Report On

IPL Win Probability Predictor using Logistic Regression

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Fourth Year Computer Engineering

by

Mohit Raje (Roll No. 33)
Pranit Patil (Roll No.32)
Pranav Maurya (Roll No.29)
Parth Gharat (Roll No.23)

Mentor
Dr. Megha Trivedi

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

(A.Y. 2023-24)

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the Course Project entitled **IPL Win Probability Predictor Using Logistic Regression** is a bonafide work of **Mohit Raje (Roll No. 33) , Pranit Patil (Roll No.32) , Pranav Maurya (Roll No.29) , Parth Gharat (Roll No.23)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in Semester VII of Fourth Year **"Computer Engineering" .**

____

Dr. Megha Trivedi
Mentor

Contents

Abstract

The Indian Premier League (IPL) is one of the most popular and competitive T20 cricket leagues in the world. The outcome of each match is determined by various factors, and predicting the win probability for a team is of great interest to cricket enthusiasts, team management, and sports analysts. This project introduces an IPL Win Probability Predictor that leverages logistic regression, a statistical model well-suited for binary classification tasks.

In this predictive model, historical IPL match data is collected and pre-processed to extract relevant features, such as team performance indicators, player statistics, venue conditions, and toss outcomes. These features are used to train a logistic regression model, which, after appropriate training, can estimate the win probability for each team during a live match.

The logistic regression model, as a binary classifier, predicts which team is more likely to win based on real-time updates, helping cricket fans, teams, and bookmakers make informed decisions. Additionally, this system can be used to enhance the in-game experience for viewers, providing an interactive win probability graph that changes dynamically during the match.

Introduction

1.1 Introduction

Cricket, often regarded as a dynamic sport filled with uncertainties and ever-shifting fortunes, serves as a fertile ground for the application of advanced data analytics and predictive modeling. Within the realm of cricketing competitions, the Indian Premier League (IPL) stands as an epitome of high-intensity, T20 cricket, where each ball bowled can drastically alter the course of a match. The burning question that lingers is: How can we, with a high degree of accuracy, predict the likely victor in a game marked by its unpredictability and complexity?

To tackle this intriguing challenge, we present the "IPL Win Probability Predictor Using Logistic Regression" project. This venture explores the cutting-edge domain of predictive analytics and leverages logistic regression, a powerful statistical model for binary classification, to offer cricket enthusiasts, team management, and sports analysts a robust tool for estimating the win probability of each IPL team during live matches.

The Twenty20 (T20) format, as witnessed in the IPL, demands a granular understanding of numerous variables, including team composition, player statistics, pitch conditions, and even the outcome of the coin toss. It is within this data-rich environment that the IPL Win Probability Predictor, fortified by logistic regression, operates. Our model transforms these multitudinous variables into quantitative insights by capturing intricate relationships, ultimately leading to a real-time estimation of the likely outcome of the match.

1.2 Problem Statement

The Indian Premier League (IPL) is a dynamic and unpredictable T20 cricket competition. The problem at hand is to develop an IPL Win Probability Predictor using logistic regression, which accurately estimates the likelihood of each team's victory in real-time during IPL matches. This involves collecting and preprocessing diverse match-related data, implementing logistic regression for prediction, and delivering an intuitive user interface to enhance the cricket-watching experience.

1.3 Objectives

1. Real-Time Prediction Engine: Develop a real-time prediction engine capable of providing dynamic win probability estimates during live IPL matches. Ensure the system updates predictions promptly as new data becomes available..

2. User-Friendly Interface: Create an intuitive and user-friendly web-based interface that allows cricket enthusiasts, teams, and analysts to access and understand win probability estimates easily. The interface should be responsive and interactive.

3. Data Collection and Preparation: Gather a comprehensive dataset of historical IPL match data, focusing on key features such as team performance, player statistics, pitch conditions, and toss outcomes. Clean and preprocess this data to ensure its quality and consistency for modeling.

4. Accurate Predictions: Develop a prediction model that gives cricket fans the ability to make informed decisions about match outcomes, with the aim of achieving good accuracy**.**

5. Interactive Match Insights: Integrate interactive elements into the predictor, such as live commentary and match insights, to offer users a more immersive experience and a deeper understanding of the game.

1.4 Scope

The scope of the "IPL Win Probability Predictor Using Logistic Regression" project encompasses the development of a user-centric, real-time tool to estimate the win probability of IPL cricket teams during live matches. This project involves data collection, logistic regression model implementation for prediction accuracy, a user-friendly web interface, feedback integration, cross-platform accessibility, and user education. It also explores potential commercial opportunities for enhancing the user experience, with a primary goal of providing cricket enthusiasts and professionals with an interactive, accurate, and engaging tool during IPL matches.

Literature Survey

2.1 Survey of Existing System

In the realm of sports analytics, the Indian Premier League (IPL) Win Probability Predictor has gained significant attention and importance. Existing systems for predicting IPL match outcomes primarily leverage statistical and machine learning techniques, with logistic regression being one of the foundational models. These systems commonly collect and analyze a plethora of data, including historical team performance, player statistics, pitch conditions, and head-to-head match records. They utilize logistic regression to model the relationship between these variables and the likelihood of a team winning a particular match. Logistic regression is favored due to its simplicity, interpretability, and ability to provide probability estimates, which are essential for estimating win probabilities. Furthermore, many existing systems incorporate real-time data feeds during matches, allowing for dynamic model updates and continuous win probability predictions as the game progresses, thus enhancing their accuracy.

Several key challenges faced by these systems include the dynamic and unpredictable nature of T20 cricket, where a single over or a key player's performance can significantly influence the outcome. Consequently, many systems incorporate advanced machine learning algorithms, such as ensemble methods or deep learning, to improve prediction accuracy. Moreover, the reliance on historical data for training models can lead to model stagnation over time, as teams and players evolve. To address this, some systems employ transfer learning techniques or adaptative modeling strategies to account for changing team dynamics and player performances. The IPL Win Probability Predictor remains a dynamic and evolving field, with ongoing research aimed at improving the robustness and accuracy of predictions for the world's most-watched T20 cricket league.

2.2 Limitations Existing System or Research Gap

The existing systems for IPL Win Probability Prediction, although valuable, come with several limitations and research gaps. Firstly, these systems heavily rely on historical data and may not adequately account for real-time variables such as player injuries, form fluctuations, or tactical decisions made during a match. As a result, there is a research gap in developing more dynamic models that can incorporate and adapt to these real-time factors, improving the accuracy of win probability predictions.

Secondly, while logistic regression is a popular choice for its simplicity, it may not capture the intricate nuances of T20 cricket, where explosive and unpredictable gameplay often defies linear modeling. There is an opportunity for research to explore the integration of more sophisticated machine learning techniques, like deep learning or reinforcement learning, to better model the complexities of the game. These methods may enable the prediction of critical turning points within a match and the ability to adjust win probability estimates accordingly. Overall, addressing these limitations and research gaps is crucial for advancing the field of IPL Win Probability Prediction and enhancing its practical application for cricket fans, coaches, and sports analysts.

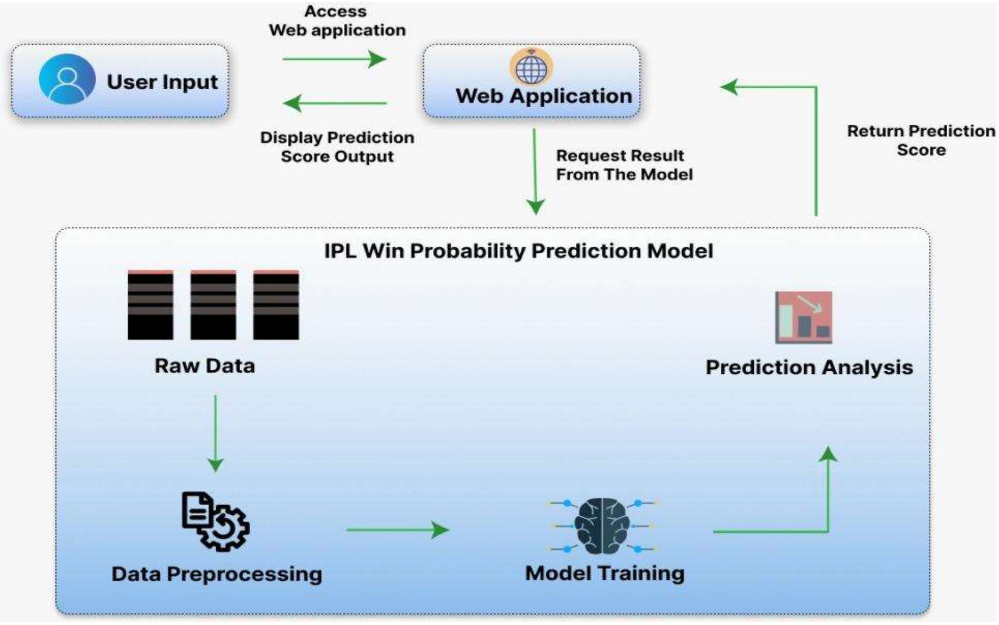
Proposed System (eg New Approach of Data Summarization)

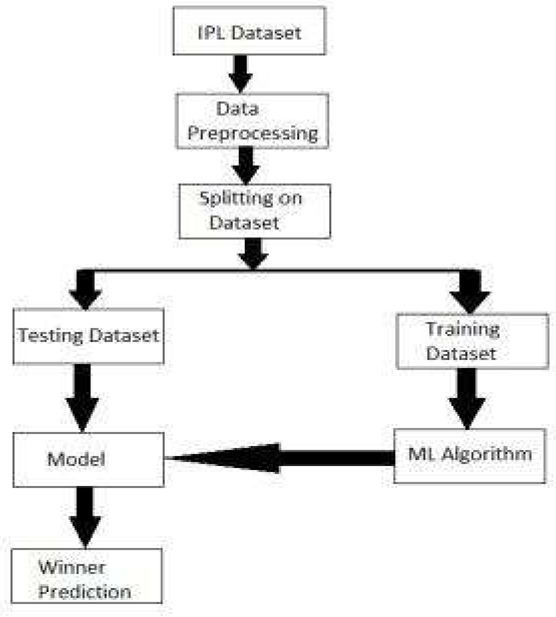
3.1 Introduction

IPL teams go through rigorous planning, player selection, and strategic decision-making to secure their victory. Predicting the outcome of these high-stakes matches is a challenging yet essential task for teams, fans, and analysts. In this context, the proposed system, the "IPL Win Predictor Using Logistic Regression," aims to provide a data-driven solution to forecast the probability of victory for IPL teams using advanced statistical techniques.The Indian Premier League is known for its unpredictability, with teams consisting of international stars and young talents from around the world. While team composition and strategy play a crucial role in the outcome of a match, data-driven approaches have proven to be valuable in understanding and predicting match results. The proposed system leverages the power of logistic regression, a widely used statistical method for binary classification, to model and predict the likelihood of a team winning an IPL match.

3.2 Architecture/Framework/Block Diagram

Flow of System





3.3 Algorithm & Process Design

1. Data Collection and Preprocessing: Gather historical IPL match data from reliable sources, including details on team performance, player statistics, pitch conditions, and match location. Clean and preprocess the data, handling missing values, outliers, and formatting inconsistencies.
2. Feature engineering: Create new features that may impact match outcomes, such as home advantage indicators, team strength, and recent performance metrics.
3. Data Splitting: Divide the historical data into two sets: a training set and a testing set. The training set will be used to train the logistic regression model, while the testing set will be used to evaluate its performance.
4. Feature Selection: Utilize domain knowledge and statistical analysis to identify the most relevant features that influence match outcomes. This might include batting averages, bowling economy rates, head-to-head statistics, and team performance metrics. Perform feature scaling if necessary to ensure all features have a consistent scale.
5. Logistic Regression Model Building:Implement a logistic regression algorithm, a supervised learning technique for binary classification. The binary classification task in this context is to predict whether a team will win or lose an IPL match. Train the logistic regression model on the

training data using the selected features. The model will learn the relationship between these features and match outcomes.

6. Model Evaluation:Assess the performance of the logistic regression model using the testing data. Common evaluation metrics for binary classification include accuracy, precision, recall, F1-score, and the receiver operating characteristic (ROC) curve.
7. User Interface Development: Create a user-friendly interface where users (fans, analysts, team managers) can input match-related data. This might include selecting teams, choosing locations, and specifying player lineups. Display the predicted probability of winning for each team, along with relevant statistics and insights.

3.4 Details of Hardware & Software

Hardware:

- Intel Processor
- 8 GB RAM

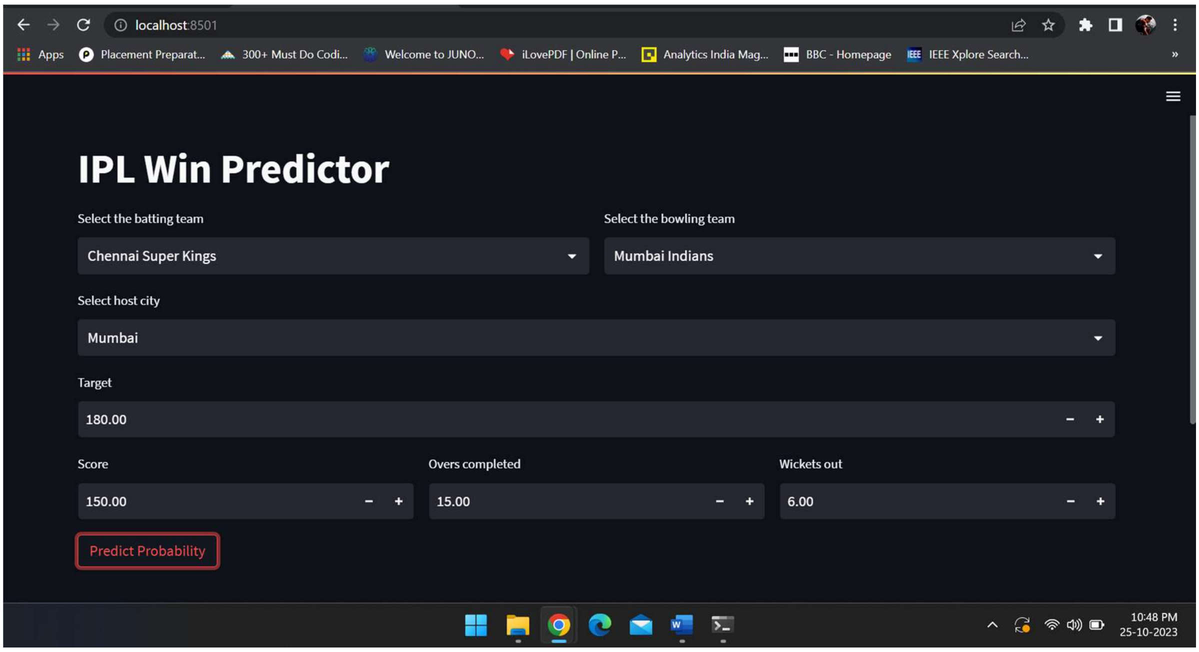
Software:

- streamlit
- pandas
- numpy
- sklearn

3.5 Code:

```
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'))
])
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
pipe1 = Pipeline(steps=[
    ('step1',trf),
    ('step2',RandomForestClassifier())
])
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred1)
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','wickets_in_over','lose','win']]
    return temp_df,target
```

3.6 Experiment and Results for Validation and Verification



IPL Win Predictor

Select the batting team: Chennai Super Kings

Select the bowling team: Mumbai Indians

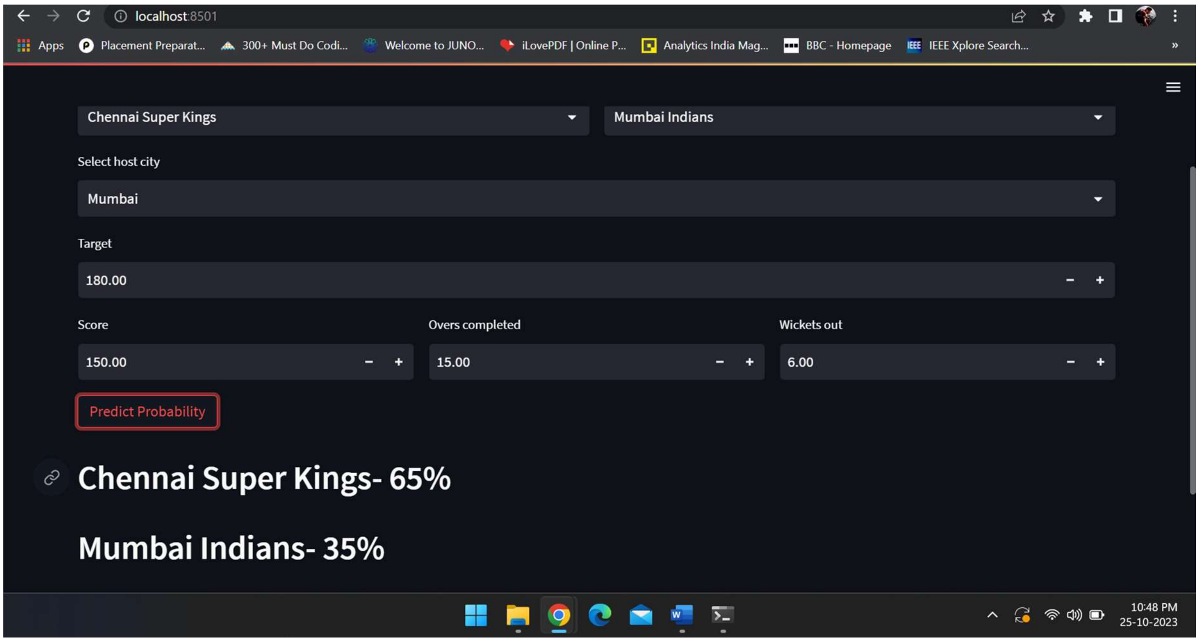
Select host city: Mumbai

Target: 180.00

Score: 150.00

Overs completed: 15.00

Wickets out: 6.00



Chennai Super Kings

Mumbai Indians

Select host city: Mumbai

Target: 180.00

Score: 150.00

Overs completed: 15.00

Wickets out: 6.00

Chennai Super Kings- 65%

Mumbai Indians- 35%

3.7 Analysis :

Predicting the winner of the Indian Premier League (IPL) using logistic regression involves several steps. First we collect historical IPL match data, including team names, venues, toss winners, and detailed match statistics. After cleaning and preprocessing the data, we can create a binary classification model, with the target variable being the match winner (either Team A or Team B). Features like team names, venue, and various statistics should serve as independent variables.OneHot Encoder is also used to convert the categorical values into binary ones. It's crucial to split your dataset into training and testing sets, then fit a logistic regression model. Evaluate the model using common classification metrics like accuracy, precision, recall, and F1 score.

3.8 Conclusion and Future work :

In conclusion, building an IPL match winner predictor using logistic regression is a complex task that involves multiple steps in data preparation, model creation, and evaluation. While logistic regression is a suitable choice for binary classification tasks like predicting the winning team (Team A or Team B) in an IPL match.Ultimately, an IPL win predictor using logistic regression can serve as a valuable exercise in data analysis and machine learning, offering insights into the factors that influence match outcomes in the IPL. However, it's essential to acknowledge the inherent unpredictability of sports and consider this model as one tool among others for understanding and analyzing cricket match results.

While the mini project has achieved its initial goals, there are several areas for potential future work and improvements:

- **Feature Engineering**: Continue to refine and expand the set of features used in the model. Consider incorporating more detailed player statistics, team performance trends, and historical head-to-head performance between teams. Developing new and insightful features can lead to better predictions.
- **Advanced Algorithm**s: Experiment with more advanced machine learning algorithms, such as ensemble methods (e.g., Random Forests, Gradient Boosting) or deep learning models, to improve predictive accuracy. Different algorithms may capture complex relationships in the data that logistic regression cannot.
- **Time Series Analysis**: Consider time series analysis to account for the temporal nature of IPL matches. This can help in capturing season-wise trends and identifying patterns that change over time.
- **Sentiment Analysis**: Integrate sentiment analysis of social media and news data to gauge the public sentiment and its potential impact on team performance and match outcomes.

3.9 References

https://www.researchgate.net/publication/344227524_IPL_Match_Prediction_using_Machine_Learning

https://www.jetir.org/view?paper=JETIR2106333

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4062937