

काशी हिन्दू
विश्वविद्यालय



BANARAS HINDU
UNIVERSITY

Rajiv Gandhi South Campus, Mirzapur,
Uttar Pradesh

Minor Project Report

on

IPL MATCH WINNER PREDICTION USING MACHINE LEARNING

Submitted By: Vikash Kumar Mishra

Master Of Computer Application

(3rd sem 2021-23)

Roll No: 21430MCA022

Under Supervision of Dr. Manoj Kumar Mishra

(Guide and Course Coordinator)

Department of Computer Science, Institute of Science
BANARAS HINDU UNIVERSITY
VARANASI (U.P.)

Contents

1. Acknowledgement
2. Introduction
3. Machine Learning in Indian Premier League
4. The Proposed Work
5. Prediction of IPL in machine learning
6. Need for this project
7. Important codes and model training
8. UI Building
9. Output
10. Advantages
11. Disadvantages
12. Conclusion
13. Reference

Acknowledgement

I would like to express my special thanks and gratitude to my course coordinator and guide **Dr. Manoj Kumar Mishra Sir** who gave me the golden opportunity to do this wonderful project on the topic **IPL MATCH WINNER PREDICTION**, which also helped me in doing a lot of research and I came to know about so many new things. I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finishing this project within the limited time span.

Introduction

Machine learning and Data Science are one of the fastest-growing technological fields. This field results in amazing changes in the medical field, production, robotics etc. The main reason for the advancement in this field is the increase in the computational power and availability of large amounts of data. In Data Science, this data is analyzed and made suitable for creating machine learning models and products.

In today's article, we are going to discuss the IPL team win prediction. Based on some match stats, we're predicting who is the winner of an IPL match. Through this project, you will get familiar with the exploratory data analysis and feature engineering techniques that need to be applied to process data

Machine Learning in Indian Premier League

Some interesting machine-learning works have also been performed on data acquired from Indian Premier League matches. In a study , the Naive Bayesian classifier was used to classify the performance of all-rounder players (bowler plus batsman) into four various non-overlapping categories, viz., a performer, a batting all-rounder, a bowling all-rounder or an underperformer by being based on their strike rate and economy rate. Stepwise multinomial logistic regression (SMLR) was used to extract the essential predictors. When validated, the Naive Bayesian model was able to classify 66.7% of the all-rounders correctly. The same authors later published a work in which a Artificial Neural Network model was used to predict the performance of bowlers based on their performance in the first three seasons of IPL . When the predicted results were validated with actual performance of the players in season four, the developed ANN model had an accuracy of 71.43%.

Although not related to IPL, a study performed at University College London in the area of predicting the outcome of a Twenty20 match would be a healthy addition as the literature work in the Twenty20 domain. The study made use of Naive Bayes, Logistic Regression, Random Forests, Gradient Boosting algorithms to predict the outcome of English County cricket matches. Two models were developed, each was given input of two different sets of features. The team only related features were input to the first model, while team and players related features were input to the second model. The study was concluded with Naive Bayes outperforming all other algorithms with the first model giving out average prediction accuracy of 62.4% and second model giving average prediction accuracy of 63.9%, i.e 64% average accuracy with 2009-2014 data and 63.8% average accuracy with 2010-2014 data.

The Proposed Work

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, T-20 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 sports, most of the prediction job is done using regression or classification tasks, both of which come under supervised learning. In simple terms, it is a prediction model which is learned by the learning algorithm from a set of dataset. Based on the type of output supervised learning is divided further into two categories, regression, and classification. In Regression, the output is a continuous value; however, classification deals with discrete kind of output. For predicting continuous values, Linear Regression appeared to be quite effective, and for classification problems like predicting the outcome of matches or classifying players, learning algorithms like Naive Bayes, Logistic Regression, Neural Networks, Random Forests were found being used in most of the previous studies. 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. Regression 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.

Prediction of IPL in machine learning

Dataset The official website of Indian Premier League [35] was the primary source of data for this study. The data was scraped from the site and maintained in a Comma Separated Values (CSV) format. The initial dataset had many features including date, season, home team, away team, toss winner, man of the match, venue, umpires, referee, home team score, away team score, powerplay score, overs details when team reached milestone of multiple of 50 (i.e., 50 runs, 100 runs, 150 runs), playing 11 players, winner and won by details. In a single season, a team has to play with other teams on two occasions, i.e., once as a home team and next time as an away team. For example, once KKR plays with CSK in its home stadium (Eden Gardens) next time they play against CSK in their home stadium (M Chinnaswamy Stadium). So, while making the dataset, the concept of home team and away team was considered to prevent the redundancy. The Indian Premier League has just been 11 years old, which is why only 634 matches data were available after the pre-processing. This number is considerably less with comparison to the data available relating to the test or ODI formats. Due to certain difficulties with some ongoing team franchises, in some seasons the league has seen the participation of new teams, and some teams have discontinued. Presence of those inactive teams in the dataset was not really necessary, but if the matches data were omitted where the inactive teams appeared, the chances were that the valuable knowledge about the teams which were still active in the league would deteriorate. For better understanding and to make the dataset look somehow cluttered-free, acronyms were used for the teams. Table 1 lists the acronyms used in the dataset.

Need of the project and application

The need for IPL winning prediction using machine learning (ML) can be driven by several factors, including:

Increased Competition: With the popularity of IPL, the level of competition among teams has increased, making accurate predictions crucial for teams and players to gain a competitive edge.

Data-driven Decisions: The vast amounts of data generated by IPL matches can provide valuable insights into player and team performance, helping teams make informed decisions about roster changes, strategy, and more.

Fan Engagement: Fans of IPL are constantly seeking new ways to engage with the tournament and predictions can provide a new source of excitement and discussion.

Improved Understanding: ML can help analysts and fans better understand the factors that contribute to winning in IPL, leading to a deeper appreciation of the sport.

Business Opportunities: IPL is a highly lucrative tournament, and accurate predictions can be used to inform betting and fantasy sports decisions, providing business opportunities for individuals and companies.

Tools and Technology

OPERATING SYSTEM

- WINDOWS 11

TECHNOLOGY

- MACHINE LEARNING

TOOL

- PYTHON
- VS CODE EDITOR

LIBRARIES

- NUMPY
- PANDAS

PACKAGES

- STREAMLIT
- PICKLE

Important Codes and Model Training

In [2]:

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

In [1]:

```
import pandas as pd
import numpy as np
```

In [3]:

```
match.head()
```

Out[3]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result_m
0	335982	Bangalore	2008-04-18	BB McCullum	Chinnaswamy Stadium	M	0 Challengers Bangalore	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	Kolkata Knight Riders	runs	1
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	0 Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings	runs	
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	Delhi	0 Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils	wickets	
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	Mumbai Indians	0 Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore	wickets	
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	Deccan Chargers	0 Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders	wickets	

In [4]:

```
match.shape
```

Out[4]:

```
(816, 17)
```

In [5]:

```
delivery.head()
```

In [5]:

```
delivery.head()
```

Out[5]:

	id	inning	over	ball	batsman	non_striker	bowler	batsman_runs	extra_runs	total_runs	non_boundary	is_wicket	dismissal_kind	player_dismissed
0	335982	1	6	5	RT Ponting	BB McCullum	AA Noffke		1	0	1	0	0	NaN
1	335982	1	6	6	BB McCullum	RT Ponting	AA Noffke		1	0	1	0	0	NaN
2	335982	1	7	1	McCollum	RT Ponting	Z Khan		0	0	0	0	0	NaN
3	335982	1	7	2	BB McCullum	RT Ponting	Z Khan		1	0	1	0	0	NaN
4	335982	1	7	3	RT Ponting	BB McCullum	Z Khan		1	0	1	0	0	NaN

In [6]:

```
total_score_df = delivery.groupby(['id','inning']).sum()['total_runs'].reset_index()
```

C:\Users\mishr\AppData\Local\Temp\ipykernel_2436\2286552851.py:: FutureWarning: The default value of numeric_only in DataFrame GroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

total_score_df = delivery.groupby(['id', 'inning']).sum()['total_runs'].reset_index()

In [7]:

```
total_score_df = total_score_df[total_score_df['inning']==1]
```

In [8]:

```
total_score_df
```

Out[8]:

	id	inning	total_runs
0	335982	1	222

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint Last Saturday at 3:27 PM Autosave Failed!

In [8]: total_score_df

Out[8]:

	id	inning	total_runs
0	335982	1	222
2	335983	1	240
4	335984	1	129
6	335985	1	165
8	335986	1	110
...
1620	1216547	1	201
1622	1237177	1	200
1624	1237178	1	131
1626	1237180	1	189
1628	1237181	1	156

816 rows × 3 columns

In [9]: match_df=match.merge(total_score_df[['id','total_runs']],left_on='id',right_on='id')

In [10]: match_df

Out[10]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner	result	result
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium		0 Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders	runs	runs
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium		0 Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings	runs	runs

816 rows × 14 columns

84°F Haze

ENG IN 07-02-2023 15:53

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint Last Saturday at 3:27 PM Autosave Failed!

In [11]: match_df['team1'].unique()

Out[11]: array(['Royal Challengers Bangalore', 'Kings XI Punjab', 'Delhi Daredevils', 'Mumbai Indians', 'Kolkata Knight Riders', 'Rajasthan Royals', 'Deccan Chargers', 'Chennai Super Kings', 'Kochi Tuskers Kerala', 'Pune Warriors', 'Sunrisers Hyderabad', 'Gujarat Lions', 'Rising Pune Supergiants', 'Rising Pune Supergiant', 'Delhi Capitals'], dtype=object)

In [12]: teams = [
 'Sunrisers Hyderabad',
 'Mumbai Indians',
 'Royal Challengers Bangalore',
 'Kolkata Knight Riders',
 'Kings XI Punjab',
 'Chennai Super Kings',
 'Rajasthan Royals',
 'Delhi Capitals']

In [13]: 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')

In [14]: match_df = match_df[match_df['team1'].isin(teams)]
match_df = match_df[match_df['team2'].isin(teams)]

In [15]: match_df.shape

Out[15]: (701, 18)

In [16]: match_df

Out[16]:

84°F Haze

ENG IN 07-02-2023 15:53

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Trusted Python 3 (ipykernel)

In [17]: `match_df = match_df[['id','city','winner','total_runs']]`

In [18]: `match_df`

Out[18]:

	id	city	winner	total_runs
0	335982	Bangalore	Kolkata Knight Riders	222
1	335983	Chandigarh	Chennai Super Kings	240
2	335984	Delhi	Delhi Daredevils	129
3	335985	Mumbai	Royal Challengers Bangalore	165
4	335986	Kolkata	Kolkata Knight Riders	110
...
811	1216547	Dubai	Royal Challengers Bangalore	201
812	1237177	Dubai	Mumbai Indians	200
813	1237178	Abu Dhabi	Sunrisers Hyderabad	131
814	1237180	Abu Dhabi	Delhi Capitals	189
815	1237181	Dubai	Mumbai Indians	156

701 rows × 4 columns

In [19]: `delivery_df = match_df.merge(delivery, on='id')`

In [20]: `delivery_df = delivery_df[delivery_df['inning'] == 2]`

In [21]: `delivery_df`

Out[21]:

	id	city	winner	total_runs_x	inning	over	ball	batsman	non_striker	bowler	... extra_runs	total_runs_y	non_boundary	is_wicket	dismissal_kind
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	0	0	0	0

84°F Haze

Search Google Chrome Microsoft Edge File Explorer Task View Taskbar 15:53 07-02-2023 ENG IN

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed! Logout

File Edit View Insert Cell Kernel Widgets Help Not Connected error Trusted Python 3 (ipykernel)

In [22]: `delivery_df['current_score'] = delivery_df.groupby('id').cumsum()['total_runs_y']`

C:\Users\mishr\AppData\Local\Temp\ipykernel_2436\1007580117.py:1: FutureWarning: The default value of numeric_only in DataFrame GroupBy.cumsum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
delivery_df['current_score'] = delivery_df.groupby('id').cumsum()['total_runs_y']

In [23]: `delivery_df`

Out[23]:

	id	city	winner	total_runs_x	inning	over	ball	batsman	non_striker	bowler	... extra_runs	total_runs_y	non_boundary	is_wicket	dismissal_kind
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	0	0	0	Nan
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	1	0	0	Nan
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	Boucher	CL White	AB Agarkar	...	1	0	0	Nan
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	Boucher	CL White	AB Agarkar	...	1	0	0	Nan
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	Boucher	CL White	AB Agarkar	...	0	0	0	Nan
...
166366	1237181	Dubai	Mumbai Indians	156	2	17	6	HH Pandya	Ishan Kishan	K Rabada	...	1	0	0	Nan
166367	1237181	Dubai	Mumbai Indians	156	2	18	1	HH Pandya	Ishan Kishan	A Nortje	...	1	0	0	Nan
166368	1237181	Dubai	Mumbai Indians	156	2	18	2	Ishan Kishan	HH Pandya	A Nortje	...	1	0	0	Nan
166369	1237181	Dubai	Mumbai Indians	156	2	18	3	HH Pandya	Ishan Kishan	A Nortje	...	0	0	1	caught

84°F Haze

Search Google Chrome Microsoft Edge File Explorer Task View Taskbar 15:53 07-02-2023 ENG IN

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed! Logout

In [24]: `delivery_df['runs_left'] = delivery_df['total_runs_x'] - delivery_df['current_score']`

In [25]: `delivery_df`

Out[25]:

	id	city	winner	total_runs_x	inning	over	ball	batsman	non_striker	bowler	... non_boundary	is_wicket	dismissal_kind	player_dismissed
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	0	0	Nan
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	0	0	Nan
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	MV Boucher	CL White	AB Agarkar	...	0	0	Nan
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	MV Boucher	CL White	AB Agarkar	...	0	0	Nan
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	MV Boucher	CL White	AB Agarkar	...	0	0	Nan
...	
166366	1237181	Dubai	Mumbai Indians	156	2	17	6	HH Pandya	Ishan Kishan	K Rabada	...	0	0	Nan
166367	1237181	Dubai	Mumbai Indians	156	2	18	1	HH Pandya	Ishan Kishan	A Nortje	...	0	0	Nan
166368	1237181	Dubai	Mumbai Indians	156	2	18	2	Ishan Kishan	HH Pandya	A Nortje	...	0	0	Nan
166369	1237181	Dubai	Mumbai Indians	156	2	18	3	HH Pandya	Ishan Kishan	A Nortje	...	0	1	caught HH Pandya
166370	1237181	Dubai	Mumbai Indians	156	2	18	4	HH Pandya	Ishan Kishan	A Nortje	...	0	0	Nan

84°F Haze 15:54 ENG IN 07-02-2023

chatgpt - Search | IPL Winning Prediction ML Adv | IPL Team Win Prediction Project | Untitled - Jupyter Notebook | Untitled - Jupyter Notebook

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed! Logout

In [26]: `delivery_df['balls_left'] = 126 - (delivery_df['over']*6 + delivery_df['ball'])`

In [27]: `delivery_df`

Out[27]:

	id	city	winner	total_runs_x	inning	over	ball	batsman	non_striker	bowler	... is_wicket	dismissal_kind	player_dismissed	fielder	
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	0	Nan	Nan	Nan
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	0	Nan	Nan	Nan
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	MV Boucher	CL White	AB Agarkar	...	0	Nan	Nan	Nan
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	MV Boucher	CL White	AB Agarkar	...	0	Nan	Nan	Nan
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	MV Boucher	CL White	AB Agarkar	...	0	Nan	Nan	Nan
...	
166366	1237181	Dubai	Mumbai Indians	156	2	17	6	HH Pandya	Ishan Kishan	K Rabada	...	0	Nan	Nan	Nan
166367	1237181	Dubai	Mumbai Indians	156	2	18	1	HH Pandya	Ishan Kishan	A Nortje	...	0	Nan	Nan	Nan
166368	1237181	Dubai	Mumbai Indians	156	2	18	2	Ishan Kishan	HH Pandya	A Nortje	...	0	Nan	Nan	Nan
166369	1237181	Dubai	Mumbai Indians	156	2	18	3	HH Pandya	Ishan Kishan	A Nortje	...	1	caught HH Pandya	AM Rahane	Nan
166370	1237181	Dubai	Mumbai Indians	156	2	18	4	HH Pandya	Ishan Kishan	A Nortje	...	0	Nan	Nan	Nan

80341 rows × 24 columns 15:54 ENG IN 07-02-2023

In [28]:

```
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0")
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x if x == "0" else "1")
wickets = delivery_df.groupby('id').cumsum()['player_dismissed'].values
delivery_df['wickets'] = 10 - wickets
delivery_df.head()
```

C:\Users\mishr\AppData\Local\Temp\ipykernel_2436\22157124.py:4: FutureWarning: The default value of numeric_only in DataFrameGroupBy.cumsum is deprecated. In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

wickets = delivery_df.groupby('id').cumsum()['player_dismissed'].values

Out[28]:

	id	city	winner	total_runs_x	inning	over	ball	bowler	non_striker	dismissal_kind	player_dismissed	fielder	extras_type	battis	
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	NaN	0	NaN	NaN Ch B
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	NaN	0	NaN	NaN Ch B
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	Boucher	MV CL White	AB Agarkar	...	NaN	0	NaN	wides Ch B
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	Boucher	CL White	AB Agarkar	...	NaN	0	NaN	wides Ch B
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	Boucher	MV CL White	AB Agarkar	...	NaN	0	NaN	NaN Ch B

5 rows × 25 columns

In [29]:

```
delivery_df.tail()
```

Out[29]:

	id	city	winner	total_runs_x	inning	over	ball	bowler	non_striker	dismissal_kind	player_dismissed	fielder	extras_type	battis
--	----	------	--------	--------------	--------	------	------	--------	-------------	----------------	------------------	---------	-------------	--------

In [30]:

```
# crr = runs/overs
delivery_df['crn'] = (delivery_df['current_score']*6)/(120 - delivery_df['balls_left'])
```

In [31]:

```
delivery_df['rrr'] = (delivery_df['runs_left']*6)/delivery_df['balls_left']
```

In [32]:

```
delivery_df
```

Out[32]:

	id	city	winner	total_runs_x	inning	over	ball	bowler	non_striker	fielder	extras_type	battting_team	bowling_team	curr	
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	NaN	NaN	Royal Challengers Bangalore	Kolkata Knight Riders
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	NaN	NaN	Royal Challengers Bangalore	Kolkata Knight Riders
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	Boucher	CL White	AB Agarkar	...	NaN	wides	Royal Challengers Bangalore	Kolkata Knight Riders
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	Boucher	CL White	AB Agarkar	...	NaN	wides	Royal Challengers Bangalore	Kolkata Knight Riders
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	Boucher	CL White	AB Agarkar	...	NaN	NaN	Royal Challengers Bangalore	Kolkata Knight Riders
...	
166366	1237181	Dubai	Mumbai Indians	156	2	17	6	HH Pandya	Ishan Kishan	K Rabada	...	NaN	NaN	Mumbai Indians	Delhi Capitals
166367	1237181	Dubai	Mumbai Indians	156	2	18	1	HH Pandya	Ishan Kishan	A Nortje	...	NaN	NaN	Mumbai Indians	Delhi Capitals
166368	1237181	Dubai	Mumbai Indians	156	2	18	2	Ishan Kishan	HH Pandya	A Nortje	...	NaN	NaN	Mumbai Indians	Delhi Capitals
166369	1237181	Dubai	Mumbai Indians	156	2	18	3	HH Pandya	Ishan Kishan	A Nortje	...	AM Rahane	NaN	Mumbai Indians	Delhi Capitals

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

```
In [33]: def result(row):
    return 1 if row['batting_team'] == row['winner'] else 0

In [34]: delivery_df['result'] = delivery_df.apply(result,axis=1)

In [35]: delivery_df
```

Out[35]:

	id	city	winner	total_runs_x	inning	over	ball	batsman	non_striker	bowler	... extras_type	batting_team	bowling_team	current_score	
121	335982	Bangalore	Kolkata Knight Riders	222	2	6	2	CL White	MV Boucher	AB Agarkar	...	NaN	Royal Challengers Bangalore	Kolkata Knight Riders	0
122	335982	Bangalore	Kolkata Knight Riders	222	2	6	3	CL White	MV Boucher	AB Agarkar	...	NaN	Royal Challengers Bangalore	Kolkata Knight Riders	1
123	335982	Bangalore	Kolkata Knight Riders	222	2	6	4	MV Boucher	CL White	AB Agarkar	...	wides	Royal Challengers Bangalore	Kolkata Knight Riders	2
124	335982	Bangalore	Kolkata Knight Riders	222	2	6	5	MV Boucher	CL White	AB Agarkar	...	wides	Royal Challengers Bangalore	Kolkata Knight Riders	3
125	335982	Bangalore	Kolkata Knight Riders	222	2	6	6	MV Boucher	CL White	AB Agarkar	...	NaN	Royal Challengers Bangalore	Kolkata Knight Riders	3
...	
166366	1237181	Dubai	Mumbai Indians	156	2	17	6	HH Pandya	Ishan Kishan	K Rabada	...	NaN	Mumbai Indians	Delhi Capitals	154
166367	1237181	Dubai	Mumbai Indians	156	2	18	1	HH Pandya	Ishan Kishan	A Nortje	...	NaN	Mumbai Indians	Delhi Capitals	155
166368	1237181	Dubai	Mumbai Indians	156	2	18	2	Ishan Kishan	HH Pandya	A Nortje	...	NaN	Mumbai Indians	Delhi Capitals	156
166369	1237181	Dubai	Mumbai Indians	156	2	18	3	HH Pandya	Ishan Kishan	A Nortje	...	NaN	Mumbai Indians	Delhi Capitals	156

jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

```
In [36]: final_df = delivery_df[['batting_team','bowling_team','city','runs_left','balls_left','wickets','total_runs_x','crr','rrr','result']]

In [37]: final_df
```

Out[37]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	rrr	result
121	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	222	88	10	222	0.000000	15.136364	0
122	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	221	87	10	222	0.181618	15.241379	0
123	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	220	86	10	222	0.352941	15.348837	0
124	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	219	85	10	222	0.514286	15.456824	0
125	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	219	84	10	222	0.500000	15.642857	0
...
166366	Mumbai Indians	Delhi Capitals	Dubai	2	18	6	156	9.058824	0.666667	1
166367	Mumbai Indians	Delhi Capitals	Dubai	1	17	6	156	9.029126	0.352941	1
166368	Mumbai Indians	Delhi Capitals	Dubai	0	16	6	156	9.000000	0.000000	1
166369	Mumbai Indians	Delhi Capitals	Dubai	0	15	5	156	8.914286	0.000000	1
166370	Mumbai Indians	Delhi Capitals	Dubai	-1	14	5	156	8.886792	-0.428571	1

80341 rows × 10 columns

```
In [38]: final_df = final_df.sample(final_df.shape[0])

In [39]: final_df.sample()
```

Out[39]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	rrr	result
161879	Delhi Capitals	Sunrisers Hyderabad	Abu Dhabi	27	40	3	162	10.125	4.05	0

Jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

```
In [40]: final_df.dropna(inplace=True)

In [41]: final_df = final_df[final_df['balls_left'] != 120]

In [42]: X = final_df.iloc[:, :-1]
y = final_df.iloc[:, -1]
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)

In [43]: X_train
```

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	rr
99569	Kings XI Punjab	Rajasthan Royals	Ahmedabad	36	62	4	191	16.034483	3.483871
32660	Royal Challengers Bangalore	Delhi Daredevils	Bangalore	24	62	1	183	16.448276	2.322581
15567	Rajasthan Royals	Sunrisers Hyderabad	Dubai	155	43	10	158	0.233769	21.627907
120421	Kolkata Knight Riders	Sunrisers Hyderabad	Hyderabad	197	112	9	209	9.000000	10.553571
158606	Royal Challengers Bangalore	Delhi Capitals	Dubai	140	38	8	196	4.097561	22.105263
...
58313	Mumbai Indians	Kings XI Punjab	Chandigarh	108	30	9	168	4.000000	21.600000
59118	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata	50	89	4	190	27.096774	3.370787
21218	Delhi Daredevils	Mumbai Indians	East London	-1	53	7	116	10.477612	-0.113208
115482	Royal Challengers Bangalore	Delhi Daredevils	Raipur	128	46	10	138	0.810811	16.695652
113542	Mumbai Indians	Royal Challengers Bangalore	Bangalore	41	80	6	151	16.500000	3.075000

62501 rows × 9 columns

```
In [44]: from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

Jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

```
In [44]: from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

trf = ColumnTransformer([
    ('trf', OneHotEncoder(sparse=False, drop='first'), ['batting_team', 'bowling_team', 'city']),
], remainder='passthrough')

In [45]: from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline

In [46]: pipe = Pipeline(steps=[
    ('step1', trf),
    ('step2', LogisticRegression(solver='liblinear'))
])

In [47]: pipe.fit(X_train, y_train)
```

C:\Users\mishr\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\preprocessing_encoders.py:828: FutureWarning: 'sparse' was renamed to 'sparse_output' in version 1.2 and will be removed in 1.4. 'sparse_output' is ignored unless you leave 'sparse' to its default value.
warnings.warn(
Pipeline
|> step1: ColumnTransformer
|> trf
|> remainder
|> OneHotEncoder
|> passthrough
|> LogisticRegression

```
In [47]: pipe
```

Jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

In [48]: `X_train.describe()`

Out[48]:

	runs_left	balls_left	wickets	total_runs_x	crr	rrr
count	62501.000000	62501.000000	62501.000000	62501.000000	62501.000000	62501.000000
mean	89.427945	68.48812	7.147166	164.952673	12.120475	11.25773
std	50.678017	33.21298	2.446635	28.808257	67.577181	13.885757
min	-16.000000	3.00000	0.000000	62.000000	-1068.000000	-4.000000
25%	49.000000	41.00000	6.000000	147.000000	4.173913	4.325581
50%	89.000000	69.00000	8.000000	165.000000	8.000000	8.114286
75%	128.000000	97.00000	9.000000	184.000000	14.368421	13.09900
max	246.000000	125.00000	10.000000	246.000000	1062.000000	396.000000

In [49]: `y_pred = pipe.predict(X_test)`

In [50]: `from sklearn.metrics import accuracy_score`
`accuracy_score(y_test,y_pred)`

Out[50]: 0.812811980032779

In [51]: `pipe.predict_proba(X_test)[126]`

Out[51]: array([0.95963647, 0.04036353])

In [52]: `teams`

Out[52]: ['Sunrisers Hyderabad', 'Mumbai Indians', 'Royal Challengers Bangalore', 'Kolkata Knight Riders', 'Kings XI Punjab', 'Chennai Super Kings', 'Rajasthan Royals', 'Delhi Capitals']

84°F Haze 15:54 07-02-2023

Jupyter Untitled Last Checkpoint: Last Saturday at 3:27 PM Autosave Failed!

In [51]: `pipe.predict_proba(X_test)[126]`

Out[51]: array([0.95963647, 0.04036353])

In [52]: `teams`

Out[52]: ['Sunrisers Hyderabad', 'Mumbai Indians', 'Royal Challengers Bangalore', 'Kolkata Knight Riders', 'Kings XI Punjab', 'Chennai Super Kings', 'Rajasthan Royals', 'Delhi Capitals']

In [53]: `delivery_df['city'].unique()`

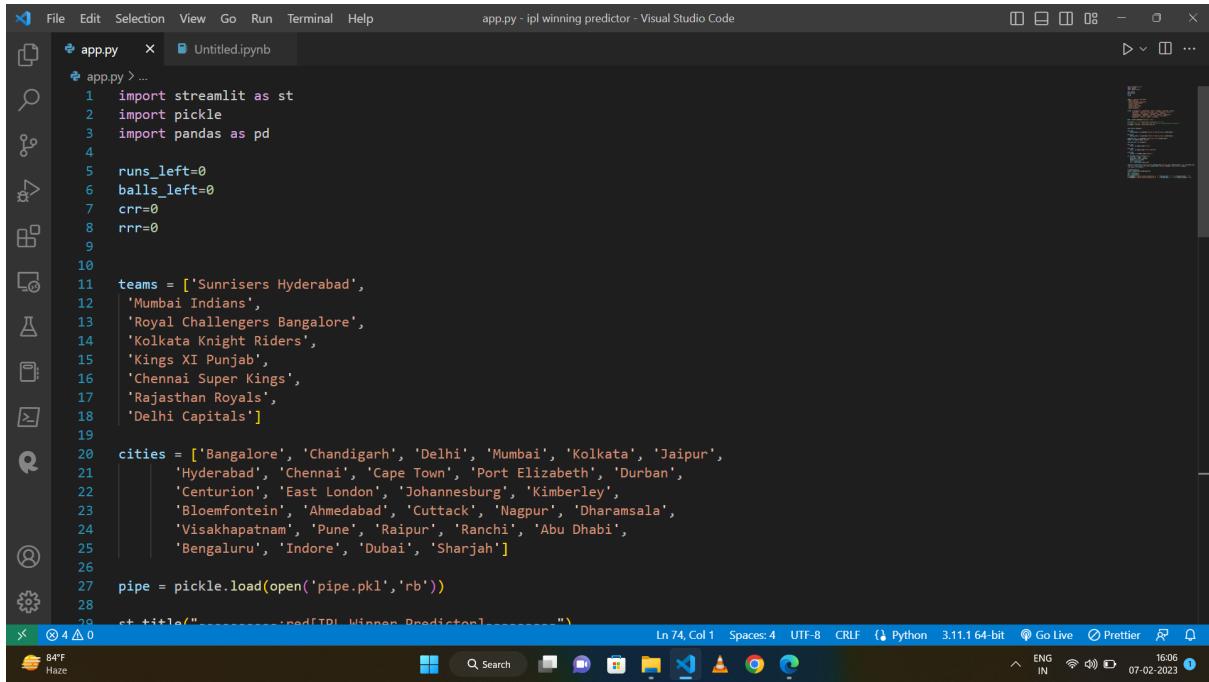
Out[53]: array(['Bangalore', 'Chandigarh', 'Delhi', 'Mumbai', 'Kolkata', 'Jaipur', 'Hyderabad', 'Chennai', 'Cape Town', 'Port Elizabeth', 'Durban', 'Centurion', 'East London', 'Johannesburg', 'Kimberley', 'Bloemfontein', 'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala', 'Visakhapatnam', 'Pune', 'Raipur', 'Ranchi', 'Abu Dhabi', 'nain', 'Bengaluru', 'Indore', 'Dubai', 'Sharjah'], dtype=object)

In [54]: `import pickle`
`pickle.dump(pipe,open('pipe.pkl','wb'))`

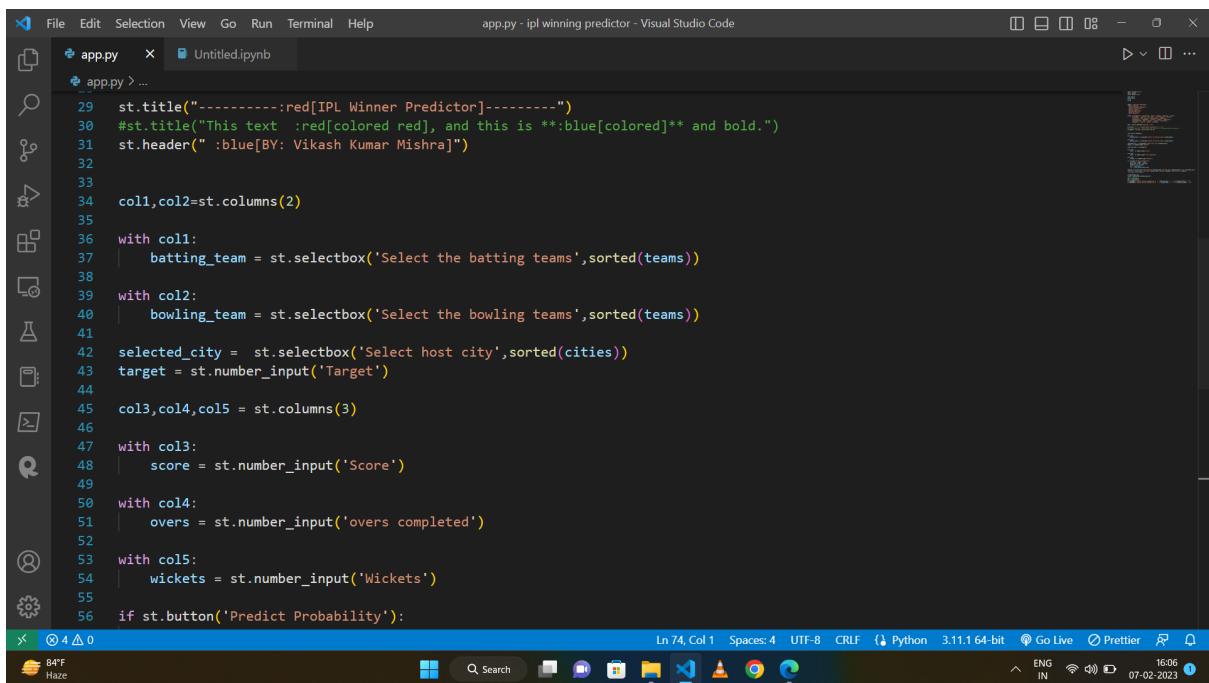
In []:

84°F Haze 15:54 07-02-2023

UI Building



```
app.py
1 import streamlit as st
2 import pickle
3 import pandas as pd
4
5 runs_left=0
6 balls_left=0
7 crr=0
8 rrr=0
9
10
11 teams = ['Sunrisers Hyderabad',
12 'Mumbai Indians',
13 'Royal Challengers Bangalore',
14 'Kolkata Knight Riders',
15 'Kings XI Punjab',
16 'Chennai Super Kings',
17 'Rajasthan Royals',
18 'Delhi Capitals']
19
20 cities = ['Bangalore', 'Chandigarh', 'Delhi', 'Mumbai', 'Kolkata', 'Jaipur',
21 'Hyderabad', 'Chennai', 'Cape Town', 'Port Elizabeth', 'Durban',
22 'Centurion', 'East London', 'Johannesburg', 'Kimberley',
23 'Bloemfontein', 'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala',
24 'Visakhapatnam', 'Pune', 'Raipur', 'Ranchi', 'Abu Dhabi',
25 'Bengaluru', 'Indore', 'Dubai', 'Sharjah']
26
27 pipe = pickle.load(open('pipe.pkl','rb'))
28
29 st.title("-----:red[IPL Winner Predictor]-----")
30 #st.title(":red[This text :red[colored red], and this is **:blue[colored]** and bold.]")
31 st.header(" :blue[BY: Vikash Kumar Mishra]")
32
33
34 col1,col2=st.columns(2)
35
36 with col1:
37     batting_team = st.selectbox('Select the batting teams',sorted(teams))
38
39 with col2:
40     bowling_team = st.selectbox('Select the bowling teams',sorted(teams))
41
42 selected_city = st.selectbox('Select host city',sorted(cities))
43 target = st.number_input('Target')
44
45 col3,col4,col5 = st.columns(3)
46
47 with col3:
48     score = st.number_input('Score')
49
50 with col4:
51     overs = st.number_input('overs completed')
52
53 with col5:
54     wickets = st.number_input('Wickets')
55
56 if st.button('Predict Probability'):
```



```
app.py
1 import streamlit as st
2 import pickle
3 import pandas as pd
4
5 runs_left=0
6 balls_left=0
7 crr=0
8 rrr=0
9
10
11 teams = ['Sunrisers Hyderabad',
12 'Mumbai Indians',
13 'Royal Challengers Bangalore',
14 'Kolkata Knight Riders',
15 'Kings XI Punjab',
16 'Chennai Super Kings',
17 'Rajasthan Royals',
18 'Delhi Capitals']
19
20 cities = ['Bangalore', 'Chandigarh', 'Delhi', 'Mumbai', 'Kolkata', 'Jaipur',
21 'Hyderabad', 'Chennai', 'Cape Town', 'Port Elizabeth', 'Durban',
22 'Centurion', 'East London', 'Johannesburg', 'Kimberley',
23 'Bloemfontein', 'Ahmedabad', 'Cutack', 'Nagpur', 'Dharamsala',
24 'Visakhapatnam', 'Pune', 'Raipur', 'Ranchi', 'Abu Dhabi',
25 'Bengaluru', 'Indore', 'Dubai', 'Sharjah']
26
27 pipe = pickle.load(open('pipe.pkl','rb'))
28
29 st.title("-----:red[IPL Winner Predictor]-----")
30 #st.title(":red[This text :red[colored red], and this is **:blue[colored]** and bold.]")
31 st.header(" :blue[BY: Vikash Kumar Mishra]")
32
33
34 col1,col2=st.columns(2)
35
36 with col1:
37     batting_team = st.selectbox('Select the batting teams',sorted(teams))
38
39 with col2:
40     bowling_team = st.selectbox('Select the bowling teams',sorted(teams))
41
42 selected_city = st.selectbox('Select host city',sorted(cities))
43 target = st.number_input('Target')
44
45 col3,col4,col5 = st.columns(3)
46
47 with col3:
48     score = st.number_input('Score')
49
50 with col4:
51     overs = st.number_input('overs completed')
52
53 with col5:
54     wickets = st.number_input('Wickets')
55
56 if st.button('Predict Probability'):
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** app.py - ipl winning predictor - Visual Studio Code
- Left Sidebar:** Includes icons for file operations like Open, Save, Find, Replace, and a search bar.
- Code Editor:** The main area contains Python code for an "ipl winning predictor". The code includes imports, function definitions, and logic for calculating winning probabilities based on team details and current match state.
- Right Sidebar:** Shows a preview of the application's output, which includes two subheaders and some placeholder text.
- Bottom Status Bar:** Ln 74, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python 3.11.1 64-bit | Go Live | Prettier | 16:06 | 07-02-2023
- Taskbar:** Shows various open applications including a browser, file explorer, and terminal.

```
app.py > ...
55 if st.button('Predict Probability'):
56     runs_left = target - score
57     balls_left = 120 - (overs*6)
58     wickets = 10 - wickets
59     crr = score/overs
60     rrr = (runs_left*6)/balls_left
61
62     input_df = pd.DataFrame({'batting_team':[batting_team], 'bowling_team':[bowling_team], 'city':[selected_city],
63                             'runs_left':[runs_left], 'balls_left':[balls_left], 'wickets':[wickets], 'total_runs_x':[target],
64                             'crr':[crr], 'rrr':[rrr]})[0]
65
66     st.table(input_df)
67     result = pipe.predict_proba(input_df)
68     #st.text(result)
69     loss = result[0][0]
70     win = result[0][1]
71     st.subheader(":red[The winning probability of ] "+batting_team + "- " + str(round(win*100)) + "%")
72     st.subheader(":red[The winning probability of ] "+bowling_team + "- " + str(round(loss*100)) + "%")
```

OUTPUT

The screenshot shows a Streamlit application titled "IPL Winner Predictor" running on localhost:8501. The interface is dark-themed with red and white text.

Inputs:

- Select the batting teams: Chennai Super Kings
- Select the bowling teams: Mumbai Indians
- Select host city: Mumbai
- Target: 196.00
- Score: 142.00
- overs completed: 16.00
- Wickets: 2.00

Buttons:

- Predict Probability (highlighted with a red border)

Data Table:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	rrr
0	Chennai Super Kings	Mumbai Indians	Mumbai	54.0000	24.0000	8.0000	196.0000	8.8750	13.5000

Output:

The winning probability of Chennai Super Kings- 78%

The winning probability of Mumbai Indians- 22%

Advantages

There are several advantages to using machine learning for IPL winning prediction, including:

Improved Accuracy: ML algorithms can analyze large amounts of data and find patterns that might be missed by human analysts, leading to more accurate predictions.

Automation: Predictive models can be automated, reducing the time and effort required for manual analysis and making the process more efficient.

Data-driven Insights: ML models can provide valuable insights into the factors that contribute to winning in IPL, allowing teams and players to make informed decisions and improve performance.

Versatility: ML models can be applied to different aspects of IPL, such as player performance, team strategy, and match outcomes, making them a versatile tool for analysis and prediction.

Scalability: ML models can be easily scaled to handle large amounts of data, making them suitable for use in large-scale sports tournaments like IPL.

Disadvantages

There are also several disadvantages to using machine learning for IPL winning prediction, including:

Limited Understanding: ML algorithms are black boxes and it can be difficult to understand how they arrived at a particular prediction, making it challenging to validate their accuracy.

Bias in Data: ML models can only be as good as the data they are trained on, and if the training data is biased, the model may produce inaccurate or unfair predictions.

Overfitting: ML models can easily overfit the training data, leading to poor performance on new or unseen data.

Complexity: ML models can be complex and difficult to develop, requiring specialized knowledge and expertise to implement effectively.

Dependence on Data: ML models rely heavily on large amounts of data, and may not work well if the data is limited or unavailable.

Contextual Limitations: ML models may not take into account important contextual factors that can affect the outcome of a match, leading to incorrect predictions.

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. A multivariate regression based model was formulated to calculate the points earned by each player based on their past performances which include (i) number of wickets taken, (ii) number of dot balls given, (iii) number of fours hit, (iv) number of sixes hit, (v) number of catches, and (vi) number of stumpings. The points awarded to each player was used to compute the relative strength of each team. 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 2008-21. The trained models were used to predict the outcome of each IPL match, 15 minutes before the game-play, immediately after the toss. The Multilayer perceptron classifier outperformed other classifiers by correctly predicting 43 out of 60, Indian Premier League matches. 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 T-20 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 its infancy stage, it is just a decade old league and has way less number of matches compared to test and one-day international formats.

References

1. Books

- Han, jiawei. Data mining: concepts and techniques/ jiawei Han, Michelin Kamber, Jian Pei.-3rd ed.
- C. M. Bishop, Pattern Recognition and Machine Learning, Springer.
- I. H. Witten et al., Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publisher.

2. Websites

- <http://www.javatpoint.com>
- <http://www.geeksforgeeks.org>
- <http://www.wikipedia.co.in>
- <http://www.stack-overflow.com>