```
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	result	dl_applied	winner	win_by_runs	win_b
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Sunrisers Hyderabad	35	
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	0	Rising Pune Supergiant	0	
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	0	Royal Challengers Bangalore	15	

```
match.shape
```

```
(756, 18)
```

```
delivery.head()
```

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_super_over	...	bye_runs	legbye_runs
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	0	...	0	0
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	0	...	0	0
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	0	...	0	0
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	0	...	0	0
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	0	...	0	0

5 rows × 21 columns

```
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='match_id')
```

```
match_df
```

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

756 rows × 20 columns

```
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 = 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_team	over	ball	batsman	...	bye_runs	legbye_
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	CH Gayle	...	0	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandeep Singh	...	0	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandeep Singh	...	0	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandeep Singh	...	0	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandeep Singh	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	RA Jadeja	...	0	
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	SR Watson	...	0	
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	SR Watson	...	0	
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	SN Thakur	...	0	
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	SN Thakur	...	0	

72413 rows × 24 columns

```
# Convert 'total_runs_y' column to numeric type
delivery_df['total_runs_y'] = pd.to_numeric(delivery_df['total_runs_y'], errors='coerce')

# Apply cumulative sum within groups
delivery_df['current_score'] = delivery_df.groupby('match_id')['total_runs_y'].cumsum()
```

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

```
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_team	over	ball	batsman	...	penalty_runs	ba
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	CH Gayle	...	0	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandeep Singh	...	0	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandeep Singh	...	0	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandeep Singh	...	0	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandeep Singh	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	2	RA Jadeja	...	0	
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	3	SR Watson	...	0	
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	4	SR Watson	...	0	
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	5	SN Thakur	...	0	
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20	6	SN Thakur	...	0	

72413 rows × 27 columns

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

```
[nan 'Mandeep Singh' 'CH Gayle' 'KM Jadhav' 'TM Head' 'Sachin Baby'
 'STR Binny' 'SR Watson' 'S Aravind' 'TS Mills' 'YS Chahal' 'AP Tare'
 'KK Nair' 'SW Billings' 'SV Samson' 'CH Morris' 'CR Brathwaite'
 'PJ Cummins' 'RR Pant' 'S Nadeem' 'PA Patel' 'JC Buttler' 'RG Sharma'
 'KH Pandya' 'KA Pollard' 'N Rana' 'M Vohra' 'AR Patel' 'SP Narine'
 'RV Uthappa' 'MJ McClenaghan' 'S Dhawan' 'DA Warner' 'MC Henriques'
 'DJ Hooda' 'Yuvraj Singh' 'BCJ Cutting' 'WP Saha' 'HM Amla' 'EJG Morgan'
 'GJ Maxwell' 'DA Miller' 'MM Sharma' 'KC Cariappa' 'C de Grandhomme'
 'G Gambhir' 'YK Pathan' 'SA Yadav' 'CR Woakes' 'I Sharma' 'AD Mathews'
 'SS Iyer' 'CJ Anderson' 'K Rabada' 'V Kohli' 'AB de Villiers' 'P Negi'
 'S Badree' 'MK Pandey' 'MJ Guptill' 'SE Marsh' 'Anureet Singh'
 'SP Jackson' 'KV Sharma' 'A Choudhary' 'MN Samuels' 'Mohammed Shami'
 'Z Khan' 'CA Lynn' 'LMP Simmons' 'HH Pandya' 'Kuldeep Yadav' 'A Mishra'
 'AT Rayudu' 'R Dravid' 'JH Kallis' 'W Jaffer' 'MV Boucher' 'B Akhil'
 'CL White' 'AA Noffke' 'SB Joshi' 'K Goel' 'JR Hopes' 'KC Sangakkara'
 'V Sehwag' 'S Chanderpaul' 'LRPL Taylor' 'BB McCullum' 'RT Ponting'
 'SC Ganguly' 'Mohammad Hafeez' 'M Kaif' 'Kamran Akmal' 'DS Lehmann'
 'L Ronchi' 'ST Jayasuriya' 'SM Pollock' 'DJ Bravo' 'MA Khote'
 'Harbhajan Singh' 'GC Smith' 'RA Jadeja' 'D Salunkhe' 'SS Tiwary'
 'AM Nayar' 'M Rawat' 'DPMD Jayawardene' 'SM Katich' 'TM Srivastava'
 'IK Pathan' 'B Chipli' 'P Kumar' 'DW Steyn' 'AM Rahane' 'RR Sarwan'
 'Salman Butt' 'AB Agarkar' 'BJ Hodge' 'LR Shukla' 'DJ Hussey' 'DB Das'
 'Umar Gul' 'AB Dinda' 'Misbah-ul-Haq' 'Shoaib Malik' 'KD Karthik'
 'PJ Sangwan' 'R Bhatia' 'Mohammad Asif' 'GD McGrath' 'SA Asnodkar'
 'HH Gibbs' 'AC Gilchrist' 'YV Takawale' 'S Vidyut' 'SP Fleming'
 'SK Raina' 'JA Morkel' 'MS Dhoni' 'CK Kapugedera' 'J Arunkumar'
 'PP Chawla' 'VRV Singh' 'DB Ravi Teja' 'SB Styris' 'SB Bangar'
 'WPUJC Vaas' 'AD Mascarenhas' 'SK Warne' 'MK Tiwary' 'TM Dilshan'
 'MF Maharoof' 'VY Mahesh' 'SR Tendulkar' 'Younis Khan' 'Niraj Patel'
 'Shahid Afridi' 'Y Venugopal Rao' 'PP Ojha' 'RP Singh' 'DT Patil'
 'A Kumble' 'LPC Silva' 'H Das' 'Sohail Tanvir' 'DR Smith' 'PR Shah'
 'SD Chitnis' 'A Nehra' 'VS Yeligati' 'S Badrinath' 'MS Gony' 'L Balaji'
 'LA Pomersbach' 'A Mukund' 'VVS Laxman' 'A Chopra' 'A Flintoff'
 'ML Hayden' 'JDP Oram' 'T Henderson' 'MM Patel' 'Kamran Khan'
 'KP Pietersen' 'R Bishnoi' 'R Vinay Kumar' 'JD Ryder' 'Joginder Sharma'
 'Yashpal Singh' 'RS Bopara' 'JP Duminy' 'RJ Quiney' 'AS Raut' 'AN Ghosh'
 'BAW Mendis' 'PC Valthaty' 'SP Goswami' 'RE van der Merwe' 'SK Trivedi'
 'MN van Wyk' 'NV Ojha' 'LA Carseldine' 'M Manhas' 'S Sohal' 'RJ Harris'
 'TL Suman' 'WA Mota' 'M Vijay' 'M Morkel' 'A Symonds' 'Shoaib Ahmed'
 'DS Kulkarni' 'C Nanda' 'SL Malinga' 'J Botha' 'A Singh' 'SS Shaikh'
 'B Lee' 'AA Bilakhia' 'Anirudh Singh' 'AA Jhunjhunwala' 'P Dogra'
 'A Uniyal' 'JM Kemp' 'R Ashwin' 'M Muralitharan' 'OA Shah' 'RS Gavaskar'
 'SE Bond' 'M Kartik' 'DP Nannes' 'MS Bisla' 'AB Barath' 'CA Pujara'
 'Y Nagar' 'S Ladda' 'GJ Bailey' 'J Theron' 'SJ Srivastava' 'R Sathish'
 'MJ Lumb' 'MD Mishra' 'Jaskaran Singh' 'KAJ Roach' 'CK Langeveldt'
 'FY Fazal' 'AC Voges' 'S Narwal' 'SW Tait' 'R Sharma' 'A Mithun'
 'Harmeet Singh' 'R McLaren' 'PD Collingwood' 'S Sriram' 'AP Dole'
 'KP Appanna' 'MR Marsh' 'Pankaj Singh' 'B Sumanth' 'AG Paunikar'
 'DJ Jacobs' 'IR Jaggi' 'DT Christian' 'AL Menaria' 'Sunny Singh'
 'MA Agarwal' 'JJ van der Wath' 'R Ninan' 'AUK Pathan' 'AJ Finch'
 'MEK Hussey' 'S Anirudha' 'S Randiv' 'TR Birt' 'Bipul Sharma' 'CA Ingram'
 'DH Yagnik' 'AC Blizzard' 'AB McDonald' 'KB Arun Karthik' 'BA Bhatt'
 'JEC Franklin' 'DL Vettori' 'RE Levi' 'DJ Harris' 'Ankit Sharma'
 'HV Patel' 'KK Cooper' 'AA Chavan' 'GB Hogg' 'F du Plessis'
 'Shakib Al Hasan' 'RN ten Doeschate' 'N Saini' 'Azhar Mahmood'
 'RJ Peterson' 'A Ashish Reddy' 'Gurkeerat Singh' 'PA Reddy' 'UBT Chand'
 'AD Russell' 'UT Yadav' 'Sunny Gupta' 'VR Aaron' 'MC Juneja' 'B Laughlin'
 'NLTC Perera' 'BMAJ Mendis' 'GH Vihari' 'SMSM Senanayake' 'KL Rahul'
```

```
# Convert 'player_dismissed' values to 0 if they are "0" (not dismissed) and to 1 otherwise (dismissed)
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x: 0 if x == "0" else 1)

# Convert the column to the integer type, handling any errors by coercing non-convertible values to NaN
delivery_df['player_dismissed'] = pd.to_numeric(delivery_df['player_dismissed'], errors='coerce')

# Drop rows with NaN values (if any)
delivery_df.dropna(subset=['player_dismissed'], inplace=True)

# Apply cumulative sum within groups
wickets = delivery_df.groupby('match_id')['player_dismissed'].cumsum().values
```

```
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0")
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x if x == "0" else "1")
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int')
wickets = delivery_df.groupby('match_id')['player_dismissed'].cumsum().values
delivery_df['wickets'] = 10 - wickets
delivery_df.head()
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsman	...	batsman_runs	extra_
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	CH Gayle	...	1	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandeep Singh	...	0	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandeep Singh	...	0	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandeep Singh	...	2	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandeep Singh	...	4	

5 rows × 28 columns

```
delivery_df.head()
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batsman	...	batsman_runs	extra_
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1	CH Gayle	...	1	
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2	Mandeep Singh	...	0	
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3	Mandeep Singh	...	0	
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4	Mandeep Singh	...	2	
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5	Mandeep Singh	...	4	

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','total_runs_x','crr','rrr','result']]
```

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

```
final_df.sample()
```

	batting_team	bowling_team	city	runs_left	balls_left	total_runs_x	cr
		Royal					

Start coding or generate with AI.

```
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	total_runs_x
100115	Chennai Super Kings	Mumbai Indians	Mumbai	169	117	173
71457	Delhi Daredevils	Kolkata Knight Riders	Pune	114	87	162
55219	Mumbai Indians	Kings XI Punjab	Chandigarh	130	82	163
51566	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata	97	72	171
46776	Kolkata Knight Riders	Mumbai Indians	Kolkata	125	111	133
...	...	...	...	...	...	...
36445	Kolkata Knight Riders	Rajasthan Royals	Ahmedabad	38	8	168
	Deccan	Chennai				

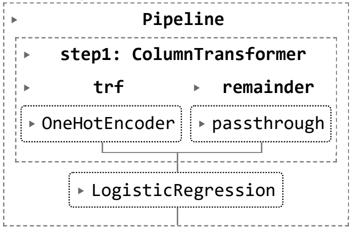
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder

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)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:868: FutureWarning: `sparse` was renamed to `sparse_outpu
  warnings.warn(
```


Pipeline
step1: ColumnTransformer
trf
remainder
OneHotEncoder
passthrough
LogisticRegression

```
X_test
```

	batting_team	bowling_team	city	runs_left	balls_left	total_runs_x	
62359	Rajasthan Royals	Deccan Chargers	Jaipur	7	4	196	
18209	Delhi Daredevils	Mumbai Indians	Delhi	126	92	176	1
69011	Mumbai Indians	Royal Challengers Bangalore	Bangalore	169	115	171	
99925	Kings XI Punjab	Kolkata Knight Riders	Kolkata	103	67	163	
133929	Mumbai Indians	Delhi Daredevils	Delhi	142	91	186	
...	...	...	...	...	...	...	
9057	Rajasthan Royals	Deccan Chargers	Hyderabad	200	109	214	
	Chennai	Rajasthan					

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

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

```
0.7955007358609574
```

```
pipe.predict_proba(X_test)[20]
```

```
array([0.18560087, 0.81439913])
```

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

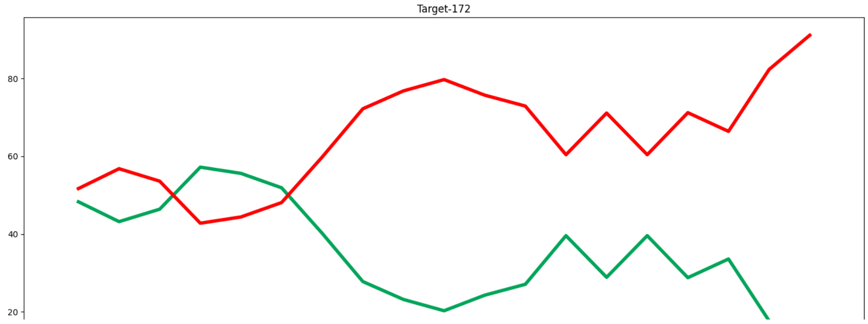
```
def match_progression(x_df,match_id,pipe):
    match = x_df[x_df['match_id'] == match_id]
    match = match[(match['ball'] == 6)]
    temp_df = match[['batting_team','bowling_team','city','runs_left','balls_left','wickets','total_runs_x','crr','rrr']].dropna()
    temp_df = temp_df[temp_df['balls_left'] != 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','lose','win','wickets_in_over']]
    return temp_df,target
```

```
temp_df,target = match_progression(delivery_df,9,pipe)
temp_df
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-77-5fc351173c15> in <cell line: 1>()
----> 1 temp_df,target = match_progression(delivery_df,9,pipe)
      2 temp_df

                                  14 frames
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py in
check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy,
force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features,
estimator, input_name)
    929         n_samples = _num_samples(array)
    930         if n_samples < ensure_min_samples:
--> 931             raise ValueError(
    932                 "Found array with %d sample(s) (shape=%s) while a"
    933                 " minimum of %d is required%s."
```

```
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))
```

```
Text(0.5, 1.0, 'Target-172')
```


Target-172