# Indian Premier League (IPL) Match Prediction using Machine Learning

**Ritesh Sharma, Devang Vogallu**
Student, Department of Computer Science and Engineering, SBJITMR, Nagpur, Maharashtra, India.
rs437718@gmail.com

## ABSTRACT

In today's date, Data Analysis is need for every data analytics to examine the sets of data to extract the useful information from it and to draw conclusion according to the information. Data Analytics techniques and algorithms are more used by the commercial industries which enables them to take precise business decisions. It is also used by the analysts and the experts to authenticate or negate experimental layouts, assumptions and conclusions. In recent years, the analytics is being used in the field of sports to predict and draw various insights. Due to the involvement of money, team spirit, city loyalty and a massive fan following, the outcomes of matches are very important for all stake holders.

In this paper, the past seven years data of IPL containing the players details, match venue details, teams, ball to ball details, is taken and analysed to draw various conclusions which help in the improvement of a player's performance. Various other features like how the venue or toss decision has influenced the winning of the match in last seven years are also predicted. Various machine learning and data extraction model are considered for prediction are Linear Regression, Decision tree and Logistic Regression. The cross-validation score and the accuracy are also calculated using various machine learning algorithms. Before prediction we must explore and visualise the data because data exploration and visualization are an important stage of predictive modelling.

**KEYWORDS:** Machine learning, Random Forest Classifier, Visualization, Cricket, Prediction**.**

## I. INTRODUCTION

Machine learning is a branch of Artificial Intelligence that aims at solving real life engineering problems. It provides the opportunity to learn without being explicitly programmed and it is based on the concept of learning from data. It is so much ubiquitously used dozen times a day that we may not even know it. The advantages of machine learning (ML) methods are that it uses mathematical models, Heuristic learning, knowledge acquisitions and decision trees for decision making. Thus, it provides controllability, observability and stability.

**Machine Learning in Sports and Cricket:**

As a result, Machine Learning is becoming quite a trend in sports analytics because of the ability of live as well as historical data. Sports analytics is the process of collecting past matches data and analysing them to extract the essential knowledge out of it with the home that it facilitates in effective decision making. Decision making may be anything including which player to buy during an auction, which player to set on the field for tomorrow's match, or something more strategic task like, building the tactics for forthcoming matches based on players previous performance.

Machine Learning can be used effectively over various occasions in sports, both on-the-field and off-the-field. When it is about on-the-field, machine learning applies to the analysis of a player's fitness level, design of offensive tactics, or decide short selection. It is also used in predicting the performance of a player, or a team, or the outcome of a match. On the other hand, the off-the-field scenario concerns the business perspective of the sports, which include understanding sales pattern and assigning prices accordingly. On-the-field analytics generally make use of supervised machine learning algorithm, example: (I) Regression for calculating the fitness of a player, (ii) Classification for predicting an outcome of a match; while off-the-field analytics concerns around performing sentiment analysis to understand people's opinion about a player or a team or a sport League. At present, Twitter has become one of the primary sources of data for sentiment analysis.

Similarly, Cricket has also been making use of sports analytics to perform prediction of outcomes of a match, while the gameplay is in progress or before the match has been begun. The game of cricket is played in various formats, i.e., One Day International, T20 and Test Matches.

**Machine Learning in IPL:**

The Indian Premier League (IPL) is a Twenty-20 cricket tournament League established with the objective of promoting cricket in India and thereby nurturing young and talented players. The League is an annual event where teams representing different Indian cities compete against each other. The teams for IPL are selected by mean of an auction. Players auctions are not a new phenomenon in the sports world. However, in India, selection of a team from pool of available players by means of auctioning of players was done in Indian Premier League (IPL) for the first time. Due to the involvement of money, team spirit, city loyalty and a massive fan following, the outcome of matches is very important for all stakeholders. This, in turn, is dependent on the complex rule governing the game, luck of the team (Toss), the ability of a players and their performances on a given day. Various other natural parameters, such as the historical data related to the players, play an integral role in predicting the outcome of a cricket match. A way of predicting the outcome of matches between various teams can aid in the team selection process. However, the varied parameters involved present significant challenges in predicting accurate results of a game. Moreover, the accuracy of prediction depends on the size of data used for the same. The tool presented in this paper can be used to evaluate the performance of players. This tool provides a visualization of player's performance. Further, several predictive models are also built for predicting the result of a match, based on each player's past performance as well as some match related data. The developed models can help decision makers during IPL matches to evaluate the strength of a team against another.

The contributions of the presented work are as follows:

- To provide the statistical analysis of players based on different characteristics
- To predict the performance of a team depending on individual player statistics
- To successfully predict the outcome of IPL matches

# II. LITERATURE SURVEY

An extensive online search produced very few articles related to winner prediction in the game of cricket IPL.

**[1]** "Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning".
2019

**Rabindra Lamsal and Ayesha Choudhary**

Proposed, a solution to calculate the weightage of team based on the player's past performance of IPL using linear regression.

Using linear regression, they also calculate player's performance in upcoming matches

A multivariate regression-based model was formulated to calculate the points earned by each player based on their past performances.

**[2]** "Predicting Players' Performance in One Day International Cricket Matches Using Machine Learning"
2018

**Kalpdrum Passi and Nirav Kumar Pandey**

They identified various factors that influence the outcome of an Indian Premier League matches.

Prediction accuracy in terms of runs scored by batsman and the number of wickets taken by the bowler in each team.

For predictive analytics, they used Weka and Dataiku. Both these tools are a collection of machines learning algorithms for data mining and provide some pre-processing functionalities.

**[3]** "A Criterion for comparing in selecting Batsmen in limited overs cricket"
2004

**GDI Barr and GS Kantor**

Defined a criterion for comparing and selecting batsmen in limited overs cricket.

They defined a new measure P(out) i.e. probability of getting out and used a two-dimensional graphical representation with Strike Rates and batting average of the batsmen.

And, to the risk-return framework used in portfolio analysis, to obtain useful, direct, and comparative

insights into batting performance, particularly in the context of the one-day game.

**[4] "**Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach**"**
2016

**Jhanwar and Paudi**
They predict the outcome of a cricket match by comparing the strengths of the two teams.

For this, they measured the performances of individual players of each team. They developed algorithms to model the performances of batsmen and bowlers where they determine the potential of player by examining his career performances and then his recent performances.

Player independent factors have also been considered to predict the outcome of a match. They show that the k-Nearest Neighbour (kNN) algorithm yields better results as compared to other classifiers.

**[5]** "Data Analytics based Deep Mayo Predictor for IPL-9"
2016

**Lakshmi, Prakash, and Patvardhan**
They present a Deep Mayo Predictor model for predicting the outcomes of the matches in IPL.

Defined Batting index and Bowling index to rank player's performance for their models to predict outcomes of IPL matches.

# III. METHODOLOGY

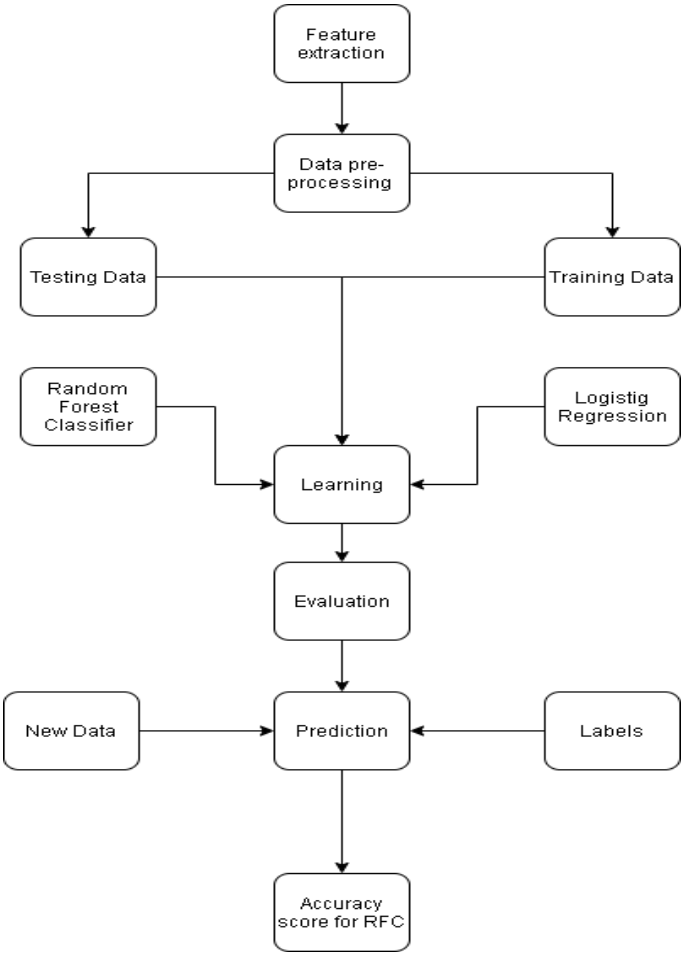


FIGURE 1: SYSTEM ARCHITECTURE

The methodology consists of 4 main stages: -

- Data Pre-processing
- Data Cleansing
- Data Preparation
- Encoding the data.

**Implementation:**

Initially, the seven IPL Seasons real-time dataset is taken in CSV format. In Data Pre-processing phase, the data is incomplete, noisy and inconsistent. Data is to be filled with missing values and correct the inconsistencies. In Data Cleansing phase, data validation is done by maintaining consistency across the dataset and data enhancement is done by adding related information to the dataset.

```
In [248]: #To check the number of columns containing null values
          null_columns=matches.isnull().sum()
          print(null_columns[null_columns > 0])

          #Removing records having null values in "winner" column
          matches=matches[matches["winner"].notna()]
          matches

          winner               3
          player_of_match      3
          umpire1              1
          umpire2              1
          umpire3            636
          dtype: int64

Out[248]:
```

FIGURE 2: CHECKING AND REMOVING NULL VALUES

The scikit-learn Label Encoder is a Python Library which converts categorical variables to a numeric variables and predictive model is created using a generic function called class_model that takes parameter model (algorithm), data, predictors input, and outcome predictable feature.

```
In [209]: encoder= LabelEncoder()
matches["team1"]=encoder.fit_transform(matches["team1"])
matches["team2"]=encoder.fit_transform(matches["team2"])
matches["winner"]=encoder.fit_transform(matches["winner"].astype(str))
matches["toss_winner"]=encoder.fit_transform(matches["toss_winner"])
matches["venue"]=encoder.fit_transform(matches["venue"])

In [210]: #outcome variable as a probability of team1 winning
matches.loc[matches["winner"]==matches["team1"],"team1_win"]=1
matches.loc[matches["winner"]!=matches["team1"],"team1_win"]=0


matches.loc[matches["toss_winner"]==matches["team1"],"team1_toss_win"]=1
matches.loc[matches["toss_winner"]!=matches["team1"],"team1_toss_win"]=0


matches["team1_bat"]=0
matches.loc[(matches["team1_toss_win"]==1) & (matches["toss_decision"]=="bat"),"team1_bat"]=1
```

**FIGURE 3: LABEL ENCODER**

The Data Preparation is significant for achieving optimal results. This involves choosing an outcome measure to evaluate different predictor variables. We manually selected a bunch of features based on the domain knowledge we had. But we have no statistical proof of the selected features being important for our dataset. Scikit-learn provides an excellent module named **feature_selection** which gives us a couple of ways to do the feature selection. First, we will check the columns if any of them represent the same values as other columns. For this, we need to create a correlation matrix to find out the relationship between columns. If the absolute value of correlation between the columns is high enough, we can say that they represent similar values.

**FIGURE 4: FEATURE SELECTION**

```
In [211]: prediction_df=matches[["team1","team2","team1_toss_win","team1_bat","team1_win","venue"]]
X=["team1","team2","team1_toss_win","team1_bat","venue"]
y=matches.team1_win
#finding the higly correlated features
correlated_features = set()
correlation_matrix = prediction_df.drop('team1_win', axis=1).corr()

for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > 0.9:
            column = correlation_matrix.columns[i]
            correlated_features.add(column)

prediction_df.drop(columns=correlated_features)
```

Out[211]:

| | team1 | team2 | team1_toss_win | team1_bat | team1_win | venue |
|---|---|---|---|---|---|---|
| 0 | 12 | 11 | 0.0 | 0.0 | 1.0 | 23 |
| 1 | 7 | 10 | 0.0 | 0.0 | 0.0 | 16 |
| 2 | 3 | 8 | 0.0 | 0.0 | 0.0 | 25 |
| 3 | 10 | 4 | 0.0 | 0.0 | 0.0 | 11 |
| 4 | 11 | 2 | 1.0 | 1.0 | 1.0 | 14 |
| ... | ... | ... | ... | ... | ... | ... |
| 631 | 2 | 11 | 0.0 | 0.0 | 0.0 | 27 |

```
In [212]: #feature selection
X = prediction_df.drop('team1_win', axis=1)
target = prediction_df['team1_win']
target=target.astype(int)

logReg=LogisticRegression(solver='lbfgs')
rfe = RFE(logReg, 20)
rfe = rfe.fit(X, target.values.ravel())
#Checking for the features of they are important
print(rfe.support_)
```

[ True True True True True]

In Data Encoding phase, label each term with short names and encode them as numerical values for predictive modelling as implemented below: -

matches.replace(['Mumbai Indians', 'Kolkata Knight Riders', 'Royal Challengers Bangalore', 'Deccan Chargers', 'Chennai Super Kings', 'Rajasthan Royals', 'Delhi Daredevils', 'Gujarat Lions', 'Kings XI Punjab', 'Sunrisers Hyderabad', 'Rising Pune Supergiant', 'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiant'] , ['MI', 'KKR', 'RCB', 'DC', 'CSK', 'RR', 'DD', 'GL', 'KXIP', 'SRH', 'RPS', 'KTK', 'PW', 'RPS'], inplace=True)

encode = {'team1': {'MI':1, 'KKR':2, 'RCB':3, 'DC':4, 'CSK':5, 'RR':6, 'DD':7, 'GL':8, 'KXIP':9, 'SRH':10, 'RPS':11, 'KTK':12, 'PW':13}, 'team2': {'MI':1, 'KKR':2, 'RCB':3, 'DC':4, 'CSK':5, 'RR':6, 'DD':7, 'GL':8, 'KXIP':9, 'SRH':10, 'RPS':11, 'KTK':12, 'PW':13},
'toss_winner': {'MI':1, 'KKR':2, 'RCB':3, 'DC':4, 'CSK':5, 'RR':6, 'DD':7, 'GL':8, 'KXIP':9, 'SRH':10, 'RPS':11, 'KTK':12, 'PW':13},

'winner': {'MI':1, 'KKR':2, 'RCB':3, 'DC':4, 'CSK':5, 'RR':6, 'DD':7, 'GL':8, 'KXIP':9, 'SRH':10, 'RPS':11, 'KTK':12, 'PW':13, 'Draw':14}}

```
In [216]: matches.replace(['Mumbai Indians','Kolkata Knight Riders','Royal Challengers Bangalore','Deccan Chargers','Chennai Super Kings',
                           'Rajasthan Royals','Delhi Daredevils','Gujarat Lions','Kings XI Punjab',
                           'Sunrisers Hyderabad','Rising Pune Supergiants','Kochi Tuskers Kerala','Pune Warriors','Rising Pune Supergiant'
                          ,['MI','KKR','RCB','DC','CSK','RR','DD','GL','KXIP','SRH','RPS','KTK','PW','RPS'],inplace=True)

encode = {'team1': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13},
          'team2': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13},
          'toss_winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13
          'winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'Draw
matches.replace(encode, inplace=True)
matches.head()
```

Out[216]:

| | id | season | city | date | team1 | team2 | toss_winner | toss_decision | result | dl_applied | winner | win_by_runs | win_by_wickets | player_of_match | v |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2017 | Hyderabad | 2017-04-05 | 10 | 3 | 3 | field | normal | 0 | 10 | 35 | 0 | Yuvraj Singh | Rajiv G Internat Sta t |
| 1 | 2 | 2017 | Pune | 2017-04-06 | 1 | 11 | 11 | field | normal | 0 | 11 | 0 | 7 | SPD Smith | Maharas G Associ Sta |
| 2 | 3 | 2017 | Rajkot | 2017-04-07 | 8 | 2 | 2 | field | normal | 0 | 2 | 0 | 10 | CA Lynn | Saura G Associ Sta |
| 3 | 4 | 2017 | Indore | 2017-04-08 | 11 | 9 | 9 | field | normal | 0 | 9 | 0 | 6 | GJ Maxwell | Holkar C Sta |
| 4 | 5 | 2017 | Bangalore | 2017-04-08 | 3 | 7 | 3 | bat | normal | 0 | 3 | 15 | 0 | KM Jadhav | Chinnas Sta |

**FIGURE 5**

The test dataset is taken as input to the learning algorithm, evaluates different scenarios like toss Winner, toss decision and team winner. Thus, new data is obtained with final prediction.

Different multi-classification algorithms such as Logistics Regression, Decision Tree and Random Forest are implemented to predict the accuracy. Out of these classification algorithm, Random Forest algorithm is proved to be the best accurate classifier.

**Import models from scikit learn module:**
from sklearn.linear_model import LogisticRegression
from sklearn.cross_validation import KFold #For K-fold cross validation

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

#Generic function for making a classification model and accessing performance:
def classification_model(model, data, predictors, outcome):
model.fit(data[predictors], data[outcome])
predictions = model.predict(data[predictors])
accuracy = metrics.accuracy_score(predictions,data[outcome])
print ('Accuracy: %s' % '{0:.3%}'. format(accuracy))

#RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)
outcome_var = ['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
classification_model(model, df,predictor_var,outcome_var)
```

- **RANDOM FOREST ACCURACY**

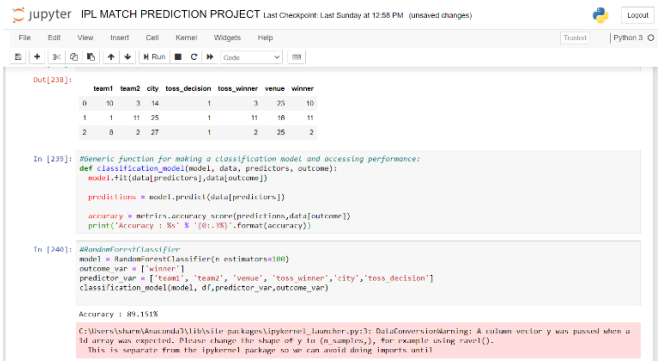


FIGURE 7: ACCURACY

**Discussion:** This screenshot shows the accuracy of Random Forest Classifier algorithm by using a Generic Function Classification Model.

# IV. RESULTS AND DISCUSSIONS

- **MATCH WINNING BY WINNING TOSS**

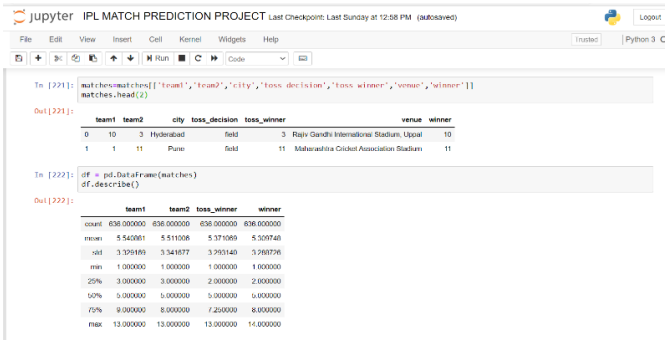


FIGURE 6: PROBABILITY OF MATCH WINNING BY WINNING TOSS

**Discussion:** This screenshot takes the different fields i.e., team1, team2, city, toss_decision, toss_winner, venue, winner and predicts the winner based on the toss_winner.

- **MATCH WINNING BY FEATURES BETWEEN 2 TEAMS**


FIGURE 8: MATCH WINNING BY FEATURES BETWEEN 2 TEAMS

**Discussion:** This screenshot shows the Prediction of Winner by comparing various features between two teams.

## visualization

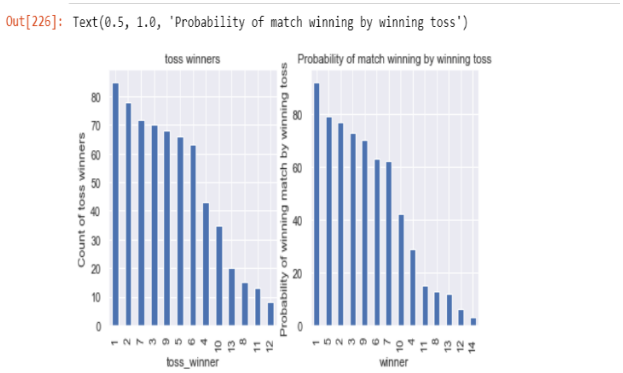


**FIGURE 9: VISUALIZATION OF MATCH WINNING BY WINNING TOSS**

**Discussion:** This screenshot is a histogram that shows two different visualizations. First is toss winners and count of toss winners. Second is probability of match winning by winning Toss.

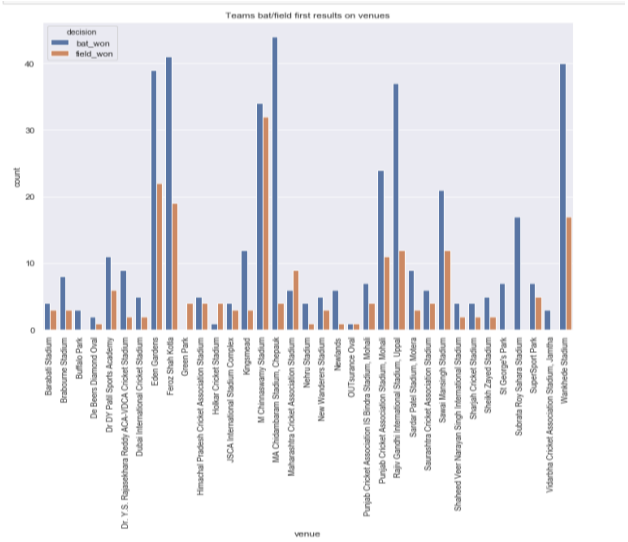


**FIGURE 10: VISUALIZATION OF MATCH WINNING BY VENUE**

**Discussion:** This screenshot is a histogram that shows the visualization of Probability of Match Winning by Venue

# V. CONCLUSION AND FUTURE SCOPE

Selection of the best team for a cricket match plays a significant role for the team's victory. The main goal of this paper is to analyse the IPL cricket data and predict the Winning Team. Here, three classification algorithms are used and compared to find the best accurate algorithm. The implementation tools used are Anaconda Navigator and Jupyter Notebook. Random Forest Classifier is observed to be the best accurate classifier with 89.151% to predict the best team. This knowledge will be used in future to predict the winning teams for the next series IPL matches. Hence using this prediction, Winner of match can be predicted.

# REFERENCES


I. Passi, Kalpdrum & Pandey, Niravkumar. (2018) "Predicting Players' Performance in One Day International Cricket Matches Using Machine Learning" 111-126. 10.5121/csit.2018.80310.

II. Rabindra Lamsal and AyeshaChoudhary, "Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning".

III. Lakshmi, Prakash, and Patvardhan, "Batting index and Bowling index to rank player's performance for their models to predict outcomes of IPL matches".

IV. Abhishek Naiket. Al, "Winning Prediction Analysis in One-Day-International (ODI) Cricket

Using Machine Learning Techniques", IJETCS, vol. 3, issue 2, ISSN:2455-9954, April 2018.

**V.** Jhanwar and Paudi, **"**Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach**"** 2016

**VI.** Rameshwari Lokhande and P.M. Chawan, "Live Cricket Score and Winning Prediction", International Journal of Trend in Research and Development, Volume 5(1), ISSN: 2394-9333