In [84]:

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

In [107]:

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

In [108]:

```
deliveries.head()
```

Out[108]:

	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 [109]:

```
ts_df=deliveries.groupby(['match_id','inning']).sum()['total_runs'].reset_index()#ts=toto
```

In [110]:

```
ts_df=ts_df[ts_df['inning']==1]
```

In [111]:

```
ts_df
```

Out[111]:

	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 [112]:

```
match.shape
```

Out[112]:

```
(756, 18)
```

In [113]:

```
match_df=match.merge(ts_df[['match_id','total_runs']],left_on='id',right_on='match_id')
match_df.head(3)
```

Out[113]:

	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

In [114]:

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

Out[114]:

```
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 [115]:

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

In [116]:

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

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

match_df['team1']=match_df['team1'].str.replace('Kings XI Punjab','King Panjab')
match_df['team2']=match_df['team2'].str.replace('Kings XI Punjab','King Panjab')

match_df['team1']=match_df['team1'].str.replace('Gujarat Lions','Gujarat Titans')
match_df['team2']=match_df['team2'].str.replace('Gujarat Lions','Gujarat Titans')
```

In [117]:

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

In [118]:

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

Out[118]:

```
array(['Sunrisers Hyderabad', 'Gujarat Titans',
       'Royal Challengers Bangalore', 'Kolkata Knight Riders',
       'Delhi Capitals', 'Mumbai Indians', 'Rajasthan Royals',
       'Chennai Super Kings'], dtype=object)
```

In [119]:

```
match_df.shape
```

Out[119]:

```
(502, 20)
```

In [120]:

```
match_df.head()
```

Out[120]:

	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
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Titans	Kolkata Knight Riders	Kolkata Knight Riders	field	normal
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Capitals	Royal Challengers Bangalore	bat	normal
5	6	IPL-2017	Hyderabad	09-04-2017	Gujarat Titans	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal
6	7	IPL-2017	Mumbai	09-04-2017	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	normal

In [121]:

```
match_df=match_df[match_df['dl_applied']==0] #dl_applied is 0 means where no rainfalls du
```

In [122]:

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

In [123]:

```
dl_df=match_df.merge(deliveries,on='match_id')# dl is deliveries
```

In [124]:

```
dl_df=dl_df[dl_df['inning']==2]
```

In [125]:

```
dl_df.sample()
```

Out[125]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	b
76761	508	Mumbai	Mumbai Indians	173	2	Delhi Daredevils	Mumbai Indians	2	

1 rows × 24 columns

In [126]:

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

In [127]:

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

In [128]:

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

In [129]:

```
dl_df.sample()
```

Out[129]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
111454	11318	Mumbai	Rajasthan Royals	191	2	Rajasthan Royals	Mumbai Indians	15

1 rows × 27 columns

In [130]:

```
#create the row for wrickets
dl_df['player_dismissed'] = dl_df['player_dismissed'].fillna("0")# Nan convert into 0
dl_df['player_dismissed'] = dl_df['player_dismissed'].apply(lambda x:x if x == "0" else "
dl_df['player_dismissed'] = dl_df['player_dismissed'].astype('int')# change into int
wickets = dl_df.groupby('match_id').cumsum()['player_dismissed'].values
dl_df['wickets'] = 10 - wickets
dl_df.head()
```

Out[130]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1

5 rows × 28 columns

In [131]:

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

In [132]:

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

In [133]:

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

In [181]:

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

In [182]:

```
final_df = dl_df[['batting_team','bowling_team','city','runs_left','balls_left','wickets'
```

In [183]:

```
final_df = final_df.sample(final_df.shape[0])
```

In [184]:

```
final_df.sample()
```

Out[184]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	
89723	Mumbai Indians	Royal Challengers Bangalore	Mumbai	81	54	8	170	8.090

In [185]:

```
final_df.shape
```

Out[185]:

```
(56775, 10)
```

In [186]:

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

In [187]:

```
final_df.head(2)
```

Out[187]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	
51195	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata	78	16	5	190	6.4(
88281	Sunrisers Hyderabad	Royal Challengers Bangalore	Bangalore	99	33	5	227	8.8:

In [188]:

```
final_df['batting_team']=final_df['batting_team'].str.replace('Delhi Daredevils','Delhi C
final_df['bowling_team']=final_df['bowling_team'].str.replace('Delhi Daredevils','Delhi C
```

In [189]:

```
final_df['batting_team']=final_df['batting_team'].str.replace('Gujarat Lions','Gujarat Ti
final_df['bowling_team']=final_df['bowling_team'].str.replace('Gujarat Lions','Gujarat Ti
```

In [190]:

```
final_df['bowling_team']=final_df['bowling_team'].str.replace('Deccan Chargers','Sunriser
final_df['batting_team']=final_df['batting_team'].str.replace('Deccan Chargers','Sunriser
```

In [191]:

```
final_df['bowling_team']=final_df['bowling_team'].str.replace('Kings XI Punjab','King Par
final_df['batting_team']=final_df['batting_team'].str.replace('Kings XI Punjab','King Par
```

In [192]:

```
final_df['bowling_team'].unique()
```

Out[192]:

```
array(['Kolkata Knight Riders', 'Royal Challengers Bangalore',
       'Chennai Super Kings', 'Gujarat Titans', 'Sunrisers Hyderabad',
       'Rajasthan Royals', 'Mumbai Indians', 'Delhi Capitals'],
      dtype=object)
```

In [193]:

```
final_df['batting_team'].unique()
```

Out[193]:

```
array(['Royal Challengers Bangalore', 'Sunrisers Hyderabad',
       'Delhi Capitals', 'Mumbai Indians', 'Kolkata Knight Riders',
       'Rajasthan Royals', 'Chennai Super Kings', 'Gujarat Titans'],
      dtype=object)
```

In [194]:

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56596 entries, 51195 to 110880
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   batting_team  56596 non-null  object 
 1   bowling_team  56596 non-null  object 
 2   city          55885 non-null  object 
 3   runs_left     56596 non-null  int64  
 4   balls_left    56596 non-null  int64  
 5   wickets       56596 non-null  int32  
 6   total_runs_x  56596 non-null  int64  
 7   crr           56596 non-null  float64
 8   rrr           56596 non-null  float64
 9   result        56596 non-null  int64  
dtypes: float64(2), int32(1), int64(4), object(3)
memory usage: 4.5+ MB
```

In [195]:

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

In [196]:

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55885 entries, 51195 to 110880
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   batting_team  55885 non-null  object 
 1   bowling_team  55885 non-null  object 
 2   city          55885 non-null  object 
 3   runs_left     55885 non-null  int64  
 4   balls_left    55885 non-null  int64  
 5   wickets       55885 non-null  int32  
 6   total_runs_x  55885 non-null  int64  
 7   crr           55885 non-null  float64
 8   rrr           55885 non-null  float64
 9   result        55885 non-null  int64  
dtypes: float64(2), int32(1), int64(4), object(3)
memory usage: 4.5+ MB
```

In [197]:

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

In [198]:

```
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 [199]:

```
X_train
```

Out[199]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	
72660	Royal Challengers Bangalore	Mumbai Indians	Mumbai	109	71	9	187	9
11069	Delhi Capitals	Mumbai Indians	Mumbai	40	20	4	162	7
83204	Rajasthan Royals	Mumbai Indians	Mumbai	25	12	5	187	9
32317	Chennai Super Kings	Royal Challengers Bangalore	Chennai	2	6	5	161	8
39499	Rajasthan Royals	Delhi Capitals	Jaipur	17	19	6	151	7
...	...	...	...	...	...	...	...	
88781	Kolkata Knight Riders	Sunrisers Hyderabad	Hyderabad	30	29	8	142	7
83322	Royal Challengers Bangalore	Kolkata Knight Riders	Bangalore	48	83	8	111	10
11279	Rajasthan Royals	Chennai Super Kings	Jaipur	16	46	8	109	7
67469	Mumbai Indians	Rajasthan Royals	Kolkata	33	21	6	165	8
39218	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata	89	51	7	163	6

44708 rows × 9 columns

In [200]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

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

In [201]:

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

In [202]:

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

In [203]:

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

Out[203]:

```
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 [204]:

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

In [205]:

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

Out[205]:

```
0.819808535385166
```

In [206]:

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

Out[206]:

```
array([0.99584111, 0.00415889])
```

In [207]:

```
def match_summary(row):
    print("Batting Team-" + row['batting_team'] + " | Bowling Team-" + row['bowling_team'
```

In [208]:

```
def match_progression(x_df,match_id,pipe):
    match = x_df[x_df['match_id'] == match_id]
    match = match[(match['ball'] == 6)]
    temp_df = match[['batting_team','bowling_team','city','runs_left','balls_left','wicke
    temp_df = temp_df[temp_df['balls_left'] != 0]
    result = pipe.predict_proba(temp_df)
    temp_df['lose'] = np.round(result.T[0]*100,1)
    temp_df['win'] = np.round(result.T[1]*100,1)
    temp_df['end_of_over'] = range(1,temp_df.shape[0]+1)

    target = temp_df['total_runs_x'].values[0]
    runs = list(temp_df['runs_left'].values)
    new_runs = runs[:]
    runs.insert(0,target)
    temp_df['runs_after_over'] = np.array(runs)[:-1] - np.array(new_runs)
    wickets = list(temp_df['wickets'].values)
    new_wickets = wickets[:]
    new_wickets.insert(0,10)
    wickets.append(0)
    w = np.array(wickets)
    nw = np.array(new_wickets)
    temp_df['wickets_in_over'] = (nw - w)[0:temp_df.shape[0]]

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

In [210]:

```
temp_df,target = match_progression(dl_df,74,pipe)# 74 match number
temp_df
```

Target- 178

Out[210]:

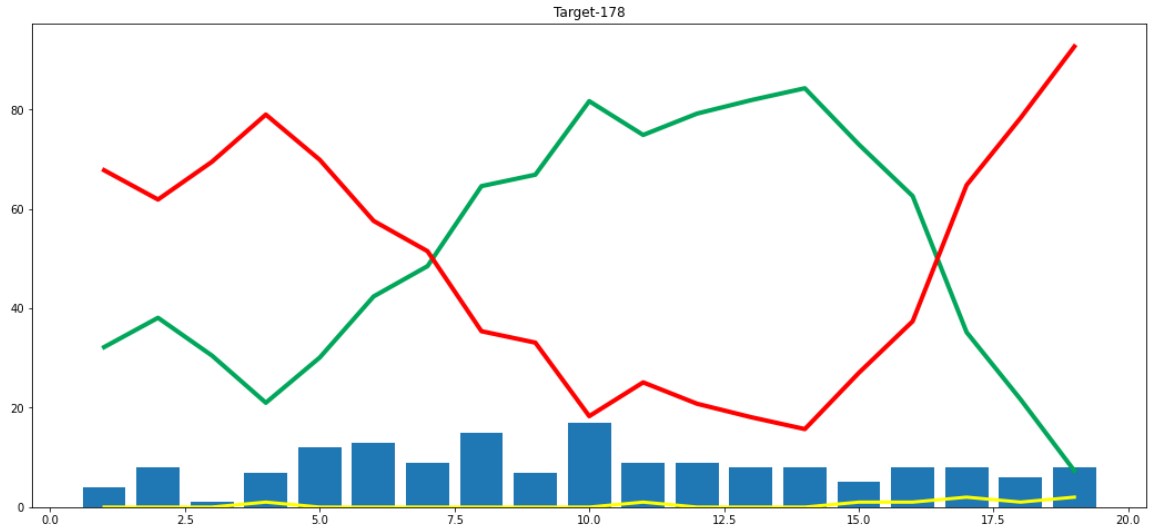
	end_of_over	runs_after_over	wickets_in_over	lose	win
9507	1	4	0	67.8	32.2
9515	2	8	0	61.9	38.1
9521	3	1	0	69.5	30.5
9527	4	7	1	79.0	21.0
9533	5	12	0	69.9	30.1
9539	6	13	0	57.6	42.4
9545	7	9	0	51.5	48.5
9553	8	15	0	35.4	64.6
9559	9	7	0	33.1	66.9
9566	10	17	0	18.3	81.7
9572	11	9	1	25.1	74.9
9578	12	9	0	20.8	79.2
9584	13	8	0	18.1	81.9
9590	14	8	0	15.7	84.3
9596	15	5	1	27.0	73.0
9603	16	8	1	37.4	62.6
9609	17	8	2	64.8	35.2
9615	18	6	1	78.3	21.7
9621	19	8	2	92.7	7.3

In [212]:

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

Out[212]:

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



In [171]:

```
teams
```

Out[171]:

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

In [172]:

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

Out[172]:

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

In [ ]:

```
```

In [ ]:

```
```