In [69]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [70]:

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

In [71]:

```
matches.shape,deliveries.shape
```

Out[71]:

```
((756, 18), (179078, 21))
```

In [72]:

```
matches.head()
```

Out[72]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result
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

In [73]:

```
deliveries.head()
```

Out[73]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	i
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	

5 rows × 21 columns

In [74]:

```
#Grouping the data of 1st innings and 2nd innings
total_runs_df=deliveries.groupby(['match_id','inning']).sum()['total_runs'].reset_index(
```

In [75]:

```
total_runs_df
```

Out[75]:

	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
...	...	...	...
1523	11413	2	170
1524	11414	1	155
1525	11414	2	162
1526	11415	1	152
1527	11415	2	157

1528 rows × 3 columns

Capturing only first inning

In [76]:

```
total_runs_df=total_runs_df[total_runs_df['inning']==1]
```

In [77]:

```
total_runs_df.head()
```

Out[77]:

	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 [78]:

```
total_runs_df['total_runs']=total_runs_df['total_runs'].map(lambda X:X+1)
```

In [79]:

```
total_runs_df
```

Out[79]:

	match_id	inning	total_runs
0	1	1	208
2	2	1	185
4	3	1	184
6	4	1	164
8	5	1	158
...	...	...	...
1518	11347	1	144
1520	11412	1	137
1522	11413	1	172
1524	11414	1	156
1526	11415	1	153

756 rows × 3 columns

In [80]:

```
total_runs_df.shape
```

Out[80]:

```
(756, 3)
```

In [81]:

```
matches.columns
```

Out[81]:

```
Index(['id', 'Season', 'city', 'date', 'team1', 'team2', 'toss_winner',
       'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',
       'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',
       'umpire3'],
      dtype='object')
```

In [82]:

```
matches.head()
```

Out[82]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result
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

Merging two Datasets matches and total_runs_df

where left side merging is done on id of matches and right join is done on match_id of total_runs_df

In [83]:

```
match_df=matches.merge(total_runs_df[['match_id','total_runs']],left_on='id',right_on='m
```

In [84]:

```
match_df
```

Out[84]:

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

756 rows × 20 columns

In [85]:

```
match_df['team1'].unique()
```

Out[85]:

```
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 [86]:

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

In [87]:

```
match_df['team1']=match_df['team1'].str.replace('Delhi Daredevils','Delhi Capitals')
match_df['team2']=match_df['team2'].str.replace('Delhi Daredevils','Delhi Capitals')
```

In [88]:

```
match_df['team1']=match_df['team1'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
match_df['team2']=match_df['team2'].str.replace('Deccan Chargers','Sunrisers Hyderabad')
```

We are going to consider only frequently occuring teams which are mentioned in Teams list

In [89]:

```
match_df=match_df[match_df['team1'].isin(Teams)]
match_df=match_df[match_df['team2'].isin(Teams)]
```

In [90]:

```
match_df.shape
```

Out[90]:

```
(641, 20)
```

In [91]:

```
match_df[match_df['dl_applied']==1].style.background_gradient(cmap='plasma')
```

Out[91]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision
56	57	IPL-2017	Bangalore	17-05-2017	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field
99	100	IPL-2008	Delhi	17-05-2008	Delhi Capitals	Kings XI Punjab	Delhi Daredevils	bat
102	103	IPL-2008	Kolkata	18-05-2008	Kolkata Knight Riders	Chennai Super Kings	Kolkata Knight Riders	bat
119	120	IPL-2009	Cape Town	19-04-2009	Kings XI Punjab	Delhi Capitals	Delhi Daredevils	field
122	123	IPL-2009	Durban	21-04-2009	Kings XI Punjab	Kolkata Knight Riders	Kolkata Knight Riders	field
148	149	IPL-2009	Centurion	07-05-2009	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat
280	281	IPL-2011	Kolkata	07-05-2011	Chennai Super Kings	Kolkata Knight Riders	Chennai Super Kings	bat
290	291	IPL-2011	Bangalore	14-05-2011	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field
488	489	IPL-2014	Delhi	10-05-2014	Delhi Capitals	Sunrisers Hyderabad	Sunrisers Hyderabad	field
536	537	IPL-2015	Visakhapatnam	22-04-2015	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field
567	568	IPL-2015	Hyderabad	15-05-2015	Sunrisers Hyderabad	Royal Challengers Bangalore	Sunrisers Hyderabad	bat
625	626	IPL-2016	Bangalore	18-05-2016	Royal Challengers Bangalore	Kings XI Punjab	Kings XI Punjab	field
641	7899	IPL-2018	Jaipur	11-04-2018	Rajasthan Royals	Delhi Capitals	Delhi Daredevils	field
653	7911	IPL-2018	Kolkata	21-04-2018	Kolkata Knight Riders	Kings XI Punjab	Kings XI Punjab	field
667	7925	IPL-2018	Delhi	02-05-2018	Delhi Capitals	Rajasthan Royals	Rajasthan Royals	field

Ignoring matches where dl method is applied

In [92]:

```
match_df=match_df[match_df['dl_applied']==0]
```

Considering only match id, city, winner, total runs

In [93]:

```
match_df=match_df[['match_id','city','winner','total_runs']]
```

In [94]:

```
match_df['winner'].unique()
```

Out[94]:

```
array(['Sunrisers Hyderabad', 'Royal Challengers Bangalore',
       'Mumbai Indians', 'Kings XI Punjab', 'Kolkata Knight Riders',
       'Delhi Daredevils', 'Chennai Super Kings', 'Rajasthan Royals',
       'Deccan Chargers', nan, 'Delhi Capitals'], dtype=object)
```

In [95]:

```
match_df=match_df[match_df['winner'].isin(Teams)]
```

In [96]:

```
match_df
```

Out[96]:

	match_id	city	winner	total_runs
0	1	Hyderabad	Sunrisers Hyderabad	208
4	5	Bangalore	Royal Challengers Bangalore	158
6	7	Mumbai	Mumbai Indians	179
7	8	Indore	Kings XI Punjab	149
9	10	Mumbai	Mumbai Indians	159
...	...	...	...	...
751	11347	Mumbai	Mumbai Indians	144
752	11412	Chennai	Mumbai Indians	137
753	11413	Visakhapatnam	Delhi Capitals	172
754	11414	Visakhapatnam	Chennai Super Kings	156
755	11415	Hyderabad	Mumbai Indians	153

542 rows × 4 columns

In [97]:

```
#Merging match_df with delivery df
```

In [98]:

```
delivery_df=match_df.merge(deliveries,on="match_id")
```

In [99]:

```
delivery_df
```

Out[99]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	c
0	1	Hyderabad	Sunrisers Hyderabad	208	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
1	1	Hyderabad	Sunrisers Hyderabad	208	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
2	1	Hyderabad	Sunrisers Hyderabad	208	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
3	1	Hyderabad	Sunrisers Hyderabad	208	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
4	1	Hyderabad	Sunrisers Hyderabad	208	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
...	...	...	...	...	...	...	...	
129935	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129936	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129937	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129938	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129939	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	

129940 rows × 24 columns

In [100]:

```
delivery_df=delivery_df[delivery_df['inning']==2]
```

In [101]:

```
delivery_df.head()
```

Out[101]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 24 columns

Current score of particular match

In [102]:

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

In [103]:

```
delivery_df.head()
```

Out[103]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 25 columns

In [104]:

```
delivery_df['runs_left']=delivery_df['total_runs_x']-delivery_df['current_score']
```

In [105]:

```
delivery_df.head()
```

Out[105]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 26 columns

In [106]:

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

In [107]:

```
delivery_df.head()
```

Out[107]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 27 columns

In [108]:

```
delivery_df['player_dismissed'].unique()[:2]
```

Out[108]:

```
array([nan, 'Mandeep Singh'], dtype=object)
```

Here player dismissed means 0 else 1

In [109]:

```
delivery_df['player_dismissed']=delivery_df['player_dismissed'].fillna(0)
```

In [110]:

```
delivery_df
```

Out[110]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	c
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	
129935	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129936	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129937	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129938	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129939	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	

63071 rows × 27 columns

In [111]:

```
delivery_df['player_dismissed']=delivery_df['player_dismissed'].apply(lambda x:x if x==0
```

In [112]:

```
delivery_df['player_dismissed'].unique()
```

Out[112]:

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

Wickets fallen

In [113]:

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

In [114]:

```
delivery_df['wickets']=10-wickets
```

In [115]:

```
delivery_df
```

Out[115]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	c
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	
129935	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129936	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129937	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129938	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	
129939	11415	Hyderabad	Mumbai Indians	153	2	Chennai Super Kings	Mumbai Indians	

63071 rows × 28 columns

Current run rate

In [116]:

```
delivery_df['cur_run_rate']=(delivery_df['current_score']*6)/(120-delivery_df['balls_lef
```

In [117]:

```
delivery_df.head()
```

Out[117]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 29 columns

In [118]:

```
delivery_df['required_run_rate']=round((delivery_df['runs_left']*6)/(delivery_df['balls_
```

In [119]:

```
delivery_df.head()
```

Out[119]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	208	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 30 columns

In [120]:

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

In [121]:

```
delivery_df['result']=delivery_df.apply(resultfun,axis=1)
```

In [122]:

```
delivery_df.sample(10)
```

Out[122]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_tea
1581	12	Bangalore	Mumbai Indians	143	2	Mumbai Indians	Roy Challenge Bangalo
11499	86	Mumbai	Mumbai Indians	104	2	Mumbai Indians	Rajastha Roya
39263	248	Chennai	Chennai Super Kings	184	2	Royal Challengers Bangalore	Chenn Super King
108676	7921	Jaipur	Sunrisers Hyderabad	153	2	Rajasthan Royals	Sunrise Hyderaba
77774	486	Cuttack	Kings XI Punjab	232	2	Chennai Super Kings	Kings Punja
101090	619	Visakhapatnam	Kings XI Punjab	125	2	Kings XI Punjab	Mumb Indiar
79371	494	Ranchi	Chennai Super Kings	149	2	Chennai Super Kings	Rajastha Roya
93585	559	Chennai	Mumbai Indians	159	2	Mumbai Indians	Chenn Super King
98999	597	Chandigarh	Mumbai Indians	190	2	Kings XI Punjab	Mumb Indiar
104693	7903	Kolkata	Sunrisers Hyderabad	146	2	Sunrisers Hyderabad	Kolkata Knig Ride

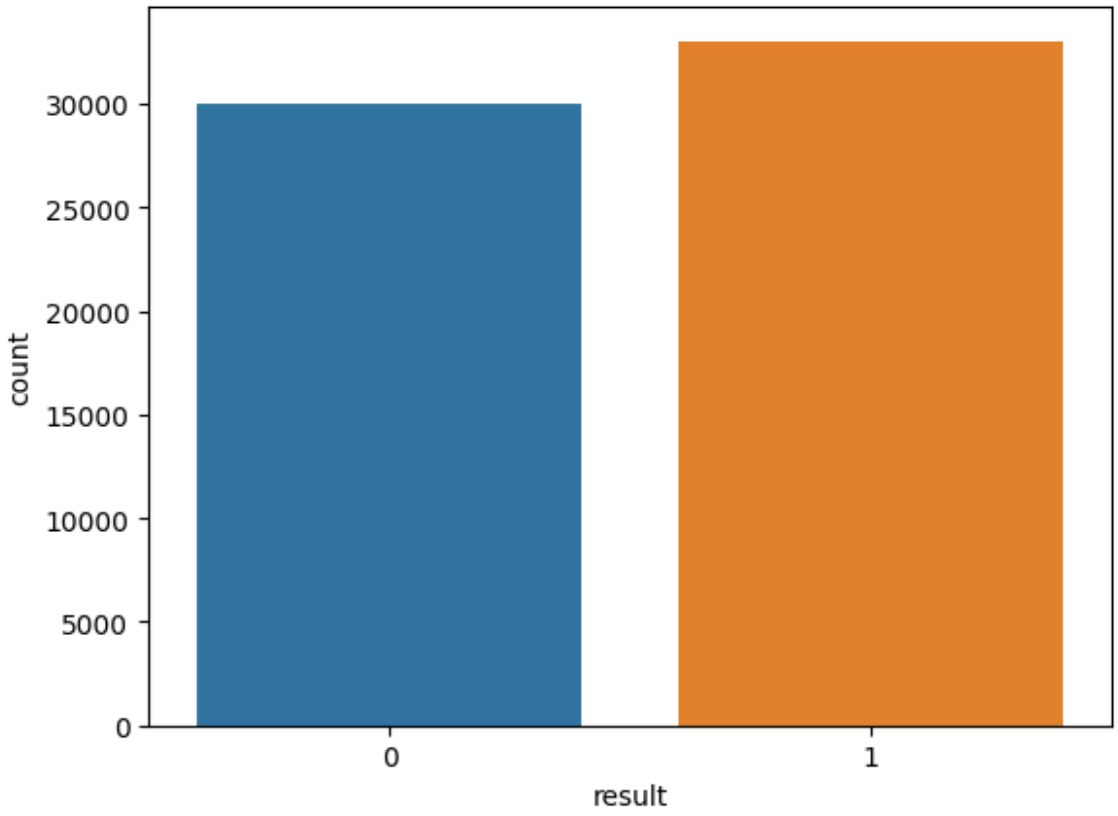
10 rows × 31 columns

In [123]:

```
sns.countplot(x=delivery_df['result'])
```

Out[123]:

```
<AxesSubplot: xlabel='result', ylabel='count'>
```



In [124]:

```
final_df=delivery_df[['batting_team','bowling_team','city','runs_left','balls_left','wic
```

In [125]:

```
final_df.sample(5)
```

Out[125]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	cur_run_rate
15453	Chennai Super Kings	Royal Challengers Bangalore	Chennai	40	36	6	6.214286
99892	Kings XI Punjab	Kolkata Knight Riders	Kolkata	153	105	9	4.800000
19553	Rajasthan Royals	Delhi Daredevils	Centurion	38	26	5	6.765957
105047	Royal Challengers Bangalore	Rajasthan Royals	Bengaluru	52	11	5	9.908257
124803	Sunrisers Hyderabad	Kolkata Knight Riders	Hyderabad	131	101	10	10.736842

In [126]:

```
final_df.isnull().sum()
```

Out[126]:

```
batting_team          0
bowling_team          0
city                712
runs_left             0
balls_left            0
wickets               0
cur_run_rate          0
required_run_rate     5
result                0
dtype: int64
```

In [127]:

```
final_df=final_df.dropna()
```

In [128]:

```
final_df.isnull().sum()
```

Out[128]:

```
batting_team          0
bowling_team          0
city                  0
runs_left             0
balls_left            0
wickets               0
cur_run_rate          0
required_run_rate     0
result                0
dtype: int64
```

In [129]:

```
final_df=final_df[final_df['balls_left']!=0]
```

In [130]:

```
final_df.head()
```

Out[130]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	cur_run_rate	requ
125	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	119	10	6.0	
126	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	118	10	3.0	
127	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	117	10	2.0	
128	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	205	116	10	4.5	
129	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	201	115	10	8.4	

In [131]:

```
x=final_df.drop(['result'],axis=1)
y=final_df['result']
```

In [132]:

```
from sklearn.model_selection import train_test_split
```

In [133]:

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.20,random_state=1)
```

In [134]:

```
xtrain.shape,ytrain.shape
```

Out[134]:

```
((49720, 8), (49720,))
```

In [137]:

```
xtrain.columns
```

Out[137]:

```
Index(['batting_team', 'bowling_team', 'city', 'runs_left', 'balls_left',
       'wickets', 'cur_run_rate', 'required_run_rate'],
      dtype='object')
```

In [135]:

```
xtest.shape,ytest.shape
```

Out[135]:

```
((12431, 8), (12431,))
```

In [136]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn import metrics
```

Categorical columns

In [138]:

```
cat_col=final_df.select_dtypes(include=['O']).columns
cat_col
```

Out[138]:

```
Index(['batting_team', 'bowling_team', 'city'], dtype='object')
```

In [141]:

```
cf=ColumnTransformer([('trf',OneHotEncoder(sparse=False,drop="first"),['batting_team','b
```

In [143]:

```
pipe=Pipeline(steps=[('step1',cf),
                     ('step2',LogisticRegression(solver='liblinear'))])
pipe.fit(xtrain,ytrain)
```

Out[143]:

```
Pipeline(steps=[('step1',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('trf',
                                                  OneHotEncoder(drop='firs
t',
                                                                sparse=Fal
se),
                                                  ['batting_team',
                                                   'bowling_team', 'cit
y'])])),
                ('step2', LogisticRegression(solver='liblinear'))])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [144]:

```
ypred=pipe.predict(xtest)
```

In [145]:

```
pipe.predict_proba(xtest)[10]
```

Out[145]:

```
array([0.04690986, 0.95309014])
```

In [147]:

```
pipe1=Pipeline(steps=[('step1',cf),
                      ('step2',RandomForestClassifier())])
pipe1.fit(xtrain,ytrain)
```

Out[147]:

```
Pipeline(steps=[('step1',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('trf',
                                                  OneHotEncoder(drop='firs
t',
                                                                sparse=Fal
se),
                                                  ['batting_team',
                                                   'bowling_team', 'cit
y'])])),
                ('step2', RandomForestClassifier())])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [149]:

```
pipe.predict_proba(xtest)[10]
```

Out[149]:

```
array([0.04690986, 0.95309014])
```

In [150]:

```
import pickle
pickle.dump(pipe,open("pipe.pickle","wb"))
```

In [ ]:

```

```