

1. INTRODUCTION

1.1 Overview

Cricket, especially the Twenty20 format, has maximum uncertainty, where a single over can completely change the momentum of the game. With millions of people following the Indian Premier League (IPL), developing a model for predicting the outcome of its matches is a real-world problem. A cricket match depends upon various factors, and in this work, the factors which significantly influence the outcome of a Twenty20 cricket match are identified. Each player's performance in the field is considered to find out the overall weight of the team. A multivariate regression based solution is proposed to calculate points for each player in the league and the overall weight of a team is computed based on the past performance of the players who have appeared most for the team. Finally, a dataset is modeled based on the identified seven factors which influence the outcome of an IPL match. Six machine learning models were trained and used for predicting the outcome of each IPL match till 2019, 15 minutes before the gameplay, immediately after the toss. Three of the trained models were seen to be correctly predicting more than 40 matches, with Multilayer Perceptron outperforming all other models with an impressive accuracy of 71.66%.

1.2 Purpose

With technology growing more and more advanced in the last few years, an in-depth acquisition of data has become relatively easy. As a result, Machine Learning is becoming quite a trend in sports analytics because of the availability of live as well as historical data. Sports analytics is the process of collecting past matches data and analyzing them to extract the essential knowledge out of it, with a hope that it facilitates in effective decision making. Decision making may be anything including which player to buy during an auction, which player to set on the field for tomorrow's match, or something more strategic task like, building the tactics for forthcoming matches based on players' previous performances. When it is about on-the-field, machine learning applies to the analysis of a player's fitness level, design of offensive tactics, or decide shot selection. It is also used in predicting the performance of a player or a team, or the outcome of a match. Similarly, cricket has also been making use of sports analytics to perform prediction of outcome of a match, while the gameplay is in progress or before the match has even begun. Even problem like predicting runs or wickets of a player for a match, based on his/her past performance is an interesting problem to work on.

2 LITERATURE SURVEY

2.1 Existing problem

Since the dawn of the IPL in 2008, it has attracted viewers all around the globe. High level of uncertainty and last moment nail biters has drawn the fans to watch the matches in large numbers. Within a short period, IPL has become the highest revenue generating league of cricket. With all this, the amount of data being generated in terms of matches revenue scores etc has also become huge. Analysing such vast amounts of data would give great insights in forecasting match results ,top scores and wicket takers etc.

The objective of this solution is to create dashboard that visualizes following capabilities and also forecast the future results

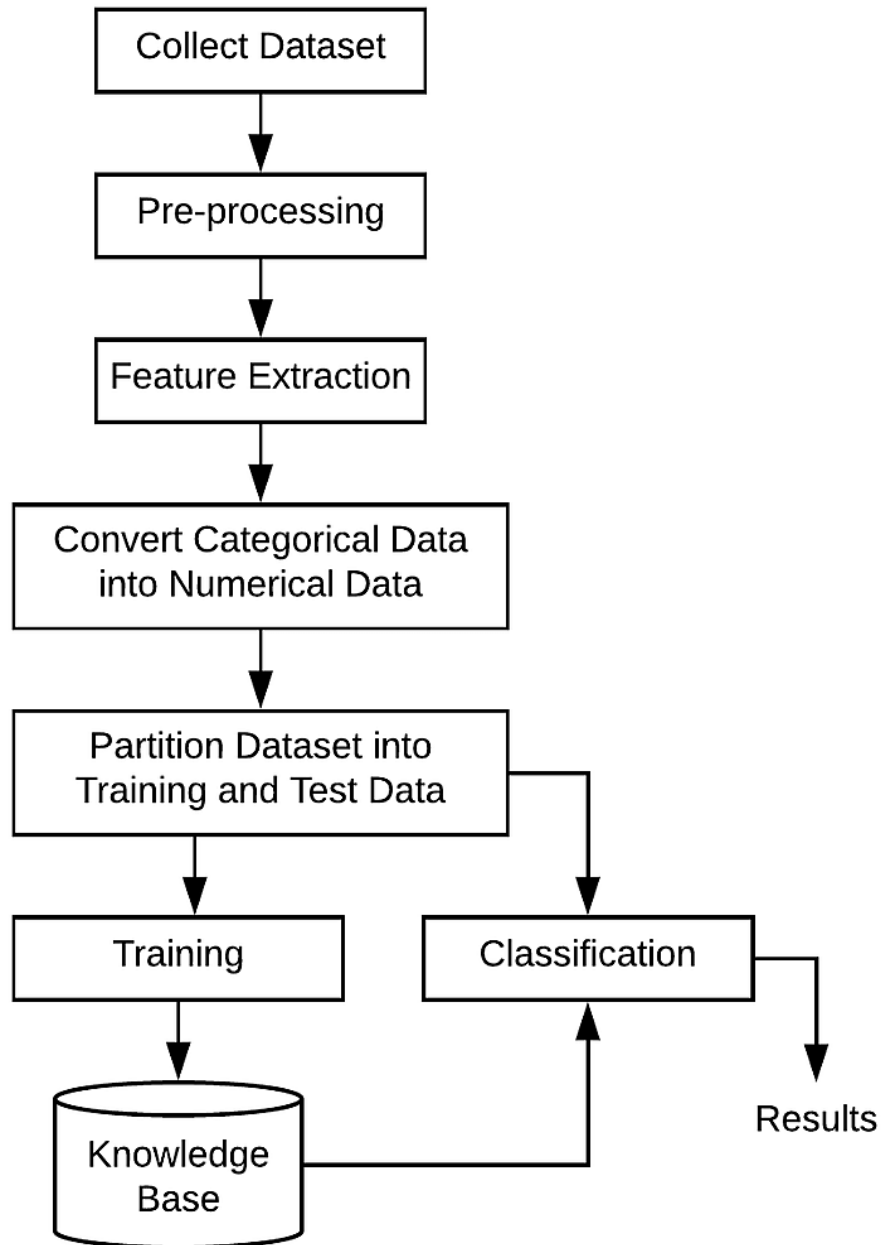
1. To find the team that won the most number of matches in the entire IPL.
2. To find the team that lost the most number of matches in the entire IPL.
3. Does winning a toss increase the chances of victory.
4. To find the player with the most player of the match awards.
5. To find the city that hosted the maximum number of IPL matches.
6. To find the most winning team for each season.
7. To find the on-field umpire with the maximum number of IPL matches.
8. To find the biggest victories in IPL while defending a total and while chasing a total.
9. Which team won the most matches while batting first.
10. Which team won the most matches while batting second.
11. List of teams which have won matches by most runs cumulatively

2.2 Proposed solution

The literature survey concluded that there was a need for a machine learning model which could predict the outcome of an IPL match before the game begins. Among all formats of cricket, Twenty20 format sees a lot of turnarounds in the momentum of the game. An over can completely change a game. Hence, predicting an outcome for a Twenty20 game is quite a challenging task. Besides, developing a prediction model for a league which is wholly based on auction is another hurdle. IPL matches cannot be predicted simply by making use of statistics over historical data solely. Because of players going under auctions, the players are bound to change their teams; which is why the ongoing performance of every player must be taken into consideration while developing a prediction model. In this work, the various factors that affect the outcome of a cricket match were analyzed, and it was observed that home team, away team, venue, toss winner, toss decision, home team weight, away team weight, influence the win probability of a team. The proposed prediction model makes use of **Multivariate Classification** to calculate points of each player in the league and compute the overall strength of each team based on the past performance of the players who have appeared most for the team.

3 THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

Hardware Requirements

1. 8/16+ GB RAM
2. Intel/AMD/MAC Processor (Anything Preferred)
3. Graphic card (Anything Preferred)

Software Requirements

1. IBM Watson Studio
2. IBM Cloud Pak for Data
3. IBM Cognos Analytics
4. Jupyter Notebook
5. Data sets

4 EXPERIMENTAL INVESTIGATIONS

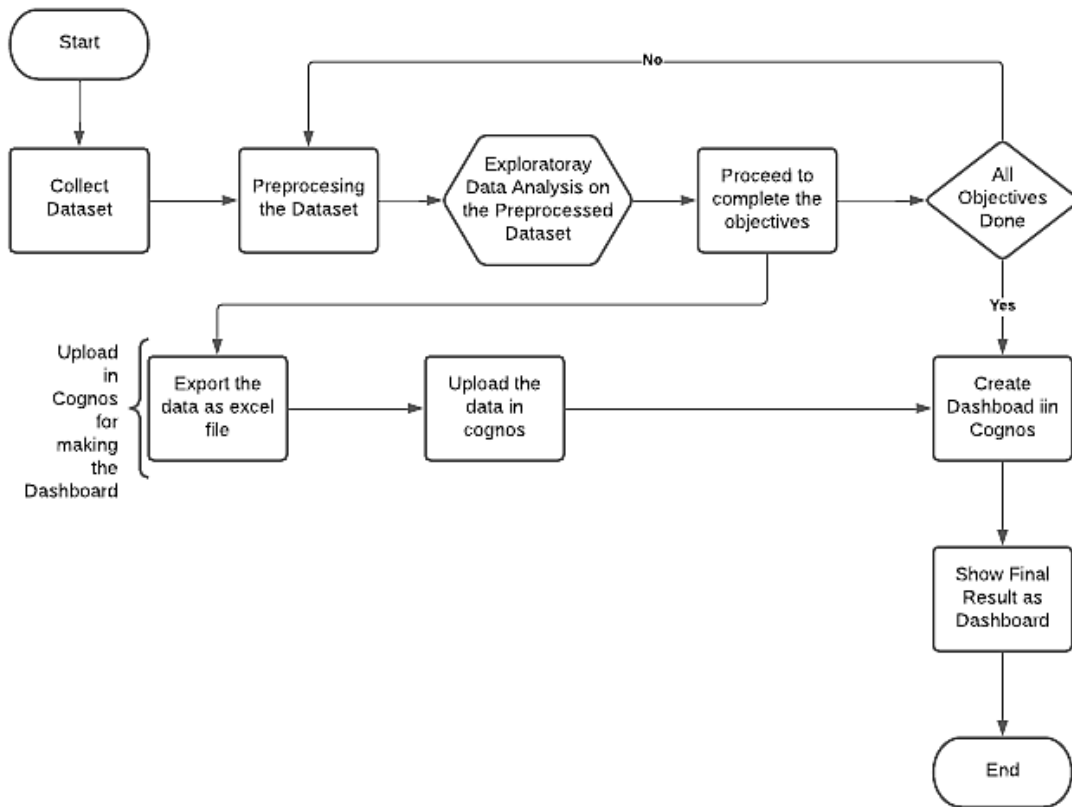
While analysiing the solution for this problem we have found some unwanted and some duplicate data that need to be processed in a manner such that to get a accurate results. So, that we have removed some unwanted data also merged the duplicate data. In our project there are such constratints which can affect the analysis such as

- There are some IPL Teams which changes their name at some seasons Eg: Delhi Daredevils, Royal Challengers Bangalore, Sunrises Hyderabad etc...
- There are some new IPL Teams which played for only few ipl seasons Eg: Kochi Tuskers Kerala etc..
- there is also teams like Pune Warriors played for some season.
- there is also some teams like chennai and Rajasthan got banned for couple of years.
- There evolved a 2 teams Gujarat Lions and Rising Pune Spergiants which was replacing the Chennai and Rajasthan.

These are some factors that may affect our analysis. So, we have sorted those issues by data extraction using "Pandas" library in python.

We also created a relationship with the atributes and constraints using the data sets which we obtained from the data extraction process. As every attribute and constraints are related to each other it is easy to figure out the solutions for the required problem statements.

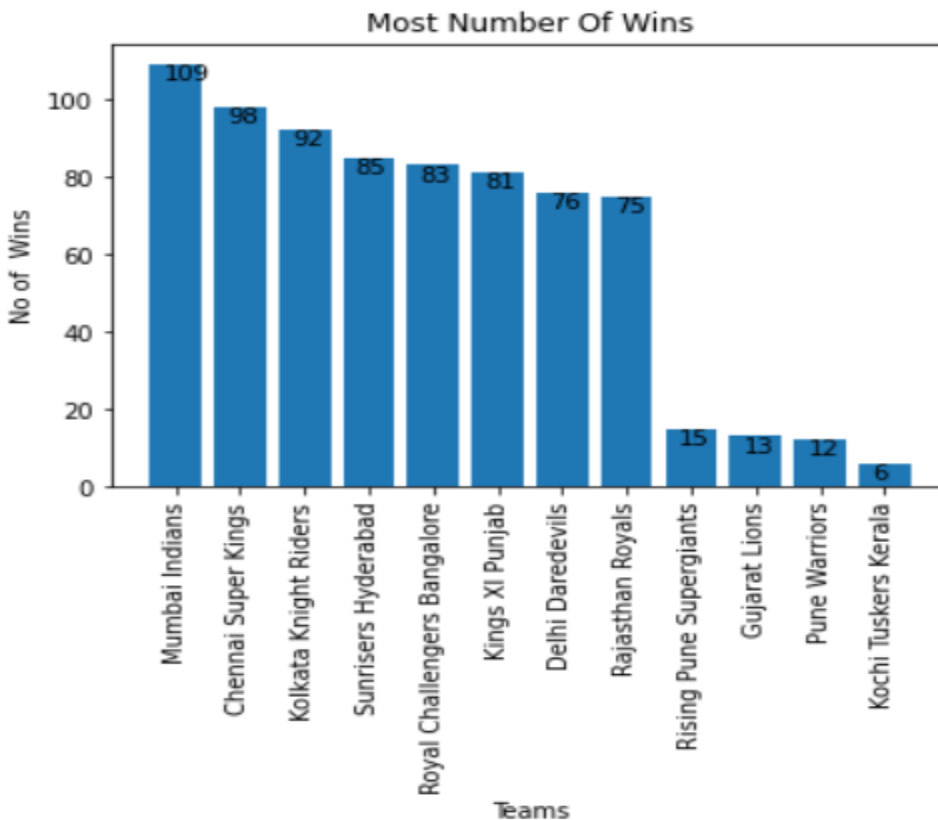
5 FLOWCHART



6 RESULT

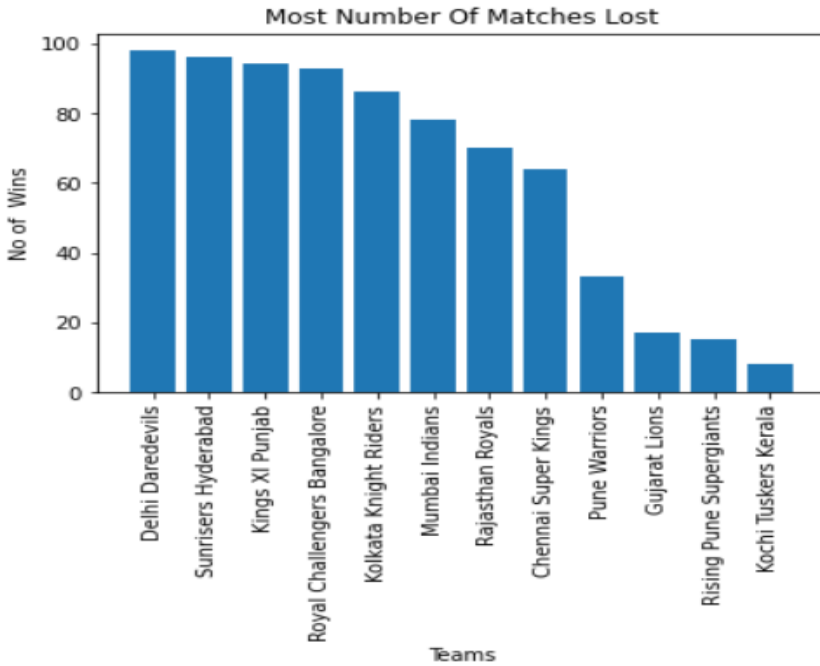
#1.To find the team that won the most number of matches in the entire IPL.

MUMBAI Indians



#2.To find the team that lost the most number of matches in the entire IPL.

Delhi Daredevils



#3.Does winning a toss increase the chances of victory.

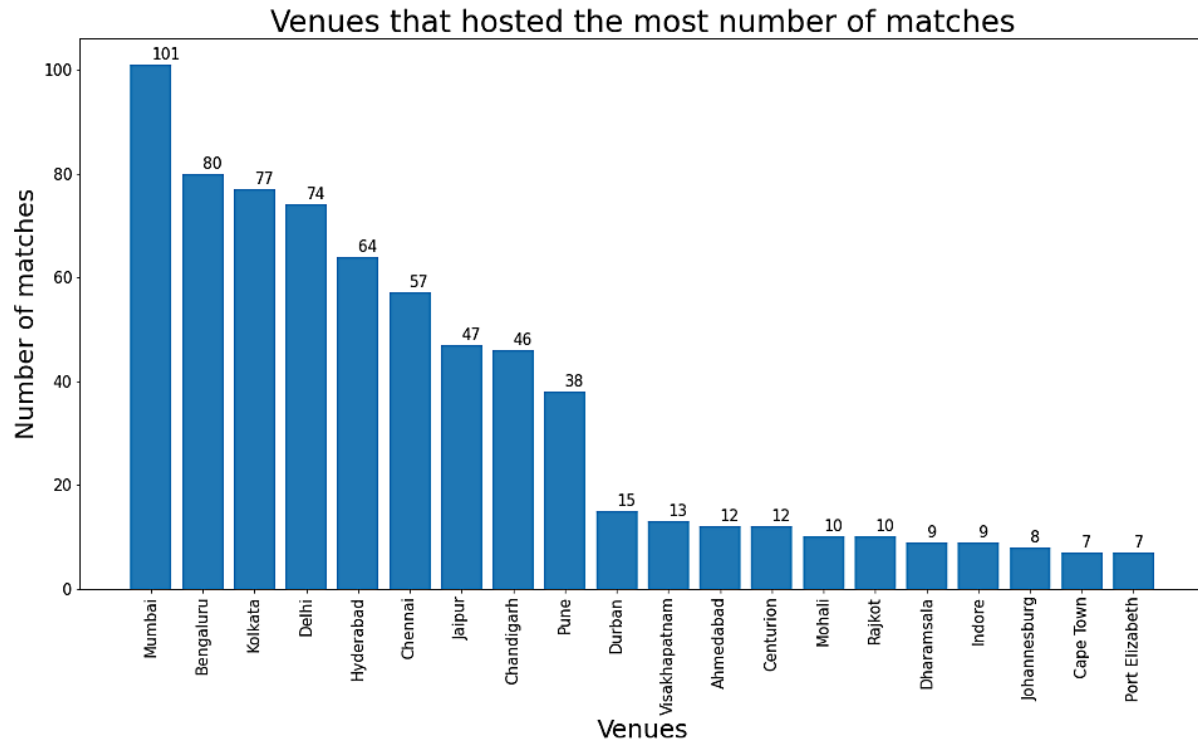
The number of times the team winning toss have won: 393

The probability of winning if won the toss: 0.52

#4.To find the player with the most player of the match awards.

	player_of_match	No_Of_Awards
35	CH Gayle	21
10	AB de Villiers	20
167	RG Sharma	17
42	DA Warner	17
137	MS Dhoni	17
222	YK Pathan	16
201	SR Watson	15
191	SK Raina	14
56	G Gambhir	13
123	MEK Hussey	12

#5.To find the city that hosted the maximum number of IPL matches.



#6.To find the most winning team for each season.

	year	team	wins
0	2008	Rajasthan Royals	13
0	2009	Delhi Daredevils	10
0	2010	Mumbai Indians	11
0	2011	Chennai Super Kings	11
0	2012	Kolkata Knight Riders	12
0	2013	Mumbai Indians	13
0	2014	Kings XI Punjab	12
0	2015	Chennai Super Kings	10
0	2016	Sunrisers Hyderabad	11
0	2017	Mumbai Indians	12
0	2018	Chennai Super Kings	11
0	2019	Mumbai Indians	11

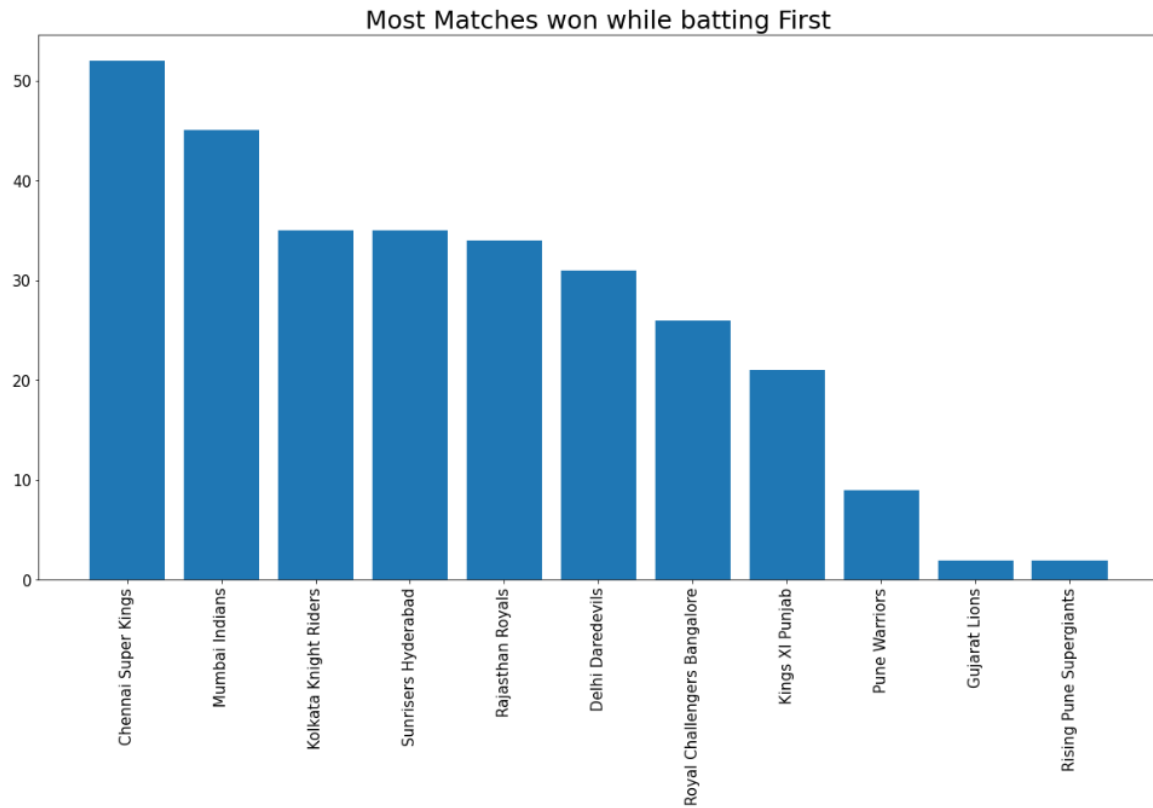
#7.To find the on-field umpire with the maximum number of IPL matches.

	umpire	matches
2	S Ravi	106.0
0	HDPK Dharmasena	87.0
11	C Shamshuddin	73.0
3	AK Chaudhary	58.0
59	SJA Taufel	55.0

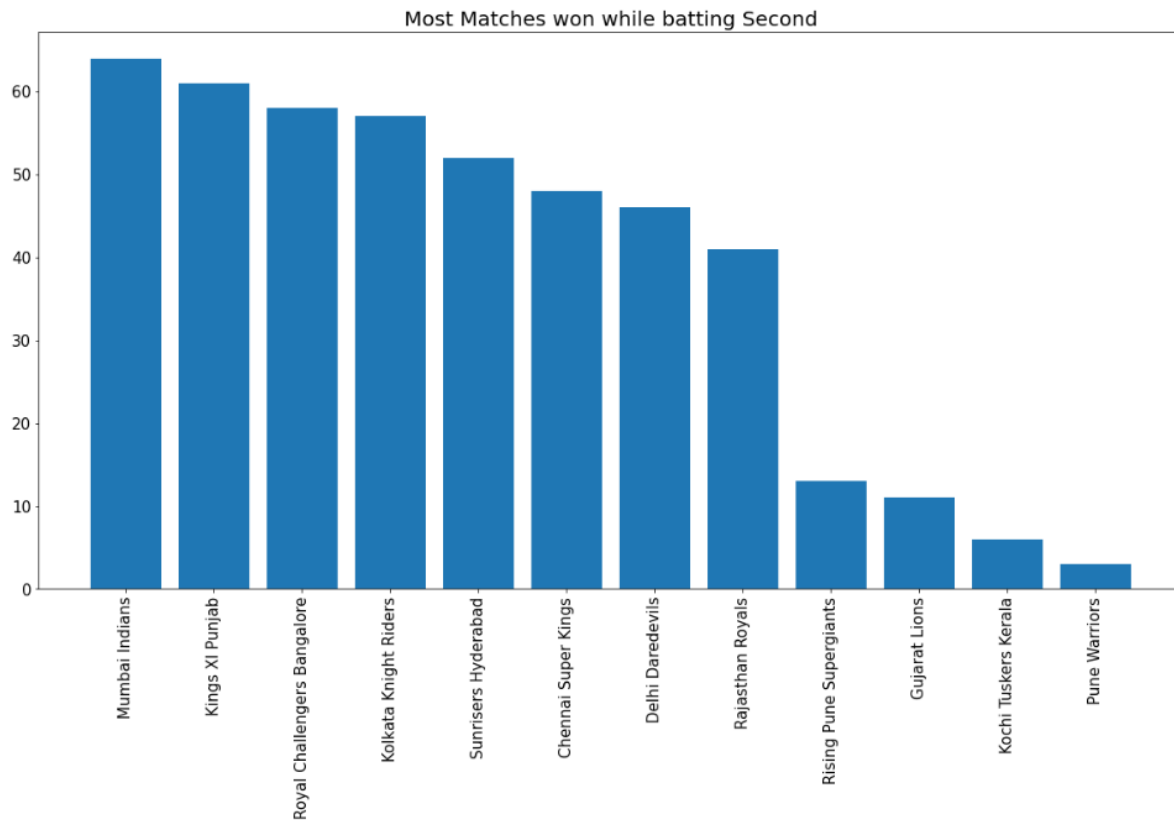
#8.To find the biggest victories in IPL while defending a total and while chasing a total.

	victory_margin	instances
1	6	85
2	7	80
3	5	71
4	8	54
5	4	41
6	9	37
7	3	18
8	10	11
9	2	6
10	1	3

#9. Which team won the most matches while batting first.



#10. Which team won the most matches while batting second.



#11. List of teams which have won matches by most runs cumulatively.

	winner	win_by_runs
0	Mumbai Indians	146
1	Royal Challengers Bangalore	144
2	Kolkata Knight Riders	140
3	Royal Challengers Bangalore	138
4	Royal Challengers Bangalore	130
5	Sunrisers Hyderabad	118
6	Kings XI Punjab	111
7	Rajasthan Royals	105
8	Mumbai Indians	102
9	Mumbai Indians	98

7 ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. The records of upcoming IPL seasons can be managed in a chronological order.
2. Any upcoming problem statements can be sorted out easily.
3. More efficient analysis
4. Time complexity is less as data have been processed multiple times.

DISADVANTAGES:

1. Acquires more space because results for different problems stores the attributes and constraints separately.
2. Any changes in teams or updation in teams may affect the original datasets.
3. Many duplicate values can affect the results.
4. Cleaning of data sets must be done in each updation to avoid false results and to increase the accuracy the results.

8 APPLICATIONS

1. It is used in Analysis of large datasets of IPL seasons
2. Accessing the statistics of any constraints is much easier.
3. Very useful to update the performance of a individual player or teams.
4. It is very useful business investors to invest their money in effective team by analysing their progress with help of this analysis.

9 CONCLUSION

In this study, the various factors that influence the outcome of an Indian Premier League matches were identified. The seven factors which significantly influence the result of an IPL match include the home team, the away team, the toss winner, toss decision, the stadium, and the respective teams' weight. Various classification-based machine learning algorithms were trained on the IPL dataset designed for this study. The dataset contained all the match data since the beginning of Indian Premier League till 2018. The accuracy of the MLP classifier would have improved further if the team weight was calculated immediately after the end of each match. Because this is the only way, the classifier gets fed with real-time performance of the

participating teams. The Twenty20 format of cricket carries a lot of randomness, because a single over can completely change the ongoing pace of the game. Indian Premier League is still at infancy stage, it is just a decade old league and has way less number of matches compared to test and one-day international formats. Hence, designing a machine learning model for predicting the match outcome of an auction-based Twenty20 format premier league with an accuracy of 72.66% and F1 score of 0.72 is highly satisfactory at this stage.

10 FUTURE SCOPE

In future many more enhancements can be made in this analysis. We may expect a IPL Predictor which is useful to acquire the results with the few known inputs, it can be achieved in several ways like web scrapping or machine trained bots which helps to predict the outcome by the analysed data sets. Web scrapping is also a technique which fetches the data from different websites and bounded into single data sets and processed. The processed data can be trained using machine language and the chat application bot can be created using the entities and constraints for the problem statements and the bot is trained to respond the requests of users expectation, This can be achieved in the upcoming days.

11 BIBLIOGRAPHY

1. <https://www.kaggle.com/nowke9/ipldata?select=matches.csv>
2. <https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=stories-get-started-dashboards>
3. Predicting Results of Indian Premier League T-20 Matches using Machine Learning (Author: Shilpi Agrawal , Suraj Pal Singh)
4. <https://www.analyticsvidhya.com/blog/2021/05/the-data-science-behind-ipl/>
5. Exploratory Data Analysis of IPL Matches-Part I | by Bipin P. | Towards Data Science
6. Predictive Analysis of an IPL Match | by Geet Pithadia | Towards Data Science

APPENDIX

Source Code:

#1.To find the team that won the most number of matches in the entire IPL.

```
most_no_of_wins = df.groupby('winner').apply(lambda x: x).reset_index()
most_no_of_wins = most_no_of_wins.groupby('winner').count()
most_no_of_wins = most_no_of_wins.city.reset_index(name='No_Of_Wins')
most_no_of_wins = most_no_of_wins.sort_values(by='No_Of_Wins',ascending=False)
```

2.To find the team that lost the most number of matches in the entire IPL.

```
# 1. Who played most matches
```

```
# 2. Who not in winner
```

```
# Kochi Tuskers Kerala Delhi Capitals
```

```
test_keys = dd.team1.unique()
```

```
lost = {test_keys[i]: 0 for i in range(len(test_keys))}
```

```
t1, t2, win = dd.team1, dd.team2, dd['winner']
```

```
for t1,t2, win in zip(t1,t2,win):
```

```
    if t1 == win:
```

```
        lost[t2] += 1
```

```
    elif t2 == win:
```

```
        lost[t1] += 1
```

```
team_lost = pd.DataFrame(lost.items(), columns={'Matches Lost', 'Team'})
```

```
team_lost.sort_values(['Matches Lost'], axis = 0, ascending = False, inplace = True, na_position = 'last')
```

```
team_lost.rename(columns={'Matches Lost':'Teams', 'Team':'Matches Lost'}, inplace=True)
```

```
team_lost.sort_values(['Matches Lost'], axis = 0, ascending = False, inplace = True, na_position = 'last')
```

3.Does winning a toss increase the chances of victory.

```
dd = df
```

```
win_count, team = 0, []
```

```
# counting the winner count
```

```
for index, value in df.iterrows():
```

```
    if(value['toss_winner']==value['winner']):    win_count += 1
```

```
print(f'The number of times the team winning toss have won: {win_count}')
```

```
prob = win_count/len(df)
```

```
print('The probability of winning if won the toss: {:.2f}' .format(prob))
```

