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

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

```
match.head()
```

	id	season	city	date	match_type	player_of_match	venue	team1	team2	toss_winner	toss_decision	
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Shah Kotla	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhede Stadium	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	C
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Eden Gardens	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	

Next steps: Generate code with match | View recommended plots | New interactive sheet

```
match.shape
```

```
(1095, 20)
```

```
delivery.head()
```

	match_id	inning	batting_team	bowling_team	over	ball	batter	bowler	non_striker	batsman_runs	extra_runs	total_runs	ext
0	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	1	SC Ganguly	P Kumar	BB McCullum	0	1	1	
1	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	2	BB McCullum	P Kumar	SC Ganguly	0	0	0	
2	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	3	BB McCullum	P Kumar	SC Ganguly	0	1	1	
3	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	4	BB McCullum	P Kumar	SC Ganguly	0	0	0	
4	335982	1	Kolkata Knight Riders	Royal Challengers Bangalore	0	5	BB McCullum	P Kumar	SC Ganguly	0	0	0	

```
total_score_df = delivery.groupby(['match_id','inning']).sum()['total_runs'].reset_index() #we use reset_index to convert it into a data
```

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

```
total_score_df
```

	match_id	inning	total_runs
0	335982	1	222
2	335983	1	240
4	335984	1	129
6	335985	1	165
8	335986	1	110
...	...	...	...
2207	1426307	1	214
2209	1426309	1	159
2211	1426310	1	172
2213	1426311	1	175
2215	1426312	1	113

1095 rows × 3 columns

Next steps: Generate code with total_score_df | View recommended plots | New interactive sheet

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

```
match_df
```

	id	season	city	date	match_type	player_of_match	venue	team1	team2	toss_winner	...	result
0	335982	2007/08	Bangalore	2008-04-18	League	BB McCullum	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	...	runs
1	335983	2007/08	Chandigarh	2008-04-19	League	MEK Hussey	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	...	runs
2	335984	2007/08	Delhi	2008-04-19	League	MF Maharoof	Feroz Shah Kotla	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	...	wickets
3	335985	2007/08	Mumbai	2008-04-20	League	MV Boucher	Wankhede Stadium	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	...	wickets
4	335986	2007/08	Kolkata	2008-04-20	League	DJ Hussey	Eden Gardens	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	...	wickets
...	...	...	...	...	...	...	...	...	...	...	...	...
1090	1426307	2024	Hyderabad	2024-05-19	League	Abhishek Sharma	Rajiv Gandhi International Stadium, Uppal, Hyd...	Punjab Kings	Sunrisers Hyderabad	Punjab Kings	...	wickets
1091	1426309	2024	Ahmedabad	2024-05-21	Qualifier 1	MA Starc	Narendra Modi Stadium, Ahmedabad	Sunrisers Hyderabad	Kolkata Knight Riders	Sunrisers Hyderabad	...	wickets
1092	1426310	2024	Ahmedabad	2024-05-22	Eliminator	R Ashwin	Narendra Modi Stadium, Ahmedabad	Royal Challengers Bengaluru	Rajasthan Royals	Rajasthan Royals	...	wickets
1093	1426311	2024	Chennai	2024-05-24	Qualifier 2	Shahbaz Ahmed	MA Chidambaram Stadium, Chepauk, Chennai	Sunrisers Hyderabad	Rajasthan Royals	Rajasthan Royals	...	runs
1094	1426312	2024	Chennai	2024-05-26	Final	MA Starc	MA Chidambaram Stadium, Chepauk, Chennai	Sunrisers Hyderabad	Kolkata Knight Riders	Sunrisers Hyderabad	...	wickets

1095 rows × 22 columns

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

```
array(['Royal Challengers Bangalore', 'Kings XI Punjab',
       'Delhi Daredevils', 'Mumbai Indians', 'Kolkata Knight Riders',
       'Rajasthan Royals', 'Deccan Chargers', 'Chennai Super Kings',
       'Kochi Tuskers Kerala', 'Pune Warriors', 'Sunrisers Hyderabad',
       'Gujarat Lions', 'Rising Pune Supergiants',
       'Rising Pune Supergiant', 'Delhi Capitals', 'Punjab Kings',
       'Lucknow Super Giants', 'Gujarat Titans',
       'Royal Challengers Bengaluru'], 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
```

```
(839, 22)
```

```
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	batter	bowler	non_striker	bats
124	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	1	R Dravid	AB Dinda	W Jaffer	
125	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	2	W Jaffer	AB Dinda	R Dravid	
126	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	3	W Jaffer	AB Dinda	R Dravid	
127	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	4	W Jaffer	AB Dinda	R Dravid	
128	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	5	R Dravid	AB Dinda	W Jaffer	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
199873	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	5	SS Iyer	AK Markram	VR Iyer	
199874	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	6	VR Iyer	AK Markram	SS Iyer	
199875	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	1	VR Iyer	Shahbaz Ahmed	SS Iyer	
199876	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	2	SS Iyer	Shahbaz Ahmed	VR Iyer	
199877	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	3	VR Iyer	Shahbaz Ahmed	SS Iyer	

96385 rows × 20 columns

Next steps: Generate code with `delivery_df` View recommended plots New interactive sheet

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

	total_runs_y
124	1
125	2
126	2
127	3
128	4
...	...
199873	110
199874	111
199875	112
199876	113
199877	114

96385 rows × 1 columns

```
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	batter	...	extra_runs	total_runs
124	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	1	R Dravid	...	0	
125	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	2	W Jaffer	...	1	
126	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	3	W Jaffer	...	0	
127	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	4	W Jaffer	...	0	
128	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	5	R Dravid	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
199873	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	5	SS Iyer	...	0	
199874	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	6	VR Iyer	...	0	
199875	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	1	VR Iyer	...	0	
199876	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	2	SS Iyer	...	0	
199877	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	3	VR Iyer	...	0	

96385 rows × 23 columns

```
# Step 1: Create a column marking if a wicket has fallen
delivery_df['wicket_fallen'] = delivery_df['player_dismissed'].notnull().astype(int)

# Step 2: Cumulative sum of wickets per match and inning
delivery_df['cumulative_wickets'] = delivery_df.groupby(['match_id', 'inning'])['wicket_fallen'].cumsum()

# Step 3: Calculate remaining wickets
delivery_df['wickets'] = 10 - delivery_df['cumulative_wickets']

# Step 4: Clip negative values (ensure minimum is 0)
delivery_df['wickets'] = delivery_df['wickets'].clip(lower=0)

# Optional: View a few rows
delivery_df[['match_id','winner', 'inning','total_runs_x', 'over', 'ball', 'player_dismissed', 'wicket_fallen', 'cumulative_wickets', 'w
```

	match_id	winner	inning	total_runs_x	over	ball	player_dismissed	wicket_fallen	cumulative_wickets	wickets
124	335982	Kolkata Knight Riders	2	222	0	1	NaN	0	0	10
125	335982	Kolkata Knight Riders	2	222	0	2	NaN	0	0	10
126	335982	Kolkata Knight Riders	2	222	0	3	NaN	0	0	10
127	335982	Kolkata Knight Riders	2	222	0	4	NaN	0	0	10
128	335982	Kolkata Knight Riders	2	222	0	5	NaN	0	0	10
129	335982	Kolkata Knight Riders	2	222	0	6	NaN	0	0	10
130	335982	Kolkata Knight Riders	2	222	0	7	NaN	0	0	10
131	335982	Kolkata Knight Riders	2	222	1	1	R Dravid	1	1	9
132	335982	Kolkata Knight Riders	2	222	1	2	NaN	0	1	9
133	335982	Kolkata Knight Riders	2	222	1	3	NaN	0	1	9
134	335982	Kolkata Knight Riders	2	222	1	4	NaN	0	1	9
135	335982	Kolkata Knight Riders	2	222	1	5	NaN	0	1	9
136	335982	Kolkata Knight Riders	2	222	1	6	NaN	0	1	9

```
# Step 1: Mark wicket if player_dismissed is not null
delivery_df['wicket_fallen'] = delivery_df['player_dismissed'].notnull().astype(int)

# Step 2: Cumulative sum of wickets per match and inning
delivery_df['cumulative_wickets'] = delivery_df.groupby(['match_id', 'inning'])['wicket_fallen'].cumsum()

# Step 3: Wickets remaining
delivery_df['wickets'] = (10 - delivery_df['cumulative_wickets']).clip(lower=0)
```

```
delivery_df.head()
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batter	...	is_wicket	player_dismiss
124	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	1	R Dravid	...	0	N
125	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	2	W Jaffer	...	0	N
126	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	3	W Jaffer	...	0	N
127	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	4	W Jaffer	...	0	N
128	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	5	R Dravid	...	0	N

5 rows × 26 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']
```

```
delivery_df
```

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball	batter	...	dismissal_kind	field
124	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	1	R Dravid	...	NaN	Na
125	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	2	W Jaffer	...	NaN	Na
126	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	3	W Jaffer	...	NaN	Na
127	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	4	W Jaffer	...	NaN	Na
128	335982	Bangalore	Kolkata Knight Riders	222	2	Royal Challengers Bangalore	Kolkata Knight Riders	0	5	R Dravid	...	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	
199873	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	5	SS Iyer	...	NaN	Na
199874	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	9	6	VR Iyer	...	NaN	Na
199875	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	1	VR Iyer	...	NaN	Na
199876	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	2	SS Iyer	...	NaN	Na
199877	1426312	Chennai	Kolkata Knight Riders	113	2	Kolkata Knight Riders	Sunrisers Hyderabad	10	3	VR Iyer	...	NaN	Na

96385 rows × 28 columns

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

final_df.sample()
```

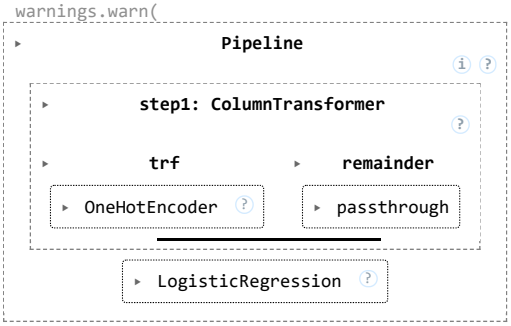
	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	rrr	result
193337	Royal Challengers	Sunrisers	Hyderabad	70	55	10	186	9.876923	8.618182	1

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

final_df.dropna(inplace=True)

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_runs_x	crr	rrr
57415	Kings XI Punjab	Mumbai Indians	Mumbai	40	28	7	163	8.021739	8.571429
174291	Royal Challengers Bangalore	Rajasthan Royals	Dubai	72	70	8	149	9.240000	6.171429
27285	Deccan Chargers	Kolkata Knight Riders	Mumbai	73	67	9	161	9.962264	6.537313
74507	Delhi Daredevils	Kolkata Knight Riders	Raipur	133	120	10	136	inf	6.650000
70959	Sunrisers Hyderabad	Kings XI Punjab	Hyderabad	73	72	8	123	6.250000	6.083333
...	...	...	...	...	...	...	...	...	...
44600	Delhi Daredevils	Deccan Chargers	Delhi	84	58	7	168	8.129032	8.689655
161614	Royal Challengers Bangalore	Kolkata Knight Riders	Abu Dhabi	7	54	8	84	7.000000	0.777778
106986	Chennai Super Kings	Kings XI Punjab	Chandigarh	102	101	8	130	8.842105	6.059406

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

trf = ColumnTransformer([
    ('trf', OneHotEncoder(sparse_output=False, drop='first', handle_unknown='ignore'),
     ['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'))
])


X_train.replace([np.inf, -np.inf], np.nan, inplace=True)
X_train.fillna(0, inplace=True)


pipe.fit(X_train,y_train)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/compose/_column_transformer.py:1667: FutureWarning:
The format of the columns of the 'remainder' transformer in ColumnTransformer.transformers_ will change in version 1.7 to match the
At the moment the remainder columns are stored as indices (of type int). With the same ColumnTransformer configuration, in the futur
To use the new behavior now and suppress this warning, use ColumnTransformer(force_int_remainder_cols=False).

  warnings.warn(
```




```
for df in [X_train, X_test]:
    df.replace([np.inf, -np.inf], np.nan, inplace=True)
    df.fillna(0, inplace=True)


y_pred = pipe.predict(X_test)


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

```
0.8143987604449118
```

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

```
array([0.21614466, 0.78385534])
```

```
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]
    if match.empty:
        print(f"No data found for match ID {match_id}")
        return None, None

    match = match[(match['ball'] == 6)]
    if match.empty:
        print(f"No data with ball == 6 for match ID {match_id}")
        return None, None

    # Check for required columns
    required_cols = ['batting_team', 'bowling_team', 'city', 'runs_left', 'balls_left',
                     'wickets', 'total_runs_x', 'crr', 'rrr']
    missing_cols = [col for col in required_cols if col not in match.columns]
    if missing_cols:
        print("Missing columns:", missing_cols)
        return None, None  # Stop function if required columns are missing

    # Continue with processing if data is valid
    temp_df = match[required_cols].dropna()
    if temp_df.empty:
        print(f"No valid data after dropna for match ID {match_id}")
        return None, None  # return if dropna removes all rows

    # Rest of function continues here
    temp_df.replace([np.inf, -np.inf], np.nan, inplace=True)
    temp_df.dropna(inplace=True)
    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]]

    temp_df = temp_df[['end_of_over', 'runs_after_over', 'wickets_in_over', 'lose', 'win']]
    return temp_df, target


print(delivery_df.columns.tolist())
```

```
['match_id', 'city', 'winner', 'total_runs_x', 'inning', 'batting_team', 'bowling_team', 'over', 'ball', 'batter', 'bowler', 'non_st
```

```
# If 'total_runs' doesn't exist but 'total_runs_x' does:
if 'total_runs_x' in match.columns:
    match = match.rename(columns={'total_runs_x': 'total_runs'})


if 'total_runs_x' in delivery_df.columns and 'total_runs' not in delivery_df.columns:
    delivery_df.rename(columns={'total_runs_x': 'total_runs'}, inplace=True)


print(delivery_df['match_id'].dtype)
print(delivery_df['match_id'].unique()[:5])
```

```
int64
[335982 335983 335984 335985 335986]
```

```
match_id = 335982
if match_id not in delivery_df['match_id'].values:
    print(f"Match ID {match_id} does not exist in the dataset."
```

```
else:
    print(f"Match ID {match_id} found!")
```

```
Match ID 335982 found!
```

```
delivery_df['total_runs_x'] = delivery_df['total_runs']


temp_df, target = match_progression(delivery_df,336003, pipe)
print(temp_df)
```

```
      end_of_over  runs_after_over  wickets_in_over  lose   win
4891            1               16                0  49.5  50.5
4897            2                7                1  61.9  38.1
4903            3                2                2  86.3  13.7
4909            4                4                1  92.5   7.5
4915            5                2                0  94.0   6.0
4922            6                4                0  94.7   5.3
4928            7                7                0  94.7   5.3
4934            8                3                0  95.8   4.2
4940            9                7                1  97.7   2.3
4946           10                5                0  98.2   1.8
4952           11                6                0  98.5   1.5
4958           12               17                0  97.3   2.7
4964           13                5                0  98.2   1.8
4970           14               16                0  96.8   3.2
4976           15               12                0  96.1   3.9
4982           16               11                0  96.2   3.8
4988           17               18                0  89.1  10.9
```