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

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

```
match.head()
```

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

```
match.shape
```

```
(756, 18)
```

```
delivery.head()
```

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bo
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	
3	1	1	Sunrisers	Royal	1	4	DA	S Dhawan	

```
total_score_df = delivery.groupby(['match_id','inning']).sum()['total_runs'].reset_index()
```

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

```
total_score_df
```

	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

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

```
match_df
```

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

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

```
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)
```

```
teams = [
    'Sunrisers Hyderabad',
    'Mumbai Indians',
```

```
        'Royal Challengers Bangalore',
        'Kolkata Knight Riders',
        'Kings XI Punjab',
        'Chennai Super Kings',
        'Rajasthan Royals',
        'Delhi Capitals'
    ]

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


    match_df.shape
```

```
        (641, 20)
```

```
    match_df.shape
```

```
        (641, 20)
```

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


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


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


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


    delivery_df
```

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

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

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

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

```
delivery_df
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_t
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunr Hydera
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunr Hydera
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunr Hydera
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunr Hydera
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunr Hydera

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

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_tea
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderabac
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderabac
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderabac
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderabac
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderabac

5 rows × 28 columns

```
delivery_df.head()
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_tea
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderaba
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderaba
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderaba
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderaba
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunriser Hyderaba

5 rows × 28 columns

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

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

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

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

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

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

final_df.sample()
```

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_run
	Kolkata Knight	Sunrisers					

```
final_df.dropna(inplace=True)

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

```
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)
```

```
X_train
```

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total
148310	Kings XI Punjab	Chennai Super Kings	Mohali	75	71	10	
117096	Mumbai Indians	Kolkata Knight Riders	Mumbai	73	49	7	
11463	Kolkata Knight Riders	Rajasthan Royals	Jaipur	152	90	8	
138150	Rajasthan Royals	Chennai Super Kings	Chennai	81	38	5	
88451	Delhi Daredevils	Chennai Super Kings	Abu Dhabi	124	59	5	
...	...	...	...	...	...	...	
63495	Kings XI Punjab	Mumbai Indians	Mumbai	78	50	8	
141209	Chennai Super Kings	Rajasthan Royals	Jaipur	158	116	9	

```
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')
```

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

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

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

```
Pipeline(steps=[('step1',
                 ColumnTransformer(remainder='passthrough',
```

```
                                        transformers=[('trf',
                                                       OneHotEncoder(drop='first',
                                                                     sparse=False),
                                                       ['batting_team',
                                                        'bowling_team', 'city'])])),
                      ('step2', LogisticRegression(solver='liblinear'))])
```

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

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

```
0.8050318873081506
```

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

```
array([0.32093402, 0.67906598])
```

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

```
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','wicket
    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
```

```
temp_df,target = match_progression(delivery_df,74,pipe)
temp_df
```

Target- 178

	end_of_over	runs_after_over	wickets_in_over	lose	win
10459	1	4	0	56.6	43.4
10467	2	8	0	51.2	48.8
10473	3	1	0	58.0	42.0
10479	4	7	1	69.5	30.5
10485	5	12	0	59.4	40.6
10491	6	13	0	47.1	52.9
10497	7	9	0	41.2	58.8
10505	8	15	0	27.4	72.6
10511	9	7	0	25.3	74.7
10518	10	17	0	13.8	86.2
10524	11	9	1	19.2	80.8
10530	12	9	0	15.7	84.3
10536	13	8	0	13.5	86.5
10542	14	8	0	11.6	88.4
10548	15	5	1	20.2	79.8
10555	16	8	1	28.7	71.3
10561	17	8	2	54.9	45.1
10567	18	6	1	70.1	29.9
10573	19	8	2	89.3	10.7

```
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))
```

```
teams
```

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

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

```
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)
```

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