In [1]:

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

In [2]:

```
match = pd.read_csv('matches.csv')  #This project is a classification problem
```

In [3]:

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

In [4]:

```
match
```

Out[4]:

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

756 rows × 18 columns

In [5]:

```
delivery
```

Out[5]:

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

179078 rows × 21 columns

In [6]:

```
match.shape
```

Out[6]:

```
(756, 18)
```

In [7]:

```
delivery.head() #every match's every ball detail
```

Out[7]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_
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 [8]:

```
#sorting how many runs were scored by a team in a match
delivery.groupby(['match_id','inning']).sum()['total_runs']

#here we can see that in 1st match - 1st team scored 207 runs; whereas
#2nd team scored 172 runs....similarly the same applies to all teams below in the list
```

Out[8]:

```
match_id  inning
1         1         207
          2         172
2         1         184
          2         187
3         1         183
                   ...
11413     2         170
11414     1         155
          2         162
11415     1         152
          2         157
Name: total_runs, Length: 1528, dtype: int64
```

In [9]:

```
total_score_df = delivery.groupby(['match_id','inning']).sum()['total_runs'].reset_index()
total_score_df
#converting this into a dataframe
```

Out[9]:

	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

In [10]:

```
total_score_df[total_score_df['inning'] == 1 ]
#extracting the total runs scored by 1st team in every match
```

Out[10]:

	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
...	...	...	...
1518	11347	1	143
1520	11412	1	136
1522	11413	1	171
1524	11414	1	155
1526	11415	1	152

756 rows × 3 columns

In [11]:

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

In [12]:

```
match.merge(total_score_df[['match_id','total_runs']], left_on = 'id', right_on = 'match_id

#merging the total runs scored by 1st team and match_id, in every match with the 'match' da
```

Out[12]:

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

756 rows × 20 columns

In [13]:

```
match_df = match.merge(total_score_df[['match_id','total_runs']], left_on = 'id', right_on
```

Now there are some ipl teams which used to play before, but don't play now. So we will take only those

teams which were common throughout the whole IPL, from starting. Also, we'll merge the details of the teams whose names have been changed, eg: Deccan Chargers is renamed to Sunrisers Hyderabad

In [14]:

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

Out[14]:

```
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 [15]:

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

]
```

here we are replacing the old team names with the new ones

In [16]:

```
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 [17]:

```
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 [18]:

```
match_df = match_df[match_df['team1'].isin(teams)]
match_df = match_df[match_df['team2'].isin(teams)]
#keep the names/rows of only those teams whose names are mentioned ...
#...in the 'teams' list; for the columns team1 and team2 in match_df dataframe
```

In [19]:

```
match_df.shape
```

Out[19]:

```
(641, 20)
```

Now we want to keep data for only those matches which are not rain affected,i.e where DLS or duckworth lewis method is not applied

In [20]:

```
match_df['dl_applied'].value_counts()
```

Out[20]:

```
0    626
1     15
Name: dl_applied, dtype: int64
```

In [21]:

```
match_df[match_df['dl_applied'] == 1] #these are those 15 matches which are rain affected,
```

Out[21]:

	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

In [22]:

```
match_df[match_df['dl_applied'] == 0] #matches without DLS
```

Out[22]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Capitals	Royal Challengers Bangalore	bat
6	7	IPL-2017	Mumbai	09-04-2017	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field
7	8	IPL-2017	Indore	10-04-2017	Royal Challengers Bangalore	Kings XI Punjab	Royal Challengers Bangalore	bat
9	10	IPL-2017	Mumbai	12-04-2017	Sunrisers Hyderabad	Mumbai Indians	Mumbai Indians	field
...	...	...	...	...	...	...	...	...
751	11347	IPL-2019	Mumbai	05-05-2019	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field
752	11412	IPL-2019	Chennai	07-05-2019	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat
753	11413	IPL-2019	Visakhapatnam	08-05-2019	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field
754	11414	IPL-2019	Visakhapatnam	10-05-2019	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field
755	11415	IPL-2019	Hyderabad	12-05-2019	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat

626 rows × 20 columns

In [23]:

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

In [24]:

```
match_df
```

Out[24]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Capitals	Royal Challengers Bangalore	bat
6	7	IPL-2017	Mumbai	09-04-2017	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field
7	8	IPL-2017	Indore	10-04-2017	Royal Challengers Bangalore	Kings XI Punjab	Royal Challengers Bangalore	bat
9	10	IPL-2017	Mumbai	12-04-2017	Sunrisers Hyderabad	Mumbai Indians	Mumbai Indians	field
...	...	...	...	...	...	...	...	...
751	11347	IPL-2019	Mumbai	05-05-2019	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field
752	11412	IPL-2019	Chennai	07-05-2019	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat
753	11413	IPL-2019	Visakhapatnam	08-05-2019	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field
754	11414	IPL-2019	Visakhapatnam	10-05-2019	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field
755	11415	IPL-2019	Hyderabad	12-05-2019	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat

626 rows × 20 columns

In [25]:

```
match_df[['match_id','city','winner','total_runs']] #filtering out only the required column
```

Out[25]:

	match_id	city	winner	total_runs
0	1	Hyderabad	Sunrisers Hyderabad	207
4	5	Bangalore	Royal Challengers Bangalore	157
6	7	Mumbai	Mumbai Indians	178
7	8	Indore	Kings XI Punjab	148
9	10	Mumbai	Mumbai Indians	158
...	...	...	...	...
751	11347	Mumbai	Mumbai Indians	143
752	11412	Chennai	Mumbai Indians	136
753	11413	Visakhapatnam	Delhi Capitals	171
754	11414	Visakhapatnam	Chennai Super Kings	155
755	11415	Hyderabad	Mumbai Indians	152

626 rows × 4 columns

In [26]:

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

In [27]:

```
match_df.merge(delivery,on='match_id') #merging the required columns with the delivery data
```

Out[27]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
0	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
1	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
2	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
3	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
4	1	Hyderabad	Sunrisers Hyderabad	207	1	Sunrisers Hyderabad	Royal Challengers Bangalore	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

149578 rows × 24 columns

In [28]:

```
delivery_df = match_df.merge(delivery,on='match_id')
```

In [29]:

```
delivery_df[delivery_df['inning'] == 2]  #since we'll be predicting this when 2nd team is p
```

Out[29]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 24 columns

In [30]:

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

In [31]:

```
delivery_df.shape
```

Out[31]:

```
(72413, 24)
```

In [32]:

```
#total_runs_y states how many runs were made on each ball
```

Now we'll find the cumulative sum of scored runs to find the current runs and runs needed to achieve the target

In [33]:

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

Out[33]:

```
125         1
126         1
127         1
128         3
129         7
         ...
149573    152
149574    154
149575    155
149576    157
149577    157
Name: total_runs_y, Length: 72413, dtype: int64
```

In [34]:

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

In [35]:

```
delivery_df
```

Out[35]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 25 columns

In [36]:

```
delivery_df.columns
```

Out[36]:

```
Index(['match_id', 'city', 'winner', 'total_runs_x', 'inning', 'batting_tea
m',
       'bowling_team', 'over', 'ball', 'batsman', 'non_striker', 'bowler',
       'is_super_over', 'wide_runs', 'bye_runs', 'legbye_runs', 'noball_run
s',
       'penalty_runs', 'batsman_runs', 'extra_runs', 'total_runs_y',
       'player_dismissed', 'dismissal_kind', 'fielder', 'current_score'],
      dtype='object')
```

In [37]:

```
delivery_df['actual_target_score'] = 1
```

In [38]:

```
delivery_df
```

Out[38]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 26 columns

In [39]:

```
delivery_df['actual_target_score'] = delivery_df['total_runs_x'] + delivery_df['actual_targ
#since a team requires 1 run extra than the orginal target to win the match, therefore, we
#calculating the actual score that the team requires to win
```

In [40]:

```
delivery_df
```

Out[40]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 26 columns

In [41]:

```
delivery_df['actual_target_score'] - delivery_df['current_score']
```

Out[41]:

```
125       207
126       207
127       207
128       205
129       201
         ...
149573      1
149574     -1
149575     -2
149576     -4
149577     -4
Length: 72413, dtype: int64
```

In [42]:

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

In [43]:

```
delivery_df
```

Out[43]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad
...	...	...	...	...	...	...	...
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians

72413 rows × 27 columns

finding out the number of balls left, after each ball thrown

In [44]:

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

In [45]:

```
delivery_df
```

Out[45]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 28 columns

now we'll find out the number of wickets left after each ball

In [46]:

```
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0") #convert the
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x if x ==
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int') #converting
```

In [47]:

```
delivery_df
```

Out[47]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 28 columns

In [48]:

```
wickets = delivery_df.groupby('match_id').cumsum()['player_dismissed'].values
#calculating cumlative s
```

In [49]:

```
delivery_df['wickets_left'] = 10 - wickets #wickets left
delivery_df
```

Out[49]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 29 columns

Now we will find the current run rate and the required run rate

In [50]:

```
#crr = runs/overs

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

In [51]:

```
delivery_df
```

Out[51]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 30 columns

In [52]:

```
#rrr = runs_left / overs left

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

In [53]:

```
delivery_df
```

Out[53]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 31 columns

In [54]:

```
#determing if the team batting 2nd won the match or no

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

In [55]:

```
delivery_df.apply(result,axis=1)
```

Out[55]:

```
125       0
126       0
127       0
128       0
129       0
         ..
149573    0
149574    0
149575    0
149576    0
149577    0
Length: 72413, dtype: int64
```

In [56]:

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

In [57]:

```
delivery_df
```

Out[57]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	ove
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	
...	...	...	...	...	...	...	...	.
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	2

72413 rows × 32 columns

In [58]:

```
#Extracting the final required columns for our data
delivery_df[['batting_team','bowling_team','city','runs_left','balls_left','wickets_left','
```

Out[58]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets_left	actual_target_
125	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	119	10	
126	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	118	10	
127	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	207	117	10	
128	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	205	116	10	
129	Royal Challengers Bangalore	Sunrisers Hyderabad	Hyderabad	201	115	10	
...	...	...	...	...	...	...	
149573	Chennai Super Kings	Mumbai Indians	Hyderabad	1	4	5	
149574	Chennai Super Kings	Mumbai Indians	Hyderabad	-1	3	5	
149575	Chennai Super Kings	Mumbai Indians	Hyderabad	-2	2	4	
149576	Chennai Super Kings	Mumbai Indians	Hyderabad	-4	1	4	
149577	Chennai Super Kings	Mumbai Indians	Hyderabad	-4	0	3	

72413 rows × 10 columns

In [59]:

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

In [60]:

```
#shuffling all the rows so that our model doesn not contain bias
final_df = final_df.sample(final_df.shape[0])
```

In [87]:

```
final_df.sample()  #prints any random delivery's instance.
```

Out[87]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets_left	actual_tai
104064	Sunrisers Hyderabad	Delhi Daredevils	Visakhapatnam	153	103	10	

In [62]:

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

Out[62]:

```
batting_team             0
bowling_team             0
city                   832
runs_left                0
balls_left               0
wickets_left             0
actual_target_score      0
crr                      0
rrr                      5
result                   0
dtype: int64
```

In [63]:

```
final_df.dropna(inplace=True)
```

In [64]:

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

In [65]:

```
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 [66]:

```
X_train
```

Out[66]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets_left	actual_target
122674	Chennai Super Kings	Kolkata Knight Riders	Chennai	25	10	6	
81145	Kings XI Punjab	Royal Challengers Bangalore	Bangalore	96	60	9	
58927	Kings XI Punjab	Rajasthan Royals	Jaipur	142	79	8	
71528	Delhi Daredevils	Kolkata Knight Riders	Pune	43	17	5	
146388	Royal Challengers Bangalore	Delhi Capitals	Delhi	38	18	5	
...	...	...	...	...	...	...	
13779	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata	111	96	9	
47084	Royal Challengers Bangalore	Mumbai Indians	Mumbai	84	46	5	
38546	Kings XI Punjab	Kolkata Knight Riders	Chandigarh	178	109	9	
4865	Delhi Daredevils	Sunrisers Hyderabad	Delhi	106	66	8	
44482	Royal Challengers Bangalore	Deccan Chargers	Nagpur	54	34	7	

57073 rows × 9 columns

In [67]:

```
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 [68]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
```

In [69]:

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

In [70]:

```
pipe.fit(X_train,y_train)
```

Out[70]:

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

In [71]:

```
y_pred = pipe.predict(X_test)
```

In [72]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[72]:

```
0.8001261475926834
```

In [73]:

```
pipe.predict_proba(X_test)[65]
```

Out[73]:

```
array([0.61701498, 0.38298502])
```

In [74]:

```
def match_progression(x_df,match_id,pipe):
    match = x_df[x_df['match_id'] == match_id] #extracting match_id
    match = match[(match['ball'] == 6)] #defining number of balls in a match
    new_df = match[['batting_team','bowling_team','city','runs_left','balls_left','wickets_

    t1 = new_df['batting_team'].unique() #print the names of batting teams and bowling team
    t2 = new_df['bowling_team'].unique()
    print(t1 +' '+ 'vs' +' '+ t2)

    new_df = new_df[new_df['balls_left'] != 0]
    result = pipe.predict_proba(new_df) #

    new_df['lose'] = np.round(result.T[0]*100,1) #finding lose probability after each over
    new_df['win'] = np.round(result.T[1]*100,1) #finding win probability after each over
    new_df['end_of_over'] = range(1,new_df.shape[0]+1) #completed over number

    target = new_df['actual_target_score'].values[0]
    runs = list(new_df['runs_left'].values)
    new_runs = runs[:]
    runs.insert(0,target)
    new_df['runs_after_over'] = np.array(runs)[:-1] - np.array(new_runs) #runs after each o

    wickets = list(new_df['wickets_left'].values)
    new_wickets = wickets[:]
    new_wickets.insert(0,10)
    wickets.append(0)
    w = np.array(wickets)
    nw = np.array(new_wickets)
    new_df['wickets_in_over'] = (nw - w)[0:new_df.shape[0]] #wickets after each over

    print("Target-",target)
    new_df = new_df[['end_of_over','runs_after_over','wickets_in_over','lose','win']]
    return new_df,target
```

```
In [75]:
```

```
new_df,target = match_progression(delivery_df,74,pipe)
new_df #data for match with match_id = 74
```

```
['Royal Challengers Bangalore vs Chennai Super Kings']
Target- 179
```

```
Out[75]:
```

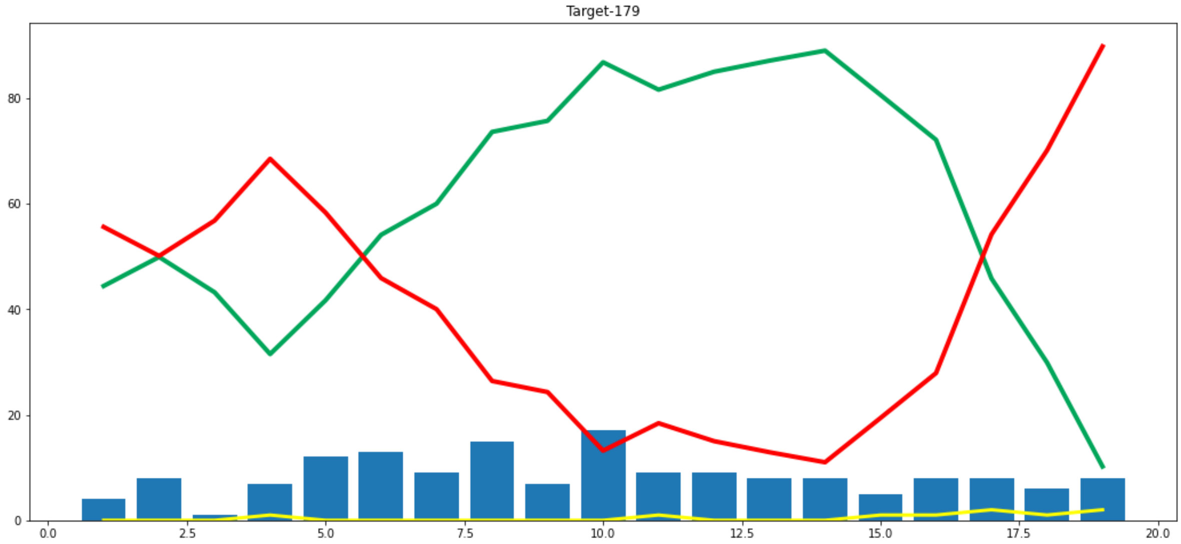
	end_of_over	runs_after_over	wickets_in_over	lose	win
10459	1	4	0	55.6	44.4
10467	2	8	0	50.1	49.9
10473	3	1	0	56.8	43.2
10479	4	7	1	68.5	31.5
10485	5	12	0	58.3	41.7
10491	6	13	0	45.9	54.1
10497	7	9	0	40.0	60.0
10505	8	15	0	26.4	73.6
10511	9	7	0	24.3	75.7
10518	10	17	0	13.2	86.8
10524	11	9	1	18.4	81.6
10530	12	9	0	15.0	85.0
10536	13	8	0	12.9	87.1
10542	14	8	0	11.0	89.0
10548	15	5	1	19.4	80.6
10555	16	8	1	27.9	72.1
10561	17	8	2	54.2	45.8
10567	18	6	1	70.1	29.9
10573	19	8	2	89.8	10.2

In [76]:

```
#Plotting graph for match with match_id = 74
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(new_df['end_of_over'],new_df['wickets_in_over'],color='yellow',linewidth=3)
plt.plot(new_df['end_of_over'],new_df['win'],color='#00a65a',linewidth=4)
plt.plot(new_df['end_of_over'],new_df['lose'],color='red',linewidth=4)
plt.bar(new_df['end_of_over'],new_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[76]:

```
Text(0.5, 1.0, 'Target-179')
```



In [77]:

```
new_df,target = match_progression(delivery_df,50,pipe)
new_df #data for match with match_id = 50
```

```
['Mumbai Indians vs Kings XI Punjab']
Target- 231
```

Out[77]:

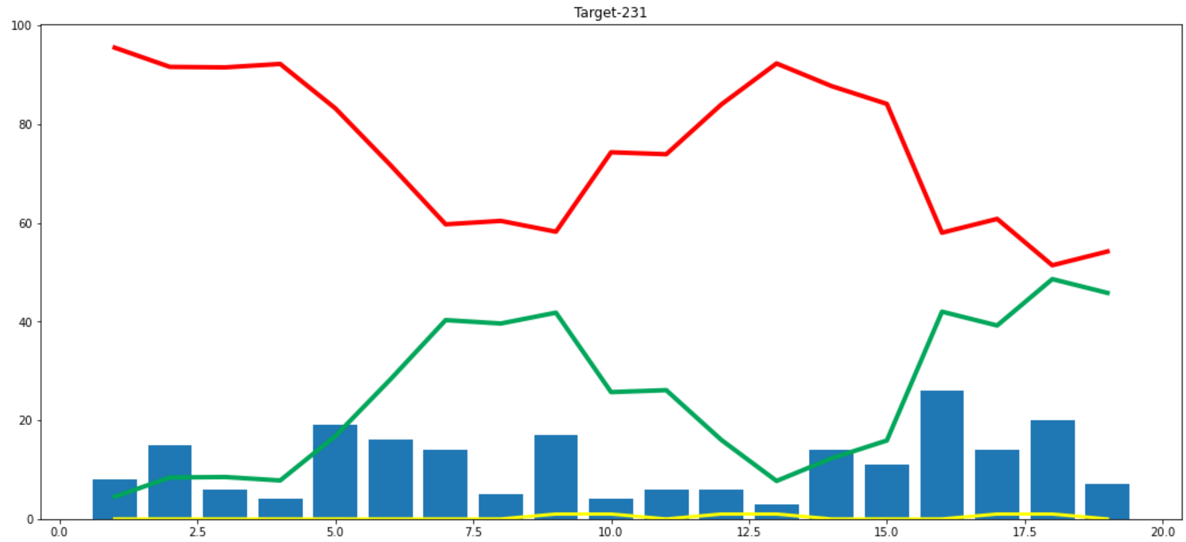
	end_of_over	runs_after_over	wickets_in_over	lose	win
6215	1	8	0	95.5	4.5
6222	2	15	0	91.6	8.4
6228	3	6	0	91.5	8.5
6234	4	4	0	92.2	7.8
6241	5	19	0	83.2	16.8
6247	6	16	0	71.7	28.3
6253	7	14	0	59.7	40.3
6259	8	5	0	60.4	39.6
6265	9	17	1	58.2	41.8
6271	10	4	1	74.3	25.7
6277	11	6	0	73.9	26.1
6284	12	6	1	84.0	16.0
6290	13	3	1	92.3	7.7
6296	14	14	0	87.7	12.3
6302	15	11	0	84.1	15.9
6308	16	26	0	58.0	42.0
6315	17	14	1	60.8	39.2
6322	18	20	1	51.4	48.6
6328	19	7	0	54.2	45.8

In [78]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(new_df['end_of_over'],new_df['wickets_in_over'],color='yellow',linewidth=3)
plt.plot(new_df['end_of_over'],new_df['win'],color='#00a65a',linewidth=4)
plt.plot(new_df['end_of_over'],new_df['lose'],color='red',linewidth=4)
plt.bar(new_df['end_of_over'],new_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[78]:

```
Text(0.5, 1.0, 'Target-231')
```



In [79]:

```
new_df,target = match_progression(delivery_df,200,pipe)
new_df #data for match with match_id = 200
```

```
['Kolkata Knight Riders vs Delhi Daredevils']
Target- 178
```

Out[79]:

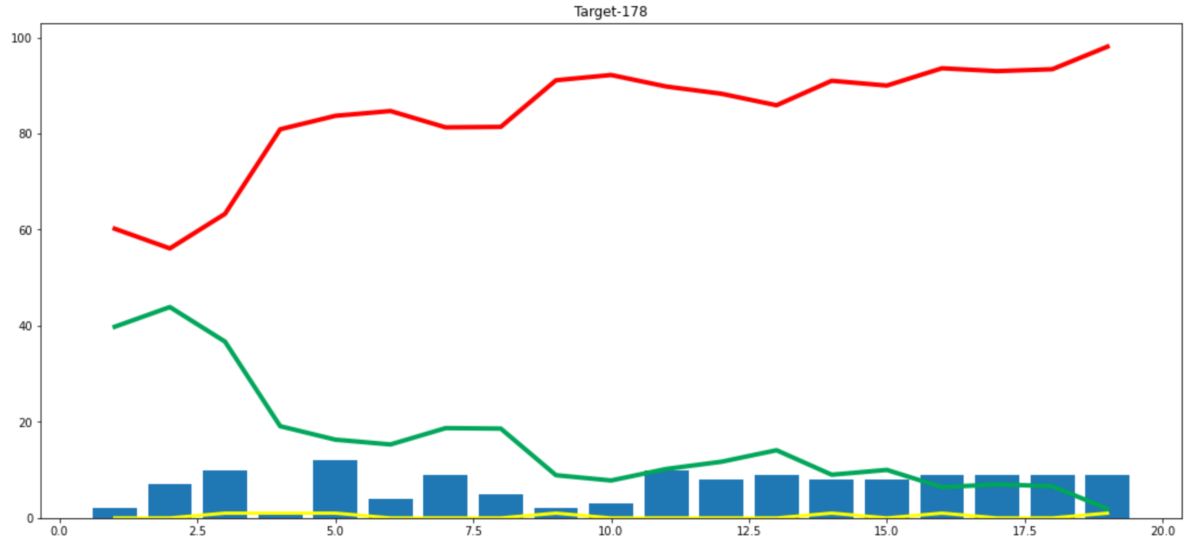
	end_of_over	runs_after_over	wickets_in_over	lose	win
39526	1	2	0	60.2	39.8
39533	2	7	0	56.1	43.9
39540	3	10	1	63.3	36.7
39546	4	1	1	80.9	19.1
39552	5	12	1	83.7	16.3
39558	6	4	0	84.7	15.3
39564	7	9	0	81.3	18.7
39570	8	5	0	81.4	18.6
39576	9	2	1	91.1	8.9
39582	10	3	0	92.2	7.8
39589	11	10	0	89.8	10.2
39596	12	8	0	88.3	11.7
39602	13	9	0	85.9	14.1
39608	14	8	1	91.0	9.0
39615	15	8	0	90.0	10.0
39621	16	9	1	93.6	6.4
39628	17	9	0	93.0	7.0
39634	18	9	0	93.4	6.6
39640	19	9	1	98.1	1.9

In [80]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(new_df['end_of_over'],new_df['wickets_in_over'],color='yellow',linewidth=3)
plt.plot(new_df['end_of_over'],new_df['win'],color='#00a65a',linewidth=4)
plt.plot(new_df['end_of_over'],new_df['lose'],color='red',linewidth=4)
plt.bar(new_df['end_of_over'],new_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[80]:

```
Text(0.5, 1.0, 'Target-178')
```



In [81]:

```
new_df,target = match_progression(delivery_df,69,pipe)
new_df #data for match with match_id = 69
```

```
['Mumbai Indians vs Kings XI Punjab']
Target- 183
```

Out[81]:

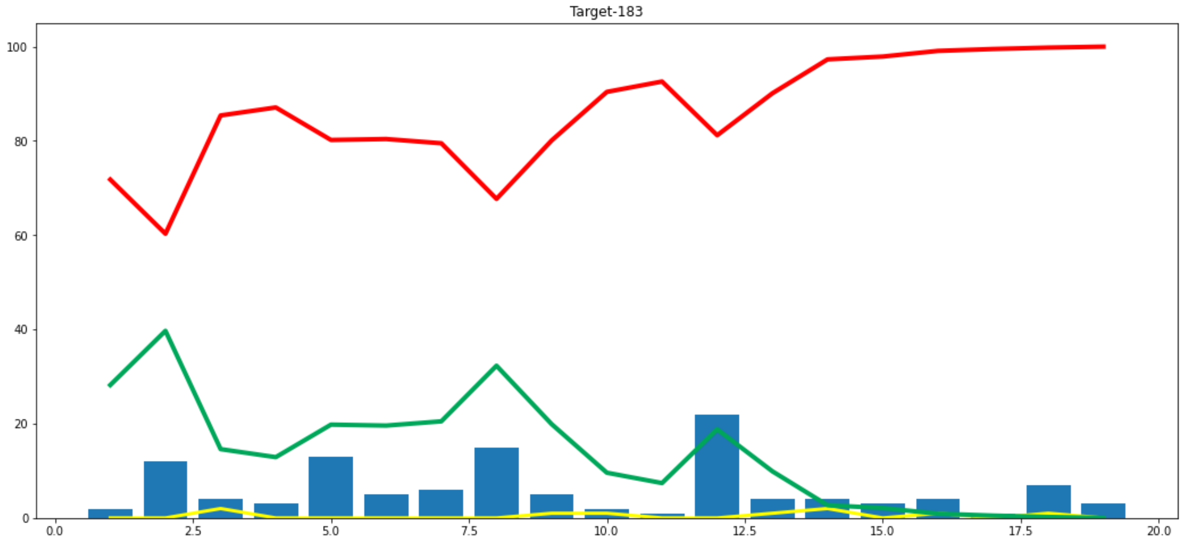
	end_of_over	runs_after_over	wickets_in_over	lose	win
9298	1	2	0	71.8	28.2
9304	2	12	0	60.3	39.7
9311	3	4	2	85.4	14.6
9317	4	3	0	87.1	12.9
9324	5	13	0	80.2	19.8
9330	6	5	0	80.4	19.6
9336	7	6	0	79.5	20.5
9343	8	15	0	67.7	32.3
9350	9	5	1	80.1	19.9
9356	10	2	1	90.4	9.6
9362	11	1	0	92.6	7.4
9368	12	22	0	81.2	18.8
9374	13	4	1	90.1	9.9
9380	14	4	2	97.3	2.7
9386	15	3	0	97.9	2.1
9392	16	4	1	99.1	0.9
9398	17	1	0	99.5	0.5
9404	18	7	1	99.8	0.2
9411	19	3	0	100.0	0.0

In [82]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(new_df['end_of_over'],new_df['wickets_in_over'],color='yellow',linewidth=3)
plt.plot(new_df['end_of_over'],new_df['win'],color='#00a65a',linewidth=4)
plt.plot(new_df['end_of_over'],new_df['lose'],color='red',linewidth=4)
plt.bar(new_df['end_of_over'],new_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[82]:

```
Text(0.5, 1.0, 'Target-183')
```



In [83]:

```
teams
```

Out[83]:

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

In [84]:

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

Out[84]:

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

In [85]:

```
import pickle
pickle.dump(pipe,open('pipe.pkl','wb'))
```

In [ ]:

In [ ]: