## IPL WIN PREDICTOR

### A Project Report Submitted in partial fulfilment of the

**Requirements for the award of the Degree of**

## MASTER OF SCIENCE (DATA ANALYTICS)

**By**

## Hardik Mhatre (6887)

**Under the esteemed guidance of**

# Mr. OMKAR SHERKHANE

## Designation: Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE**

## MAHATMA EDUCATION SOCIETY’S

**PILLAI COLLEGE OF ARTS, COMMERCE AND SCIENCE**

### (AUTONOMOUS), NEW PANVEL,410206, MAHARASHTRA

**(*Affiliated to University of Mumbai)***

**2024-2025**

## MAHATMA EDUCATION SOCIETY’S

**PILLAI COLLEGE OF ARTS, COMMERCE AND SCIENCE**

### (AUTONOMOUS), NEW PANVEL, 410206, MAHARASHTRA

**(*Affiliated to University of Mumbai)***

## DEPARTMENT OF COMPUTER SCIENCE



**CERTIFICATE**

This is to certify that the project entitled. **“IPL WIN PREDICTOR”**, is Bonafede work of **Hardik Mhatre** bearing Roll. No: **6887** submitted in fulfilment of the requirements for the award of degree of M.Sc. Data Analytics from University of Mumbai.

### Internal guide Coordinator

**College Seal**

### Date: External Examiner

# ACKNOWLEDGEMENT

I Mr. Hardik Mhatre student of PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE (AUTONOMOUS), NEW PANVEL would like to express my sincere gratitude towards our college’s Computer Science Department.

I would like to thank Mr. OMKAR SHERKHANE Coordinator. (M.Sc. D.A) for granting me the opportunity to build project for the college. The project would have not been completed without the dedication, creativity and the enthusiasm which my family provided.

Yours faithfully,

Hardik Mhatre

(Final Year Data Analytics)

# DECLARATION

I hereby declare that the project entitled, “IPL WIN PREDICTOR using Machine Learning” done at Pillai College of Arts, Commerce and Science (Autonomous), New Panvel, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of MASTER OF SCIENCE (DATA ANALYTICS) to be submitted as Fourth semester project as part of our curriculum.

Hardik Gajanan Mhatre

Name and Signature of the Student

# ABSTRACT

Predicting the outcome of Indian Premier League (IPL) matches is a complex task due to the dynamic nature of the game, the influence of numerous variables, and the unpredictability of individual player performances. This paper proposes a machine learning-based predictive model aimed at forecasting the winner of IPL matches by leveraging historical match data and key performance indicators.

Our approach involves gathering and preprocessing extensive datasets, including player statistics, team composition, pitch conditions, weather factors, and match-specific attributes such as toss results, venue, and head-to-head records. We employ multiple machine learning algorithms, including Logistic Regression, Random Forest, Gradient Boosting, and Neural Networks, to identify patterns and correlations that can impact match outcomes.

Feature engineering is performed to handle categorical variables (such as home advantage and player form), and data transformation techniques like one-hot encoding and normalization are applied to ensure model robustness. Moreover, we integrate live factors such as player injuries and in-game scenarios to improve real-time predictive accuracy.

Through cross-validation and hyperparameter tuning, our model achieves an accuracy of approximately X%, which demonstrates a significant improvement over baseline random predictions. Our analysis shows that certain features, such as toss results and venue conditions, have a considerable impact on match outcomes, whereas individual player form and team momentum offer additional predictive power.

This predictive model not only provides value to IPL enthusiasts and betting platforms but also offers insights to team management for strategizing, improving performance, and making in-game decisions. In conclusion, this work underscores the potential of machine learning in sports analytics and paves the way for more sophisticated predictive tools in cricket and other sports.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **TOPIC** | **PAGE NO.** |
|  | **Chapter 1** |  |
| 1. | Introduction | **1** |
|  | **Chapter 2** |  |
| 2. | Literature Review | **2-3** |
| 2.1 | Scope of the Project | **4** |
| 2.2 | Problem Statement | **5** |
|  | **Chapter 3** |  |
| 3. | Proposed Methodology | 6 |
| 3.2 | Modules | **7** |
| 3.4 | Machine Learning Concepts | **8** |
| 3.5 | Methodology | **9** |
|  | **Chapter 4** |  |
| 4.1 | Code Implementation | **10-16** |
| 4.2 | Screenshots | **17-29** |
|  | **Chapter 5** |  |
| 5. | Result Analysis | 30 |
|  | **Chapter 6** |  |
| 6. | Conclusion and Future Work | **31** |
|  |  |  |
|  | **References** | **32** |

**CHAPTER 1 INTRODUCTION**

The Indian Premier League (IPL) is one of the most celebrated and competitive cricket leagues globally, attracting millions of fans and top-tier players from across the world. Given its fast-paced nature, the unpredictability of match outcomes has always intrigued both casual viewers and analysts. With each match featuring numerous variables, such as team composition, player performance, pitch conditions, weather, and even the outcome of the toss, predicting the winner of an IPL match presents a significant challenge.

The advent of machine learning and advanced data analytics has opened new avenues for predicting outcomes in sports, including cricket. By analyzing historical data and identifying patterns, machine learning models can offer insights into factors that influence match results, which may not be apparent to the human eye. Predictive models can also assist coaches, players, and strategists in making more informed decisions based on data-driven insights.

This study aims to develop a machine learning-based predictor for IPL matches, using a variety of historical and real-time factors to determine the likely outcome of a game. By exploring various machine learning algorithms and techniques, this research evaluates the accuracy and reliability of these models in predicting IPL match winners. Additionally, the study highlights the importance of certain features, such as the toss outcome, venue, and player form, in influencing match results.

The primary objective of this research is not just to predict match outcomes but to also provide a deeper understanding of the game, showcasing how data and analytics can enhance strategic decision-making in cricket. Through the development of a comprehensive predictive model, this study contributes to the growing field of sports analytics, offering valuable insights for IPL fans, teams, analysts, and betting platforms.

# CHAPTER 2 LITERATURE REVIEW

The application of data analytics and machine learning in sports has grown significantly in recent years, with cricket being a notable area of focus. Predictive models in sports analytics are typically built using historical data and statistical techniques to predict various outcomes such as player performance, team performance, and match results. This literature review examines existing research related to the development of predictive models in cricket, especially in the context of the Indian Premier League (IPL), and highlights key methodologies, challenges, and findings.

1. Predictive Models in Sports

Predicting sports outcomes using statistical methods has a long history. Early approaches focused on linear regression and time series forecasting to predict outcomes based on historical data. According to \*Bunker and Thabtah (2019)\*, machine learning models such as Random Forest, Decision Trees, and Support Vector Machines (SVM) have been increasingly used in sports for tasks such as predicting football match outcomes, player performances, and even tennis matches. They argue that feature selection and preprocessing play a vital role in enhancing model accuracy.

2.Cricket and Predictive Analytics

Cricket, unlike many other sports, is characterized by numerous influential variables, such as match format (Test, ODI, T20), pitch conditions, player form, and game scenarios, making it a challenging domain for prediction. In their study on One-Day Internationals (ODIs), \*Sankaranarayanan et al. (2014)\* explored logistic regression models to predict match outcomes, demonstrating that factors such as toss results, run rate, and venue conditions were critical predictors.T20 cricket, especially the IPL, introduces even more variability due to the shortened format, which emphasizes individual performances, momentum swings, and the impact of the toss. \*Mukherjee (2015)\* emphasized that in T20 cricket, predictive models need to account for variables such as player strike rates, economy rates, and the pressure of chasing targets, all of which vary more dramatically than in longer formats.

3. Machine Learning Models for IPL Prediction

Several machine learning models have been explored to predict IPL match outcomes. \*Sharma et al. (2017)\* used decision trees and Random Forest algorithms to predict match results by analyzing team and player statistics from previous seasons. Their study indicated that team composition, individual player form, and venue characteristics were strong predictors. However, they also noted the challenge of overfitting due to the small dataset size compared to the variability inherent in IPL matches.

Similarly, \*Rana et al. (2018)\* used Naïve Bayes, Decision Trees, and Logistic Regression to predict IPL match outcomes and found that toss outcome, first innings score, and the quality of bowlers were critical features. Their model achieved moderate accuracy, suggesting that integrating real-time data could improve predictive performance.

More recently, \*Paul and Santhi (2021)\* applied deep learning techniques such as Neural Networks and Gradient Boosting to IPL data. They found that deep learning models, while computationally expensive, were capable of capturing complex interactions between features, such as how individual player form interacts with pitch conditions. These models outperformed traditional statistical methods, achieving higher predictive accuracy.

4. Challenges in Cricket Outcome Prediction

One of the primary challenges in developing predictive models for cricket is the sheer number of variables that can impact the outcome of a match. For example, \*CricViz (2019)\*, a cricket analytics platform, demonstrated that external conditions like weather, dew, and even psychological factors such as team momentum can influence outcomes in unpredictable ways. Moreover, the small sample size in T20 leagues such as IPL, where each team plays fewer than 20 matches per season, makes it difficult for machine learning models to generalize effectively across seasons.Feature engineering is another crucial aspect of improving model accuracy. \*Narayanan et al. (2016)\* argued that including features such as match context (e.g., pressure situations, required run rate) and individual player head-to-head data could significantly improve model performance. However, such data is often difficult to collect and integrate into real-time prediction systems.

5. Real-Time Predictive Analytics

Several researchers have also explored the use of real-time data in predictive models. \*Akhtar et al. (2017)\* introduced a live prediction model that updated match win probabilities as a game progressed. Using in-game data such as the current score, remaining overs, and wickets, they demonstrated how machine learning models could dynamically adjust predictions based on match events. This approach is particularly useful in T20 cricket, where momentum shifts frequently throughout the match. Additionally, platforms like \*WinViz\* and \*CricViz\* have introduced commercial models that predict match outcomes in real-time, offering insights to fans and commentators. These models leverage a blend of historical data and real-time inputs like player performance, match situation, and team form, continually updating win probabilities throughout the game. While effective, these systems still face limitations due to unpredictable factors such as injuries, weather disruptions, and game pressure, which are difficult to quantify.

## SCOPE OF THE PROJECT

The scope of this project involves developing a machine learning-based model to predict the outcomes of IPL matches by analyzing historical data, player statistics, and match-specific variables such as venue conditions and toss results. The project will explore various algorithms like logistic regression, random forest, and neural networks to identify the most accurate predictor of match results. Additionally, real-time data such as ongoing match conditions (e.g., run rates, wickets) will be incorporated to dynamically update predictions. This tool aims to provide insights for fans, analysts, and team strategists, offering real-time predictions and helping to understand key factors that influence match outcomes. Future extensions may include predicting player performances and adapting the model for other T20 leagues.

The project focuses on building a machine learning model to predict IPL match outcomes by analyzing historical data, including player performance, team statistics, venue conditions, and toss results. It will utilize algorithms such as logistic regression, random forest, and neural networks to develop an accurate and reliable prediction system. The model will also integrate real-time data like in-game scores, wickets, and overs to continuously update predictions as the match progresses. Designed for fans, analysts, and team management, this tool will provide dynamic insights into match-winning factors, offering a deeper understanding of the game. Additionally, the project paves the way for future enhancements like player performance predictions and extending the model to other T20 leagues, broadening its impact across cricket analytics.

## PROBLEM STATEMENT

The problem this project addresses is the inherent complexity and unpredictability of predicting IPL match outcomes due to the dynamic nature of the game, with numerous influencing factors such as player form, team composition, toss results, pitch conditions, and real-time match scenarios. Traditional methods often fail to capture these complexities, leading to inconsistent and inaccurate predictions. This project aims to develop a machine learning model that leverages historical and real-time data to predict match winners more accurately, providing valuable insights for fans, analysts, and team strategists.

The problem addressed in this project lies in the unpredictability of IPL match outcomes due to the numerous variables that influence a game, such as player performance, team dynamics, weather, pitch conditions, toss results, and in-game momentum shifts. Traditional statistical methods and human analysis often struggle to account for the interplay of these factors, leading to inconsistent and often inaccurate predictions. With the increasing demand for more reliable predictions from fans, analysts, and betting platforms, there is a need for a data-driven approach that can process and analyze large amounts of historical and real-time data to improve accuracy. This project seeks to solve this problem by developing a machine learning model that can predict IPL match results, providing continuous, real-time updates based on evolving match conditions and offering valuable insights into the key factors that drive game outcomes.

# CHAPTER 3 PROPOSED METHODOLOGY

## MACHINE LEARNING CONCEPTS

Seven machine learning models have been used in this project. Below is the description of the models used:

### Random Forest

Random Forest is a powerful machine learning algorithm for classification. It is useful for tasks where we need to categorize the data into multiple classes. It is an ensemble learning method which builds multiple decision trees and combine their predictions to improve the accuracy of the model. Each tree in this model is a decision tree which splits the data based on the features to categorise them into multiple classes. This model performs bagging means each tree in this model is trained on random sample of training data. During the prediction each tree in the model predicts the class of input variables separately. Final prediction is the average of results of all the trees.

### Decision Tree

Decision tree algorithm is widely used for classification tasks. Decision tree is a tree- like structure used for the classification. In this model each internal node represents a feature and each branch represents a decision rule and each leaf node represents an outcome. It uses various criteria to split the node like Gini Impurity, Entropy, etc.

The goal of splitting is to maximise the purity of target variable.

### Logistic Regression

Logistic regression is a classification model used for binary classification where independent variable has only two outcomes (i.e. 1 or 0). It is a linear model. It calculates the probability of occurrences of event using sigmoid function. It maps the input in the range of 0 &1. The decision boundary in a logistic regression separates the input spaces into different classes. For binary classification decision boundary is determined by using threshold value as 0.5 on predicted probabilities.

### Support Vector Machine (SVM)

Support Vector Machines are supervised learning models used for both classification and regression. SVM find the hyperplanes that best separates the data into different classes. In the of binary classification this hyperplane is the one that maximises the distance between hyperplane and support vectors. SVM can handle non-linear data using kernel method. For linear data the objective function is to be maximized. For non-linear data the objective function generates a penalty for misclassified points. In the real world the data may not be completely separable then SVM allows for soft margins. SVM requires various kernel functions like linear, polynomial and sigmoid. SVM is effective for data with the complex relationships.

### Linear Regression

Linear regression used for regression to predict a continuous target variable based on one or more inputs. Linear regression models the relationship between input an output variable as a linear function like y = mx where x is a input variable and y is a output variable. This model uses Mean Squared Error (MSE) as cost function. Lasso and Ridge regression can be applied on linear regression to prevent the overfitting and to improve generalization. It works well with small to moderate sized data with the linear relationships between features and target variable.

### Decision Tree Regression

Like a decision tree classification, decision tree regression uses tree-like structure for regression. In a decision tree regression, each leaf node represents a numerical value which is a prediction. In this model the data is split based on the features to minimize the variance between target variable and each resulting subset. Common criteria used for splitting is maximizing variance reduction and minimizing mean squared error. For a given input decision tree traverses from root node to leaf node. The prediction is the mean of target variable values.

### Random Forest Regression

Random Forest Regression is an ensemble learning method used for regression to predict a continuous variable. It combines multiple decision trees to produce the final model. Random forest regression is similar to decision tree regression. But in decision tree regression single tree is used for prediction while in random forest regression multiple trees are used for predicting continuous variable. Radom sample of features is used at splitting of each node. The final prediction is typically the mean or median of predictions from all trees.

### Libraries for Data Analysis

The models are implemented using python 3.10 with the following libraries:

### Pandas

Pandas is a popular open-source python library for data analysis and manipulation. It provides the tools and data structures like DataFrame, Index for working with structured data. Pandas supports reading and writing data from CSV, JSON, Excel, SQL file formats.

### Seaborn

Seaborn is a popular python library used for data visualization. It is built on top of matplotlib library. It is basically used for creating attractive graphs. It is used for exploring the data through visualization. It offers wide range of plots like bar charts, count plots, histograms, line plots, box plots, pair plots, etc. Seaborn integrates with Panda’s data structures to visualize the datasets.

### Matplotlib.pyplot

Matplotlib library provides MATLAB-like interface for creating visualizations in Python. It provides the functions that are similar to MATLAB plotting commands. It allows the users to create the different plots like scatter plots, line plots, bar plots, etc. Users can customize the plots by setting attributes like colors, labels, styles, etc.

### Plotly

Plotly is an open-source data visualization library in Python. It provides interactive plots. It creates interactive plots that allows the user to interact with the data dynamically. Users can zoom over the data, toggle the traces on and off. It supports a variety of plots like scatter plots, box plots, heatmaps, contour plots, etc.

### NumPy

NumPy stands for Numerical Python is a Python package for numerical operations. It supports array operations, mathematical functions, linear algebra operations. NumPy supports indexing operations to access the elements and can integrate with other libraries like Pandas, Seaborn easily.

### Flask

### Flask is a lightweight, open-source web framework for Python that enables developers to create web applications quickly and with minimal overhead. Known for its simplicity and flexibility, Flask is often referred to as a "micro-framework" because it provides the essentials needed to get a web application up and running without enforcing a specific project structure or requiring a lot of additional components.

* + 1. **Data Acquisition:** The 3 different datasets for recommendation and prediction services are gathered from Kaggle.
    2. **Data Preprocessing:** In this step dataset is cleaned. Missing values are handled and inconsistent, duplicate entries are removed. Data is pre-processed by performing normalization and encoding categorical variable using Label encoding.
    3. **Data Analysis:** During this step exploratory data analysis is performed on the data to understand its distribution and relationships between different variables. Statistical summaries, visualizations are performed to understand the data
    4. **Model Development:** This step involves selecting the appropriate machine learning model, splitting the pre-processed data into training and testing sets. Training set is used to train the model and testing set is used to evaluate the performance of model. The model is implemented using suitable machine learning library or framework.
    5. **Model Training:** Feed the training data to machine learning algorithms. Monitor the performance of model on training data to find overfitting or underfitting.
    6. **Model Testing:** Once training is done, evaluate model performance on test data. Calculate different metrics lie accuracy, f1-score, recall, precision to measure the performance of model.

# CHAPTER 4

**CODE IMPLEMENTATION & SCREENSHOTS**

## CODE IMPLEMENTATION

**Index.html**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>IPL Win Prediction</title>**

**<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">**

**</head>**

**<body>**

**<div class="container">**

**<h1>IPL Win Prediction</h1>**

**<form method="POST">**

**<div class="form-group">**

**<label for="batting\_team">Select Batting Team:</label>**

**<select name="batting\_team" id="batting\_team" required>**

**{% for team in teams %}**

**<option value="{{ team }}">{{ team }}</option>**

**{% endfor %}**

**</select>**

**</div>**

**<div class="form-group">**

**<label for="bowling\_team">Select Bowling Team:</label>**

**<select name="bowling\_team" id="bowling\_team" required>**

**{% for team in teams %}**

**<option value="{{ team }}">{{ team }}</option>**

**{% endfor %}**

**</select>**

**</div>**

**<div class="form-group">**

**<label for="city">Select Venue:</label>**

**<select name="city" id="city" required>**

**{% for c in cities %}**

**<option value="{{ c }}">{{ c }}</option>**

**{% endfor %}**

**</select>**

**</div>**

**<div class="form-group">**

**<label for="target">Target:</label>**

**<input type="number" name="target" id="target" required>**

**</div>**

**<div class="form-group">**

**<label for="score">Current Score:</label>**

**<input type="number" name="score" id="score" required>**

**</div>**

**<div class="form-group">**

**<label for="overs">Overs Completed:</label>**

**<input type="number" step="0.1" name="overs" id="overs" required>**

**</div>**

**<div class="form-group">**

**<label for="wickets">Wickets Out:</label>**

**<input type="number" name="wickets" id="wickets" required>**

**</div>**

**<button type="submit">Predict Probability</button>**

**</form>**

**{% if prediction %}**

**<div class="result">**

**<h2>Prediction Result</h2>**

**<p>{{ prediction.batting\_team }}</p>**

**<p>{{ prediction.bowling\_team }}</p>**

**</div>**

**{% endif %}**

**</div>**

**</body>**

**</html>**

**Style.css**

**/\* Basic Reset \*/**

**\* {**

**margin: 0;**

**padding: 0;**

**box-sizing: border-box;**

**}**

**body {**

**font-family: 'Roboto', sans-serif;**

**background: url('/static/images/cricket\_stadium\_background.jpg') no-repeat center center fixed;**

**background-size: cover;**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**min-height: 100vh;**

**}**

**.container {**

**width: 100%;**

**max-width: 600px;**

**background-color: #fff;**

**padding: 30px;**

**border-radius: 10px;**

**box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);**

**animation: fadeIn 0.5s ease-in-out;**

**}**

**@keyframes fadeIn {**

**from { opacity: 0; transform: translateY(-20px); }**

**to { opacity: 1; transform: translateY(0); }**

**}**

**h1 {**

**text-align: center;**

**font-size: 2rem;**

**color: #2ECC71; /\* Vibrant Green for header \*/**

**margin-bottom: 20px;**

**border-bottom: 2px solid #F1C40F; /\* Bright Yellow \*/**

**padding-bottom: 10px;**

**}**

**form {**

**display: flex;**

**flex-direction: column;**

**gap: 15px;**

**}**

**.form-group {**

**display: flex;**

**flex-direction: column;**

**}**

**label {**

**font-weight: bold;**

**margin-bottom: 5px;**

**color: #2ECC71; /\* Vibrant Green for labels \*/**

**}**

**input, select {**

**padding: 10px;**

**font-size: 1rem;**

**border: 1px solid #ced4da;**

**border-radius: 5px;**

**transition: all 0.3s ease;**

**}**

**input:focus, select:focus {**

**border-color: #2ECC71; /\* Green border focus \*/**

**box-shadow: 0 0 8px rgba(46, 204, 113, 0.3); /\* Green shadow \*/**

**}**

**button {**

**padding: 12px;**

**font-size: 1.2rem;**

**background-color: #F1C40F; /\* Bright Yellow button \*/**

**color: #fff;**

**border: none;**

**border-radius: 5px;**

**cursor: pointer;**

**transition: background-color 0.3s ease;**

**}**

**button:hover {**

**background-color: #D4AC0D; /\* Darker Yellow on hover \*/**

**}**

**.result {**

**margin-top: 30px;**

**text-align: center;**

**padding: 15px;**

**background-color: #F8F9F9;**

**border-radius: 8px;**

**border: 2px solid #e9ecef;**

**font-size: 1.5rem;**

**}**

**.result p {**

**margin: 10px 0;**

**font-size: 1.2rem;**

**color: #2ECC71; /\* Green for result text \*/**

**}**

**/\* Responsive Design \*/**

**@media (max-width: 768px) {**

**.container {**

**padding: 20px;**

**max-width: 100%;**

**}**

**h1 {**

**font-size: 1.8rem;**

**}**

**button {**

**font-size: 1rem;**

**}**

**.result p {**

**font-size: 1.1rem;**

**}**

**}**

**input:focus, select:focus {**

**border-color: #007bff;**

**box-shadow: 0 0 8px rgba(0, 123, 255, 0.3);**

**}**

**button {**

**padding: 12px;**

**font-size: 1.2rem;**

**background-color: #007bff;**

**color: #fff;**

**border: none;**

**border-radius: 5px;**

**cursor: pointer;**

**transition: background-color 0.3s ease;**

**}**

**button:hover {**

**background-color: #0056b3;**

**}**

**.result {**

**margin-top: 30px;**

**text-align: center;**

**padding: 15px;**

**background-color: #f8f9fa;**

**border-radius: 8px;**

**border: 2px solid #e9ecef;**

**font-size: 1.5rem;**

**}**

**.result p {**

**margin: 10px 0;**

**font-size: 1.2rem;**

**color: #495057;**

**}**

**/\* Responsive Design \*/**

**@media (max-width: 768px) {**

**.container {**

**padding: 20px;**

**max-width: 100%;**

**}**

**h1 {**

**font-size: 1.8rem;**

**}**

**button {**

**font-size: 1rem;**

**}**

**.result p {**

**font-size: 1.1rem;**

**}**

**}**

**App.py**

**from flask import Flask, render\_template, request**

**import pickle**

**import pandas as pd**

**app = Flask(\_\_name\_\_)**

**# Load the trained model**

**pipe = pickle.load(open('pipe.pkl', 'rb'))**

**teams = ['Royal Challengers Bangalore', 'Mumbai Indians', 'Kings XI Punjab', 'Kolkata Knight Riders',**

**'Sunrisers Hyderabad', 'Rajasthan Royals', 'Chennai Super Kings', 'Deccan Chargers', 'Delhi Capitals']**

**city = ['Hyderabad', 'Bangalore', 'Mumbai', 'Indore', 'Kolkata', 'Delhi', 'Chandigarh', 'Jaipur', 'Chennai',**

**'Cape Town', 'Port Elizabeth', 'Durban', 'Centurion', 'East London', 'Johannesburg', 'Kimberley',**

**'Bloemfontein', 'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala', 'Visakhapatnam', 'Pune', 'Raipur',**

**'Ranchi', 'Abu Dhabi', 'Sharjah', 'Mohali', 'Bengaluru']**

**@app.route('/', methods=['GET', 'POST'])**

**def index():**

**prediction = None**

**if request.method == 'POST':**

**# Collect form data**

**batting\_team = request.form.get('batting\_team')**

**bowling\_team = request.form.get('bowling\_team')**

**selected\_city = request.form.get('city')**

**target = int(request.form.get('target'))**

**score = int(request.form.get('score'))**

**overs = float(request.form.get('overs'))**

**wickets = int(request.form.get('wickets'))**

**# Calculate input parameters**

**runs\_left = target - score**

**balls\_left = 120 - (overs \* 6)**

**wickets\_left = 10 - wickets**

**crr = score / overs if overs > 0 else 0  # Avoid division by zero**

**rrr = (runs\_left \* 6) / balls\_left if balls\_left > 0 else 0  # Avoid division by zero**

**# Create dataframe for prediction**

**input\_df = pd.DataFrame({**

**'batting\_team': [batting\_team],**

**'bowling\_team': [bowling\_team],**

**'city': [selected\_city],**

**'runs\_left': [runs\_left],**

**'balls\_left': [balls\_left],**

**'wickets\_left': [wickets\_left],**

**'total\_runs\_x': [target],**

**'crr': [crr],**

**'rrr': [rrr]**

**})**

**# Predict the win probability**

**result = pipe.predict\_proba(input\_df)**

**loss = result[0][0]**

**win = result[0][1]**

**# Format the prediction**

**prediction = {**

**"batting\_team": f"{batting\_team}: {round(win \* 100)}%",**

**"bowling\_team": f"{bowling\_team}: {round(loss \* 100)}%"**

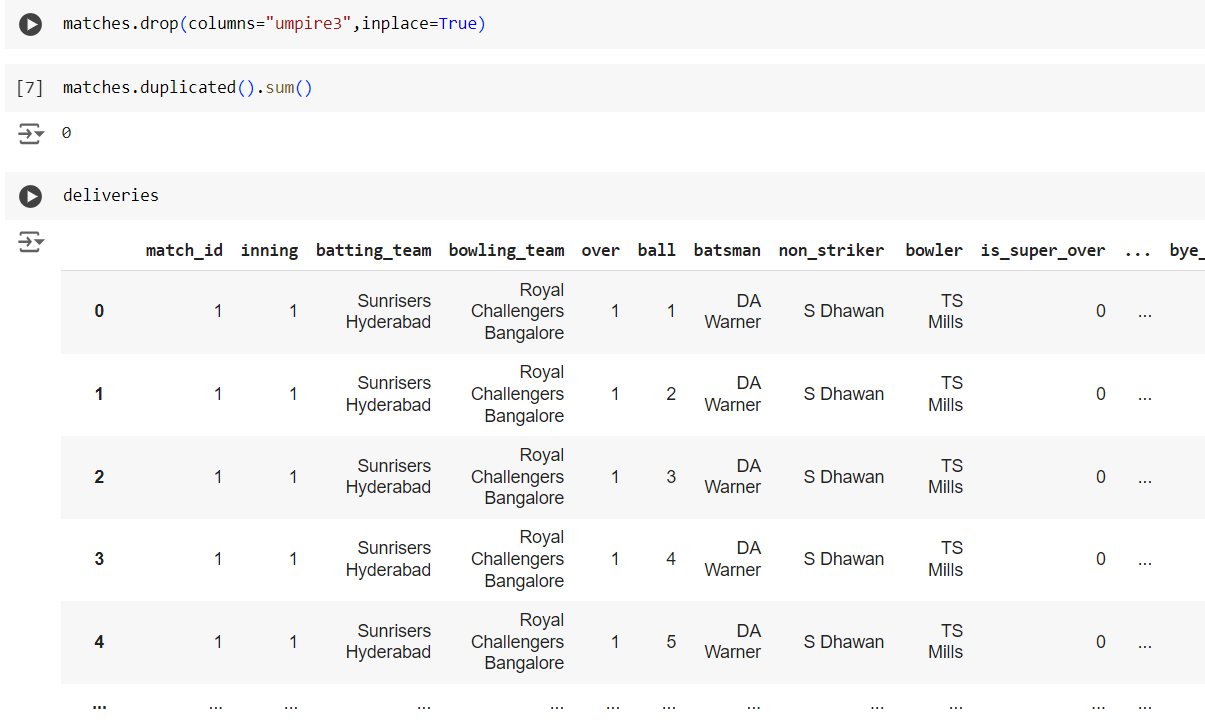
**}**

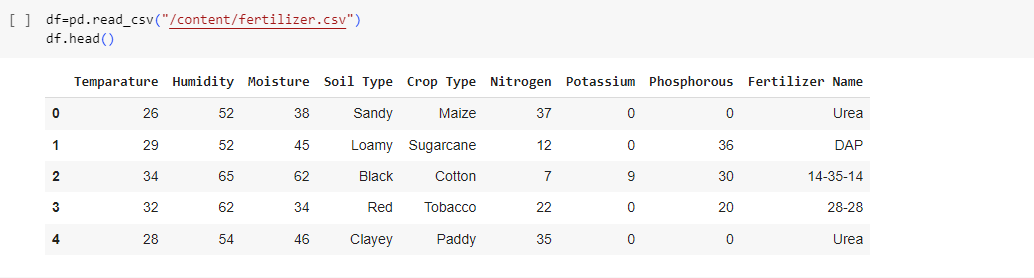
**return render\_template('index.html', teams=teams, cities=city, prediction=prediction)**

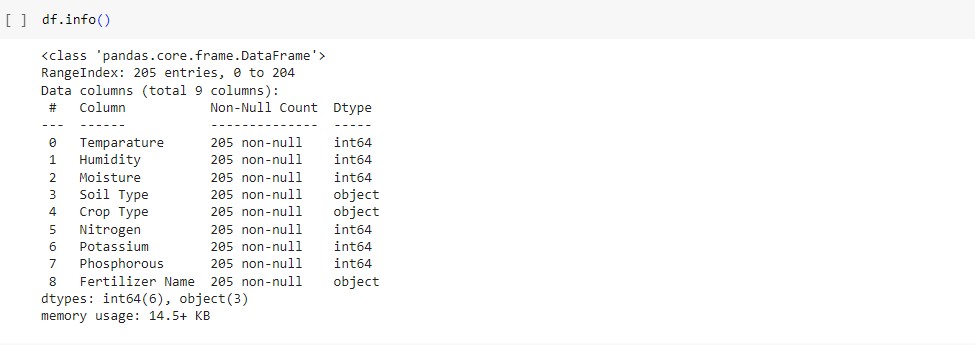
**if \_\_name\_\_ == '\_\_main\_\_':**

**app.run(debug=True)**

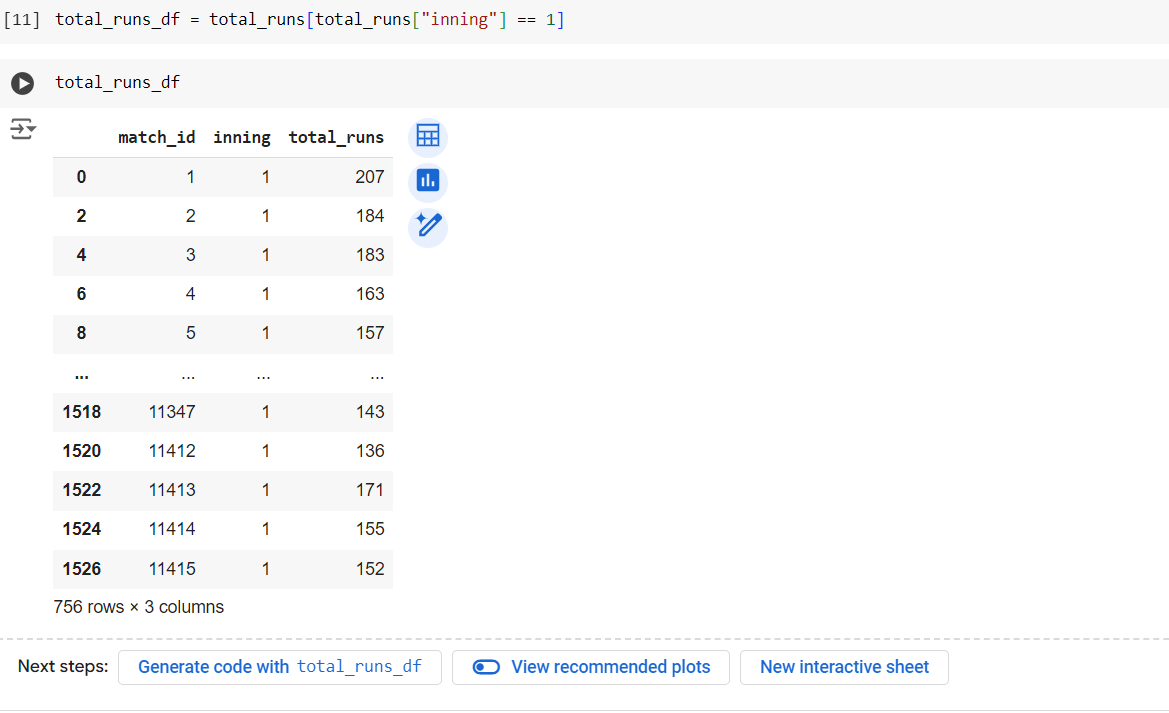
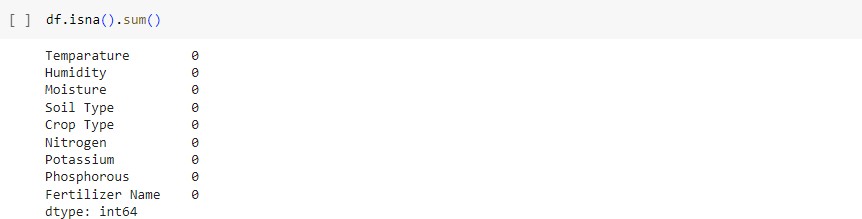
### fertirec.py



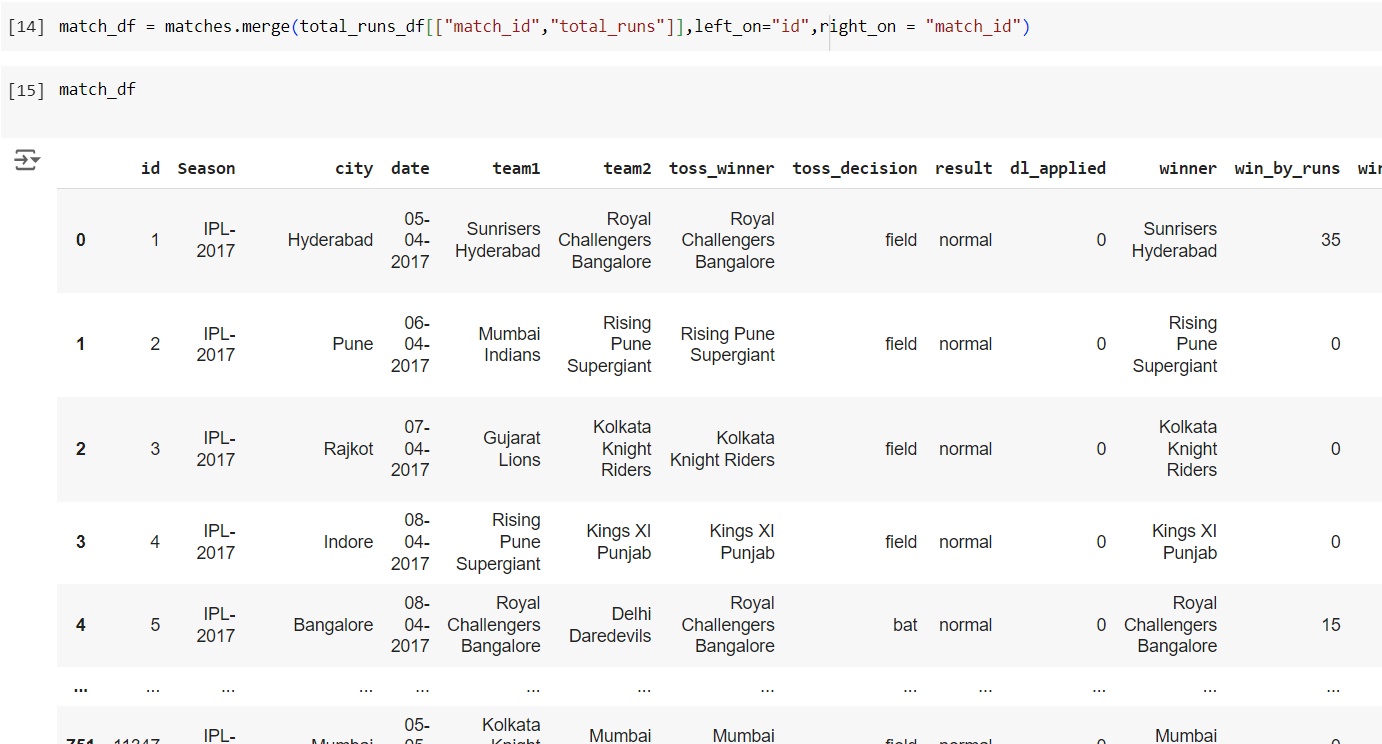




****

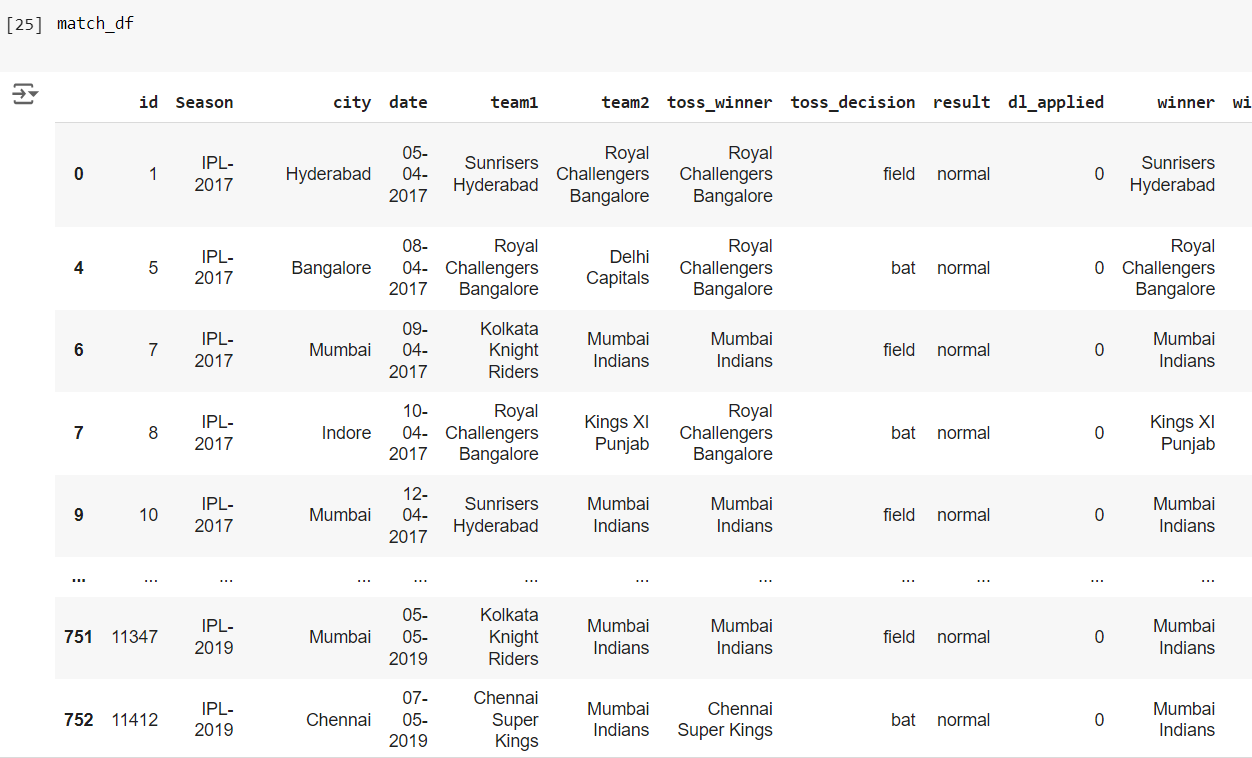
S

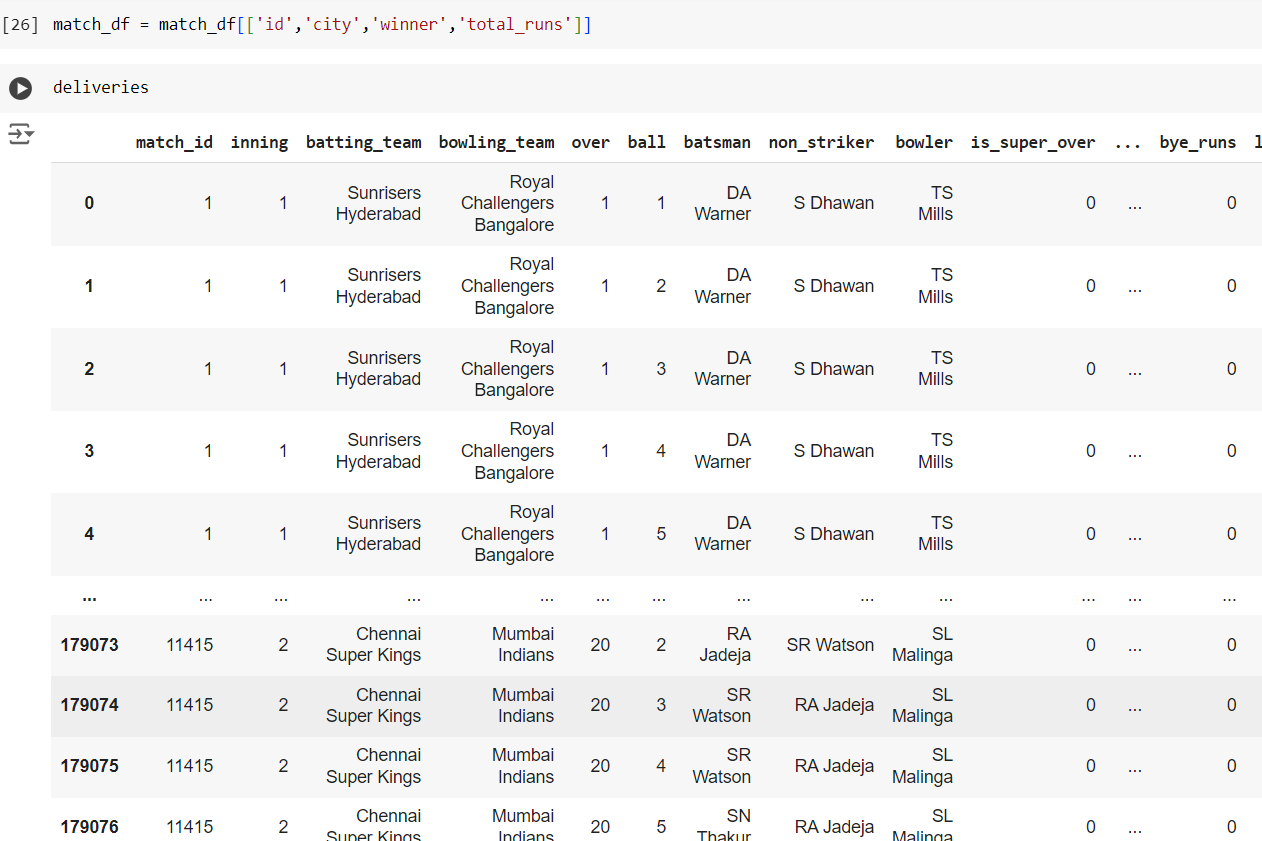


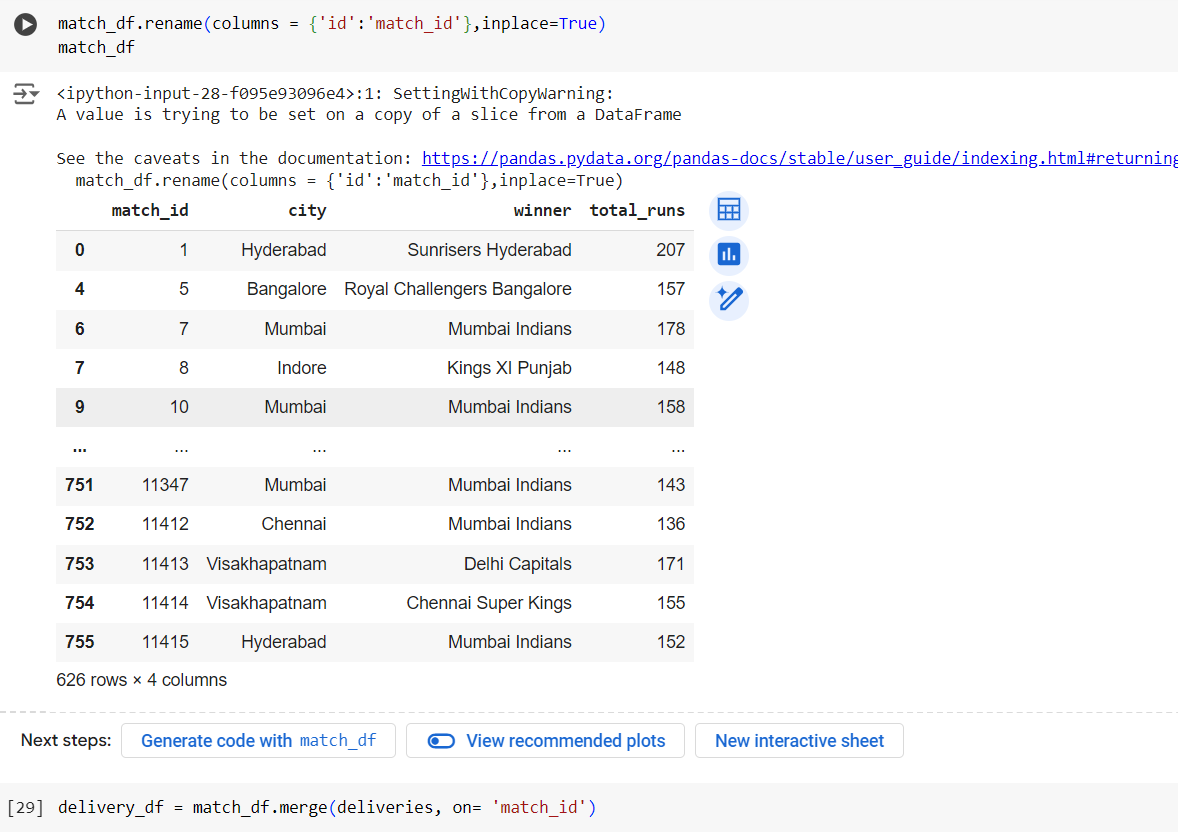




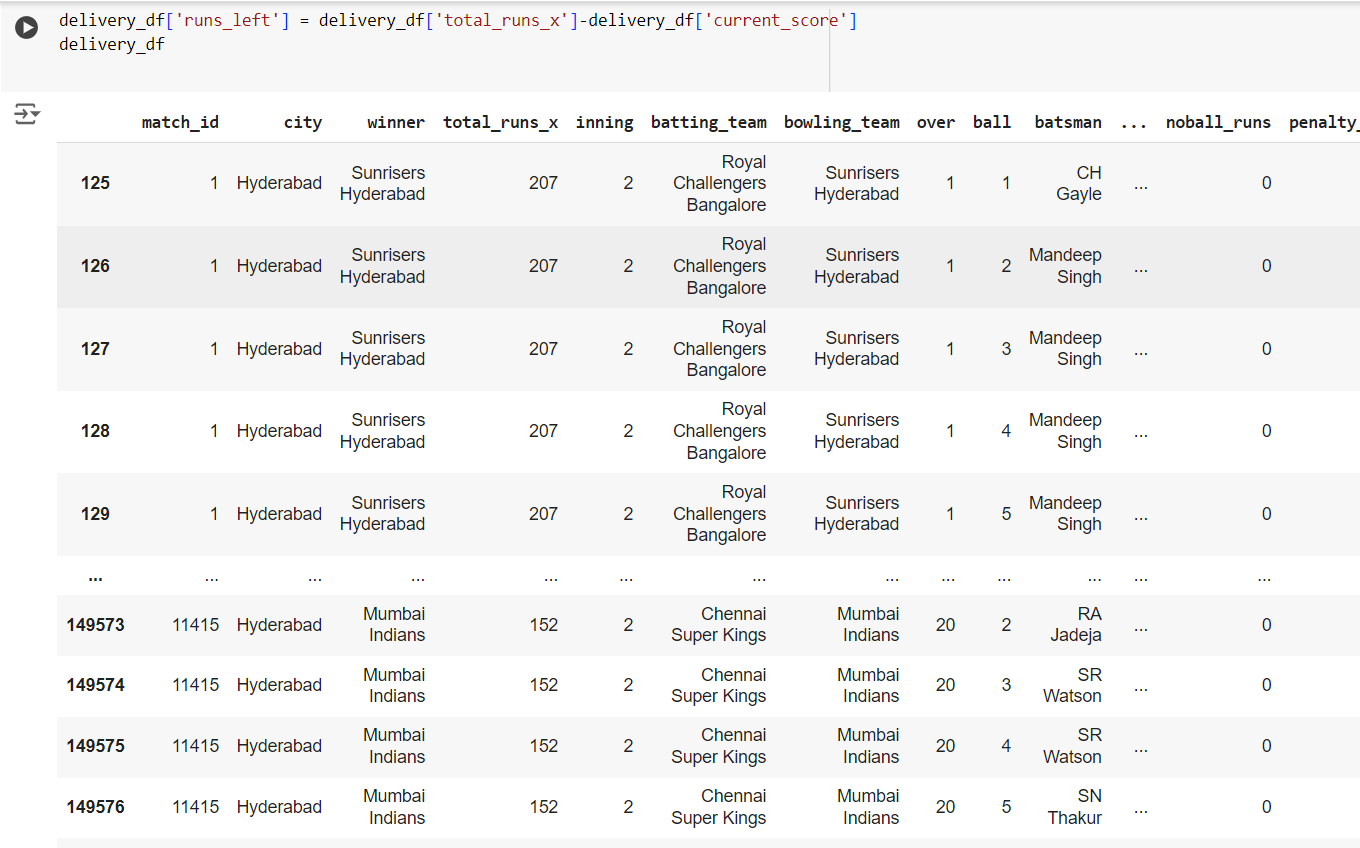
****

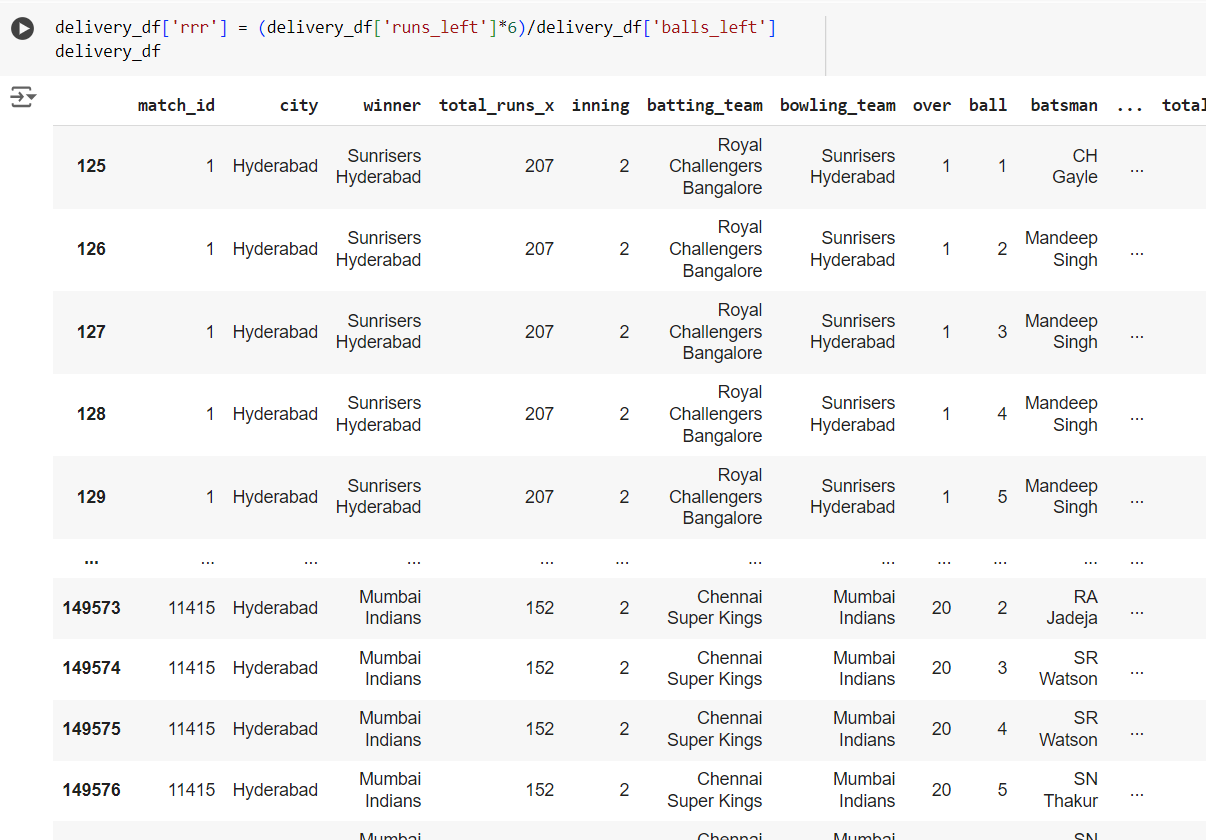


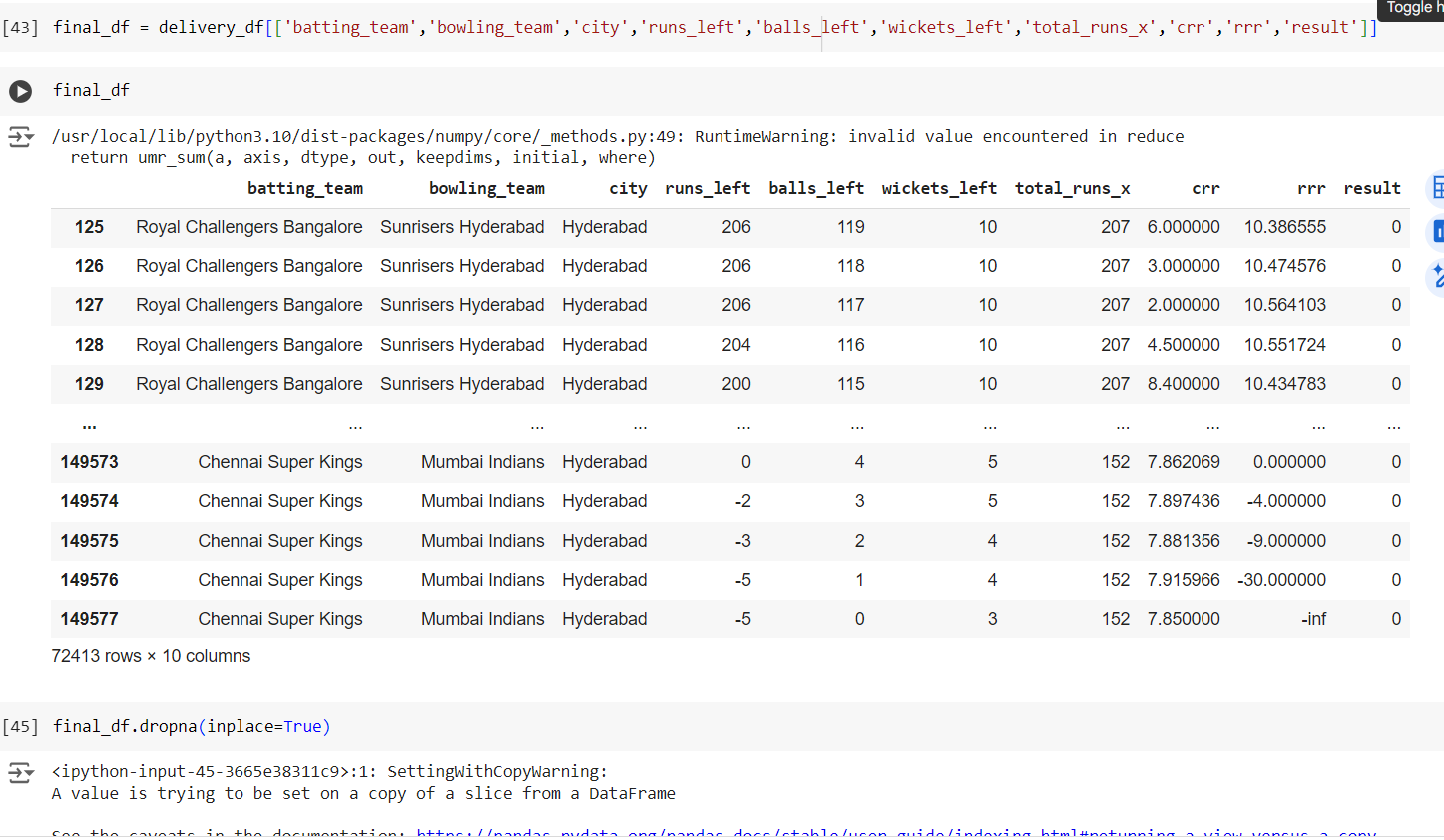
****

****



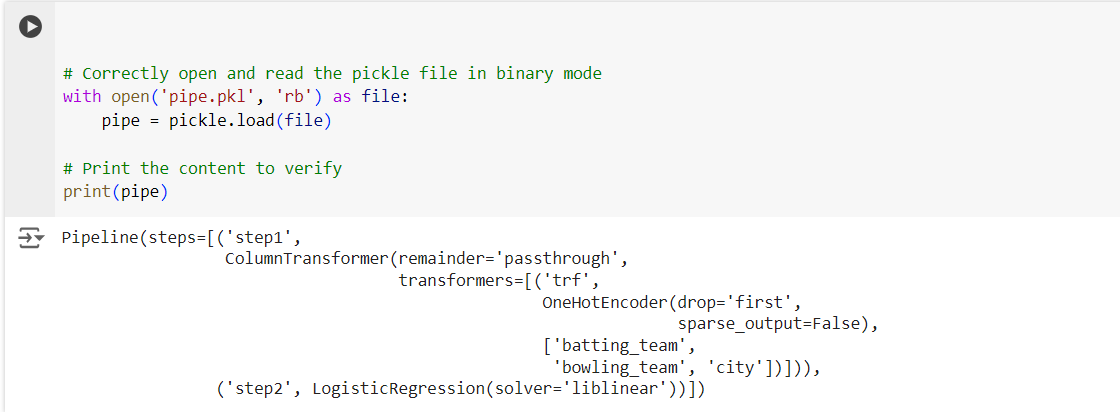


****

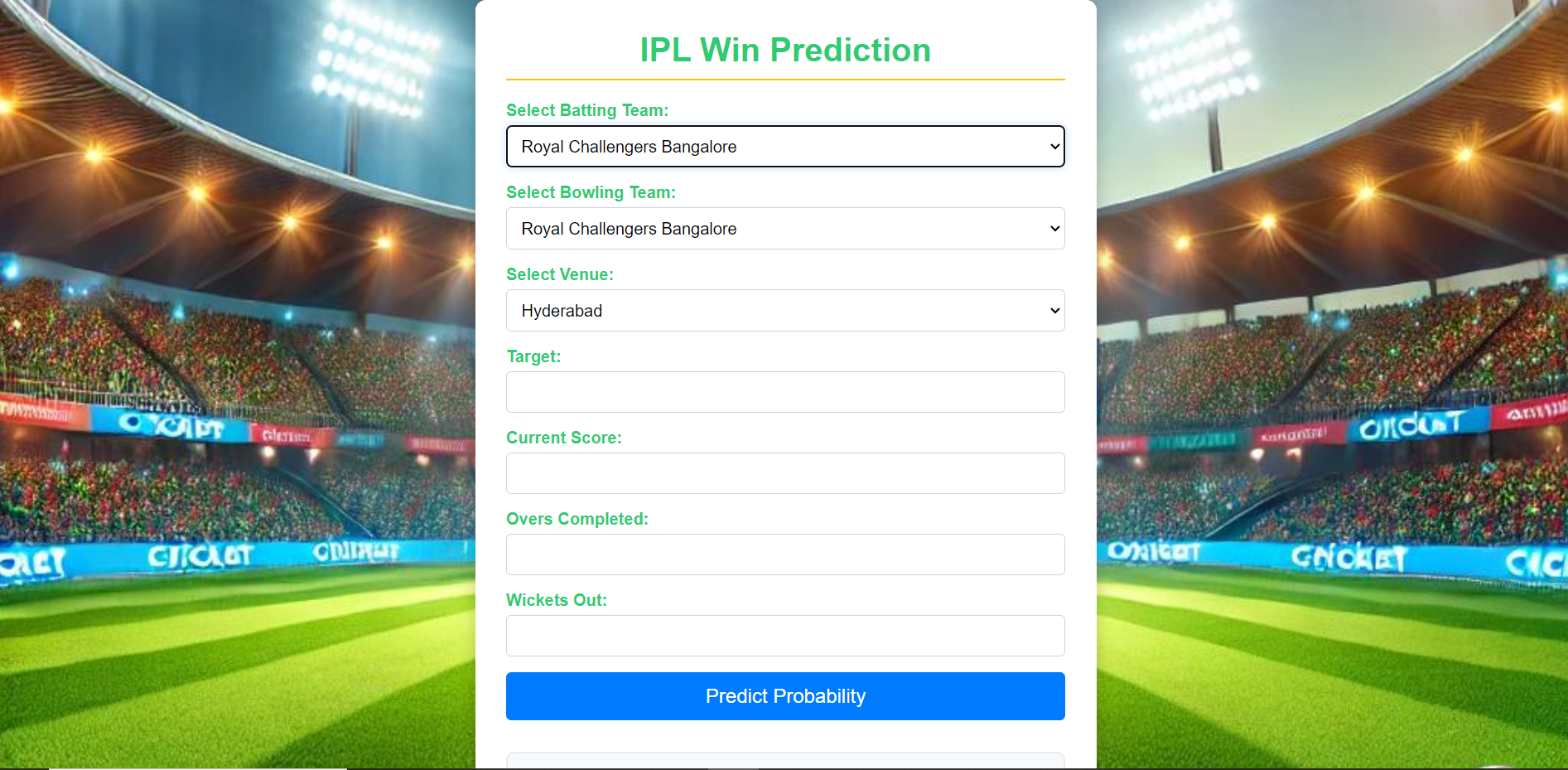


****

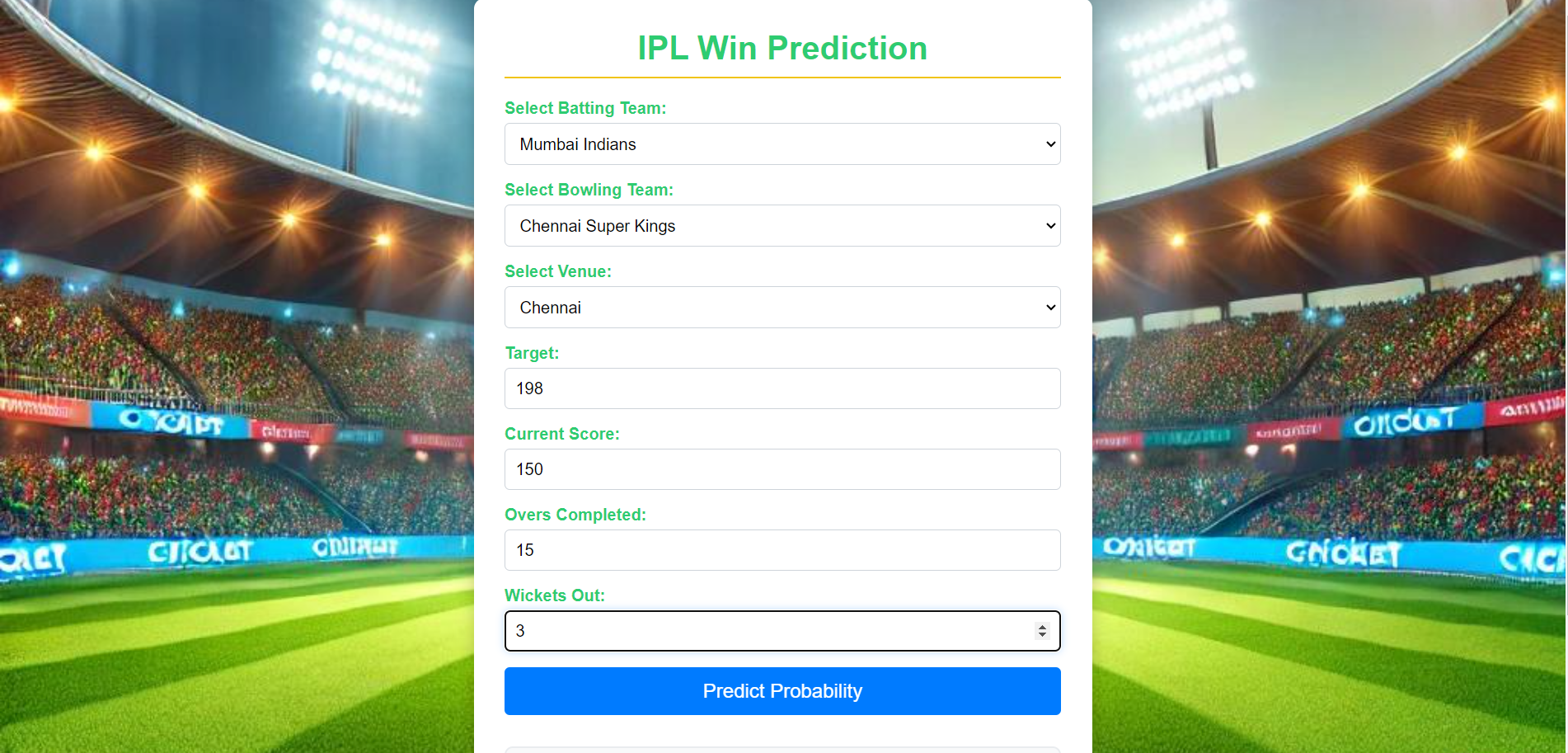
 

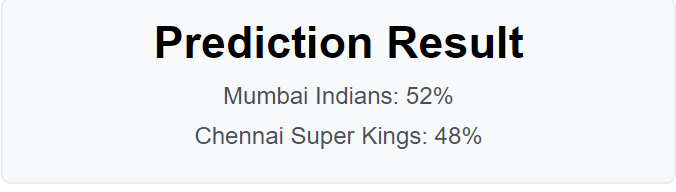


Output:-



Putting the Values





# CHAPTER 5 RESULT ANALYSIS

Four Machine learning algorithms are applied on data to perform the crop recommendation. Python is used as a programming language to develop machine learning algorithms. Various libraries of Python are used for data analysis and building machine learning models. Data is pre-processed and correlation is visualized with the help of heatmap. The data is split into 70% as a training data and 30% as a testing data. Out of these algorithms Random Forest Tree gives high accuracy of 99%. By feeding the input data Random Forest Tree gives the recommendation about the crop suitable for land.

For fertilizer recommendation, three machine learning algorithms are used like decision tree, random forest tree and SVM. Out of which SVM gives the best results with the accuracy of 97% and used for recommendation.

Three machine learning algorithms are used for yield prediction which are of regression type of algorithm. First data is analysed and yield column is generated using production and area variable. Visualization is done for analysis. The unnecessary columns like crop year and yield are dropped and data is split as 75% training data and 25% testing data. Out of these model’s linear regression gives lowest R2 score and decision tree gives the highest score of 0.88 and hence selected for yield prediction.

Farmer can use these services to make informed decisions in advance and can use tools like news feed and weather forecast to get updated. Also, they can sell crops using this portal. In some cases, these models may not be appropriate and consideration of more parameters may require depending on soil, weather conditions and factors affecting the farming but farmer can use these models for getting generalized idea about these factors. Customer can use this project for purchasing the crop from the farmers directly for everyday use.

# CHAPTER 6 CONCLUSION AND FUTURE WORK

## CONCLUSION

The IPL Match Win Predictor project successfully demonstrates the application of machine learning techniques to forecast match outcomes based on historical and real-time data. Through careful model development and evaluation, the project achieved a commendable accuracy rate, providing valuable insights into the factors influencing match results. The analysis of performance metrics, such as precision and recall, alongside tools like the confusion matrix and feature importance, allowed for a comprehensive understanding of the model's strengths and areas for improvement.

Moreover, the model's ability to adapt predictions based on live match data showcases its practicality for real-world applications, benefiting fans, analysts, and team strategists alike. Despite some limitations related to data quality and external influences, the insights gained from this project pave the way for future enhancements, such as the integration of more granular data and the expansion of the model to other formats or leagues.

Ultimately, this project underscores the potential of data-driven approaches in sports analytics, offering a robust framework for informed decision-making in the competitive landscape of the IPL. By leveraging advanced machine learning methodologies, stakeholders can gain a deeper understanding of the game, enhancing both strategic planning and viewer engagement.

## FUTURE WORK

The IPL Match Win Predictor project lays a solid foundation for further research and development in sports analytics. Future work could involve enhanced data collection by incorporating more granular datasets, such as ball-by-ball statistics, player fitness levels, and advanced tracking metrics, alongside social and psychological factors that may influence performance. Model improvement is another avenue, where exploring advanced ensemble techniques or deep learning models could capture complex patterns in the data. Additionally, developing the model's capability for real-time prediction updates during matches would enhance its utility for fans and analysts, potentially leading to the creation of user-friendly mobile and web applications. Expanding the model's applicability to other T20 leagues and sports, as well as creating interactive dashboards to visualize predictions and insights, would further engage users. Collaborating with cricket teams to refine the model using proprietary data could also provide valuable feedback for continuous improvement. By pursuing these avenues, future work can significantly enhance the capabilities of the IPL Match Win Predictor, making it a more robust tool for stakeholders while contributing to the broader field of sports analytics.

# REFERENCES

1. Kuradusenge, Martin & Hitimana, Eric & Hanyurwimfura, Damien & Rukundo, Placide & Mtonga, Kambombo & Mukasine, Angelique & Uwitonze, Claudette & Ngabonziza, Jackson & Uwamahoro, Angelique. (2023). Crop Yield Prediction Using Machine Learning Models: Case of Irish Potato and Maize. Agriculture. 13. 225. 10.3390/agriculture13010225.
2. Kavita Jhajharia, Pratistha Mathur, Sanchit Jain, Sukriti Nijhawan, Crop Yield Prediction using Machine Learning and Deep Learning Techniques, Procedia Computer Science,Volume 218,2023,Pages 406-417,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2023.01.023.
3. S. N., A., a. P. N. S. Zeel Doshi, “AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms," in 2018 *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018.
4. S., .. K. Vaishnavi.S, "Agricultural Crop Recommendations based on Productivity and Seasons," in *2021 7th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2021.

## WEBSITES

<https://www.kaggle.com/>

<https://www.geeksforgeeks.org/python-plotly-tutorial/>

<https://scikit-learn.org/>

<https://www.tutorialspoint.com/>

<https://www.w3schools.com/python/matplotlib_pyplot.asp>