In [1]:
```
import numpy as np
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:
```
matches = pd.read_csv("C:/Users/HP/Desktop/project-Win Predictor/matches.csv")
matches
```

Out[2]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_a
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	
...	...	...	...	...	...	...	...	...	...	...
751	11347	IPL-2019	Mumbai	05-05-2019	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	
752	11412	IPL-2019	Chennai	07-05-2019	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	normal	
753	11413	IPL-2019	Visakhapatnam	08-05-2019	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field	normal	
754	11414	IPL-2019	Visakhapatnam	10-05-2019	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	normal	
755	11415	IPL-2019	Hyderabad	12-05-2019	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	normal	

756 rows × 18 columns

In [3]:
```
matches.shape
```

Out[3]:
```
(756, 18)
```

In [4]:
```
deliveries = pd.read_csv("C:/Users/HP/Desktop/project-Win Predictor/deliveries.csv")
deliveries
```

Out[4]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	
...	...	...	...	...	...	...	...	...	...	
179073	11415	2	Chennai Super Kings	Mumbai Indians	20	2	RA Jadeja	SR Watson	SL Malinga	
179074	11415	2	Chennai Super Kings	Mumbai Indians	20	3	SR Watson	RA Jadeja	SL Malinga	
179075	11415	2	Chennai Super Kings	Mumbai Indians	20	4	SR Watson	RA Jadeja	SL Malinga	
179076	11415	2	Chennai Super Kings	Mumbai Indians	20	5	SN Thakur	RA Jadeja	SL Malinga	
179077	11415	2	Chennai Super Kings	Mumbai Indians	20	6	SN Thakur	RA Jadeja	SL Malinga	

179078 rows × 21 columns

In [5]:
```
total_score=deliveries.groupby(['match_id','inning']).sum()['total_runs'].reset_index(
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\3490800198.py:1: FutureWarning: The defa
ult value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future version,
numeric_only will default to False. Either specify numeric_only or select only column
s which should be valid for the function.
  total_score=deliveries.groupby(['match_id','inning']).sum()['total_runs'].reset_ind
ex()
```

In [6]:
```
total_score.head()
```

Out[6]:

	match_id	inning	total_runs
0	1	1	207
1	1	2	172
2	2	1	184
3	2	2	187
4	3	1	183

In [7]:
```
total_score.shape
```

Out[7]: (1528, 3)

In [8]:
```
total_score= total_score[total_score['inning']==1]
```

In [9]:
```
total_score.head()
```

Out[9]:

	match_id	inning	total_runs
0	1	1	207
2	2	1	184
4	3	1	183
6	4	1	163
8	5	1	157

In [10]:
```
df = matches.merge(total_score[['match_id','total_runs']],left_on = 'id',right_on = 'ma
```

In [11]:
```
df
```

Out[11]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_a
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	
...	...	...	...	...	...	...	...	...	...	
751	11347	IPL-2019	Mumbai	05-05-2019	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	
752	11412	IPL-2019	Chennai	07-05-2019	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	normal	
753	11413	IPL-2019	Visakhapatnam	08-05-2019	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field	normal	
754	11414	IPL-2019	Visakhapatnam	10-05-2019	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	normal	
755	11415	IPL-2019	Hyderabad	12-05-2019	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	normal	

756 rows × 20 columns

In [12]:
```
df.head()
```

Out[12]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	I
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	C

In [13]:
```
df['team1'].unique()
```

Out[13]:
```
array(['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',
       'Rising Pune Supergiant', 'Royal Challengers Bangalore',
       'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',
       'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',
       'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiants',
       'Delhi Capitals'], dtype=object)
```

In [14]:
```
teams = [
'Sunrisers Hyderabad',
'Mumbai Indians',
'Royal Challengers Bangalore',
'Kolkata Knight Riders',
'Kings XI Punjab',
'Chennai Super Kings',
'Rajasthan Royals',
'Delhi Capitals'
]
```

In [15]:
```
df['team1']= df['team1'].str.replace('Delhi Daredevils','Delhi Capitals')
df['team2']= df['team2'].str.replace('Delhi Daredevils','Delhi Capitals')
```

In [16]:
```
df['team1']= df['team1'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
df['team2']= df['team2'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
```

In [17]:
```
df = df[df['team1'].isin(teams)]
df = df[df['team2'].isin(teams)]
```

In [18]:
```
df.head()
```

Out[18]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Capitals	Royal Challengers Bangalore	bat	normal	0
6	7	IPL-2017	Mumbai	09-04-2017	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0
7	8	IPL-2017	Indore	10-04-2017	Royal Challengers Bangalore	Kings XI Punjab	Royal Challengers Bangalore	bat	normal	0
9	10	IPL-2017	Mumbai	12-04-2017	Sunrisers Hyderabad	Mumbai Indians	Mumbai Indians	field	normal	0

In [19]:
```
df['dl_applied'].value_counts()
```

Out[19]:
```
0    626
1     15
Name: dl_applied, dtype: int64
```

In [20]:
```
df = df[df['dl_applied']==0]
```

In [21]:
```
df.head()
```

Out[21]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Capitals	Royal Challengers Bangalore	bat	normal	0
6	7	IPL-2017	Mumbai	09-04-2017	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal	0
7	8	IPL-2017	Indore	10-04-2017	Royal Challengers Bangalore	Kings XI Punjab	Royal Challengers Bangalore	bat	normal	0
9	10	IPL-2017	Mumbai	12-04-2017	Sunrisers Hyderabad	Mumbai Indians	Mumbai Indians	field	normal	0

In [22]:
```
df = df[['match_id','city','winner','total_runs']]
```

In [23]:
```
df.shape
```

Out[23]: (626, 4)

In [24]:
```
delivery_df = df.merge(deliveries,on = 'match_id')
```

In [25]:
```
delivery_df.shape
```

Out[25]: (149578, 24)

In [26]:
```
delivery_df
```

Out[26]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	bats
0	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	W
1	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	W
2	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	W
3	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	W
4	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	W
...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	J
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	W
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	W
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	T
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	T

149578 rows × 24 columns

In [27]:
```
delivery_df.columns
```

Out[27]:
```
Index(['match_id', 'city', 'winner', 'total_runs_x', 'inning', 'batting_team',
       'bowling_team', 'over', 'ball', 'batsman', 'non_striker', 'bowler',
       'is_super_over', 'wide_runs', 'bye_runs', 'legbye_runs', 'noball_runs',
       'penalty_runs', 'batsman_runs', 'extra_runs', 'total_runs_y',
       'player_dismissed', 'dismissal_kind', 'fielder'],
      dtype='object')
```

In [28]:
```
delivery_df=delivery_df[delivery_df['inning']==2]
```

In [29]:
```
delivery_df['current_score']= delivery_df.groupby('match_id').cumsum()['total_runs_y']
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\4178504166.py:1: FutureWarning: The defa
ult value of numeric_only in DataFrameGroupBy.cumsum is deprecated. In a future versi
on, numeric_only will default to False. Either specify numeric_only or select only co
lumns which should be valid for the function.
  delivery_df['current_score']= delivery_df.groupby('match_id').cumsum()['total_runs_
y']
C:\Users\HP\AppData\Local\Temp\ipykernel_672\4178504166.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['current_score']= delivery_df.groupby('match_id').cumsum()['total_runs_
y']
```

In [30]:
```
delivery_df
```

Out[30]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	bats
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mar
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mar
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mar
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mar
...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	J
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	W
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	W
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	T
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	T

72413 rows × 25 columns

In [31]:
```
delivery_df['runs_left'] = delivery_df['total_runs_x']-delivery_df['current_score']
```

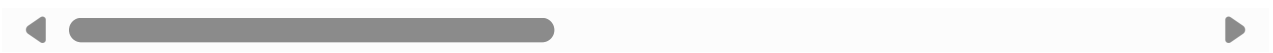
```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\2802540191.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['runs_left'] = delivery_df['total_runs_x']-delivery_df['current_score']
```

In [32]:
```
delivery_df
```

Out[32]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	bats
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mar
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mar
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mar
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mar
...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	J
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	W
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	W
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	T
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	T

72413 rows × 26 columns



In [33]:
```
delivery_df['balls_left'] = 126 - (delivery_df['over']*6 + delivery_df['ball'])
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\517047302.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['balls_left'] = 126 - (delivery_df['over']*6 + delivery_df['ball'])
```

In [34]:
```
delivery_df.head()
```

Out[34]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsma
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	C Gay
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandee Sing
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandee Sing
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandee Sing
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandee Sing

5 rows × 27 columns

In [35]:
```
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0")
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x if x
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int')
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\1172123056.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0")
C:\Users\HP\AppData\Local\Temp\ipykernel_672\1172123056.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x
if x =="0" else "1")
C:\Users\HP\AppData\Local\Temp\ipykernel_672\1172123056.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int')
```

In [36]:
```
wickets = delivery_df.groupby('match_id').cumsum()['player_dismissed'].values
delivery_df['wickets']= 10 - wickets
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\711636818.py:1: 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('match_id').cumsum()['player_dismissed'].values
C:\Users\HP\AppData\Local\Temp\ipykernel_672\711636818.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['wickets']= 10 - wickets
```

In [37]:
```
delivery_df
```

Out[37]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	bats
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mar
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mar
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mar
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mar
...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	J
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	W
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	W
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	T
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	T

72413 rows × 28 columns

In [38]:
```
delivery_df ['Crr'] = delivery_df['current_score']*6/(120-delivery_df['balls_left'])
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\1851297851.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df ['Crr'] = delivery_df['current_score']*6/(120-delivery_df['balls_left'])
```

In [39]:
```
delivery_df.head(1)
```

Out[39]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsma
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	C Gay

1 rows × 29 columns

In [40]:
```
delivery_df['rrr']= delivery_df['runs_left']*6/delivery_df['balls_left']
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\915821988.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['rrr']= delivery_df['runs_left']*6/delivery_df['balls_left']
```

In [41]:
```
delivery_df.head(5)
```

Out[41]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsma
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	C Gay
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandee Sing
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandee Sing
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandee Sing
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandee Sing

5 rows × 30 columns

In [42]:
```
delivery_df.shape
```

Out[42]: (72413, 30)

In [43]:
```
def abc(row):
    return 1 if row['batting_team']== row['winner'] else 0
```

In [44]:
```
delivery_df['result'] = delivery_df.apply(abc,axis = 1)
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_672\145264440.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  delivery_df['result'] = delivery_df.apply(abc,axis = 1)
```

In [45]:
```
delivery_df.head(2)
```

Out[45]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsma
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	C Gay
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandee Sing

2 rows × 31 columns

In [46]:
```
delivery_df.columns
```

Out[46]:
```
Index(['match_id', 'city', 'winner', 'total_runs_x', 'inning', 'batting_team',
       'bowling_team', 'over', 'ball', 'batsman', 'non_striker', 'bowler',
       'is_super_over', 'wide_runs', 'bye_runs', 'legbye_runs', 'noball_runs',
       'penalty_runs', 'batsman_runs', 'extra_runs', 'total_runs_y',
       'player_dismissed', 'dismissal_kind', 'fielder', 'current_score',
       'runs_left', 'balls_left', 'wickets', 'Crr', 'rrr', 'result'],
      dtype='object')
```

In [47]:
```
delivery_df = delivery_df[['batting_team','bowling_team','city','runs_left','balls_lef
```

In [48]:
```
delivery_df
```

Out[48]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
125	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	206	119	10	1	6.000000	10.38
126	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	206	118	10	0	3.000000	10.47
127	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	206	117	10	0	2.000000	10.56
128	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	204	116	10	2	4.500000	10.55
129	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	200	115	10	4	8.400000	10.43
...	...	...	...	...	...	...	...	...	
149573	Chennai Super Kings	Mumbai Indians	Hyderabad	0	4	5	1	7.862069	0.00
149574	Chennai Super Kings	Mumbai Indians	Hyderabad	-2	3	5	2	7.897436	-4.00
149575	Chennai Super Kings	Mumbai Indians	Hyderabad	-3	2	4	1	7.881356	-9.00
149576	Chennai Super Kings	Mumbai Indians	Hyderabad	-5	1	4	2	7.915966	-30.00
149577	Chennai Super Kings	Mumbai Indians	Hyderabad	-5	0	3	0	7.850000	

72413 rows × 10 columns

In [50]:
```
final_df = delivery_df.sample(delivery_df.shape[0])
```

In [51]:
```
final_df
```

Out[51]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
96408	Royal Challengers Bangalore	Chennai Super Kings	Ranchi	63	34	8	0	5.232558	1
59228	Delhi Daredevils	Royal Challengers Bangalore	Bangalore	49	21	5	1	6.545455	1
139394	Kolkata Knight Riders	Royal Challengers Bangalore	Bengaluru	58	22	6	2	9.306122	1
7652	Delhi Daredevils	Rajasthan Royals	Delhi	104	105	9	0	10.000000	
46778	Kolkata Knight Riders	Mumbai Indians	Kolkata	121	109	10	4	6.545455	
...	...	...	...	...	...	...	...	...	
66714	Kings XI Punjab	Rajasthan Royals	Chandigarh	44	-1	2	0	6.595041	-26
109155	Royal Challengers Bangalore	Chennai Super Kings	Chennai	118	93	9	0	6.666667	
10300	Kings XI Punjab	Delhi Daredevils	Chandigarh	37	32	6	0	8.250000	
115126	Kolkata Knight Riders	Sunrisers Hyderabad	Hyderabad	129	105	10	0	5.200000	
132498	Kolkata Knight Riders	Rajasthan Royals	Kolkata	60	53	7	0	7.522388	

72413 rows × 10 columns

In [ ]:
```
# Train test split
```

In [93]:
```
x = final_df.iloc [:,:-1]
y = final_df['result']
```

In [94]:
```
x
```

Out[94]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
96408	Royal Challengers Bangalore	Chennai Super Kings	Ranchi	63	34	8	0	5.232558	1
59228	Delhi Daredevils	Royal Challengers Bangalore	Bangalore	49	21	5	1	6.545455	1
139394	Kolkata Knight Riders	Royal Challengers Bangalore	Bengaluru	58	22	6	2	9.306122	1
7652	Delhi Daredevils	Rajasthan Royals	Delhi	104	105	9	0	10.000000	
46778	Kolkata Knight Riders	Mumbai Indians	Kolkata	121	109	10	4	6.545455	
...	...	...	...	...	...	...	...	...	
66714	Kings XI Punjab	Rajasthan Royals	Chandigarh	44	-1	2	0	6.595041	-26
109155	Royal Challengers Bangalore	Chennai Super Kings	Chennai	118	93	9	0	6.666667	
10300	Kings XI Punjab	Delhi Daredevils	Chandigarh	37	32	6	0	8.250000	
115126	Kolkata Knight Riders	Sunrisers Hyderabad	Hyderabad	129	105	10	0	5.200000	
132498	Kolkata Knight Riders	Rajasthan Royals	Kolkata	60	53	7	0	7.522388	

71342 rows × 9 columns

In [95]:
```
y
```

Out[95]:
```
96408     1
59228     0
139394    1
7652      1
46778     1
         ..
66714     0
109155    0
10300     1
115126    1
132498    1
Name: result, Length: 71342, dtype: int64
```

In [96]:
```
x_train.isnull().sum()
```

Out[96]:
```
batting_team    0
bowling_team    0
city            0
runs_left       0
balls_left      0
wickets         0
total_runs_y    0
Crr             0
rrr             0
dtype: int64
```

In [82]:
```
final_df.dropna(inplace = True)
```

In [83]:
```
final_df.isnull().sum()
```

Out[83]:
```
batting_team    0
bowling_team    0
city            0
runs_left       0
balls_left      0
wickets         0
total_runs_y    0
Crr             0
rrr             0
result          0
dtype: int64
```

In [97]:
```
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 = 10
```

In [98]:
```
x_train
```

Out[98]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
100458	Chennai Super Kings	Kings XI Punjab	Mumbai	64	26	4	1	10.340426	1
91508	Kolkata Knight Riders	Chennai Super Kings	Ranchi	82	62	5	1	6.827586	
77276	Delhi Daredevils	Mumbai Indians	Delhi	145	104	10	0	6.000000	
31534	Royal Challengers Bangalore	Delhi Daredevils	Johannesburg	118	96	9	0	4.000000	
47356	Deccan Chargers	Chennai Super Kings	Mumbai	50	22	4	6	5.632653	1
...	...	...	...	...	...	...	...	...	
86728	Mumbai Indians	Kolkata Knight Riders	Abu Dhabi	146	99	10	1	4.857143	
108067	Rajasthan Royals	Mumbai Indians	Mumbai	99	59	8	1	8.655738	1
21057	Royal Challengers Bangalore	Chennai Super Kings	Port Elizabeth	138	86	8	1	7.235294	
83302	Kolkata Knight Riders	Royal Challengers Bangalore	Ranchi	46	47	7	1	5.671233	
21827	Kolkata Knight Riders	Rajasthan Royals	Cape Town	55	35	5	1	6.705882	

57073 rows × 9 columns

In [99]:
```
x_test
```

Out[99]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
8410	Rajasthan Royals	Kings XI Punjab	Jaipur	74	58	7	4	8.903226	7.6
34211	Kolkata Knight Riders	Royal Challengers Bangalore	Kolkata	123	108	10	1	6.000000	6.8
81184	Kings XI Punjab	Royal Challengers Bangalore	Bangalore	13	21	8	6	9.757576	3.7
435	Delhi Daredevils	Royal Challengers Bangalore	Bangalore	78	59	7	0	7.770492	7.9
18528	Chennai Super Kings	Rajasthan Royals	Chennai	38	19	7	4	10.277228	12.0
...	...	...	...	...	...	...	...	...	
31307	Kolkata Knight Riders	Chennai Super Kings	Centurion	129	80	9	1	8.850000	9.6
73727	Kolkata Knight Riders	Rajasthan Royals	Jaipur	61	39	4	2	6.148148	9.3
111626	Kings XI Punjab	Sunrisers Hyderabad	Hyderabad	88	38	5	6	7.097561	13.8
25031	Kings XI Punjab	Royal Challengers Bangalore	Durban	51	31	7	1	6.337079	9.8
35946	Chennai Super Kings	Delhi Daredevils	Delhi	29	25	6	1	9.852632	6.9

14269 rows × 9 columns

In [100]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

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

In [88]:
```
x_test.isnull().sum()
```

Out[88]:
```
batting_team    0
bowling_team    0
city            0
runs_left       0
balls_left      0
wickets         0
total_runs_y    0
Crr             0
rrr             0
dtype: int64
```

In [109]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
```

In [91]:
```
final_df = final_df[final_df['balls_left'] !=0]
```

In [92]:
```
final_df
```

Out[92]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_y	Crr	
96408	Royal Challengers Bangalore	Chennai Super Kings	Ranchi	63	34	8	0	5.232558	1
59228	Delhi Daredevils	Royal Challengers Bangalore	Bangalore	49	21	5	1	6.545455	14
139394	Kolkata Knight Riders	Royal Challengers Bangalore	Bengaluru	58	22	6	2	9.306122	15
7652	Delhi Daredevils	Rajasthan Royals	Delhi	104	105	9	0	10.000000	5
46778	Kolkata Knight Riders	Mumbai Indians	Kolkata	121	109	10	4	6.545455	6
...	...	...	...	...	...	...	...	...	
66714	Kings XI Punjab	Rajasthan Royals	Chandigarh	44	-1	2	0	6.595041	-264
109155	Royal Challengers Bangalore	Chennai Super Kings	Chennai	118	93	9	0	6.666667	7
10300	Kings XI Punjab	Delhi Daredevils	Chandigarh	37	32	6	0	8.250000	6
115126	Kolkata Knight Riders	Sunrisers Hyderabad	Hyderabad	129	105	10	0	5.200000	7
132498	Kolkata Knight Riders	Rajasthan Royals	Kolkata	60	53	7	0	7.522388	6

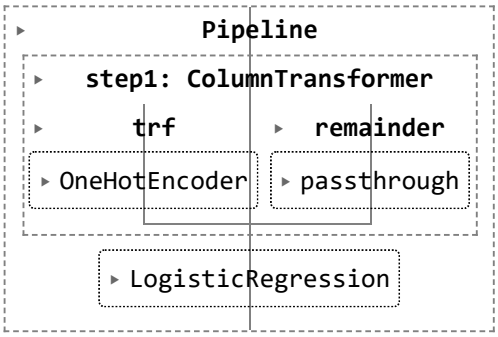
71342 rows × 10 columns

In [117]:
```
pipe = Pipeline(steps = [
    ('step1',trf),
    ('step2',LogisticRegression(solver = 'liblinear'))
])
```

In [118]:
```
pipe.fit(x_train,y_train)
```

```
D:\Users\HP\anaconda3\Lib\site-packages\sklearn\preprocessing\_encoders.py:972: 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(
```

Out[118]:



In [119]:
```
y_pred = pipe.predict(x_test)
```

In [120]:
```
y_pred.shape
```

Out[120]:
```
(14269,)
```

In [121]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[121]:
```
0.8032798374097694
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: