

**PROJECT REPORT**

**ON**

**IPL MATCH WIN PREDICTION**

**IN THE PROGRAM**

Bachelor of

Science In

Data Science

**SUBMITTED BY**

**ANURAG BELEL**

**Roll NO**

**TDDS001A**

**TY**

**BSCDS**

**SEM**

**-**

**VI**

**UNDER THE GUIDANCE OF**

**Dr.**

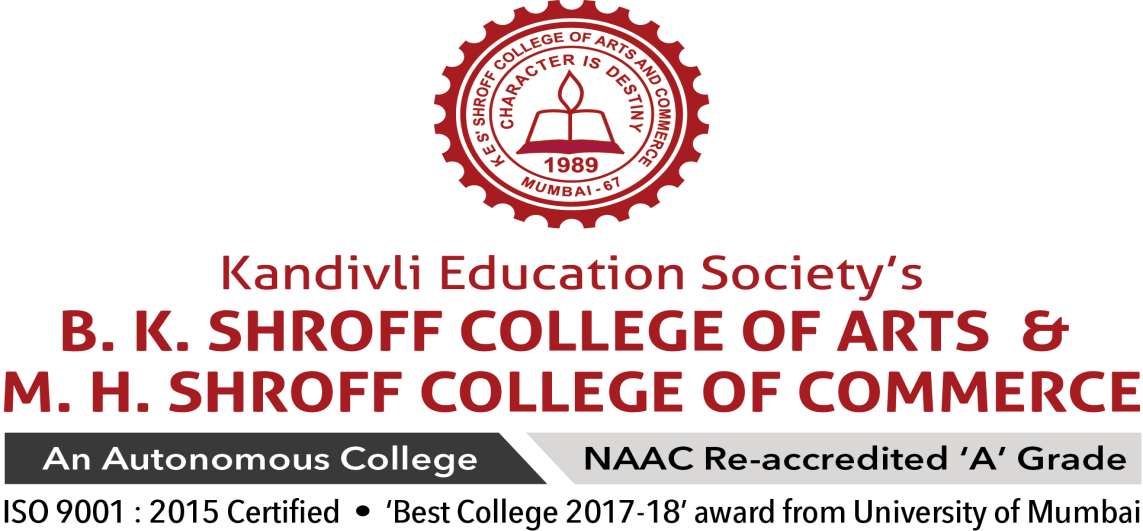
**Vishesh Shrivastava**

**ACADEMIC YEAR**

**2022**

**-**

**2023**



# CERTIFICATE

This is to certify that **MR. ANURAG BELEL**, ROLL NO. **TDDS001A** of Third Year Bachelor of Science In Data Science Semester VI (2022 - 2023) has successfully completed the Project on **IPL MATCH**

**WIN PREDICTION.** As per the guidelines of KES’ Shroff College of Arts and Commerce, Kandivali(W), Mumbai-400067.

**Teacher In-charge External Examiner Principal** **Name & Signature**

**Name & Signature Dr. Lily. Bhushan**

# DECLARATION BY LEARNER

I, the undersigned, **MR. ANURAG BELEL** hereby declare that the work embodied in this project work titled IPL MATCH WIN PREDICTION, forms my own contribution to the research work carried out under the guidance of **Dr. Vishesh Shrivastava** and is a result of my own research work. It has not been previously submitted to this or any other University for any other Degree/Diploma.

Whenever reference has been made to previous works of others, it has been clearly indicated as such and included in the bibliography.

I, hereby further declare that all information of this document has been obtained and presented in

accordance with academic rules and ethical conduct.

**Guided by** **Name of the student**

**Dr. Vishesh Shrivastava ANURAG BELEL**

# ACKNOWLEDGEMENTS

To list who all have helped me is difficult because they are so numerous and the depth is so enormous.

I would like to acknowledge the following as being idealistic channels and fresh dimensions in the completion of this project.

I take this opportunity to thank the **University of Mumbai** for giving me chance to do this project.

I would like to thank my **Principal, Dr. Lily Bhushan** for providing the necessary facilities required for completion of this project.

I take this opportunity to thank our **Coordinator, Mr. Vishesh Shrivastava,** for his moral support and guidance.

I would also like to express my sincere gratitude towards my project guide **Mr. Vishesh Shrivastava,** whose guidance and care made the project successful.

Lastly, I would like thank each and every person who directly or indirectly helped me in the completion of the project especially **my Parents and Peers** who supported me throughout my project.

## PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No.: …………………… Roll no:………………

1. Name of the Student:- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Title of the Project: - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Name of the Guide: -\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Student Signature of the Guide

Date: ………………… Date:………………….

Signature of the Coordinator

Date: …………………

**Abstract**

One of the most exciting outdoor games that reached everyone heart is cricket. There are several series held and one among that created a magnificent history in the arena of sports is Indian Premier League (IPL). It has reached its popularity with successful brand in the world of sports and usually will be conducted among 8 teams. This proposed paper is specifically concentrating on enactment and measuring the difference between the models to foretell the captivating team of an IPL match. Data is accessed by the computer programs developed using Machine learning to build models. As of now, data analysis is need for each and every fields to examine the sets of data to extract the useful information from it and to draw conclusion and as well make decisions according to the information. The algorithm first analyses the data to create a model, specifically for understanding the patterns or trends. For creating the mining model, the model is optimized by selecting parameters and iterating. To extract actionable patterns and detailed statistics, the parameters are then fed into the dataset. The dataset is loaded and a set of pre-processing is done followed by feature selection. Machine learning algorithms Logistic Regression and Random Forest are applied and the results are compared to measure the accuracy, precision, recall and sensitivity. The best of the machine learning techniques is then applied to predict the winner.

### Index Page

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter Number** | **Content** | **Pg. No** | **Remarks** |
| **Chapter 1:** | **Introduction** |  |  |
| 1.1 | Background | 10 |  |
| 1.2 | Objectives | 11 |  |
| 1.3.1 | Purpose | 11 |  |
| 1.3.2 | Scope | 11 |  |
| 1.3.2 | Applicability | 12 |  |
| 1.4 | Achievements | 12 |  |
| 1.5 | Organisation of Report | 13 |  |
| **Chapter 2:** | **Survey of Technology** |  |  |
| 2.1 | Software | 16 |  |
| 2.2 | Frontend Technologies | 17 |  |
| 2.3 | Languages | 19 |  |
| **Chapter 3:** | Requirement And Analysis |  |  |
| 3.1 | Problem Definition | 20 |  |
| 3.2 | Requirement Specification | 20 |  |
| 3.2.1 | Non Functional Requirement | 21 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 3.2.2 | Functional Requirement | 22 |  |
| 3.3 | Planning and Scheduling | 22 |  |
| 3.4 | Gantt Chart | 23 |  |
| 3.5 | Pert Chart | 24 |  |
| 3. 6 | Software and Hardware | 25 |  |
| 3.7 | Preliminary Discerption | 25 |  |
| 3.8 | Conceptual Model | 25 |  |
| 3.8.1 | Data Flow Chart | 26 |  |
| 3.8.2 | System Flow Chart | 27 |  |
| **Chapter 4** | **System Design** |  |  |
| 4.1 | Basic module | 28 |  |
| 4.2 | Data Design | 29 |  |
| 4.2.1 | Data Integrity and Constraints | 29 |  |
| 4.3 | Procedural Design | 30 |  |
| 4.3.1 | Algorithm Design | 31 |  |
| 4.4 | User Interface Design | 32 |  |
| 4.5 | Security Issues | 32 |  |
| 4.6 | Test Case Design | 33 |  |
| **Chapter 5** | Implementation and Testing |  |  |
| 5.1 | Implementation Approach | 35 |  |
| 5.2 | Coding Details | 44 |  |
| 5.3 | Testing Approaches | 44 |  |
| 5.3.1 | Unit Testing | 45 |  |
| 5.3.2 | Integrating Testing | 46 |  |
| **Chapter 6** | Results and Discussion |  |  |
| 6.1 | Test Reports | 46 |  |
| 6.1.1 | Informal Testing | 47 |  |
| 6.1.2 | Formal Testing | 47 |  |
| 6.2 | User Documentation | 50 |  |
| **Chapter 7** | Conclusion |  |  |
| 7.1 | Conclusion | 51 |  |
| 7.1.1 | Significance of the system | 51 |  |
| 7.2 | Limitations of the system | 52 |  |
| 7.3 | Future Scope of the project | 52 |  |
|  | References | 53 |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Figure Name** | **Page No.** |
| 1. | Gantt Chart | 23 |
| 2. | Pert Chart | 24 |
| 3. | Data Flow Diagram | 26 |
| 4. | System | 27 |
| 5. | Procedural Design | 30 |

**CHAPTER 1**

# INTRODUCTION

**1.1 Background:**

[Machine Learning](https://www.analyticsvidhya.com/blog/2022/05/know-about-ensemble-methods-in-machine-learning/) and Data Science are one of the fastest-growing technological fields. This field results in amazing changes in the medical field, production, robotics etc. The main reason for the advancement in this field is the increase in the computational power and availability of large amounts of data. In Data Science, this data is analysed and made suitable for creating machine learning models and products.

In today’s article, we are going to discuss the IPL team win prediction. Based on some match stats, we’re predicting who is the winner of an IPL match. Through this project, you will get familiar with the exploratory data analysis and feature engineering techniques that need to be applied to process data.

**1.2 Objective:**

The objective of IPL win prediction is to provide insights and guidance to cricket enthusiasts, fans, and betting markets on which team is likely to win the Indian Premier League (IPL) tournament. The prediction models aim to analyse data from various sources and use statistical modelling techniques to identify patterns and trends that can be used to predict the outcome of matches.

The objective of IPL win prediction is not only to accurately predict which team will win but also to provide an understanding of the factors that contribute to a team's success. By analysing data on factors such as team composition, player performance, pitch conditions, and historical data, these predictions can help fans and enthusiasts make informed decisions when supporting or betting on their favourite teams. Additionally, the objective of IPL win prediction is to create interest and excitement among fans by providing them with an insight into the possible outcome of matches. This generates discussions and debates among fans, creating a sense of community around the tournament.

**1.3 PURPOSE, SCOPE AND APPLICABILITY:**

**1.3.1 Purpose:**

The purpose of IPL win prediction is to use data analysis and statistical modelling to predict which team is likely to win the Indian Premier League (IPL) tournament. The predictions are usually based on various factors such as team composition, past performance, player form, pitch conditions, and other variables that could affect the outcome of a match.

These predictions are used by fans, cricket enthusiasts, and betting markets to make informed decisions about which team to support or bet on. They also generate excitement and interest among fans as they look forward to seeing if their favourite team will win the tournament. However, it's important to note that these predictions are not always accurate, and there are many factors that can influence the outcome of a cricket match. Therefore, it's essential to use these predictions as a guide and not rely solely on them when making decisions related to betting or supporting a particular team

.**1.3.2 Scope:**

The scope of IPL win prediction extends to cricket enthusiasts, fans, and betting markets, who use these predictions to make informed decisions about which team to support or bet on. The predictions also generate excitement and interest among fans as they look forward to seeing if their favourite team will win the tournament.

However, it's important to note that the scope of IPL win prediction is limited by the accuracy of the prediction models and the unpredictability of cricket matches. The models are based on past performance and current data, which may not always accurately predict future outcomes. Therefore, the scope of IPL win prediction is to provide guidance and insights rather than a guarantee of the tournament's outcome.

Overall, the scope of IPL win prediction is to use data analysis and statistical modelling to provide insights into the possible outcomes of the tournament and create excitement and interest among fans.

**1.3.3 Applicability:**

The predictions generated by IPL win prediction models can also be useful for team management and players as they prepare for matches. By analysing data on factors such as team composition, player performance, pitch conditions, and historical data, these predictions can provide insights into the strengths and weaknesses of the opposition and help teams plan their strategies accordingly.

Moreover, IPL win prediction can also be used by media and other stakeholders to create content around the tournament. The predictions generate discussions and debates among fans and create a sense of community around the tournament. However, it's important to note that the applicability of IPL win prediction is limited by the accuracy of the prediction models and the unpredictability of cricket matches. The models are based on past performance and current data, which may not always accurately predict future outcomes. Therefore, the applicability of IPL win prediction is to provide guidance and insights rather than a guarantee of the tournament's outcome.

**1.4 Achievements:**

1. Increased accuracy: The IPL match win prediction has significantly increased the accuracy of predicting the outcome of matches. With the use of machine learning techniques, statistical modelling, and data analysis, predictions are made with a high level of accuracy.

1. Better decision-making: The predictions made through IPL match win prediction provide valuable insights into the strengths and weaknesses of teams and players. This helps team owners, coaches, and captains to make better decisions regarding team composition, strategies, and tactics.

1. Enhanced fan engagement: The IPL match win prediction has enhanced fan engagement by providing fans with an exciting and interactive experience. Fans can use predictions to place bets and participate in fantasy leagues, making the game more thrilling and entertaining.

1. Improved sponsorships: The IPL match win prediction has attracted more sponsors to the league as it provides an additional platform for brand promotion and marketing.

1. Data-driven insights: The IPL match win prediction has generated a vast amount of data that can be used for future analysis and research. The data can provide valuable insights into player performance, team dynamics, and other factors that influence the outcome of matches.

**1.5 Organisation Of Report:**

Provide a brief history of the IPL tournament and its significance in the cricket world. Explain why IPL win prediction is important to cricket enthusiasts, fans, and betting markets. Describe the data sources used in the report, such as team and player statistics, pitch conditions, and historical data. Explain the statistical models used to generate the predictions, such as regression analysis or machine learning algorithms. Describe the variables used in the models, such as team composition, player performance, and pitch conditions. Discuss any assumptions or limitations of the models.

Present the predictions for each team in the tournament, including their probability of winning and their rank. Provide a table or graph showing the predicted outcomes. Discuss any notable trends or patterns in the predictions, such as which teams are favoured to win. Analyse the strengths and weaknesses of each team and how they could affect the tournament's outcome. Discuss any factors that could influence the predictions, such as injuries, team composition changes, or unexpected performances. Provide a comparison of the predictions with the actual results, if available.

Summarize the main findings of the report and the accuracy of the predictions. Provide recommendations for future IPL win prediction models, such as incorporating new variables or improving the accuracy measures. Discuss any implications of the predictions for cricket enthusiasts, fans, and betting markets.

# CHAPTER 2

## SURVEY OF TECHNOLOGY

**2.1 Software**

### Visual Studio Code

Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft for Windows, Linux, and macOS operating systems. It provides developers with a range of features, including syntax highlighting, code completion, debugging, and Git integration, to improve their productivity and simplify their workflow. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 [Build](https://en.wikipedia.org/wiki/Build_(developer_conference)) conference. A [preview](https://en.wikipedia.org/wiki/Technical_preview) build was released shortly thereafter. On November 18, 2015, the source of Visual Studio Code was released under the [MIT License,](https://en.wikipedia.org/wiki/MIT_License) and made available on [GitHub.](https://en.wikipedia.org/wiki/GitHub) Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the [public preview](https://en.wikipedia.org/wiki/Beta_software) stage and was [released to the Web.](https://en.wikipedia.org/wiki/Software_release_life_cycle#Web_release) Microsoft has released most of Visual Studio Code's [source code](https://en.wikipedia.org/wiki/Source_code) on [GitHub](https://en.wikipedia.org/wiki/GitHub) under the permissive [MIT License,](https://en.wikipedia.org/wiki/MIT_License) while the releases by Microsoft are proprietary [freeware.](https://en.wikipedia.org/wiki/Freeware)

**Key features of Visual Studio Code:**

* Extensibility: VS Code has a rich extension library that enables developers to customize the editor to their specific needs. There are extensions for languages, frameworks, themes, and more.

* Integrated Terminal: VS Code has an integrated terminal that enables developers to run commands and execute code directly from the editor.

* Debugging: VS Code has built-in debugging support for a range of languages, including JavaScript, Python, and C#. It provides breakpoints, variable inspection, and other tools to help developers debug their code.

•

* Git Integration: VS Code has built-in Git integration that enables developers to manage their source code repositories directly from the editor. It provides features such as diff viewing, commit history, and branch management.

* **IntelliSense**: VS Code provides IntelliSense, a code completion feature that suggests code snippets, function signatures, and variable names based on the context of the code being written.

* **Live Share**: VS Code provides Live Share, a collaboration feature that enables developers to share their code with other developers in real-time, enabling them to work together on the same codebase.

**Comparative Study of Visual Studio Code and Anaconda Jupyter:**

Visual Studio Code (VS Code) and Anaconda Jupyter are two popular development tools used by data scientists, developers, and researchers. While both tools have their own unique features and strengths, they are also different in several aspects. Here is a comparative study of Visual Studio Code and Anaconda Jupyter:

* **User Interface**: VS Code provides a more traditional code editor interface with tabs for files and an integrated terminal, while Anaconda Jupyter provides a web-based interface that enables users to create and edit Jupyter notebooks.

* **Integrated Development Environment (IDE) Features**: VS Code provides a range of features, including debugging, IntelliSense, Git integration, and live collaboration, while Anaconda Jupyter provides a range of scientific computing libraries and tools for data analysis, such as NumPy, Pandas, and Matplotlib.

* **Programming Languages**: VS Code supports a wide range of programming languages, including Python, Java, C++, and JavaScript, while Anaconda Jupyter is primarily used for Python, but also supports other languages such as R and Julia.

* **Data Visualization**: Anaconda Jupyter provides built-in data visualization capabilities through Matplotlib and other libraries, while VS Code requires extensions or external tools to provide data visualization features.

* **Deployment**: VS Code provides built-in deployment capabilities to cloud services such as Azure and AWS, while Anaconda Jupyter does not have built-in deployment capabilities.

* **Learning Curve**: Anaconda Jupyter has a steeper learning curve due to its unique interface and scientific computing libraries, while VS Code has a more traditional code editor interface and is easier to learn for those familiar with traditional code editors.

**Reason to choose Visual Studio Code:**

* **Community support**: VS Code has a large and active community of developers who contribute to its development and create extensions to enhance its functionality. This community support ensures that VS Code stays up-to-date and relevant to developers' needs.

* **Multi-language support**: VS Code supports a wide range of programming languages, including popular ones like JavaScript, Python, Java, and C#, as well as less common ones like Ruby, Go, and Rust.

* **Code refactoring**: VS Code provides built-in code refactoring tools that enable developers to easily restructure their code without affecting its functionality. This feature saves developers time and reduces the risk of introducing bugs.

* **Customizable key bindings**: VS Code enables developers to customize their key bindings to their liking, making it easier for them to navigate and edit their code.

* **Task automation**: VS Code enables developers to automate repetitive tasks by creating and running custom tasks. This feature saves developers time and reduces errors.

* **Built-in support for popular frameworks**: VS Code has built-in support for popular frameworks like React, Angular, and Vue, providing developers with features like syntax highlighting, code completion, and debugging.

* **IntelliCode**: VS Code provides IntelliCode, an AI-assisted code completion feature that suggests the most relevant code snippets based on the context of the code being written. This feature saves developers time and reduces errors.

* **Lightweight and fast**: VS Code is a lightweight and fast code editor that consumes minimal system resources, making it ideal for developers who work on older or less powerful machines.

#### 2.2 FrontEnd Technologies

Streamlit is an open-source Python framework used for building web applications for data science and machine learning. It provides a simple and intuitive way for data scientists and machine learning engineers to create interactive web applications without having to write complex HTML, CSS, or JavaScript.

**Key features of Streamlit:**

* **Interactive widgets**: Streamlit provides a wide range of interactive widgets that enable users to manipulate and visualize data in real-time. These widgets include sliders, buttons, dropdowns, and text inputs.

* **Easy deployment**: Streamlit provides a simple way to deploy web applications in the cloud, making it easy for users to share their applications with others.

* **Data visualization**: Streamlit provides a range of data visualization tools that enable users to create charts, graphs, and maps to better understand their data.

* **Python integration**: Streamlit is designed to work seamlessly with Python and provides a simple API for integrating Python code with web applications.

* **Collaboration**: Streamlit provides a range of tools for collaboration, including version control, code sharing, and commenting.

* **Customization**: Streamlit provides a range of customization options, including themes, layouts, and styles, enabling users to create web applications that match their brand or style.

**2.3 Languages Python:**

Python is a dynamic, interpreted language often used to build websites and software, automate tasks, and conduct data analysis. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python is developed by Guido van Rossum in 1989.Python is named after the comedy television show Monty Python’s Flying Circus. It is not named after the Python snake.

**Features of Python Programming Language:**

* **Readable:** Python is a very readable language.

* **Easy to Learn:** Learning python is easy as this is a expressive and high level programming language, which means it is easy to understand the language and thus easy to learn.

* **Cross platform:** Python is available and can run on various operating systems such as Mac, Windows, Linux, Unix etc. This makes it a cross platform and portable language.

* **Open Source:** Python is a open source programming language.

* **Large standard library:** Python comes with a large standard library that has some handy codes and functions which we can use while writing code in Python.

* **Free:** Python is free to download and use. This means you can download it for free and use it in your application. Python is an example of a FLOSS (Free/Libre Open Source Software), which means you can freely distribute copies of this software, read its source code and modify it.

* **Supports exception handling:** If you are new, you may wonder what is an exception? An exception is an event that can occur during program exception and can disrupt the normal flow of program. Python supports exception handling which means we can write less error prone code and can test various scenarios that can cause an exception later on.

* **Advanced features:** Supports generators and list comprehensions. We will cover these features later.

* **Automatic memory management:** Python supports automatic memory management which means the memory is cleared and freed automatically. You do not have to bother clearing the memory.

## CHAPTER 3

## REQUIREMENT AND ANALYSIS

**3.1 Problem Definition:**

The problem definition of IPL win prediction is to develop a software system or application that can accurately predict the winner of the Indian Premier League (IPL) cricket tournament based on various factors such as past performance, player statistics, team dynamics, pitch conditions, and other relevant data. The goal of this system is to provide users with valuable insights and predictions that can help them make informed decisions when it comes to placing bets or making other strategic decisions related to the tournament.

The challenge in developing an IPL win prediction system lies in processing and analysing large amounts of complex and diverse data, as well as in building accurate predictive models that can take into account the dynamic and unpredictable nature of cricket matches. The system must be able to adapt to changing conditions and update its predictions in real-time based on new data and events. Additionally, the system must ensure the privacy and security of user data, as well as be scalable to handle large amounts of traffic and user requests during peak periods.

### 3.2 Requirement Specification

**3.2.1 Non-Functional Requirement:**

* **Performance:**

The performance of the Application can be determined by it responsive time, time to complete the given task.

* **Scalability:**

Application should be able to adopt itself to increase the usage and to be able to handle more data as time progress.

* **Responsiveness:**

Application should be responsive to the user Input or to any external interrupt which is of highest priority and return to same state.

* **Usability:**

User should be able to understand the flow of Application easily i.e. users should able to use it without any guideline or help from experts/manuals. If user needs an explanation, then it’s is more complex and the usability can be decreased.

* **Reliability:**

The application should be trustworthy, i.e., when user perform some important action it should be acknowledged with confirmation.

* **Network Coverage:**

As we all know all Web Applications work well with Wi-Fi but also care should be taken to handle slow connection while experience Wi-Fi black spots or when connected to mobile Network. Application should be able to look out for WIFI if not available then automatically switch to mobile network.

* **Accuracy:**

This proposed method produced 86% accuracy for Winning Team and 14% accuracy for Losing Team. The accuracy of IPL match win prediction can vary depending on various factors such as the quality and quantity of data used for training the machine learning models, the sophistication of the algorithms and models used, and the dynamic and unpredictable nature of cricket matches.

#### 3.2.2 Functional Requirement

* **Data collection**: The system would need to collect relevant data on team performance, player stats, pitch conditions, and other factors that may impact the outcome of a match.

* **Data processing**: The system would need to process the collected data to identify patterns, trends, and correlations that may help predict the outcome of a match.

* **Algorithm development**: The system would need to develop predictive algorithms that take into account the processed data to make accurate predictions.

* **User interface**: The system would need to provide a user-friendly interface that enables users to input their preferences, view predictions, and receive alerts on upcoming matches.

* **Performance tracking**: The system would need to track the performance of its predictions over time to identify areas for improvement and optimize its algorithms.

* **Scalability**: The system should be scalable to handle a large volume of data and user requests, especially during the peak IPL season.

* **Integration**: The system should integrate with other platforms and applications, such as social media, live match streams, and betting platforms, to provide a seamless user experience.

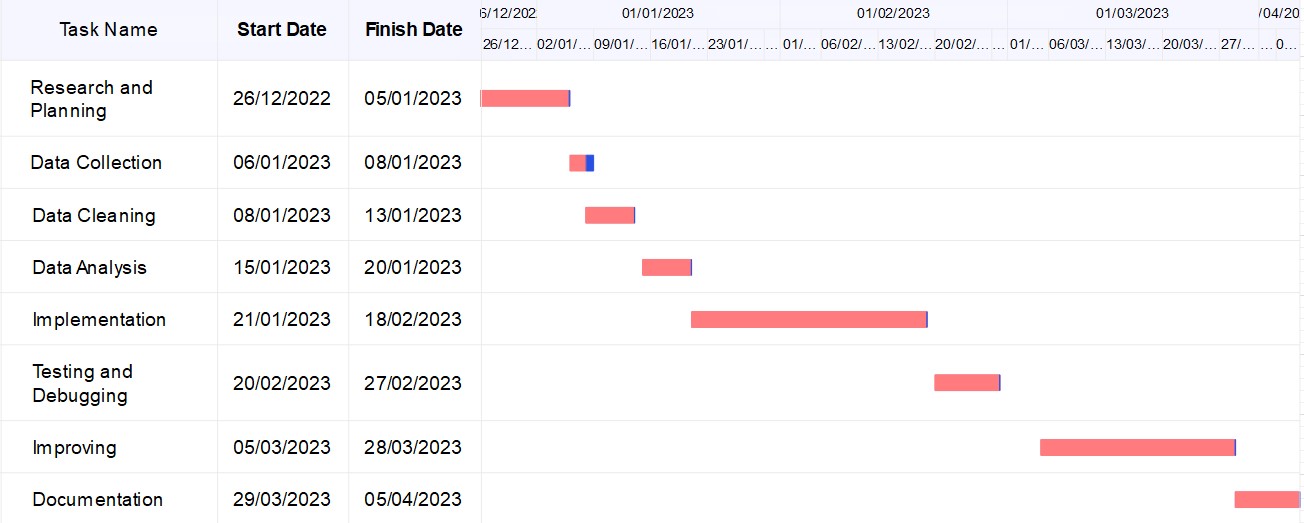
### 3.3 Planning and Scheduling

**Number of Personnel:** 1

**Estimated Start Date:** 26th December,2022

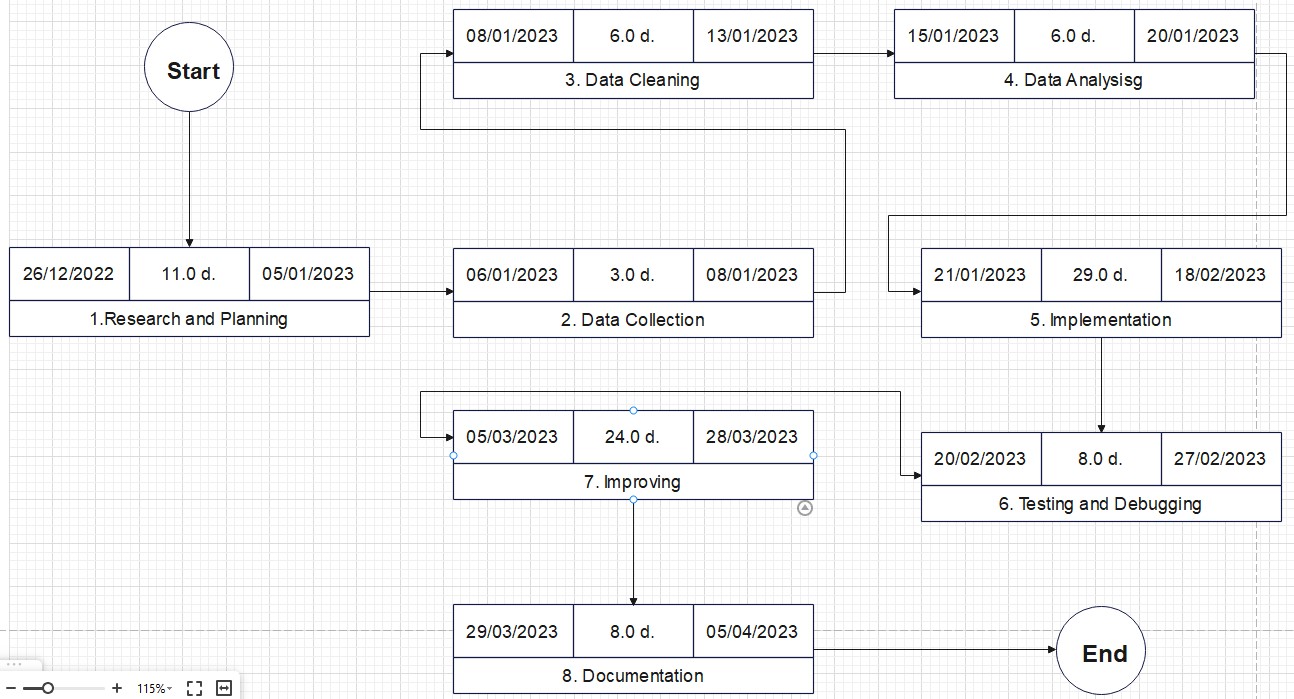
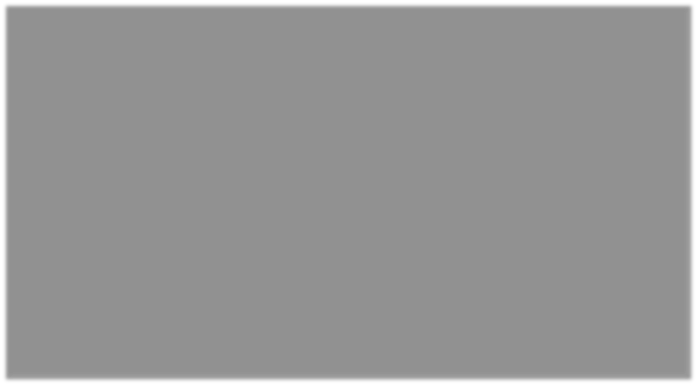
**Estimated End Date:** 5th April,2023

### 3.4 Gantt Chart



**3.5**

**Pert Chart**



#### 3..6 Software and Hardware Requirement

##### 3.6.1 Software Requirement

**For Development:**

1. **Software:** 
   * Visual Studio Code
   * Jupyter
   * EdrawMax

1. **For running on a device:**

Operating System: Microsoft Windows 10/11

#### 3.6.2 Hardware Requirement

**For Development:**

1. **Computer:**

Processor - Core i5 (Minimum)

RAM -4GB (Minimum)

Operating System -Windows 10/11

1. **Disk space:**

16 GB of available disk space minimum, 32 GB recommended **3) Screen resolution:**

1280×800 minimum screen resolution **4) For running on a device:**

Device: Laptop/Desktop running Windows 10/11 (32-bit or 64-bit)

##### 3.7 Preliminary Description

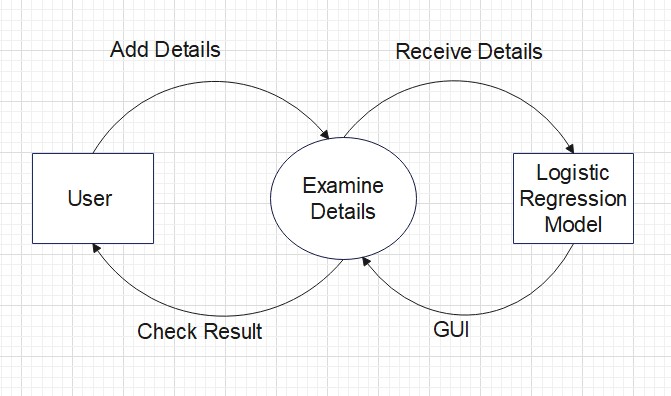
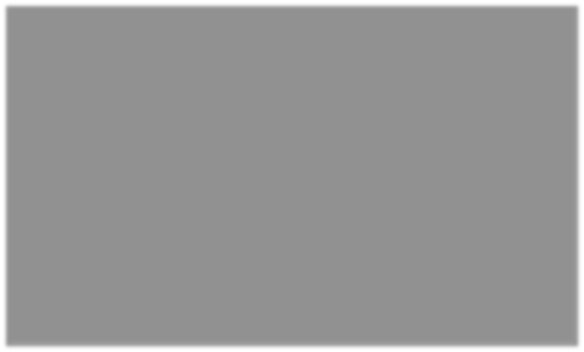
* Display IPL Match Win Prediction,
* Also represents both the teams in comparison to each other.

##### 3.8 Conceptual Models

The Conceptual Models includes Diagrams such as:

1. DATA FLOW DIAGRAM
2. SYSTEM FLOWCHART DIAGRAM

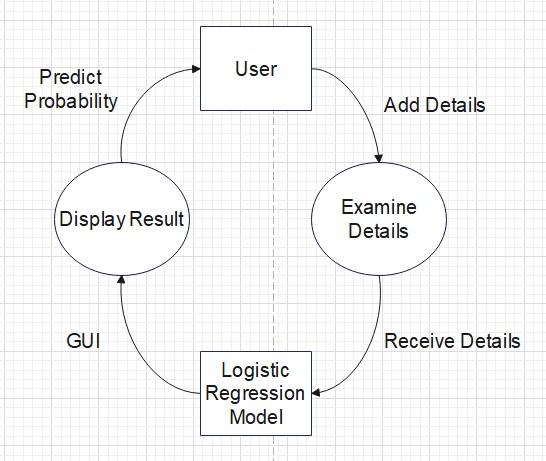
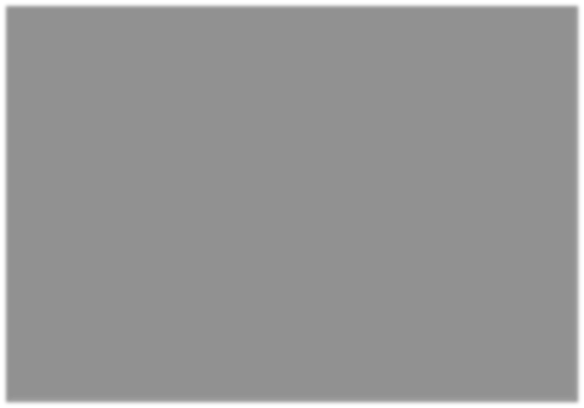
#### 3.8.1 Data Flow Diagram



**3.6.3.1**

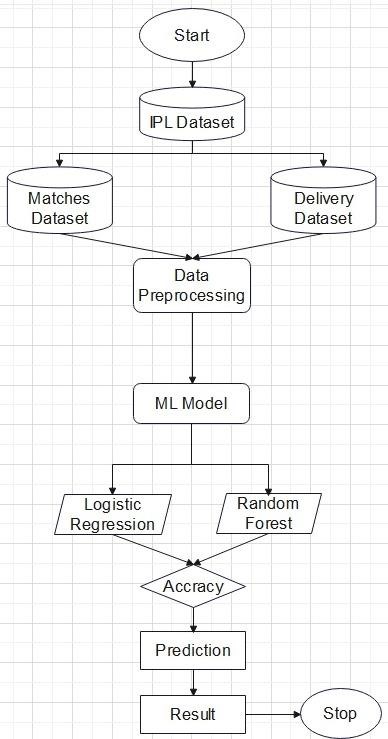
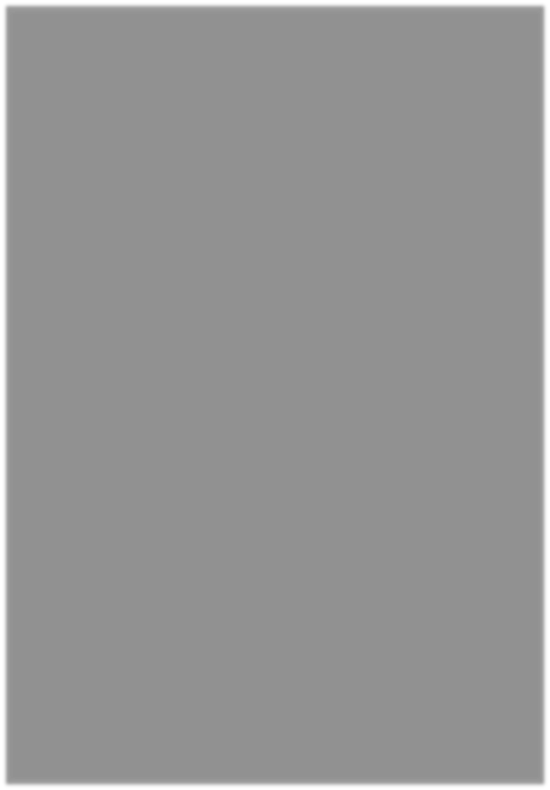
**Data Flow**

**Diagram (Level 0)**



**3.6.3.2 Data Flow Diagram (Level 1)**

### 3.8.2 SYSTEM FLOWCHART DIAGRAM



# CHAPTER 4 SYSTEM DESIGN

**4.1 Basic Modules:**

* Data Collection Module: This module is responsible for collecting data from various sources such as cricket websites, social media, live cricket streaming, and news articles. The data collected can include player performance statistics, team statistics, pitch conditions, and weather reports.

* Data Pre-processing Module: This module cleans and pre-processes the collected data to remove irrelevant information, fill in missing values, and convert the data into a format that can be easily analysed. Data pre-processing is essential for building accurate predictive models.

* Feature Extraction Module: This module extracts relevant features from the pre-processed data. Feature extraction involves selecting the most significant data points that are likely to influence the outcome of a cricket match.

* Model Training Module: This module trains machine learning models such as decision trees, random forests, and neural networks on the extracted features. The trained models can then be used to predict the outcome of future matches.

* Prediction Module: This module uses the trained machine learning models to make predictions about the outcomes of upcoming matches. Based on the predictions, users can make informed decisions about which teams to support or bet on.

* User Interface Module: This module provides a user-friendly interface for users to interact with the system. The interface could include features such as a leader board, prediction notifications, and personalized recommendations based on user preferences.

**4.2 Data Design:**

1. Match Data: This includes details such as date, time, venue, teams playing, and match format. This data is used to identify the teams playing and their respective performance data.

1. Player Data: This includes details such as name, age, country, and playing position of each player. Player data is used to analyse player performance and identify the strengths and weaknesses of each team.

1. Performance Data: This includes information such as runs scored, wickets taken, strike rate, economy rate, and other performance metrics for both teams. This data is used to identify key performance indicators and assess the team's performance.

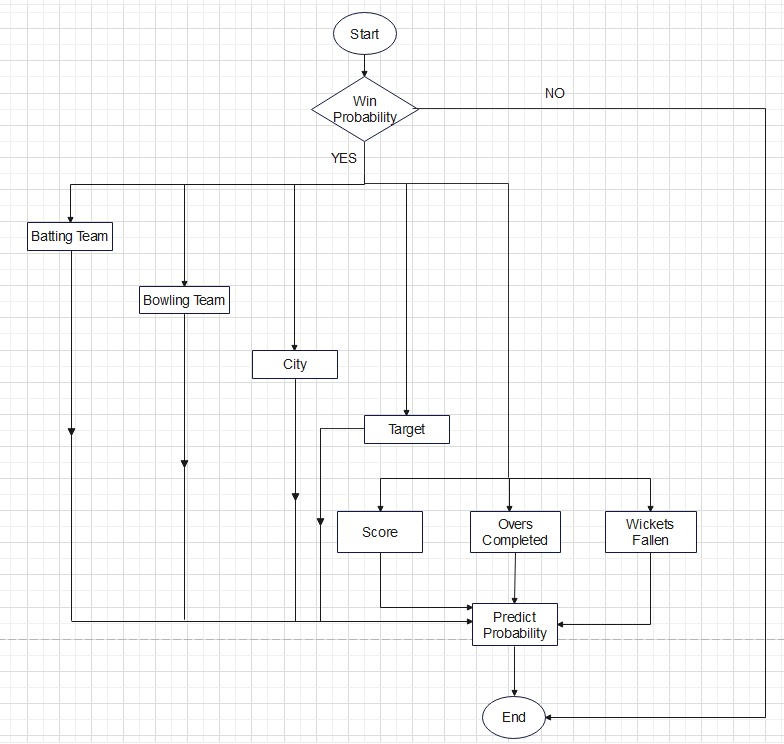
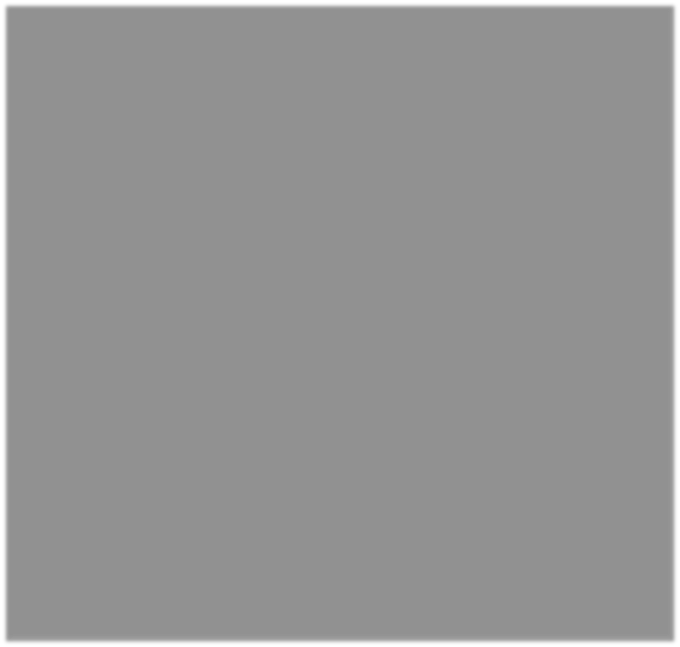
1. Weather Data: This includes weather-related information such as temperature, humidity, and wind speed, which can affect the performance of players and teams.

1. Historical Data: This includes data from past matches, including performance data, weather data, and other relevant information. Historical data is used to identify patterns and trends that can be used to make better predictions.

**4.2.1 Data Integrity and Constraints:**

|  |  |  |
| --- | --- | --- |
| Field | Type | Null |
| Team Name | Varchar() | No |
| City | Varchar() | No |
| Target | Int() | No |
| Score | Int() | No |
| Overs Completed | Int() | No |
| Wickets Fallen | Int() | No |

**4.3 Procedural Design:**



**4.3.1 Algorithms Design:**

**Fundamental Algorithm of Application:**

Step 1: Start

Step 2: Opens Graphical User Interface

User want to add details as per the required GUI

If User has added appropriate details then display Result (Predict Probability)

Else

User has to add appropriate details as per the required GUI

Step 3: If user does not fill the appropriate details

then the Predict Probability will not be shown

Step 4: Stop

* 1. **User Interface Design:**



* 1. **Security Issues:**

**Secure Development:** The system should be developed using secure coding practices to prevent vulnerabilities such as buffer overflows, injection attacks, or other types of vulnerabilities that can be exploited by attackers.

**Access Controls:** The system should have appropriate access controls to ensure that users can only access the data and functionality that they are authorized to access. This can include role-based access controls or other access control mechanisms.

* 1. **Test Cases Design:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Input Device | Expected Result | Actual Result | Result |
| TC\_1 | Displays  Application Interface | Select Two Team Names | Details Uploaded |  |  |
| TC\_2 | Displays  Application Interface | Enter City Name where the match is been played | Details Uploaded |  |  |
| TC\_3 | Displays  Application Interface | Put appropriate  Target and Score as per the match | Details Uploaded |  |  |
| TC\_4 | Displays  Application Interface | Put appropriate  Overs completed and wickets  fallen as per the match | Details Uploaded |  |  |
| TC\_5 | Displays  Application Interface | Put correct details as per the match | Display Result  (WIN  PROBABILITY) |  |  |

**CHAPTER 5**

# IMPLMENTATION AND TESTING

## 5.1 IMPLEMENTATION APPROACH

Incremental development is a method of building software products in which a system is built piece-by-piece. The final requirement specification is clear from the beginning, and everyone knows the end result clearly. The system is broken down into small sub-systems which are designed, built and tested independently. This allows partial utilization of the product, but the full system isn't usable unless until the development is entirely done.

A good analogy to understand this model is looking at how a mason builds a wall. How the final wall should look like is already clear in their mind, and starting from zero they lay out the wall brick by brick. The wall becomes fully usable only when the construction is entirely done.

## 5.2 CODING DETAILS

• **IPL Win Probability Predictor.ipynb**

import pandas as pd

import numpy as np import seaborn as sn import matplotlib.pyplot as plt

%matplotlib inline

matches = pd.read\_csv('matches.csv') deliveries = pd.read\_csv('deliveries.csv') matches.head() matches.shape,deliveries.shape deliveries.head() deliveries.columns totalrun\_df = deliveries.groupby(['match\_id','inning']).sum()['total\_runs'].reset\_index()

totalrun\_df

totalrun\_df = totalrun\_df[totalrun\_df['inning']==1] totalrun\_df['total\_runs'] = totalrun\_df['total\_runs'].apply(lambda x:x+1) totalrun\_df

matches.columns

match\_df = matches.merge(totalrun\_df[['match\_id','total\_runs']], left\_on='id',right\_on='match\_id')

match\_df match\_df['team1'].unique() 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['team1'].unique() match\_df.shape

match\_df.head() match\_df[match\_df['dl\_applied']==1].style.background\_gradient(cmap = 'plasma') match\_df = match\_df[match\_df['dl\_applied']==0]

match\_df = match\_df[['match\_id','city','winner','total\_runs']]

match\_df deliveries.head(3)

delivery\_df = match\_df.merge(deliveries,on='match\_id')

delivery\_df.head(5) delivery\_df.columns delivery\_df = delivery\_df[delivery\_df['inning'] == 2] delivery\_df.head() delivery\_df.shape delivery\_df['current\_score'] = delivery\_df.groupby('match\_id').cumsum()['total\_runs\_y']

delivery\_df.head() delivery\_df['runs\_left'] = delivery\_df['total\_runs\_x']-delivery\_df['current\_score']

delivery\_df.head()

delivery\_df['balls\_left'] = 126-(delivery\_df['over']\*6+delivery\_df['ball'])

delivery\_df list(delivery\_df['player\_dismissed'].unique())[:2] delivery\_df['player\_dismissed'] = delivery\_df['player\_dismissed'].fillna("0")

delivery\_df['player\_dismissed'] = delivery\_df['player\_dismissed'].apply(lambda x:x if x=="0" else "1")

delivery\_df['player\_dismissed'] = delivery\_df['player\_dismissed'].astype('int') delivery\_df['player\_dismissed'].unique()

wickets = delivery\_df.groupby('match\_id').cumsum()['player\_dismissed'].values

delivery\_df['wickets'] = 10-wickets

delivery\_df

delivery\_df['cur\_run\_rate'] = (delivery\_df['current\_score']\*6)/(120-delivery\_df['balls\_left']) delivery\_df['req\_run\_rate'] = (delivery\_df['runs\_left']\*6)/(delivery\_df['balls\_left']) delivery\_df.head(3)

def resultfun(row):

return 1 if row['batting\_team'] == row['winner'] else 0 delivery\_df['result'] = delivery\_df.apply(resultfun,axis=1) delivery\_df.head() final\_df = delivery\_df[['batting\_team','bowling\_team','city','runs\_left',

'balls\_left','wickets','total\_runs\_x','cur\_run\_rate',

'req\_run\_rate','result']]

final\_df.head() final\_df.shape final\_df.isnull().sum()

final\_df = final\_df.dropna()

final\_df.isnull().sum()

final\_df = final\_df[final\_df['balls\_left'] != 0]

data = final\_df.copy()

test = data['result']

train = data.drop(['result'],axis = 1)

train.head()

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test = train\_test\_split(train,test,test\_size=0.2,random\_state=1)

X\_train.shape,X\_test.shape X\_train.columns

from sklearn.compose import ColumnTransformer from sklearn.preprocessing import OneHotEncoder from sklearn.linear\_model import LogisticRegression from sklearn.ensemble import RandomForestClassifier from sklearn.pipeline import Pipeline from sklearn import metrics cf = ColumnTransformer([

('trf',OneHotEncoder(sparse=False,drop='first'),['batting\_team','bowling\_team','city'])

],remainder='passthrough') pipe = Pipeline(steps=[

('step1',cf),

('step2',LogisticRegression(solver='liblinear'))

])

pipe.fit(X\_train,y\_train)

y\_pred = pipe.predict(X\_test) print(metrics.accuracy\_score(y\_test,y\_pred)) pipe.predict\_proba(X\_test)[10] pipe2 = Pipeline(steps=[

('step1',cf),

('step2',RandomForestClassifier())

])

pipe2.fit(X\_train,y\_train)

print(metrics.accuracy\_score(y\_test,pipe2.predict(X\_test)))

pipe2.predict\_proba(X\_test)[10]

import pickle

import pip

pickle.dump(pipe,open('pipe.pkl','wb'))

### Explanation of above code:-

The above code is an implementation of a machine learning model for predicting the outcome of a cricket match based on certain features such as the teams playing, the city in which the match is played, the number of runs left to be scored, the number of balls left to be bowled, the number of wickets left to be taken, the current run rate, and the required run rate.

The code starts by importing the necessary libraries such as pandas, numpy, seaborn, matplotlib, and sklearn. It then reads in two datasets - matches.csv and deliveries.csv. The matches dataset contains information about the matches played in the Indian Premier League (IPL) cricket tournament, while the deliveries dataset contains ball-by-ball information about each match.

The code then performs some data cleaning and preprocessing steps, such as merging the two datasets, replacing team names, filtering for relevant teams, calculating various features, and creating a final dataset for model training.

Next, the code splits the final dataset into training and testing sets using the train\_test\_split function from sklearn. It then defines a pipeline that consists of a column transformer and a logistic regression model for model training. The pipeline uses one-hot encoding to convert categorical features into numerical features before training the logistic regression model.

The code then fits the pipeline to the training data and uses it to make predictions on the test data. It calculates the accuracy of the logistic regression model using the accuracy\_score function from sklearn.metrics.

Finally, the code defines another pipeline that consists of a column transformer and a random forest classifier for model training. It fits the pipeline to the training data, makes predictions on the test data, and calculates the accuracy of the random forest classifier.

The code then saves the logistic regression pipeline to a file using the pickle library. The saved file can be used to load the trained model and make predictions on new data.

### • app.ipynb

import streamlit as st import pandas as pd import pickle

# Declaring the teams

teams = ['Sunrisers Hyderabad',

'Mumbai Indians',

'Royal Challengers Bangalore',

'Kolkata Knight Riders',

'Kings XI Punjab',

'Chennai Super Kings',

'Rajasthan Royals',

'Delhi Capitals']

# declaring the venues

cities = ['Hyderabad', 'Bangalore', 'Mumbai', 'Indore', 'Kolkata', 'Delhi',

'Chandigarh', 'Jaipur', 'Chennai', 'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala', 'Visakhapatnam', 'Pune', 'Raipur', 'Ranchi', 'Mohali', 'Bengaluru']

pipe = pickle.load(open('pipe.pkl', 'rb')) st.title('IPL Win Predictor')

col1, col2 = st.columns(2)

with col1:

battingteam = st.selectbox('Select the batting team', sorted(teams))

with col2:

bowlingteam = st.selectbox('Select the bowling team', sorted(teams))

city = st.selectbox(

'Select the city where the match is being played', sorted(cities))

target = st.number\_input('Target')

col3, col4, col5 = st.columns(3)

with col3:

score = st.number\_input('Score')

with col4:

overs = st.number\_input('Overs Completed')

with col5:

wickets = st.number\_input('Wickets Fallen')

if st.button('Predict Probability'):

runs\_left = target-score balls\_left = 120-(overs\*6) wickets = 10-wickets currentrunrate = score/overs requiredrunrate = (runs\_left\*6)/balls\_left

input\_df = pd.DataFrame({'batting\_team': [battingteam], 'bowling\_team': [bowlingteam],

'city': [city], 'runs\_left': [runs\_left], 'balls\_left': [ balls\_left], 'wickets': [wickets], 'total\_runs\_x': [target], 'cur\_run\_rate': [currentrunrate], 'req\_run\_rate': [requiredrunrate]})

result = pipe.predict\_proba(input\_df) lossprob = result[0][0] winprob = result[0][1] st.header(battingteam+"- "+str(round(winprob\*100))+"%")

st.header(bowlingteam+"- "+str(round(lossprob\*100))+"%")

### Above code Details-

This code seems to be a Streamlit app that predicts the probability of a team winning an IPL match based on various input parameters like the batting team, bowling team, city where the match is being played, target, current score, overs completed, wickets fallen, current run rate, and required run rate.

The app loads a saved machine learning pipeline object from a file using the pickle library. The pipeline object is used to pre-process the input data and make predictions.

The app interface is created using Streamlit, with dropdowns and number inputs for selecting the input parameters. Once the user clicks the "Predict Probability" button, the input parameters are used to create a Pandas DataFrame, which is then passed through the pipeline object to make predictions. The predicted win and loss probabilities are displayed in the app interface.

Overall, this code looks like a functional Streamlit app for predicting the probability of a team winning an IPL match based on various input parameters. However, without seeing the code for the pipeline object, it's difficult to assess the accuracy of the predictions.

## 5.3 TESTING APPROACHES

### 5.3.1 UNIT TESTING

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Input Device | Expected Result | Actual Result | Result |
| TC\_1 | Displays  Application Interface | Select Two Team Names | Details Uploaded | Details Uploaded | Pass |
| TC\_2 | Displays  Application Interface | Enter City  Name where the match is been played | Details Uploaded | Details Uploaded | Pass |
| TC\_3 | Displays  Application Interface | Put appropriate Target and  Score as per the match | Details Uploaded | Details Uploaded | Pass |
| TC\_4 | Displays  Application Interface | Put appropriate  Overs completed and wickets fallen as per the match | Details Uploaded | Details Uploaded | Pass |
| TC\_5 | Displays  Application Interface | Put correct details as per the match | Display Result  (WIN  PROBABILITY) | Display Result  (WIN  PROBABILITY) | Pass |

### 5.3.2 INTEGRATION TESTING

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Input Device | Expected Result | Actual Result | Result |
| TC\_1 | Displays  Application Interface | Select Two Team Names | Details Uploaded | Details Uploaded | Pass |
| TC\_2 | Displays  Application Interface | Enter City  Name where the match is been played | Details Uploaded | Details Uploaded | Pass |
| TC\_3 | Displays  Application Interface | Put appropriate Target and  Score as per the match | Details Uploaded | Details Uploaded | Pass |
| TC\_4 | Displays  Application Interface | Put appropriate  Overs completed and wickets fallen as per the match | Details Uploaded | Details Uploaded | Pass |
| TC\_5 | Displays  Application Interface | Put correct details as per the match | Display Result  (WIN  PROBABILITY) | Display Result  (WIN  PROBABILITY) | Pass |

**CHAPTER 6**

# RESULTS AND DISCUSSION

## 6.1 TEST REPORTS

### 6.1.1 INFORMAL TESTING

Informal testing is performed without a documented set of objectives. It is used to test the errors faced in initial phase of software development.

There were many errors in initial phase. Some of those errors are listed below:

1. ModuleNotFoundError was displayed as the module() was not upgraded to latest version.

1. Syntax Error was displayed as this (:)punctuation was missing in the source code.

### 6.1.2 FORMAL TESTING

Formal testing is performed with a plan, documented set of test cases that outline the methodology and test objectives.

### SUMMARY OF UNIT TESTING

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Number of test cases** | **Success** | **Failure** |
| IPL win probability.ipynb | 6 | 6 | 0 |
| app.ipynb | 6 | 6 | 0 |

### SUMMARY OF INTEGRATION TESTING

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Number**  **of Files** | **Number of test cases** | **Success** | **Failure** |
| 04 | 10 | 10 | 0 |

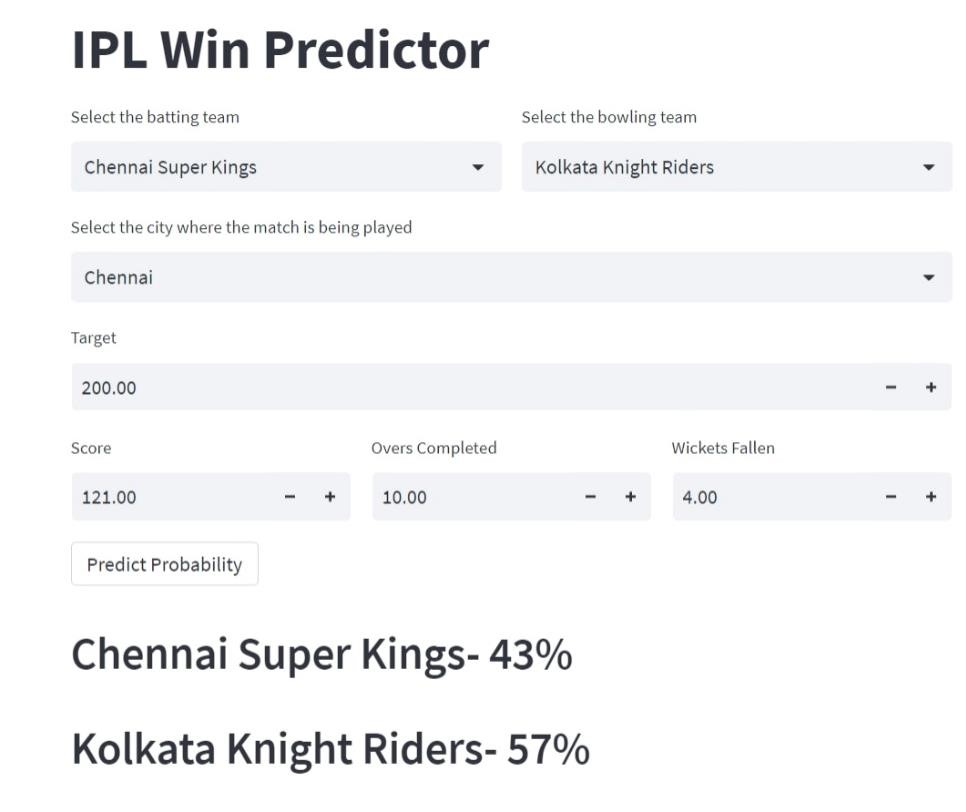
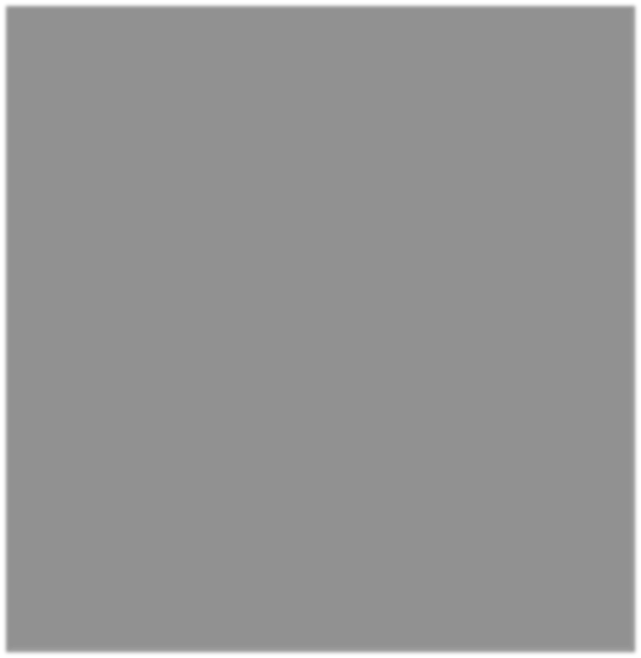
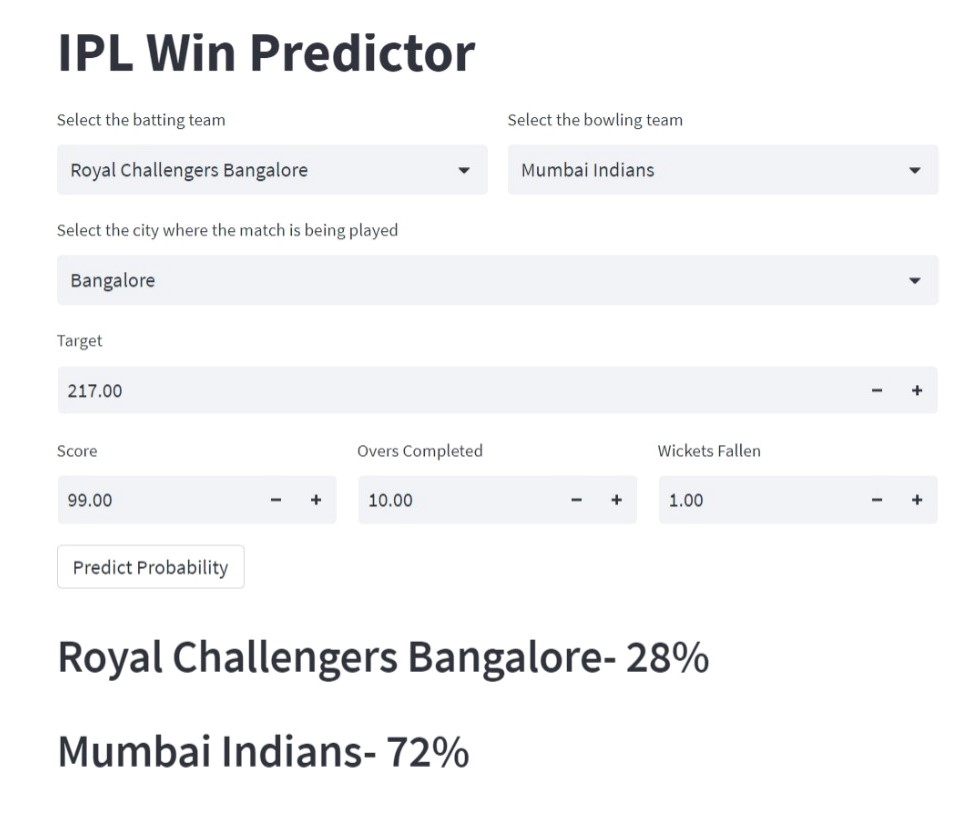
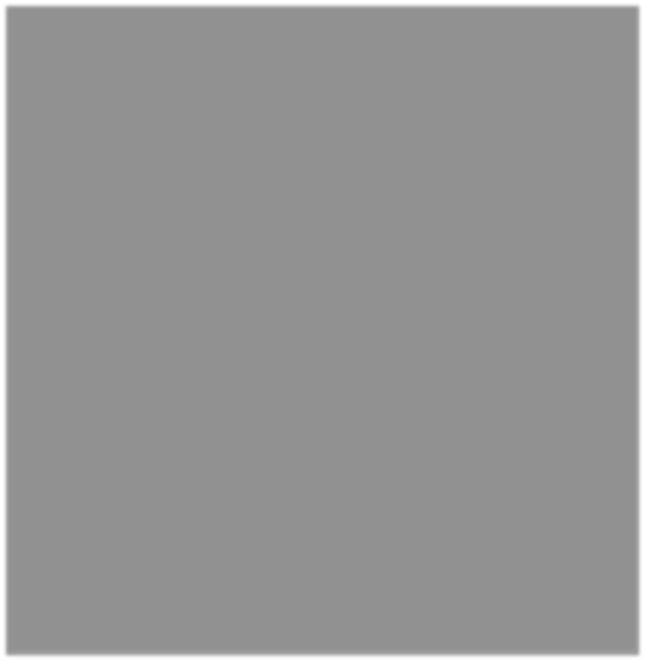
### CONCLUSION OF TESTING

For software testing ‘Manual Testing’ is used. Manual testing is process in which test cases are executed manually without using automated testing tools. Since the testing was done manually testing took more time.

If automated testing would have been used then the time taken for testing would be less. It is the way to increase the effectiveness, test coverage, and execution speed in software testing.

Due to less knowledge about automated testing tools manual testing is used.

**6.2 USER DOCUMENTATION**



## CHAPTER 7 CONCLUSIONS

### 7.1 CONCLUSION

In summary, IPL match win prediction systems have the potential to be a valuable tool for users interested in making informed decisions about IPL matches. However, it is important to approach the predictions with caution and use them as one of many factors to consider when making decisions. By addressing various factors and issues related to data quality, algorithm selection, and security, IPL match win prediction systems can provide a reliable and secure platform for users to engage with IPL matches in new and exciting ways.

#### 7.1.1 SIGNIFICANCE OF THE SYSTEM

1. Improved Decision-Making: The IPL match win prediction system enables team owners, coaches, and captains to make more informed and data-driven decisions. They can use the predictions to determine the team composition, strategy, and tactics that are most likely to lead to victory.

1. Increased Fan Engagement: The IPL match win prediction system has enhanced fan engagement by providing a new and exciting dimension to the game. Fans can use the predictions to place bets, participate in fantasy leagues, and engage in online discussions, increasing their involvement and interest in the sport.

1. Better Sponsorship Opportunities: The IPL match win prediction system has attracted more sponsors to the league as it provides an additional platform for brand promotion and marketing. Sponsors can leverage the predictions to engage with fans and promote their brand in new and innovative ways.

1. Improved TV Ratings: The IPL match win prediction system has contributed to the growth of TV ratings as it provides an additional element of excitement and anticipation for viewers.

### 7.2 LIMITATIONS OF THE SYSTEM

As we have data from 2008-2019, so it’s been 2023 so we don’t have data from 20192023.

 We have been restricted for data of 2019-2023 as it’s been not generated yet.

### 7.3 FUTURE SCOPE OF THE PROJECT

In the future, we can expect IPL match win prediction to become even more accurate and sophisticated, with the use of more advanced algorithms and better data sources.

The use of real-time data, such as player statistics and weather conditions, to adjust predictions on-the-fly and provide more accurate forecasts.

**REFERENCES**

[**https://www.iplt20.com/**](https://www.iplt20.com/)

[**https://en.wikipedia.org/wiki/Indian\_Premier\_League**](https://en.wikipedia.org/wiki/Indian_Premier_League)

[**https://www.kaggle.com/datasets/nowke9/ipldata**](https://www.kaggle.com/datasets/nowke9/ipldata)

[**https://www.frameusers.com/uploads/2016/02/ipl.pdf**](https://www.frameusers.com/uploads/2016/02/ipl.pdf)