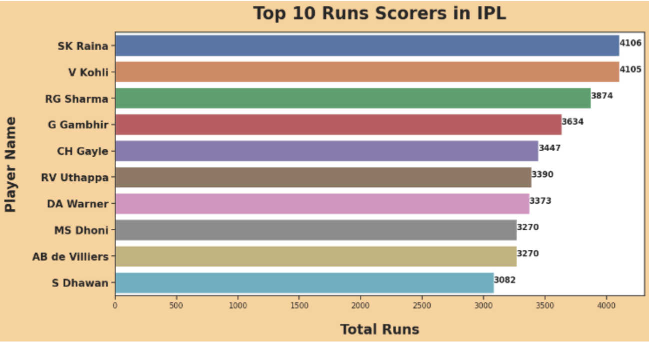


Fig. 10 Top 10 Runs Scorers in IPL

Figure 9 and figure 10 shows that:

- CH Gayle has won the highest number of the Man of the Match Awards
- SK Raina has scored the highest total runs with 4106 runs and VK Kohli with 4105 missed the top position by a whister of a single run.

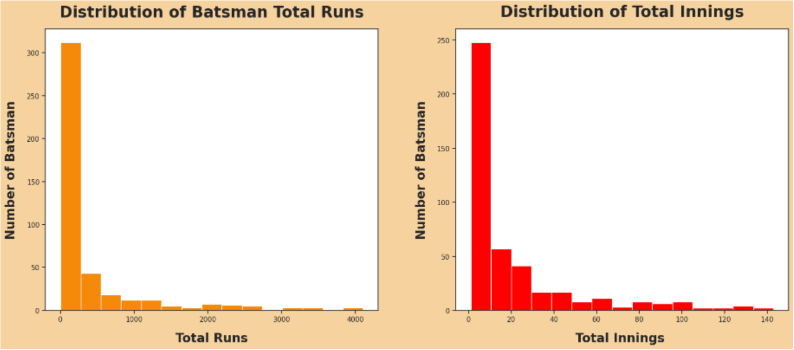


Fig 11 Distribution of Batsman Total Runs and Total Innings

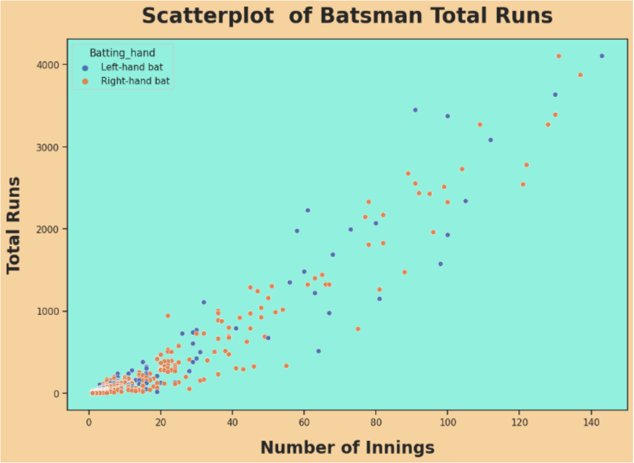


Fig 12 Scatter Plot of Batsman Total Runs

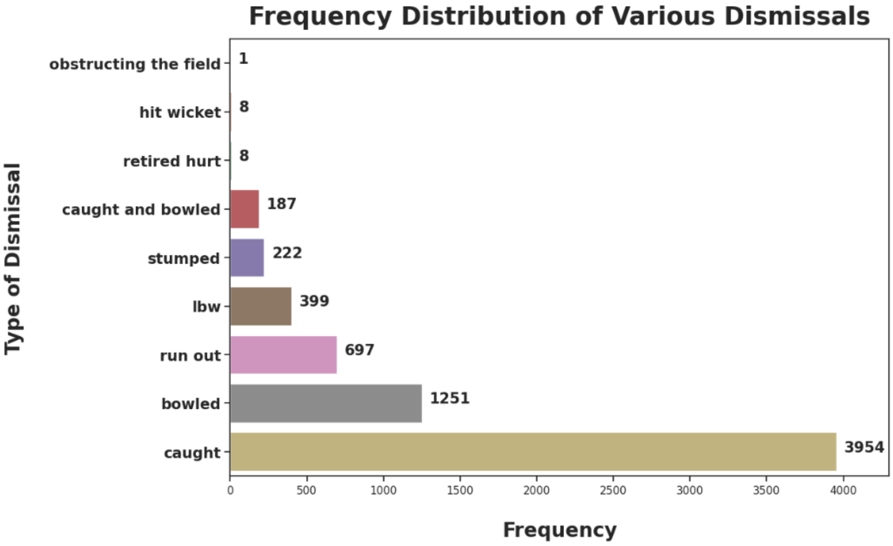


Fig 13 Frequency Distribution of Various Dismissals

Figure 11, 12 and 13 helps us to conclude that:

- The distribution of the Total Runs and Total Innings is heavily skewed to the towards the right indicating that few players are the star players.
- Majority of players have scored less than 250 runs in total and less than 10 innings.
- Most of the batsman are right handed and out of 9 players with more than 120 innings, 7 are right handed batsman.

- 3954 batsman got out by getting caught followed by bowled , run out. . Only one batsman got out by obstructing the fielder in all the IPL matches.

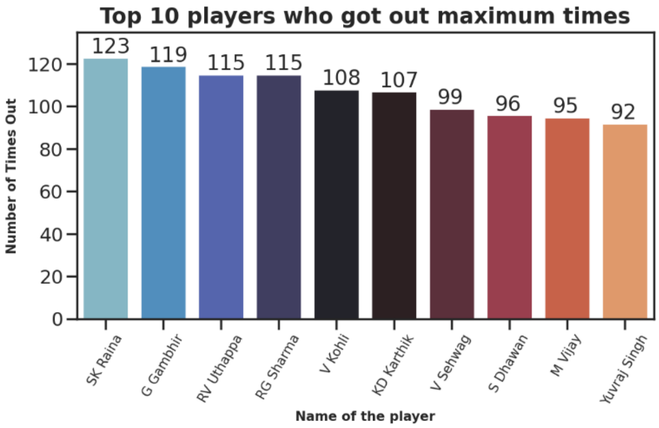


Fig 14 Top 10 players who got out maximum times

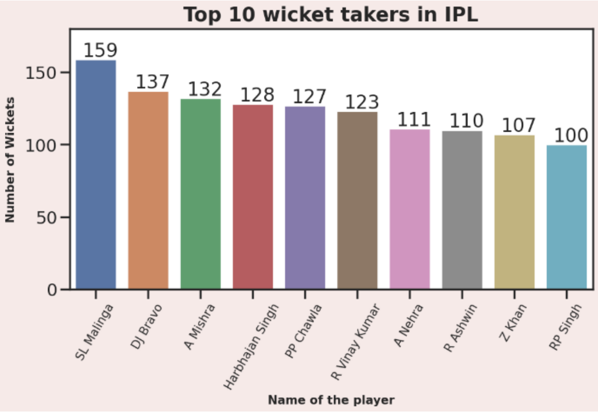


Fig 15 Top 10 wicket takers in IPL

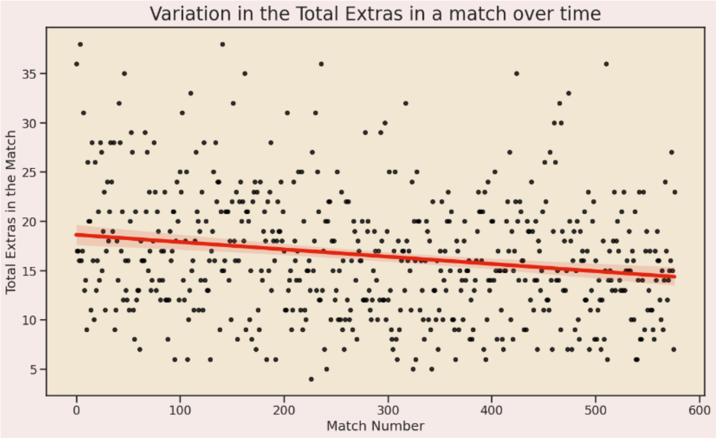


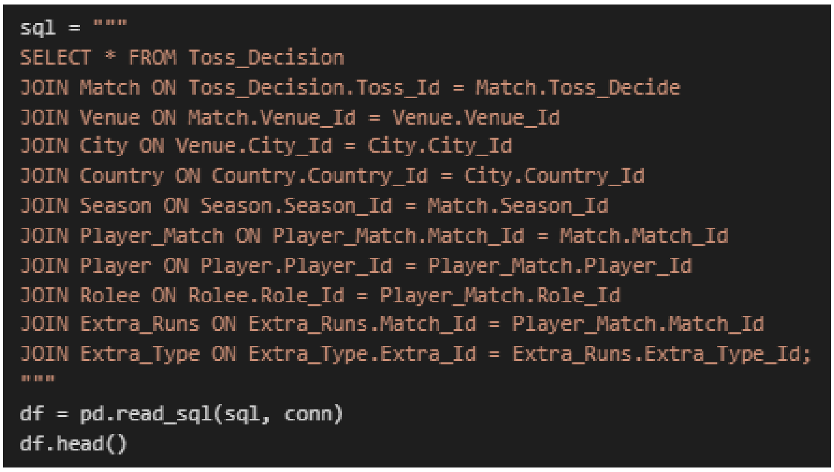
Fig 16 Variation in total extras in a match over time

From fig. 14. 15 and fig 16 we can infer that:

- SK Raina got out maximum times getting out 123 times.
- SL Malinga is the top wicket taker with 159 wickets
- There is a decreasing trend in the total extra runs in a match over time. That means bowlers are getting better and being careful not to award extra runs.

8 Joining Necessary Tables

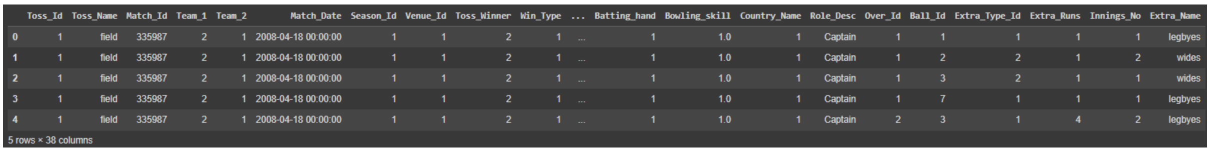
We have used SELECT, JOIN commands to JOIN the multiple tables in the database to form a csv file for further processing and regression analysis. This can be seen in fig 17.



```
sql = """
SELECT * FROM Toss_Decision
JOIN Match ON Toss_Decision.Toss_Id = Match.Toss_Decide
JOIN Venue ON Match.Venue_Id = Venue.Venue_Id
JOIN City ON Venue.City_Id = City.City_Id
JOIN Country ON Country.Country_Id = City.Country_Id
JOIN Season ON Season.Season_Id = Match.Season_Id
JOIN Player_Match ON Player_Match.Match_Id = Match.Match_Id
JOIN Player ON Player.Player_Id = Player_Match.Player_Id
JOIN Rolee ON Rolee.Role_Id = Player_Match.Role_Id
JOIN Extra_Runs ON Extra_Runs.Match_Id = Player_Match.Match_Id
JOIN Extra_Type ON Extra_Type.Extra_Id = Extra_Runs.Extra_Type_Id;
"""
df = pd.read_sql(sql, conn)
df.head()
```

Fig 17 Joining Necessary Tables to form CSV file

We remove the unimportant columns in the dataset to process the data for regression analysis. The processed dataset is shown in Fig 18.



	Toss_Id	Toss_Name	Match_Id	Team_1	Team_2	Match_Date	Season_Id	Venue_Id	Toss_Winner	Win_Type	...	Batting_hand	Bowling_skill	Country_Name	Role_Desc	Over_Id	Ball_Id	Extra_Type_Id	Extra_Runs	Innings_No	Extra_Name
0	1	field	335987	2	1	2008-04-18 00:00:00	1	1	2	1	...	1	1.0	1	Captain	1	1	1	1	1	legbyes
1	1	field	335987	2	1	2008-04-18 00:00:00	1	1	2	1	...	1	1.0	1	Captain	1	2	2	1	2	wides
2	1	field	335987	2	1	2008-04-18 00:00:00	1	1	2	1	...	1	1.0	1	Captain	1	3	2	1	1	wides
3	1	field	335987	2	1	2008-04-18 00:00:00	1	1	2	1	...	1	1.0	1	Captain	1	7	1	1	1	legbyes
4	1	field	335987	2	1	2008-04-18 00:00:00	1	1	2	1	...	1	1.0	1	Captain	2	3	1	4	2	legbyes

5 rows × 38 columns

Fig 18 Processed Dataset

9 Results

Regression Analysis: predicting Win_Margin of teams.

The steps involved in this process are:

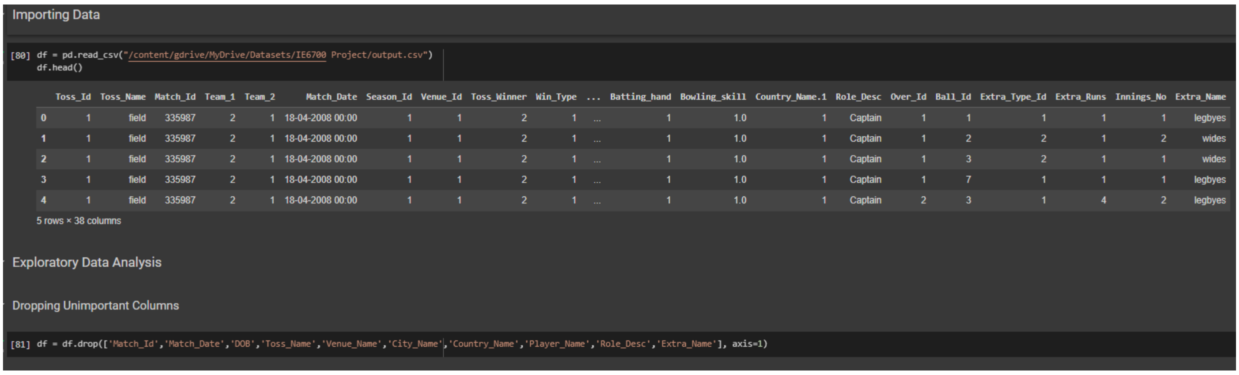
Step 1 : Importing the dataset formed on MySQL

Step 2: Dropping unimportant columns

Step 3 : Dealing with Null Values

Step 4: Splitting the data into train and test parts

Step 5: Regression Analysis



Importing Data

```
[80] df = pd.read_csv("/content/gdrive/MyDrive/Datasets/IE6700 Project/output.csv")
     df.head()
```

	Toss_Id	Toss_Name	Match_Id	Team_1	Team_2	Match_Date	Season_Id	Venue_Id	Toss_Winner	Win_Type	...	Batting_hand	Bowling_skill	Country_Name.1	Role_Desc	Over_Id	Ball_Id	Extra_Type_Id	Extra_Runs	Innings_No	Extra_Name
0	1	field	335987	2	1	18-04-2008 00:00	1	1	2	1	...	1	1.0	1	Captain	1	1	1	1	1	legbyes
1	1	field	335987	2	1	18-04-2008 00:00	1	1	2	1	...	1	1.0	1	Captain	1	2	2	1	2	wides
2	1	field	335987	2	1	18-04-2008 00:00	1	1	2	1	...	1	1.0	1	Captain	1	3	2	1	1	wides
3	1	field	335987	2	1	18-04-2008 00:00	1	1	2	1	...	1	1.0	1	Captain	1	7	1	1	1	legbyes
4	1	field	335987	2	1	18-04-2008 00:00	1	1	2	1	...	1	1.0	1	Captain	2	3	1	4	2	legbyes

5 rows × 38 columns

Exploratory Data Analysis

Dropping Unimportant Columns

```
[81] df = df.drop(['Match_Id','Match_Date','DOB','Toss_Name','Venue_Name','City_Name','Country_Name','Player_Name','Role_Desc','Extra_Name'], axis=1)
```

Fig 19 Importing the dataset and dropping unimportant columns

As in Fig 19, we have imported the dataset and dropped the unimportant columns to for further regression analysis.

The correlation matrix in Fig 20 shows that multiple columns are highly positively and negatively correlated with the win_margin column. Hence we compute the columns having high correlation.

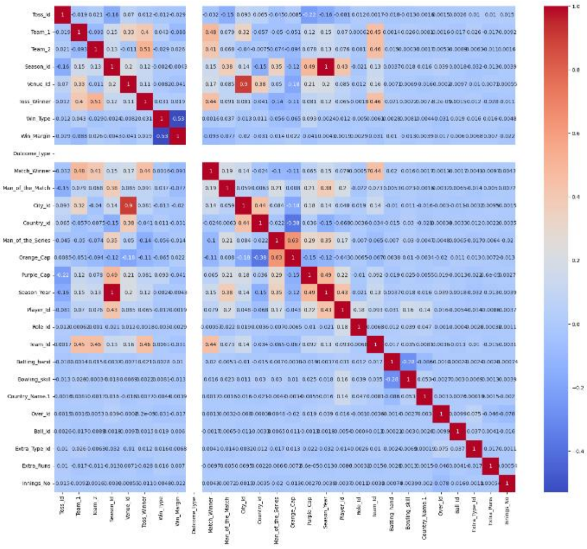
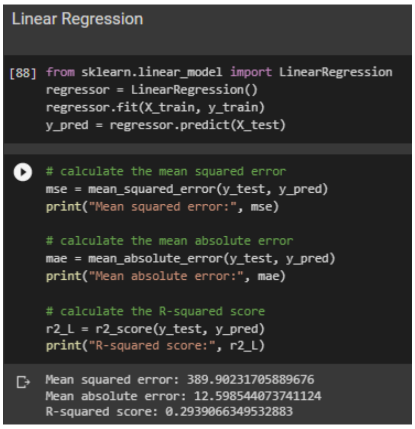


Fig 20 Correlation Matrix of Columns



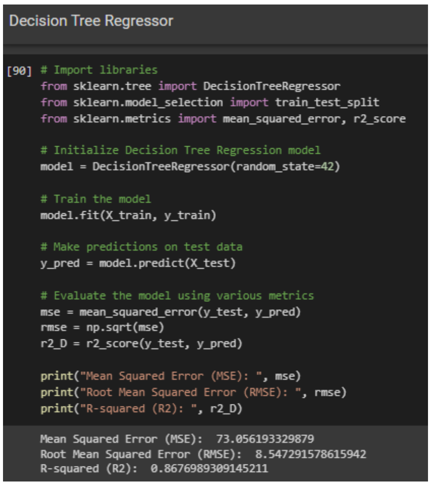
Linear Regression

```
[88] from sklearn.linear_model import LinearRegression
     regressor = LinearRegression()
     regressor.fit(X_train, y_train)
     y_pred = regressor.predict(X_test)
```

```
     # calculate the mean squared error
     mse = mean_squared_error(y_test, y_pred)
     print("Mean squared error:", mse)

     # calculate the mean absolute error
     mae = mean_absolute_error(y_test, y_pred)
     print("Mean absolute error:", mae)

     # calculate the R-squared score
     r2_L = r2_score(y_test, y_pred)
     print("R-squared score:", r2_L)
```

```
Mean squared error: 389.90231705889676
Mean absolute error: 12.598544073741124
R-squared score: 0.2939066349532883
```

Fig 21 Linear Regression model

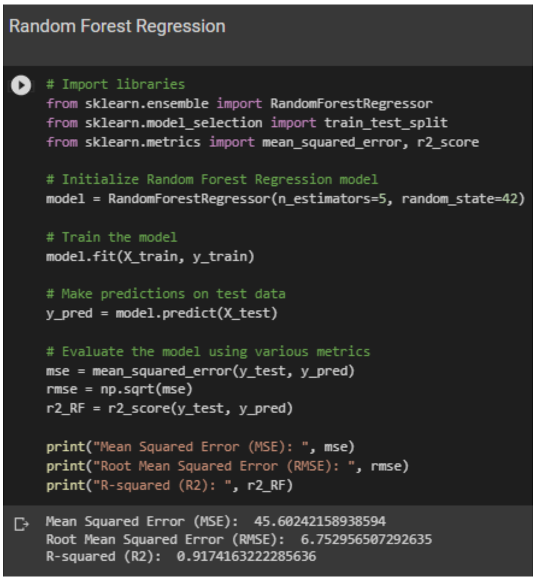


Decision Tree Regressor

```
[90] # Import libraries
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error, r2_score

     # Initialize Decision Tree Regression model
     model = DecisionTreeRegressor(random_state=42)

     # Train the model
     model.fit(X_train, y_train)

     # Make predictions on test data
     y_pred = model.predict(X_test)

     # Evaluate the model using various metrics
     mse = mean_squared_error(y_test, y_pred)
     rmse = np.sqrt(mse)
     r2_D = r2_score(y_test, y_pred)

     print("Mean Squared Error (MSE): ", mse)
     print("Root Mean Squared Error (RMSE): ", rmse)
     print("R-squared (R2): ", r2_D)
```

```
Mean Squared Error (MSE):  73.056193329879
Root Mean Squared Error (RMSE):  8.547291578615942
R-squared (R2):  0.8676989309145211
```

Fig 22 Decision Tree Regressor Model

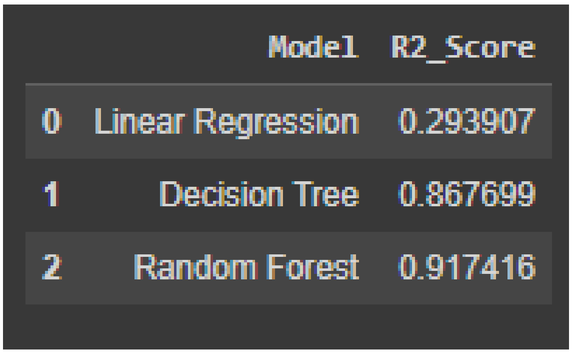


Random Forest Regression

```
# Import libraries
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Initialize Random Forest Regression model
model = RandomForestRegressor(n_estimators=5, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions on test data
y_pred = model.predict(X_test)

# Evaluate the model using various metrics
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2_RF = r2_score(y_test, y_pred)

print("Mean Squared Error (MSE): ", mse)
print("Root Mean Squared Error (RMSE): ", rmse)
print("R-squared (R2): ", r2_RF)
```

```
Mean Squared Error (MSE):  45.60242158938594
Root Mean Squared Error (RMSE):  6.752956507292635
R-squared (R2):  0.9174163222285636
```

Fig 23 Random Forest Regression



	Model	R2_Score
0	Linear Regression	0.293907
1	Decision Tree	0.867699
2	Random Forest	0.917416

Fig 24 Model Comparison

From figures 21 to 24 it is evident that the data is not linear hence, the linear regression model accuracy is very low. The decision tree model and random forest model fit very nicely with the data. We can accurately predict the win margin of a team and winning team with 91.7% accuracy. Using the parameters of an IPL match, one can accurately predict the winning team. If the margin of team is negative, the team has lost the match and if its positive, the team has one the match.

10 Conclusion

- There are 262 players from India. Most of the foreign players are from Australia and South Africa with 72 and 39 players respectively. There is a single player from Netherlands.
- The highest number of matches are played in M Chinnaswamy Stadium. followed by Eden Gardens.
- The highest number of matches are played in Mumbai followed by Bangalore and Kolkata.
- The teams who won the toss preferred to bat first in the initial seasons but from the last 3 season teams are choosing to field first and in the 2016 season field first was the predominant choice.
- CH Gayle has won 2 Orange Caps and DJ Bravo has won 2 Purple Caps.
- The least number of runs are made in the 1st over and the highest in the 18th over. The runs scored per over are increasing from 1st over to 6th over from then it is having a sharp drop and rising steadily and picking up momentum from 16th over onwards. The first six overs of an innings will be a mandatory powerplay, with only two fielders allowed outside the 30-yard circle. Beginning with the seventh over, no more than five fielders will be allowed outside the 30-yard circle. So the batsman are able to score more runs in the first 6 overs. The last 5 overs are commonly referred to as death overs. In these overs batting teams will be trying to score runs quicker than normal.
- True to its name, death overs the last 5 overs are having most wickets. As batsman try to score more run as the innings is ending, they are risking to score more runs and

getting out. The most interesting is the run-out dismissal in the last 2 overs. It may be because as tailenders of the batting team trying to rotate strike to the batsman who can bat well and getting out in the process. The most common way of getting out are caught, bowled, run out.

- Mumbai Indians won the highest number of matches in the all seasons from 2008 to 2016. Rising Pune Supergiants won the least number of matches. Pune Warriors ,Gujarat Lions, Kochi Tuskers Kerala, Rising Pune Supergiants have won a smaller number of matches. These franchises got added in the later seasons of the IPl and some were dissolved too. They could not play a greater number of matches.
- CH Gayle has won the highest number of the Man of the Match Awards.
- SK Raina has scored the highest total runs with 4106 runs and VK Kohli with 4105 missed the top position by a whister of a single run.
- The distribution of the Total Runs and Total Innings is heavily skewed to the towards the right indicating that few players are the star players. Majority of players have scored less than 250 runs in total and less than 10 innings.
- Most of the batsman are right-handed and out of 9 players with more than 120 innings, 7 are right-handed batsman.
- 3954 batsman got out by getting caught followed by bowled , run out. . Only one batsman got out by obstructing the fielder in all the IPL matches.
- SK Raina got out maximum times getting out 123 times.
- SL Malinga is the top wicket taker with 159 wickets
- There is a decreasing trend in the total extra runs in a match over time. That means bowlers are getting better and being careful not to award extra runs.

- We received a maximum accuracy of 91.7% using Random Forest model.
- With all the parameters in place one can predict Win_margin and win between two IPL teams.

11 References

1. H. Barot, A. Kothari, P. Bide, B. Ahir and R. Kankaria, "Analysis and Prediction for the Indian Premier League," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9153972.

2. Sasank Viswanadha, Kaustubh Sivalenka, Madan Gopal Jhawar and Vikram Pudi, *Dynamic Winner Prediction in Twenty20 Cricket: Based on Relative Team Strengths*.

3. Ananda Bandulasiri, *Predicting the Winner in One Day International Cricket*.

4. S Priyanka, K Vysali and K B PriyaIyer, *Prediction of Indian Premier League-IPL 2020 using Data Mining Algorithms*.

5. Shilpi Agrawal, Suraj Pal Singh and Jayash Kumar Sharma, *Predicting Results of Indian Premier League T-20Matches using Machine Learning*.

6. Amal Chaminda Kaluarachchi and Aparna S. Varde, *CricAI A Classification Based Tool to Predict the Outcome in ODI Cricket*.

7. Madan Gopal Jhanwar and Vikram Pudi, *Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach*.

8. *Cricket Stats*, [online] Available: https://stats.espncricinfo.com/ci/engine/records/index.html.

9. [online] Available: https://towardsdatascience.com/the-complete-guide-to-decision-trees-28a4e3c7be14.