```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
import numpy as np

import pandas as pd
```

```
match=pd.read_csv('/content/drive/MyDrive/Data Science Project/IPL Prediction/matches.csv
delivery=pd.read_csv('/content/drive/MyDrive/Data Science Project/IPL Prediction/deliveri
```

```
match.head()
```

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

```
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 Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	

5 rows × 21 columns

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

```
total_score_df
```

	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

```
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='mat
```

```
match_df
```

	id	Season	city	date	team1	team2	toss_winner	toss_dec
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	

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_t
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi

72413 rows × 24 columns

```
delivery_df.shape
```

```
(72413, 24)
```

```
print(delivery_df.dtypes)
```

```
match_id           int64
city              object
winner            object
total_runs_x       int64
inning             int64
batting_team      object
bowling_team      object
over               int64
ball               int64
batsman           object
non_striker       object
```

```
bowler              object
is_super_over        int64
wide_runs            int64
bye_runs             int64
legbye_runs          int64
noball_runs          int64
penalty_runs         int64
batsman_runs         int64
extra_runs           int64
total_runs_y         int64
player_dismissed    object
dismissal_kind      object
fielder             object
current_score        int64
dtype: object
```

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

```
delivery_df
```

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

72413 rows × 25 columns

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

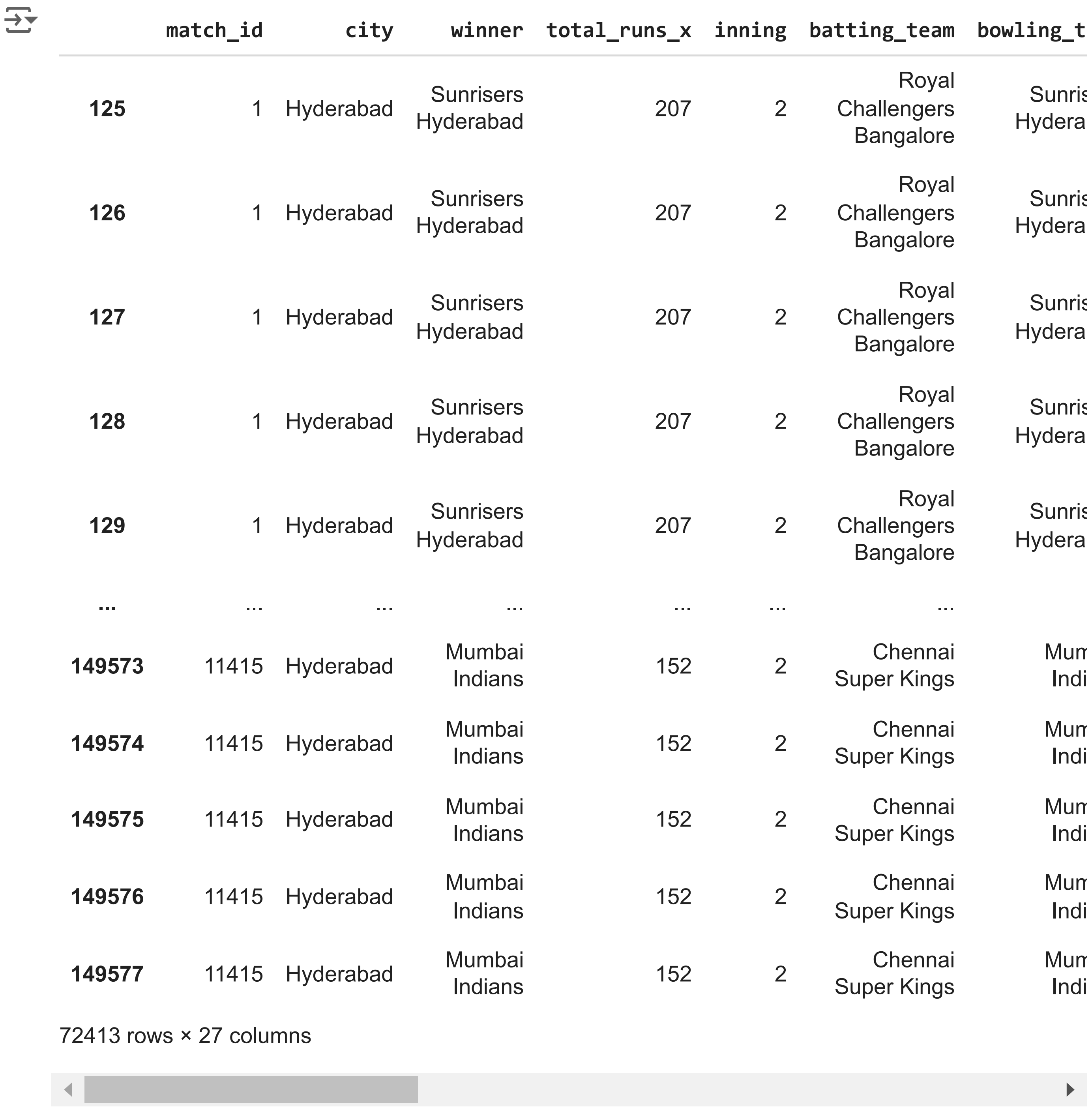
```
delivery_df
```

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

72413 rows × 26 columns

```
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	Sunris Hydera
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunris Hydera
...	...	...	...	...	...	...	
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mun Indi

72413 rows × 27 columns

```
# Step 1: Fill NaN values in 'player_dismissed' with '0'
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna('0')

# Step 2: Convert 'player_dismissed' to numeric (1 for dismissal, 0 otherwise)
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x: x if x

# Ensure the column is of integer type
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int')

# Step 3: Perform the cumulative sum operation by group
wickets= delivery_df.groupby('match_id')['player_dismissed'].cumsum().values

# Step 4: If needed, adjust the wickets to reflect the remaining wickets
delivery_df['wickets'] = 10 - wickets

# Display the head of the DataFrame
delivery_df.head()
```

	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

5 rows × 28 columns

```
delivery_df.tail()
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_tean
149573	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumba Indians
149574	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumba Indians
149575	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumba Indians
149576	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumba Indians
149577	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumba Indians

5 rows × 28 columns

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

```
#rrr=
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','wick
```

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

```
final_df.sample()
```

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs
[illegible]	Kings XI	Kolkata Knight	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

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

```
                 0
batting_team     0
bowling_team     0
city           832
runs_left        0
balls_left       0
wickets          0
total_runs_x     0
crr              0
rrr              7
result           0

dtype: int64
```

Double-click (or enter) to edit

```
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_
9412	Mumbai Indians	Kings XI Punjab	Chandigarh	66	5	1	
18686	Deccan Chargers	Royal Challengers Bangalore	Bangalore	145	106	10	
92459	Kolkata Knight Riders	Rajasthan Royals	Ahmedabad	95	65	10	
61784	Royal Challengers Bangalore	Rajasthan Royals	Bangalore	188	113	10	
11190	Kings XI Punjab	Deccan Chargers	Hyderabad	154	115	10	
...	...	...	...	...	...	...	
65186	Rajasthan Royals	Delhi Daredevils	Delhi	117	88	10	

```
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
```

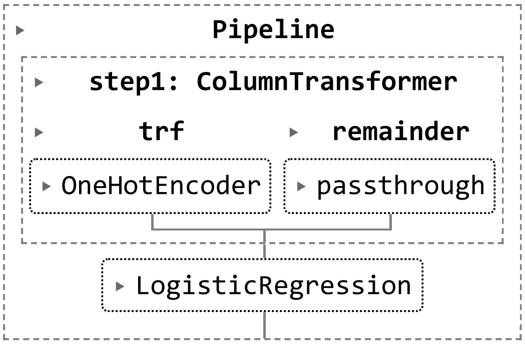
```
X_train.describe()
```

	runs_left	balls_left	wickets	total_runs_x	crr	r
count	57073.000000	57073.000000	57073.000000	57073.000000	57073.000000	57073.0000
mean	92.439560	62.792634	7.549700	165.660155	7.446988	10.3648
std	49.969024	33.299819	2.125509	29.252027	2.267280	13.6439
min	-12.000000	-2.000000	0.000000	65.000000	0.000000	-510.0000
25%	53.000000	35.000000	6.000000	147.000000	6.268657	7.1428
50%	92.000000	63.000000	8.000000	165.000000	7.487179	8.8732
75%	130.000000	92.000000	9.000000	185.000000	8.700000	10.8947
max	249.000000	119.000000	10.000000	250.000000	42.000000	582.0000

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

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:975: Futur
  warnings.warn(
```



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

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

```
0.8004064755764244
```

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

```
array([0.15330707, 0.84669293])
```

```
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','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[:]

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	57.0	43.0