In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
```

In [2]:

```python
ball_by_ball = pd.read_csv('./Data/IPL_Ball_by_Ball_2008_2022.csv')
matches_result = pd.read_csv('./Data/IPL_Matches_Result_2008_2022.csv')

ipl_2023_teams = pd.read_csv('./Data/Ipl_2023 _cricketers - Team name.csv')
ipl_2023_teams = ipl_2023_teams.rename(columns={'Teams': 'team'})

ipl_2023_venues = pd.read_csv('./Data/Ipl_2023 _cricketers - Venue.csv')
ipl_2023_venues = ipl_2023_venues.rename(columns={'Venue': 'venue'})

# ipl_2023_players = pd.read_csv('./Data/Ipl_2023 _cricketers - Players.csv')
# ipl_2023_players.drop('Team ', axis=1, inplace=True)
```

Preprocessing

- Change column names, drop unnecessary columns [in ball_by_ball, matches_result]

In [3]:

```python
ball_by_ball_orig = ball_by_ball
ball_by_ball = ball_by_ball.rename(columns={
    'ID': 'match_id',
    'ballnumber': 'ball_number',
    'non-striker': 'non_striker',
    'BattingTeam': 'batting_team',
})
ball_by_ball = ball_by_ball.loc[:, [
    'match_id',
    'innings',
    'batting_team',
    'overs',
    'ball_number',
    'batter',
    'bowler',
    'total_run',
]]
```

In [4]:

```python
matches_result_orig = matches_result
matches_result = matches_result.rename(columns={
    'ID': 'match_id',
    'Team1': 'team_1',
    'Team2': 'team_2',
    'Venue': 'venue',
})
matches_result = matches_result.loc[:, [
    'match_id',
    'team_1',
    'team_2',
    'venue',
]]
```

In [5]:

```
ball_by_ball_orig.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 225954 entries, 0 to 225953
Data columns (total 17 columns):
 #   Column             Non-Null Count   Dtype 
---  ------             --------------   ----- 
 0   ID                 225954 non-null  int64 
 1   innings            225954 non-null  int64 
 2   overs              225954 non-null  int64 
 3   ballnumber         225954 non-null  int64 
 4   batter             225954 non-null  object
 5   bowler             225954 non-null  object
 6   non-striker        225954 non-null  object
 7   extra_type         12049 non-null   object
 8   batsman_run        225954 non-null  int64 
 9   extras_run         225954 non-null  int64 
 10  total_run          225954 non-null  int64 
 11  non_boundary       225954 non-null  int64 
 12  isWicketDelivery   225954 non-null  int64 
 13  player_out         11151 non-null   object
 14  kind               11151 non-null   object
 15  fielders_involved  7988 non-null    object
 16  BattingTeam        225954 non-null  object
dtypes: int64(9), object(8)
memory usage: 29.3+ MB
```

In [6]:

```
ball_by_ball.head()
```

Out[6]:

	match_id	innings	batting_team	overs	ball_number	batter	bowler	total_run
0	1312200	1	Rajasthan Royals	0	1	YBK Jaiswal	Mohammed Shami	0
1	1312200	1	Rajasthan Royals	0	2	YBK Jaiswal	Mohammed Shami	1
2	1312200	1	Rajasthan Royals	0	3	JC Buttler	Mohammed Shami	1
3	1312200	1	Rajasthan Royals	0	4	YBK Jaiswal	Mohammed Shami	0
4	1312200	1	Rajasthan Royals	0	5	YBK Jaiswal	Mohammed Shami	0

In [7]:

```
matches_result_orig.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 950 entries, 0 to 949
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ID               950 non-null    int64  
 1   City             899 non-null    object 
 2   Date             950 non-null    object 
 3   Season           950 non-null    object 
 4   MatchNumber      950 non-null    object 
 5   Team1            950 non-null    object 
 6   Team2            950 non-null    object 
 7   Venue            950 non-null    object 
 8   TossWinner       950 non-null    object 
 9   TossDecision     950 non-null    object 
 10  SuperOver        946 non-null    object 
 11  WinningTeam      946 non-null    object 
 12  WonBy            950 non-null    object 
 13  Margin           932 non-null    float64
 14  method           19 non-null     object 
 15  Player_of_Match  946 non-null    object 
 16  Team1Players     950 non-null    object 
 17  Team2Players     950 non-null    object 
 18  Umpire1          950 non-null    object 
 19  Umpire2          950 non-null    object 
dtypes: float64(1), int64(1), object(18)
memory usage: 148.6+ KB
```

In [8]:

```
matches_result.head()
```

Out[8]:

	match_id	team_1	team_2	venue
0	1312200	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium, Ahmedabad
1	1312199	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium, Ahmedabad
2	1312198	Royal Challengers Bangalore	Lucknow Super Giants	Eden Gardens, Kolkata
3	1312197	Rajasthan Royals	Gujarat Titans	Eden Gardens, Kolkata
4	1304116	Sunrisers Hyderabad	Punjab Kings	Wankhede Stadium, Mumbai

- ## Get Venues Mapping

In [9]:

```
# [print(x) for x in np.sort(ipl_2023_venues['venue'].values)];
```

In [10]:

```
# [print(x) for x in np.sort(matches_result['venue'].unique())];
```

In [11]:

```
# Arun Jaitley Stadium [Arun Jaitley Stadium, Delhi]
# Brabourne Stadium [Brabourne Stadium, Mumbai]
# Dr DY Patil Sports Academy [Dr DY Patil Sports Academy, Mumbai]
# Eden Gardens [Eden Gardens, Kolkata]
# Himachal Pradesh Cricket Association Stadium
# M.Chinnaswamy Stadium [M Chinnaswamy Stadium]
# MA Chidambaram Stadium [MA Chidambaram Stadium, Chepauk] [MA Chidambaram Stadium,
# Maharashtra Cricket Association Stadium [Maharashtra Cricket Association Stadium,
# Narendra Modi Stadium [Narendra Modi Stadium, Ahmedabad]
# Punjab Cricket Association IS Bindra Stadium [Punjab Cricket Association IS Bindr
# Rajiv Gandhi International Stadium [Rajiv Gandhi International Stadium, Uppal]
# Sawai Mansingh Stadium
# Wankhede Stadium [Wankhede Stadium, Mumbai]
```

In [12]:

```
venue_mapping = {
 'Arun Jaitley Stadium, Delhi': 'Arun Jaitley Stadium',
 'Arun Jaitley Stadium': 'Arun Jaitley Stadium',
 'Brabourne Stadium, Mumbai': 'Brabourne Stadium',
 'Brabourne Stadium': 'Brabourne Stadium',
 'Dr DY Patil Sports Academy, Mumbai': 'Dr DY Patil Sports Academy',
 'Dr DY Patil Sports Academy': 'Dr DY Patil Sports Academy',
 'Eden Gardens, Kolkata': 'Eden Gardens',
 'Eden Gardens': 'Eden Gardens',
 'M Chinnaswamy Stadium': 'M.Chinnaswamy Stadium',
 'M.Chinnaswamy Stadium': 'M.Chinnaswamy Stadium',
 'Maharashtra Cricket Association Stadium, Pune': 'Maharashtra Cricket Association
 'Maharashtra Cricket Association Stadium': 'Maharashtra Cricket Association Stadiu
 'Narendra Modi Stadium, Ahmedabad': 'Narendra Modi Stadium',
 'Narendra Modi Stadium': 'Narendra Modi Stadium',
 'Rajiv Gandhi International Stadium, Uppal': 'Rajiv Gandhi International Stadium',
 'Rajiv Gandhi International Stadium': 'Rajiv Gandhi International Stadium',
 'Wankhede Stadium, Mumbai': 'Wankhede Stadium',
 'Wankhede Stadium': 'Wankhede Stadium',
 'Himachal Pradesh Cricket Association Stadium': 'Himachal Pradesh Cricket Associat
 'Sawai Mansingh Stadium': 'Sawai Mansingh Stadium',
 'MA Chidambaram Stadium, Chepauk': 'MA Chidambaram Stadium',
 'MA Chidambaram Stadium, Chepauk, Chennai': 'MA Chidambaram Stadium',
 'MA Chidambaram Stadium': 'MA Chidambaram Stadium',
 'Punjab Cricket Association IS Bindra Stadium, Mohali': 'Punjab Cricket Associatio
 'Punjab Cricket Association Stadium, Mohali': 'Punjab Cricket Association IS Bindr
 'Punjab Cricket Association IS Bindra Stadium': 'Punjab Cricket Association IS Bin
}
```

In [13]:

```
# np.setdiff1d(
#     np.setdiff1d(list(venue_mapping.keys()), ipl_2023_venues['venue'].values),  m
# )
```

- ## Get Teams Mapping

In [14]:

```
# print(np.array_equal(
#     np.sort(matches_result['team_1'].unique()),
#     np.sort(matches_result['team_2'].unique())
# ))

# print(np.array_equal(
#     np.sort(ball_by_ball['batting_team'].unique()),
#     np.sort(matches_result['team_1'].unique())
# ))
```

In [15]:

```
# [print(x) for x in ipl_2023_teams['team'].values];
```

In [16]:

```
# [print(x) for x in matches_result['team_1'].unique()];
```

In [17]:

```
# Rajasthan Royals
# Gujarat Titans
# Royal Challengers Bangalore
# Lucknow Super Giants
# Sunrisers Hyderabad
# Punjab Kings [Kings XI Punjab]
# Delhi Capitals [Delhi Daredevils]
# Mumbai Indians
# Chennai Super Kings
# Kolkata Knight Riders
```

In [18]:

```
team_mapping = {
 'Rajasthan Royals': 'Rajasthan Royals',
 'Gujarat Titans': 'Gujarat Titans',
 'Royal Challengers Bangalore': 'Royal Challengers Bangalore',
 'Lucknow Super Giants': 'Lucknow Super Giants',
 'Sunrisers Hyderabad': 'Sunrisers Hyderabad',
 'Mumbai Indians': 'Mumbai Indians',
 'Chennai Super Kings': 'Chennai Super Kings',
 'Kolkata Knight Riders': 'Kolkata Knight Riders',
 'Kings XI Punjab': 'Punjab Kings',
 'Punjab Kings': 'Punjab Kings',
 'Delhi Daredevils': 'Delhi Capitals',
 'Delhi Capitals': 'Delhi Capitals'
}
```

In [19]:

```
# np.setdiff1d(
#     np.setdiff1d(list(team_mapping.keys()), ipl_2023_teams['team'].values),  matc
# )
```

- ## Apply Venues/Teams Mapping [in matches_result, ball_by_ball]

In [20]:

```
matches_result.team_1 = matches_result.team_1.map(team_mapping)
print(matches_result.loc[matches_result.team_1.isnull()].shape)

matches_result.team_2 = matches_result.team_2.map(team_mapping)
print(matches_result.loc[matches_result.team_2.isnull()].shape)

matches_result.venue = matches_result.venue.map(venue_mapping)
print(matches_result.loc[matches_result.venue.isnull()].shape)

print(matches_result.shape)
print(matches_result.dropna().shape)
```

```
(99, 4)
(96, 4)
(320, 4)
(950, 4)
(525, 4)
```

```
In [21]: ball_by_ball.batting_team = ball_by_ball.batting_team.map(team_mapping)

         ball_by_ball.loc[ball_by_ball.batting_team.isnull()].shape
```

```
Out[21]: (23105, 8)
```

- ## Remove unnecessary Teams [in ball_by_ball] and Venues [in matches_result]

```
In [22]: matches_result = matches_result.dropna(subset=['team_1', 'team_2', 'venue'])
         # matches_result = matches_result.dropna(subset=['venue'])

         print(matches_result_orig.shape)
         print(matches_result.shape)
```

```
(950, 20)
(525, 4)
```

```
In [23]: ball_by_ball = ball_by_ball.dropna(subset=['batting_team'])

         print(ball_by_ball_orig.shape)
         print(ball_by_ball.shape)
```

```
(225954, 17)
(202849, 8)
```

- ## Select first 6 overs, Select innings 1 & 2, Map innings (1,2) to (0,1) [in ball_by_ball]

```
In [24]: ball_by_ball.innings.unique()
```

```
Out[24]: array([1, 2, 3, 4, 5, 6], dtype=int64)
```

```
In [25]: ball_by_ball.overs.unique()
```

```
Out[25]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19], dtype=int64)
```

```
In [26]: ball_by_ball = ball_by_ball.loc[(ball_by_ball.overs <= 5) & (ball_by_ball.innings <
         ball_by_ball.innings = ball_by_ball.innings.replace({1: 0, 2: 1})
         ball_by_ball.shape
```

```
Out[26]: (63652, 8)
```

```
In [27]: ball_by_ball.innings.unique()
```

Out[27]:

```
array([0, 1], dtype=int64)
```

In [28]:

```
ball_by_ball.overs.unique()
```

Out[28]:

```
array([0, 1, 2, 3, 4, 5], dtype=int64)
```

- ## Grouping

In [29]:

```
ball_by_ball_gb = ball_by_ball.groupby(['match_id', 'innings', 'batting_team'])
total_runs = ball_by_ball_gb['total_run'].sum()
batsmen = ball_by_ball_gb['batter'].unique()
bowlers = ball_by_ball_gb['bowler'].unique()
```

In [30]:

```
total_runs = total_runs.to_frame(name = 'total_runs').reset_index()
batsmen = batsmen.to_frame(name = 'batsmen').reset_index()
bowlers = bowlers.to_frame(name = 'bowlers').reset_index()
```

In [31]:

```
data = total_runs.merge(
    batsmen.merge(bowlers, how='right', on=['match_id','innings','batting_team']),
    how='right', on=['match_id','innings','batting_team']
)
```

In [32]:

```
data = data.merge(matches_result, on=['match_id'])
```

In [33]:

```
mask = data['batting_team'] == data['team_1']
data.loc[mask, 'bowling_team'] = data['team_2']
data.loc[~mask, 'bowling_team'] = data['team_1']
```

In [34]:

```
data = data.drop(columns=['team_1', 'team_2'])
data = data[['match_id', 'venue', 'innings', 'batting_team', 'bowling_team', 'batsm
```

In [35]:

```
# match_id == 829763, data for one innings is missing
# match_id == 829813, total_runs for one innings is 2 (probably a mistake in data e
data = data.drop(data[(data['match_id'] == 829763) | (data['match_id'] == 829813)].
```

In [36]:

```
# drop match_id as it is no longer needed
data = data.drop('match_id', axis=1)
```

In [37]:

```
data
```

Out[37]:

	venue	innings	batting_team	bowling_team	batsmen	bowlers	total
0	M.Chinnaswamy Stadium	0	Kolkata Knight Riders	Royal Challengers Bangalore	[SC Ganguly, BB McCullum, RT Ponting]	[P Kumar, Z Khan, AA Noffke]	
1	M.Chinnaswamy Stadium	1	Royal Challengers Bangalore	Kolkata Knight Riders	[R Dravid, W Jaffer, V Kohli, JH Kallis, CL Wh...	[AB Dinda, I Sharma, AB Agarkar]	
2	Punjab Cricket Association IS Bindra Stadium	0	Chennai Super Kings	Punjab Kings	[PA Patel, ML Hayden, MEK Hussey]	[B Lee, S Sreesanth, JR Hopes]	
3	Punjab Cricket Association IS Bindra Stadium	1	Punjab Kings	Chennai Super Kings	[K Goel, JR Hopes]	[JDP Oram, MS Gony]	
4	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	[L Ronchi, ST Jayasuriya, DJ Thornely, RV Utha...	[P Kumar, Z Khan, JH Kallis]	
...	...	...	...	...	...	...	
1044	Eden Gardens	1	Lucknow Super Giants	Royal Challengers Bangalore	[Q de Kock, KL Rahul, M Vohra, DJ Hooda]	[Mohammed Siraj, JR Hazlewood, Shahbaz Ahmed]	
1045	Narendra Modi Stadium	0	Royal Challengers Bangalore	Rajasthan Royals	[V Kohli, F du Plessis, RM Patidar]	[TA Boult, M Prasidh Krishna]	
1046	Narendra Modi Stadium	1	Rajasthan Royals	Royal Challengers Bangalore	[YBK Jaiswal, JC Buttler, SV Samson]	[Mohammed Siraj, JR Hazlewood, GJ Maxwell, Sha...	
1047	Narendra Modi Stadium	0	Rajasthan Royals	Gujarat Titans	[YBK Jaiswal, JC Buttler, SV Samson]	[Mohammed Shami, Yash Dayal, LH Ferguson, Rash...	

	venue	innings	batting_team	bowling_team	batsmen	bowlers	total
1048	Narendra Modi Stadium	1	Gujarat Titans	Rajasthan Royals	[WP Saha, Shubman Gill, MS Wade, HH Pandya]	[TA Boult, M Prasidh Krishna, YS Chahal]	

1046 rows × 7 columns