

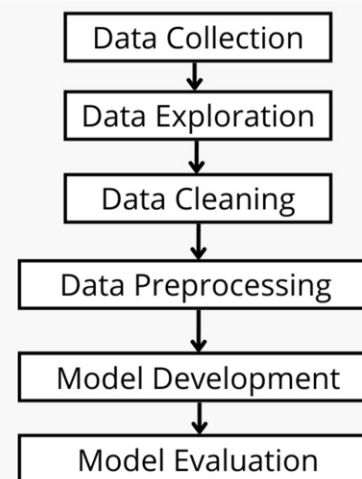
ABSTRACT

The Indian Premier Tournament (IPL) is a professional Twenty20 cricket league in India, played amongst eight teams representing eight different cities in March, April, and May of each year. Winning a cricket match is determined by a number of factors, including the team's home advantage, previous performances on that ground, records at the same venue, the players' overall experience, record against a specific opposition, and the team's overall current form, as well as the individual player's current form. The purpose of this research is to undertake a comprehensive examination of IPL data

OBJECTIVES

We intend to do detailed powerplay, batsman, and bowler analyses, as well as use machine learning models to anticipate victories, runs scored, and wickets taken by a bowler, among other things. IPL match predictor is a machine learning (ML) based prediction approach in which data sets and previous stats are trained in all dimensions, including venue, runs left, balls left, wickets left, the current run rate required run rate, and so on, with each factor having a different strength, using a Logistic regression model to predict the probability of winning the match. Then, using tableau software, we'll create a dashboard that shows Team, Player, and Location data in a dynamic way.

METHODOLOGY



EXPERIMENTAL CODE(SAMPLES)

```
In [107]: M 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']].drop
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 [108]: M temp_df,target = match_progression(delivery_df,75,pipe)
temp_df
```

REQUIREMENTS

- Python 3.7
- Matplotlib
- Pandas
- PyPlot
- Seaborn
- Sklearn
- Tableau

RESULTS

