IPL WINNING PREDICTION PROJECT

USING MACHINE LEARNING

***Project submitted in partial fulfillment of the requirement for the degree of Master of***

***Computer Application***

**By**

Disha Ghosh

(**Registration No.:** 2306151031)

**Under the Supervision of**

Dr. Manas Ranjan Senapati



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,**

**VEER SURENDRA SAI UNIVERSITY OF TECHNOLOGY,**

**BURLA, SAMBALPUR – 768018**

**[2024-2025]**



# **Certificate of Examination**

This is to certify that the project report entitled ‘‘**IPL WINNING PREDICTOR**” submitted by **Disha Ghosh** (**2306151031**) of Master of Computer Application (MCA), Veer Surendra Sai University of Technology, Odisha has been examined by us. We are satisfied with the quality and correctness of the work.

HOP, PG & Ph.D. HOD,

Department of CSE Department of CSE

VSSUT, Burla VSSUT, Burla



# **Supervisor’s Certificate**

This is to certify that the Project-I work entitled “**IPL WINNING PREDICTON**” is being submitted by **Disha Ghosh**, **2306151031**, to the Department of Computer Science & Engineering, Veer Surendra Sai University of Technology, Burla, in partial fulfillment of the requirement for the degree of Master of Computer Application (MCA) during the academic year 2024-2025. It is an original work carried out by his/her under my supervision.

**Supervisor**

# **Acknowledgment**

I would like to thank our supervisor **Dr. Manas Ranjan Senapati** for invaluable guidance, constant assistance, support, endurance and constructive suggestions for the betterment of the technical seminar.

Secondly, I would like to thank those special persons whose paper works helped us as we went through our work and helped to modify and eliminate some of the irrelevant or un-necessary stuffs.

I also wish to thank all the staff members of the department of Computer Science &Engineering for helping us directly or indirectly in completing this work successfully.

Finally, we are thankful to our friends for their continued moral and material support throughout the course and in helping us finalize the presentation.

Date: Full Signature of the candidate Place: VSSUT, Burla

## **Abstract**

Cricket is a popular sport not only in India but also in the surrounding areas the world. Specifically the T-20 format of this game is very popular in recent years. Today one of the championship named as Indian premier league (IPL) associated with this format has grown rapidly. But cricket is always said it is a game of uncertainty. Predicting the winner of the tournament or the game has an area of concern for many fans. Technology, on the other hand, is developing at an alarming rate. Machine learning algorithms are always the first choice for researchers to predict something after model training. So in this project we will predict the probable percentage of winning of team in the IPL using different supervised learning methods

**Keywords:**

1. IPL (Indian Premier League)
2. Machine Learning
3. Streamlit
4. Prediction Model
5. Win Probability
6. Data Visualization
7. Target Score
8. Current Run Rate (CRR)
9. Required Run Rate (RRR)
10. Feature Engineering

### **Contents**

Title page……………...……………………….….…………………………….…I

Certificate of Examination………….…………………………………….…….…II

Supervisor’s Certificate………………….…………………….…...…….…….... III

Acknowledgements…….…….………………………………………….………. IV

Abstract……………………….…….……………………………………………. V

1. Introduction…………………………………………………………………...1-3

1.1. Problem Statement…………………………………………………....1

1.2. Objective of the Project ………………...………………………….…1

1.3. Scope of the Project………………………………………..……….....3

1. Literature Review ……………………………………………...……………...4-5
2. Features and Functionalities…………………………………………….……..6
3. Machine Learning in Cricket………………………………………………….7-8

4.1. Machine Learning Based Methods…………………………..………..7

1. Approach and Design………………………………………………………...8-17

5.1. Data Collection…………….…………………………………………8

5.2. Data Pre-Processing…………………………………………………10

5.3. Training and Testing Splitting the Dataset…………………………..13

5.4. Algorithm……………………………………………………………14

5.5. Front-End……….…………………………………………………...15

6. Tools and Technologies………………………………………………….….17-18

6.1. Programming Languages………………………………………..…...17

6.2. Libraries and Framework…………………………………………….18

6.3. Development Environment……………………………………….….18

7. Results and Discussion……………………………………………………...19-20

8. Conclusion and Future Work…………………………………………………..21

9.References……………………………………………………………………....22

**1.Introduction**

The motivation for using machine learning in the IPL is to improve player accuracy and performance and create better strategies for the remaining game. This project leverages machine learning models to analyze real-time data inputs and provide match predictions based on historical data and match parameters. We implemented machine learning model by using the algorithm called as Logistic Regression with the help of this algorithm we predicted the probability of winning of both teams which are in the form of percentage and the displayed our result on the website. It focuses on key aspects like current score, overs completed, target score, and wickets, which influence the match's outcome.

* 1. **Problem Statement**

Prediction of the winner of an IPL match is challenging due to the many dynamic factors influencing the game's outcome, such as team performance, current score, wickets lost, and match conditions. This project aims to develop a system that can accurately predict the probability of a team winning based on real-time match data and historical IPL data.

The key challenges include:

1. **Accurate Prediction**: Predicting the batting team’s chances of winning using variables like current score, overs, wickets, and required run rates.
2. **Handling Real-time Data**: Processing live match inputs and updating predictions as the game progresses.
3. **Leveraging Historical Data**: Using past IPL match data to train the model for accurate predictions.
4. **Interactive Interface**: Providing an easy-to-use interface for users to input match details and view predictions.

The goal is to create a **real-time IPL match predictor** that helps users understand match dynamics and predicts the outcome based on the ongoing game.

**1.2 Objective of the project**

The primary objective of the **IPL Win Predictor** project is to develop an interactive web application that leverages machine learning to predict the outcome of ongoing IPL (Indian Premier League) cricket matches. The key goals of this project are:

1. **Match Outcome Prediction**:  
   To predict the probability of the batting team winning or losing based on key match parameters such as current score, target score, wickets fallen, overs completed and the performance of both teams.
2. **Real-time Data Input**:  
   To allow users to input real-time match data (such as score, overs, and wickets) and instantly receive predictions on the likelihood of each team winning the match.
3. **Data Visualization**:  
   To provide a visual representation of the prediction results through metrics and charts (e.g., pie chart), making it easier for users to understand the probabilities of each team’s success.
4. **Integration of Historical Match Data**:  
   To use historical data from previous IPL matches for training the machine learning model, ensuring that the predictions are based on patterns and trends observed in past performances.
5. **Interactive User Experience**:  
   To create an easy-to-use interface where users can select teams, enter match details, and view results with minimal effort. The app provides a dynamic and engaging experience through features like progress bars and real-time updates.
6. **Use of Machine Learning**:  
   To apply machine learning algorithms, such as classification models, to predict the win probabilities by analyzing various factors that influence a match's outcome.
7. **Educational and Analytical Tool**:  
   To serve as an educational tool for cricket enthusiasts, analysts, and team managers, helping them understand the key factors affecting match outcomes and how predictions are made based on real-time data.

**1.3 Scope of the Project**

The scope of the **IPL Win Predictor** project focuses on building a machine learning-based web application that predicts the outcome of IPL matches. The project covers the following key aspects:

1. **Prediction Model**:

The system will predict the probability of the batting team winning or losing based on real-time match data such as score, overs, wickets, and required run rates. The prediction model will be trained on historical IPL match data, leveraging machine learning techniques to generate accurate outcomes.

### **Real-time Data Processing**: Users will be able to input live match data, including the current score, overs completed, wickets lost, and target score, to receive updated predictions. The application will process this data dynamically as the match progresses, allowing users to track the shifting probabilities.

### **User Interface**: A user-friendly interface built using **Streamlit** will allow users to interact with the application. Users can select teams, enter match details, and view the predicted probabilities, along with visualizations such as pie charts showing the win/loss chances.

### **Visualization of Predictions**: The project includes visual representations of match outcomes, such as pie charts and real-time progress bars, to make the predictions more accessible and engaging for users.

### **Historical Data Integration**: The project will incorporate historical IPL match data to train the model, ensuring that predictions are based on patterns and trends observed in past matches.

### **Limited to IPL Matches**: The current scope is specifically focused on the IPL (Indian Premier League) and may not extend to other cricket formats or leagues.

### **Accuracy Improvement**: The project will provide a baseline model, but future enhancements can improve prediction accuracy by integrating more features like weather conditions, player form, and other match-specific variables.

### **Educational Tool**: The project can be used as an educational resource to demonstrate the application of machine learning in sports analytics, providing insights into match dynamics and prediction models.

### **2.Literature Review**

Cricket match outcome prediction has gained immense traction due to the popularity of cricket globally, particularly in formats like One-Day Internationals (ODIs), Test matches, and T20 leagues like the Indian Premier League (IPL). Researchers have utilized advanced machine learning (ML) techniques and data analytics to predict match outcomes with varying levels of precision. This review discusses key contributions in this area, highlighting methodologies, findings, and challenges.

Naik et al. [1] provided an in-depth analysis of ODI cricket matches by preprocessing and normalizing data to improve model accuracy. Their study emphasized the importance of factors like match conditions, toss decisions, and player performance in outcome prediction. Similarly, Singhvi et al. [2] addressed the dynamic nature of T20 cricket, focusing on match conditions and real-time data analysis. Their study demonstrated how specific features could improve model accuracy in a fast-paced format like T20. Nimmagadda et al. [3] extended this approach by employing data mining techniques for cricket score and outcome prediction, showing how precise data preprocessing can significantly impact prediction results.

The IPL, being one of the most competitive T20 leagues, has been a key focus for researchers. Studies like Gupta and Rathore [27] demonstrated that selecting specific features such as toss decision, venue, and team strength leads to improved classification models. Jain and Gupta [28] conducted a comprehensive analysis of IPL team strengths using data mining techniques, offering insights into player performances and match strategies. Authors in [4] utilized datasets from IPL seasons to train Random Forests and Multiple Linear Regression models, highlighting the importance of combining ensemble methods for better accuracy.

Several studies have explored hybrid and ensemble approaches to improve predictive accuracy. Naik et al. [9] proposed a framework combining clustering and classification techniques, achieving enhanced prediction accuracy. Agarwal and Pandey [29] explored neural networks to uncover complex feature relationships, while Khandelwal and Singh [30] demonstrated the superiority of Gradient Boosting techniques over traditional ML methods. These approaches emphasized the need for integrating multiple models to capture diverse patterns in cricket data.

A common theme in the literature is the impact of player-specific and team dynamics. Pandey and Bhattacharjee [6] explored the significance of player statistics, toss decisions, and weather conditions, revealing their critical roles in determining match outcomes. Studies such as [7] delved into factors like player form, team morale, and home advantage, providing insights into how psychological and environmental aspects influence matches. Researchers in [13] introduced novel variables, such as crowd influence and stadium conditions, to enhance model realism.

Machine learning frameworks have played a pivotal role in advancing cricket analytics. Shah et al. [14] employed convolutional neural networks (CNNs) to analyze match footage, while researchers in [18] used recurrent neural networks (RNNs) to capture temporal patterns in cricket matches. These frameworks allowed for effective handling of sequential and real-time data, making predictions more robust. The use of XGBoost and LightGBM in studies like [11] and [12] further demonstrated the importance of leveraging advanced algorithms for feature selection and model training.

External factors such as weather, pitch conditions, and dew have also been key considerations. Studies like [15] and [16] showed that integrating these variables significantly improved predictive accuracy, as they directly affect gameplay dynamics. Toss decisions, which determine the order of batting or bowling, have been highlighted as critical in several studies, including [9] and [13], as they influence team strategies and match outcomes.

Real-time prediction systems have emerged as a significant area of interest. Research in [19], [20], and [21] highlighted the potential of integrating live updates into prediction models. These studies demonstrated how real-time data processing could benefit broadcasters, analysts, and fans by providing immediate insights into the likely outcome of ongoing matches.

AI-driven methods have further revolutionized cricket outcome prediction. Agarwal and Pandey [29] showed the potential of deep learning models in analyzing intricate patterns within historical data. Shah et al. [14] extended this approach by incorporating spatial data using CNNs, while researchers in [18] focused on RNN-based models to analyze temporal dependencies in cricket matches. These advancements have paved the way for real-time predictive systems with practical applications.

Despite these advancements, limitations persist in existing models. Researchers in [10] pointed out constraints such as limited datasets and the use of traditional algorithms, advocating for more scalable solutions. Studies like [5] and [8] explored alternative metrics, including net run rate and team consistency, to refine their models further. Additionally, the importance of robust feature selection was emphasized in [9], where including only relevant variables significantly enhanced prediction efficiency.

Several works highlighted the challenges associated with predicting outcomes in formats like T20 cricket. The unpredictable nature of this format, combined with its fast pace, makes accurate prediction challenging. To address these challenges, researchers have used ensemble techniques and hybrid models, as seen in [6] and [30]. These approaches integrate multiple algorithms to handle the variability in T20 matches effectively.

In conclusion, cricket outcome prediction has seen remarkable progress, thanks to advancements in ML, AI, and data analytics. The IPL has been a prime focus area, with studies utilizing historical and real-time data to develop accurate predictive models. While challenges like limited data and unpredictable formats remain, the integration of advanced algorithms and external factors has significantly enhanced model accuracy and reliability. These studies provide insights into the evolving field of sports analytics, offering a strong foundation for future research and practical applications.

**3. Features and Functionality**

* Model Loading: The pre-trained IPL\_model.pkl is loaded into the application using Pickle. If the file is missing, the user is notified with an error message. Similarly, the final\_dataset.csv is loaded for reference, containing historical match data.
* **User Inputs:** The app allows users to input the following match parameters:
  + **Batting Team**: The team currently batting.
  + **Bowling Team**: The team currently bowling (opposing team).
  + **City**: The city where the match is being played.
  + **Target**: The target score set by the batting team.
  + **Score**: The current score of the batting team.
  + **Over and Ball Completed**: Number of overs and balls bowled so far.
  + **Wickets Gone**: Number of wickets the batting team has lost.
* **Prediction Mechanism:** Once the user has filled in the necessary datails, the app calculates various match metrices:
* **Runs Left**: The difference between the target and the current score.
* **Balls Left**: The remaining balls in the match.
* **Wickets Left**: The remaining wickets the batting team can lose.
* **Current Run Rate (CRR)**: The run rate is calculated from the current score and overs.
* **Required Run Rate (RRR)**: The required run rate is based on the target and balls remaining.

These metrics are used to generate an input dataset, which is then fed into the machine learning model to predict the probabilities of winning for both teams.

* **Prediction Results:** The output consists of:
  + **Win Probability**: Probability of the batting team winning the match.
  + **Loss Probability**: Probability of the bowling team winning the match.

Both probabilities are displayed as **metrics** in Streamlit, and the prediction is visualized in a **pie chart**, showing the proportion of each team's winning chances.

* **Match Summary:** The app also shows a table with key match statistics like runs left, balls left, and both CRR and RRR to summarize the current match state.
* **Progress Bar:** To provide visual feedback, the app includes a progress bar that simulates the likelihood of winning for the batting team, creating a dynamic and interactive experience.
* **Reset Functionality:** A reset button is provided to clear all user inputs and start a new prediction. This feature ensures a smooth experience when predicting multiple matches.

**4. Machine Learning in Cricket**

The motivation for using machine learning in the IPL is to improve player accuracy and performance and create better strategies for the remaining game. We implemented machine learning model by using the algorithm called as Logistic Regression with the help of this algorithm we predicted the probability of winning of both teams which are in the form of percentage and the displayed our result on the website deployed with the help of Heroku, in website it takes input of second inning like target, teams playing, runs scored, etc.

**4.1 Machine Learning-Based Methods**

* **Logistic Regression / Classification Models**:

Logistic Regression is a common method for classification for binary tasks, where the goal is to predict one of two outcomes—in this case, whether the batting team will win or lose. Logistic regression is used here to model the relationship between match features (e.g., runs left, wickets lost, current run rate) and the outcome of the match.

The model outputs probabilities (ranging from 0 to 1) that represent the likelihood of the batting team winning, allowing the system to predict the **probability of win** for both teams.

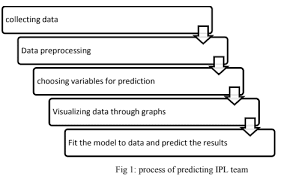
* **Random Forest**:

Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the mode of the classes (in this case, win/loss predictions) for the individual trees. Random forest provides better accuracy compared to individual decision trees.

Handling missing data, outliers, and overfitting issues can be helped by it.. (in this case, win/loss predictions)

**5. Approach and Design**

The figure below explains the approach we have taken into building the prediction model using machine learning algorithms.



. Fig 5.1: Training and testing of data

**5.1. Data Collection**

Data collection involves gathering and measuring information from many different sources. Practical machine learning solutions are developed from the data that we collect. We can use data analysis to find recurring patterns by capturing a record of past events. You use machine learning to look for trends and predict future changes.

1. First of all we want the data which we are going to pre-process and using that we
2. will train our model. So we collected the IPL data from 2008-2019 from the website
3. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
4. They include information like city, date, team1, team2, toss win, toss decision,
5. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
6. bowler, and so on. This data is saved in a csv format in an excel file
7. First of all we want the data which we are going to pre-process and using that we
8. will train our model. So we collected the IPL data from 2008-2019 from the website
9. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
10. They include information like city, date, team1, team2, toss win, toss decision,
11. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
12. bowler, and so on. This data is saved in a csv format in an excel file
13. First of all we want the data which we are going to pre-process and using that we
14. will train our model. So we collected the IPL data from 2008-2019 from the website
15. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
16. They include information like city, date, team1, team2, toss win, toss decision,
17. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
18. bowler, and so on. This data is saved in a csv format in an excel file
19. First of all we want the data which we are going to pre-process and using that we
20. will train our model. So we collected the IPL data from 2008-2019 from the website
21. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
22. They include information like city, date, team1, team2, toss win, toss decision,
23. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
24. bowler, and so on. This data is saved in a csv format in an excel file
25. First of all we want the data which we are going to pre-process and using that we
26. will train our model. So we collected the IPL data from 2008-2019 from the website
27. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
28. They include information like city, date, team1, team2, toss win, toss decision,
29. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
30. bowler, and so on. This data is saved in a csv format in an excel file
31. First of all we want the data which we are going to pre-process and using that we
32. will train our model. So we collected the IPL data from 2008-2019 from the website
33. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
34. They include information like city, date, team1, team2, toss win, toss decision,
35. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
36. bowler, and so on. This data is saved in a csv format in an excel file.

First of all we want the data which we are going to pre-process and using that we will train our model. So we collected the IPL data from 2008-2022 from the website named Kaggle. There are two files in this data set: IPL\_Ball\_by\_Ball\_2008\_2022 and IPL\_Matches\_2008\_2022. They include information like city, date, team1, team2, toss win, toss decision, winner, umpire1, umpire2, man of the match, batting team, over, balls, batter, bowler, and so on. This data is saved in a csv format in an excel file.

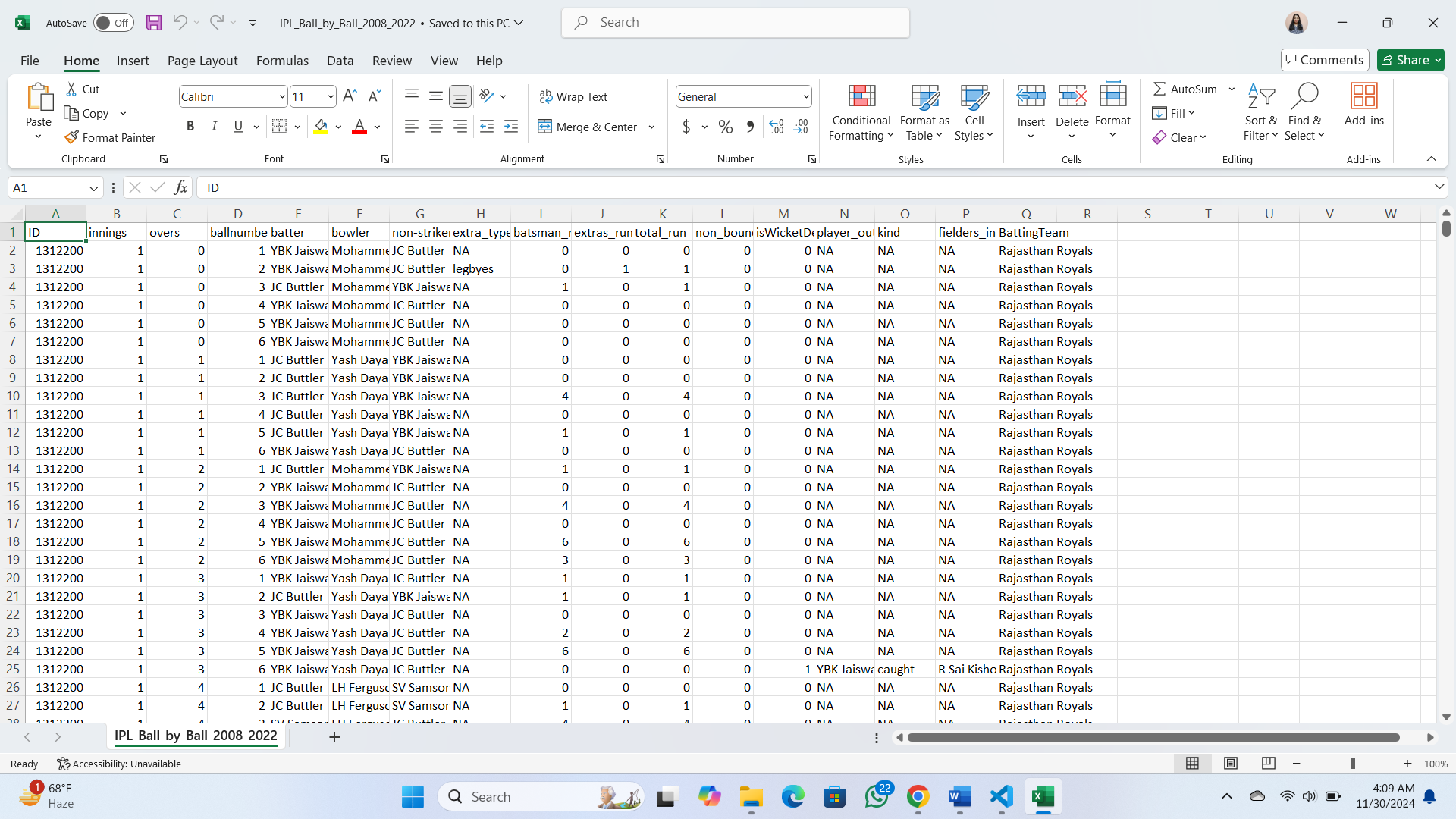


Fig 5.2: Delivery file

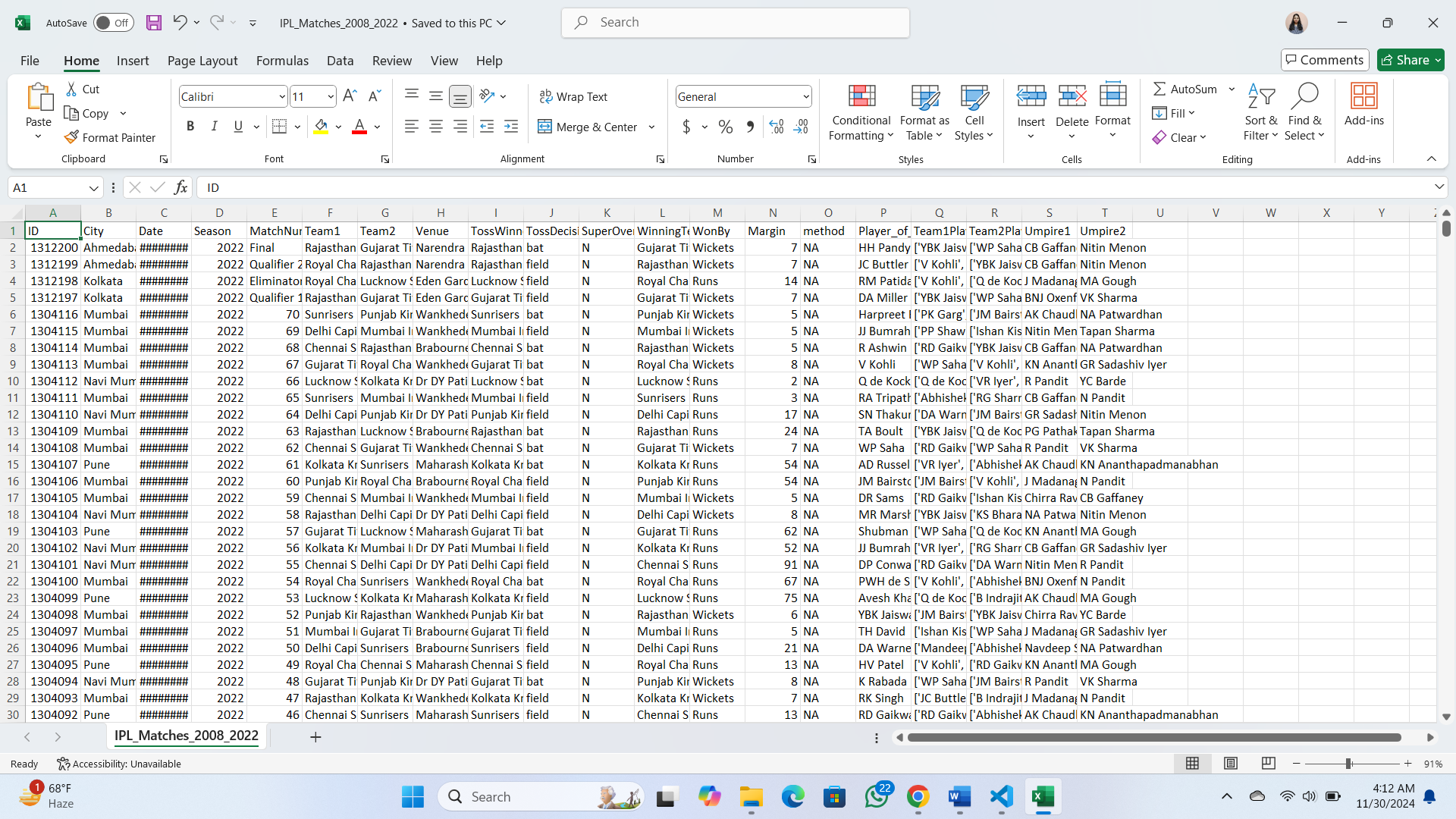


Fig 5.3: Match file

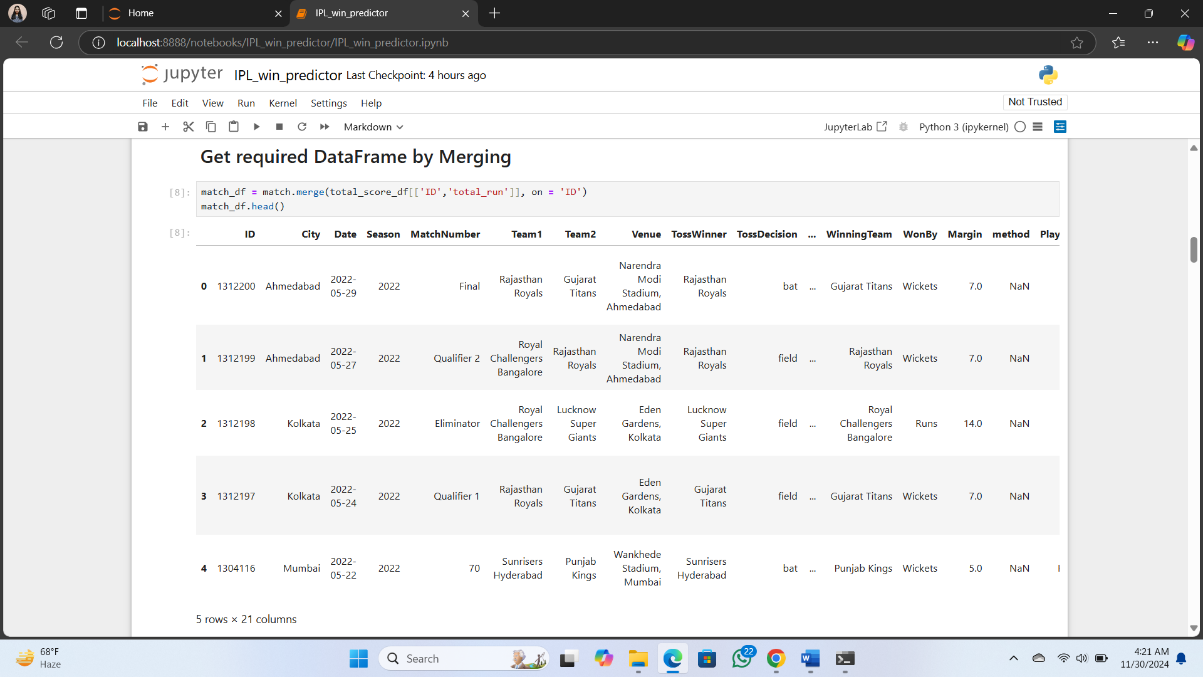
1. First of all we want the data which we are going to pre-process and using that we
2. will train our model. So we collected the IPL data from 2008-2019 from the website
3. named Kaggle. There are two files in this data set: deliveries.csv and matches.csv.
4. They include information like city, date, team1, team2, toss win, toss decision,
5. winner, umpire1, umpire2, man of the match, batting team, over, balls, batter,
6. bowler, and so on. This data is saved in a csv format in an excel file

**5.2. Data pre-processing**

No quality data, no quality results. This is the basic line behind the success of any algorithm. Because before using any algorithm we have to process the data in advance if we expect the best results from it. There are many potential problems with data such as incorrect values, inconsistencies, imperfections etc. There are many pre-processing steps to deal with these issues such as data clearing, data consolidation, data modification, data reduction, data segmentation etc. The most crucial libraries for data analysis are Numpy and Pandas. Other attributes that can be found using simple python programmers include runs left, balls left, wickets left, total runs, current run rate, and required run rate. Following that, there are some other changes that need to be made, such as in the initial years, there were different teams playing, so we need to remove those matches, and some other teams that are playing in IPL now days had different names in old IPL matches, so we need to rename them. There are to major processes involved in pre-processing which are:

**5.2.1 Data cleaning:** In various categories there are some invalid properties in the database. Merging is not possible exactly because of the proximity to these erroneous properties. Along these lines, we've attempted to replace invalid attributes with subtle attributes in a variety of categories.

**5.2.2 Choosing Right Attributes:** This progression is critical because it allows us to eliminate data categories that are irrelevant to the data pre-processing. This is estimated using feature importance.



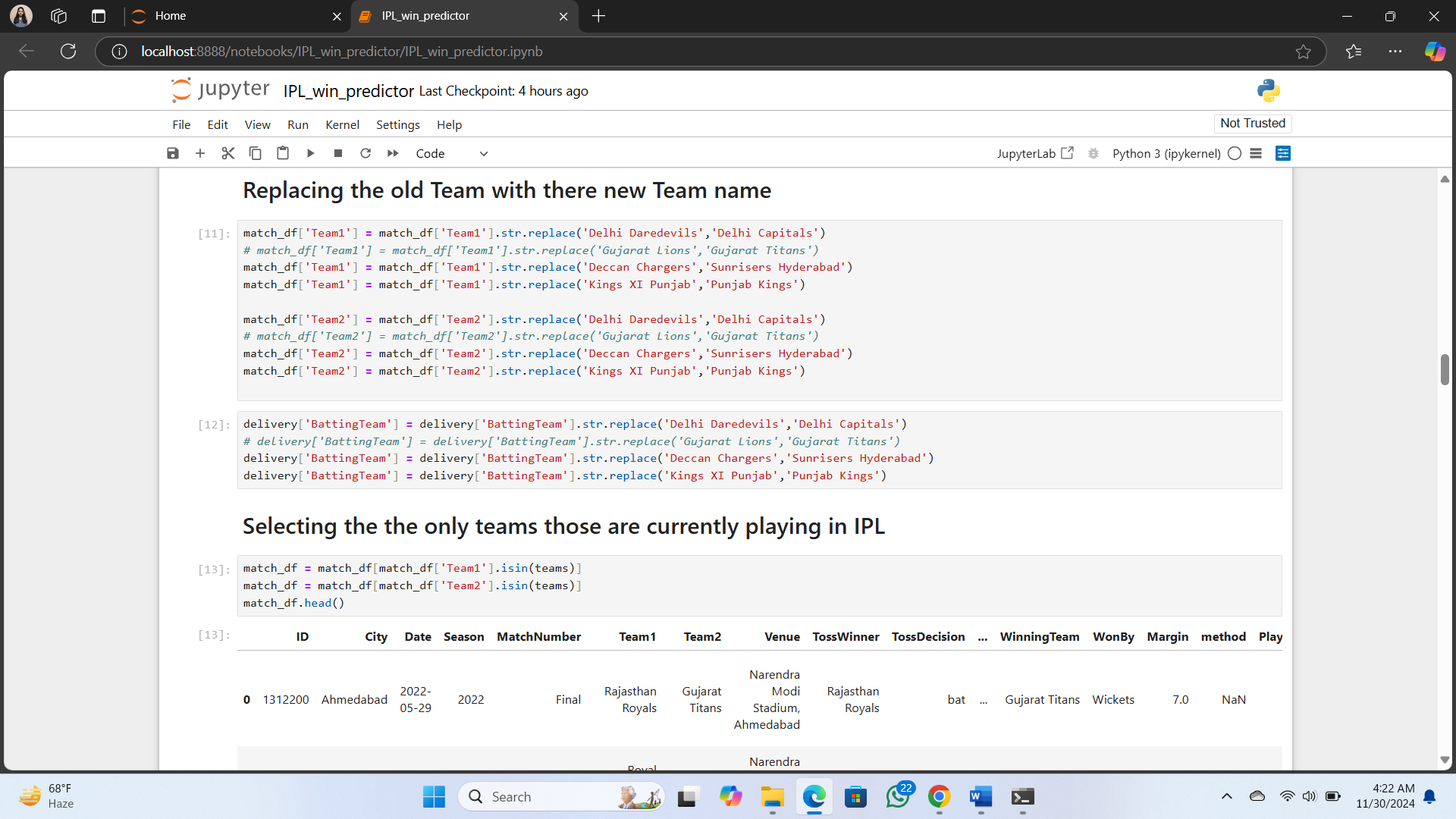
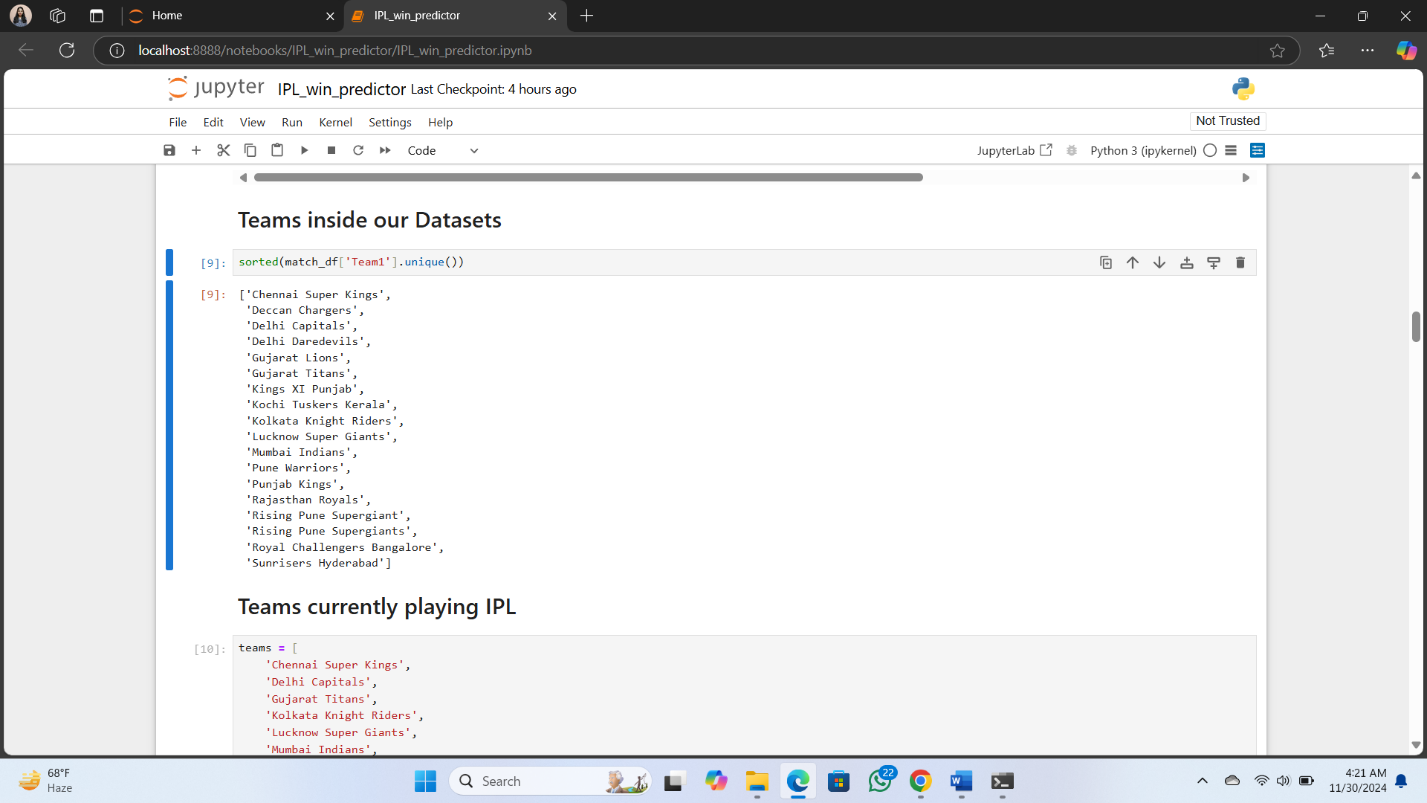
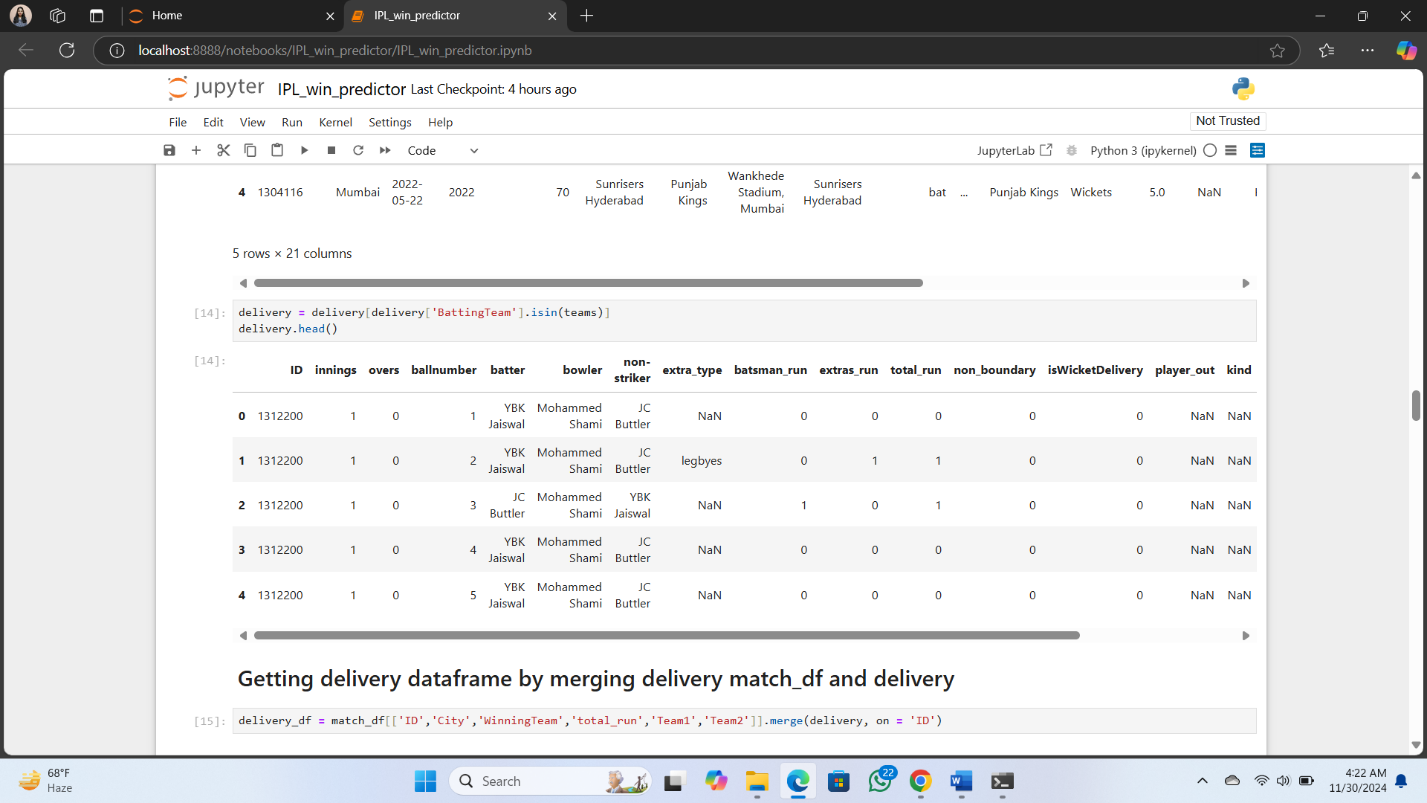


Fig 5.4: Choosing only required attributes



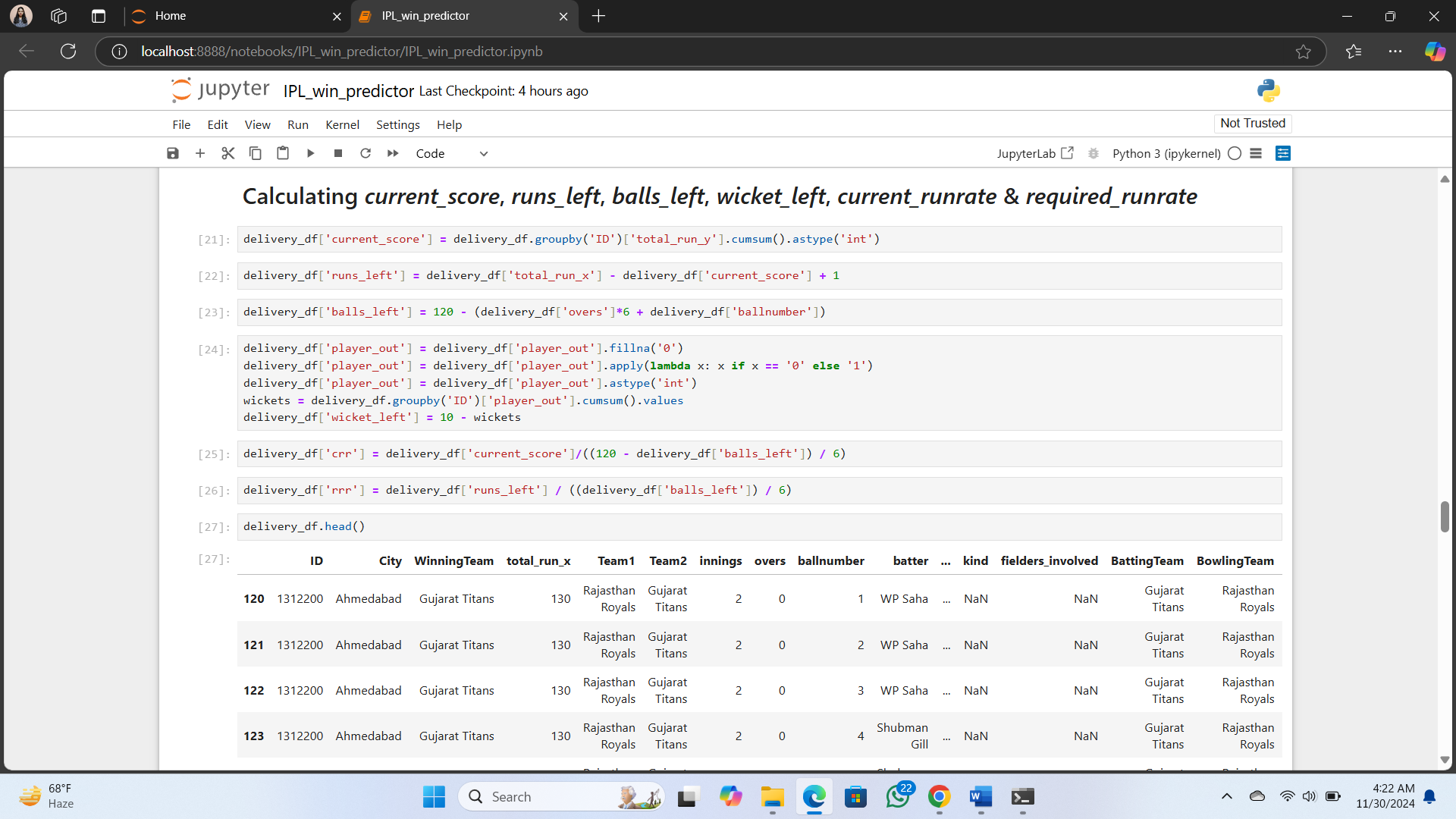


Fig 5.5: Data preprocessing code

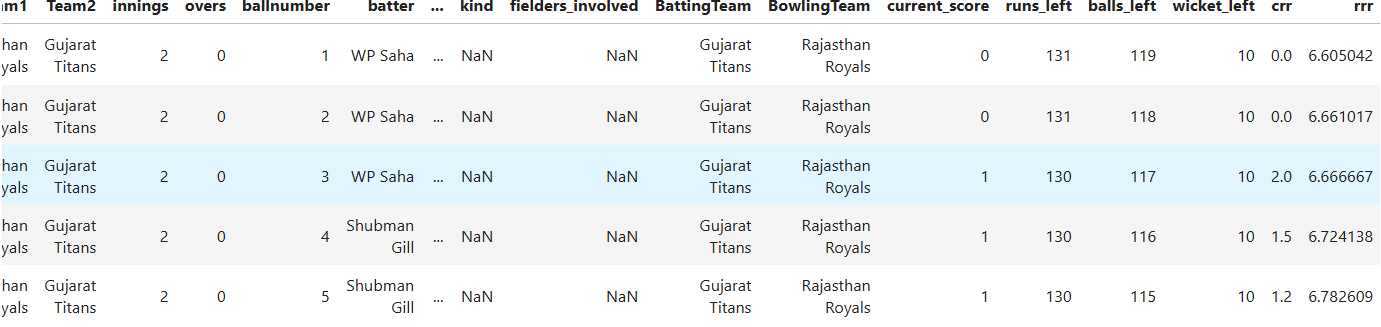
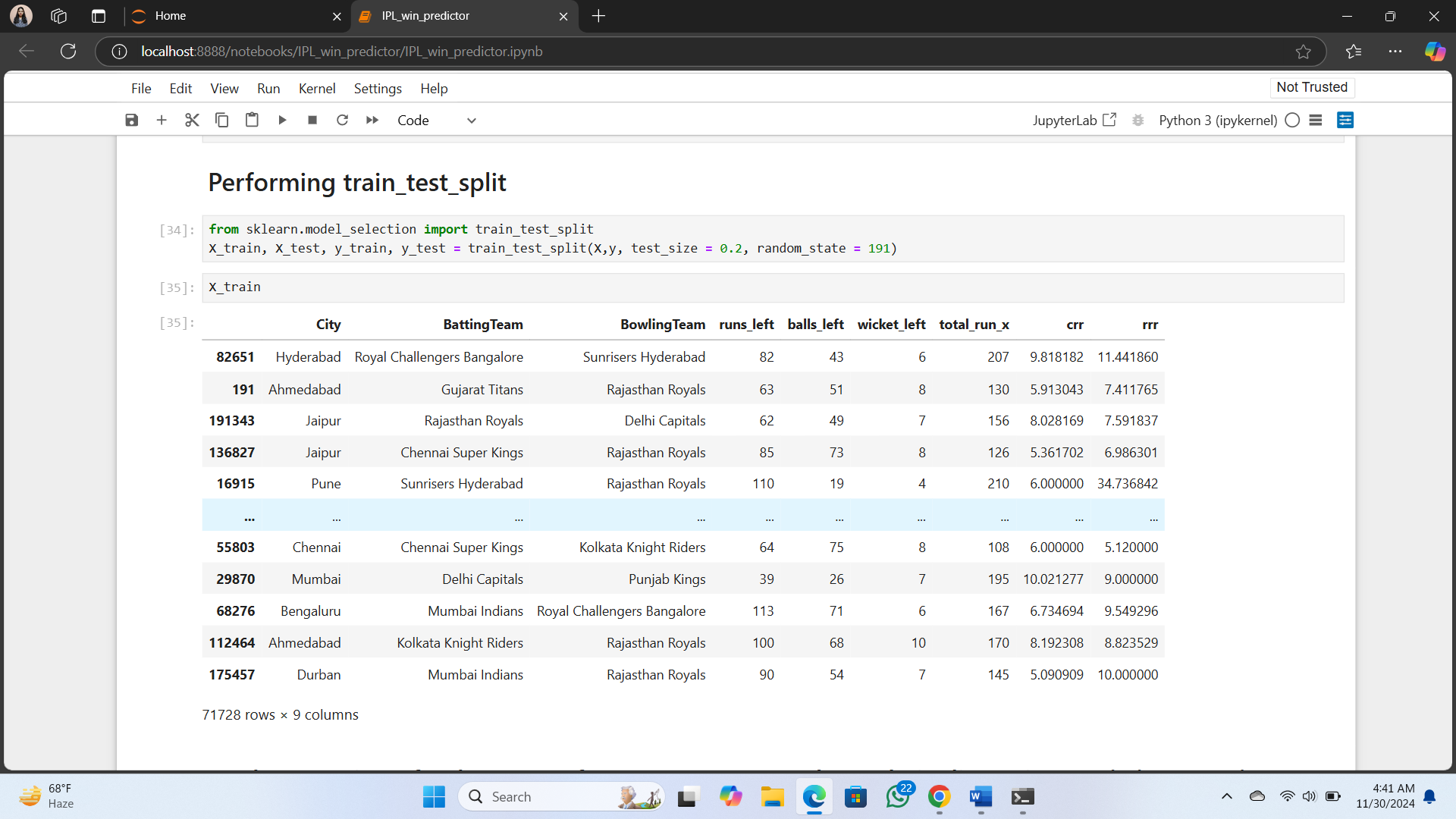


Fig 5.6: Pre-processed data

**5.3 Training and Testing Splitting the data set:**

Testing data: The machine learning algorithm is trained using this data. The data scientist gives the algorithm with input data that matches to an expected output. The model evaluates the data repeatedly in order to gain a better understanding of its behaviour and then modifies itself to accomplish the goal. Training data: A huge dataset used to train a machine learning model is referred to as training data. Prediction models that apply machine learning algorithms are taught how to extract features that are important to certain business goals using training data.



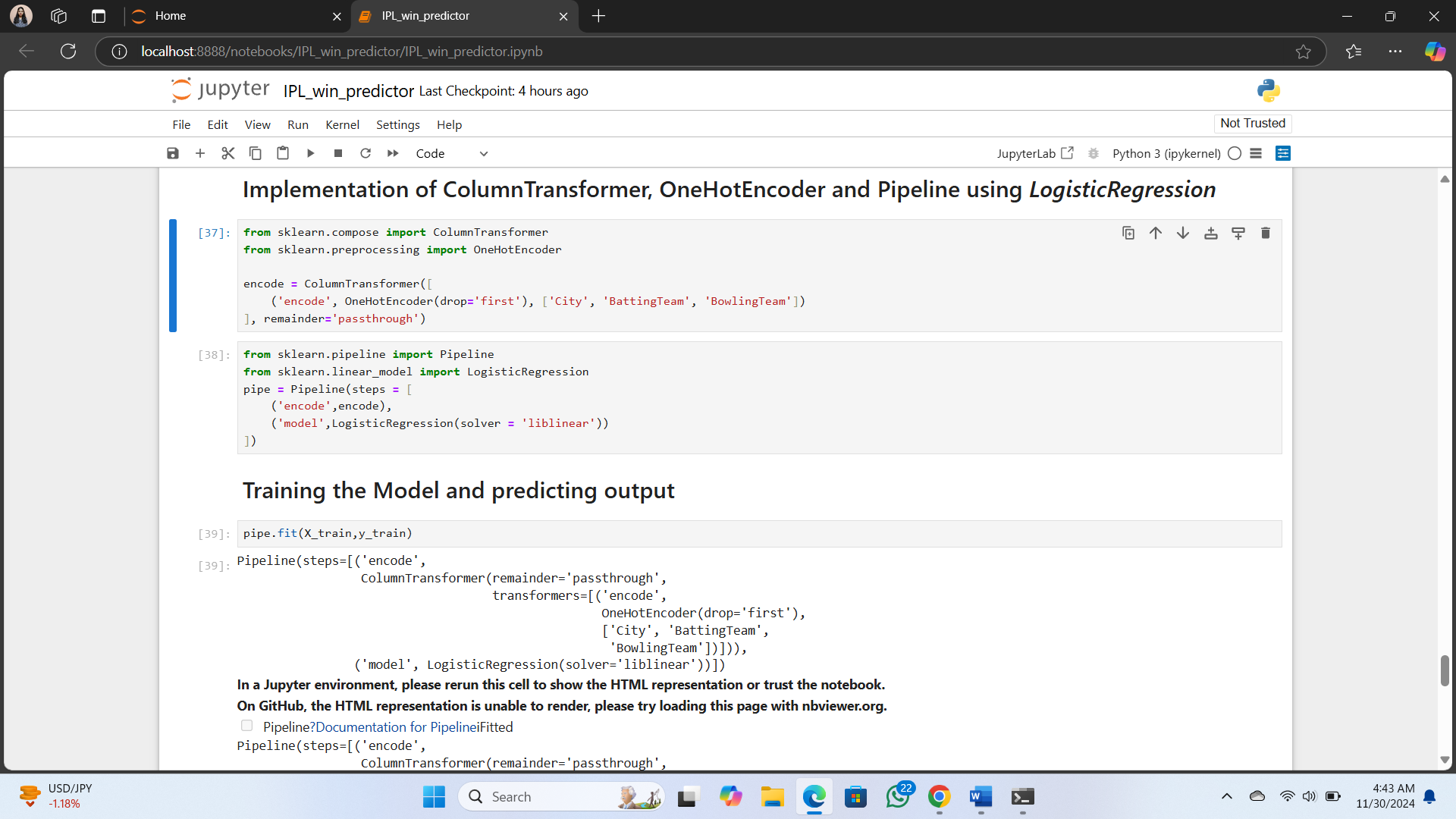
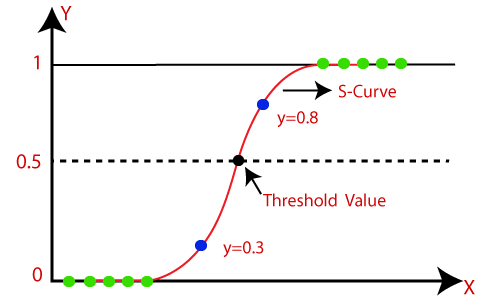
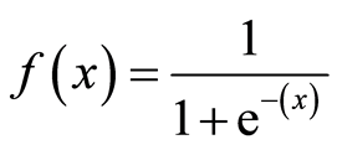


Fig 5.7: Training and testing of data

**5.4 Algorithm:**

From sklearn.linear\_model we will import Logistic regression. So the Logistic regression is the algorithm which we are going to use in this project. Logistic.



A Logistic Regression model is similar to a Linear Regression model, except that the Logistic Regression model has a more complex cost function. The Sigmoid function can be used to convert expected values to probabilities. Any real number can be converted into a number between 0 and 1 with this function. sigmoid is used to convert predictions to probabilities.

The best accuracy of this Logistic regression approach is 80%, which is fairly impressive.

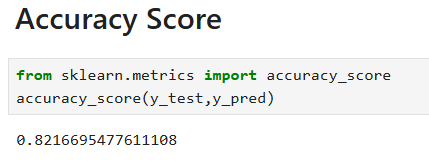
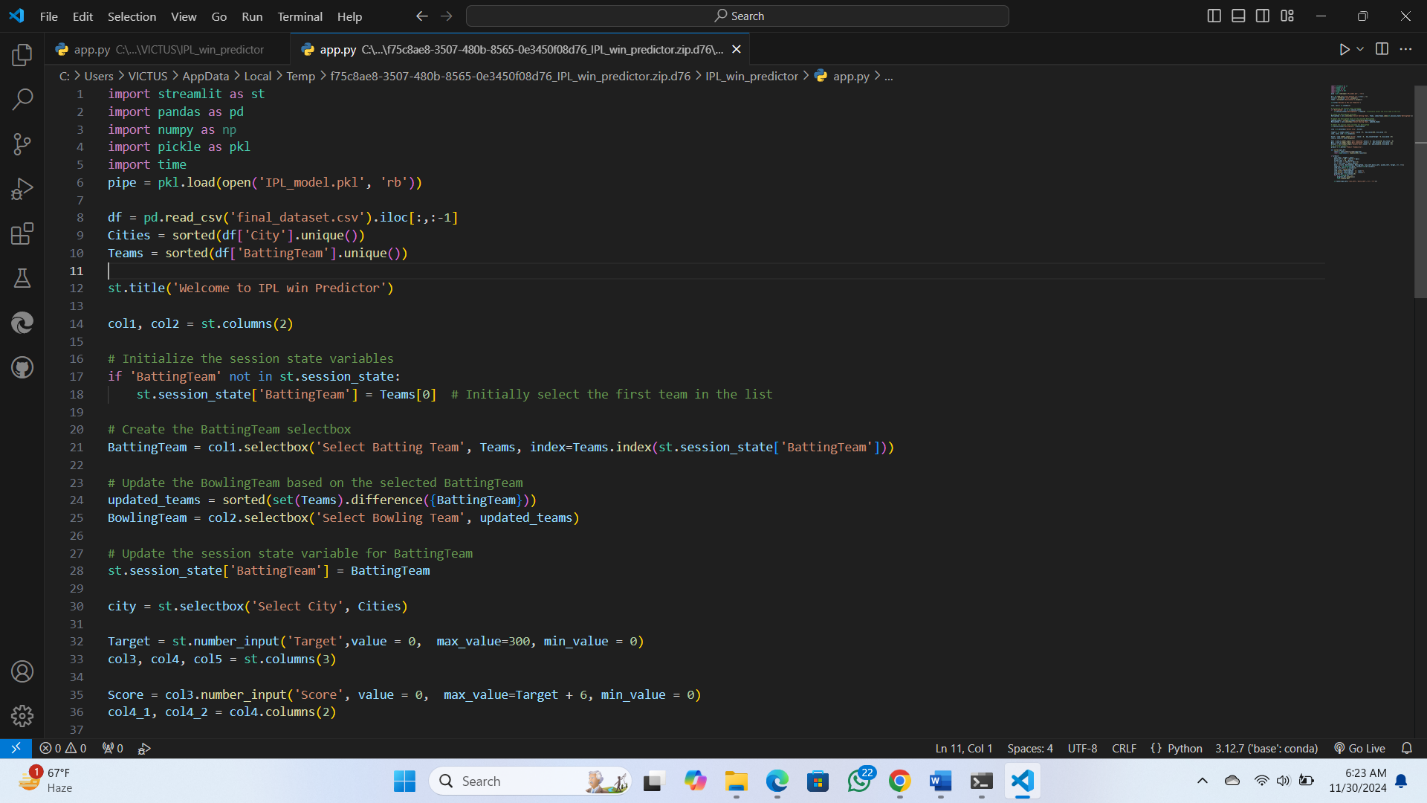


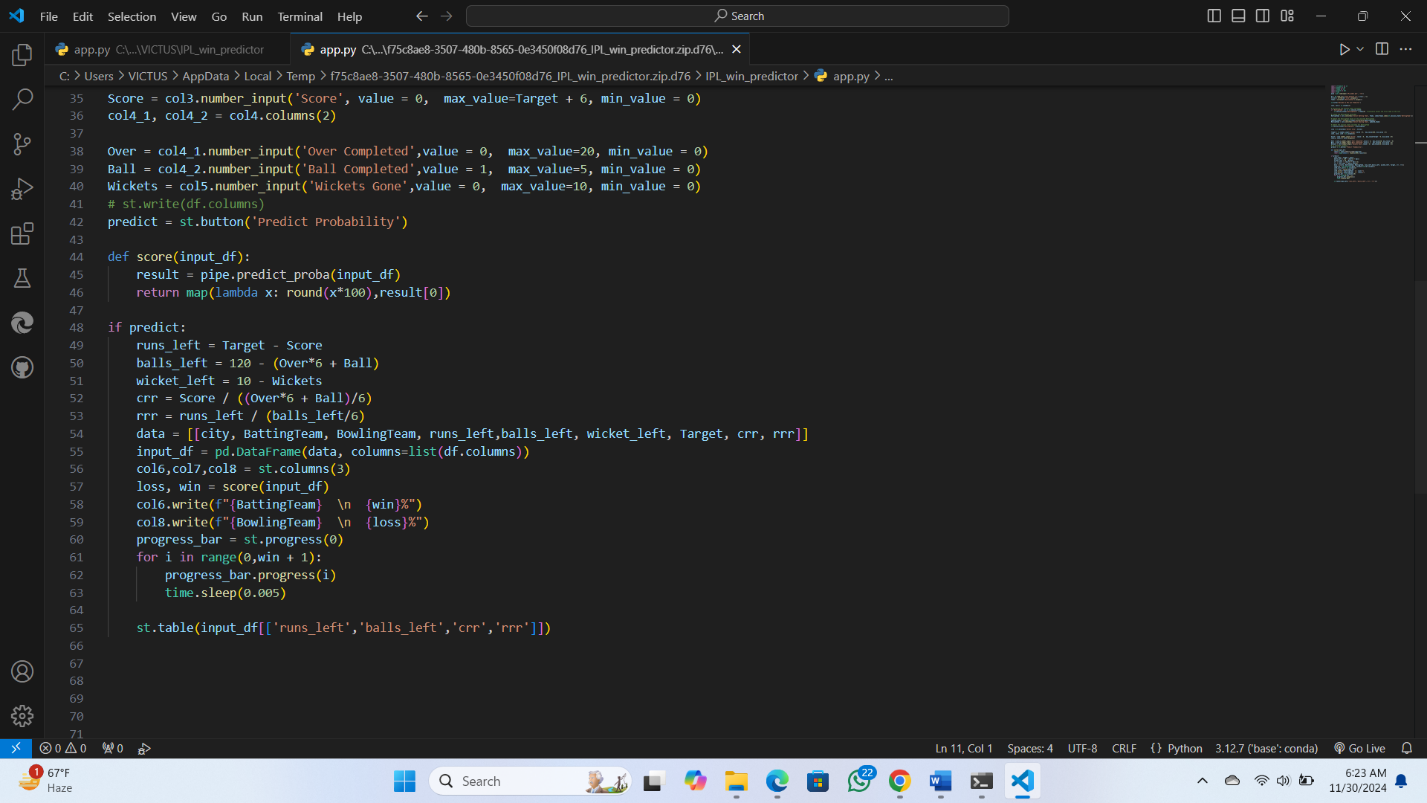
Fig 5.8: Accuracy percentage

Now after training our model using we will try our model over a match let’s say match number “1” and make a graph of whole match scenario of the complete second inning in each and every ball showing the win percentage and losing percentage per balls. And one last part of the project before going into the web development is that we will make a pipe object of our program. A pipe is a mechanism for passing data from one process to another.

**5.5 Front End:**

The IPL Win Predictor is an interactive web application we developed using Streamlit, designed to predict the probability of the batting team winning an IPL match based on real-time match conditions. The application utilizes a pre-trained machine learning model, which takes user inputs such as the batting and bowling teams, the city of the match, target score, current score, overs completed, balls bowled, and wickets lost by the batting team. These inputs are used to compute key game metrics such as runs remaining, balls left to bowl, current run rate (CRR), and required run rate (RRR), which serve as features for the model. The model then calculates the win probabilities for both the batting and bowling teams, which are displayed in percentage format. The results are accompanied by visualizations like a progress bar representing the batting team's win probability and a pie chart illustrating the probability distribution between the two teams. In addition to the predictive functionality, the application includes error handling for invalid inputs and missing files, ensuring robustness. The design emphasizes a user-friendly interface with clear guidance for input, along with the ability to reset the form for new predictions. This project not only highlights the application of machine learning in sports analytics but also demonstrates how real-time predictions and data visualizations can enhance the decision-making process for cricket analysts and enthusiasts. The IPL Win Predictor is a comprehensive tool for engaging with cricket matches in a more analytical way, offering valuable insights into match dynamics and outcomes.





After the completion of web development our website will look like this shown below:

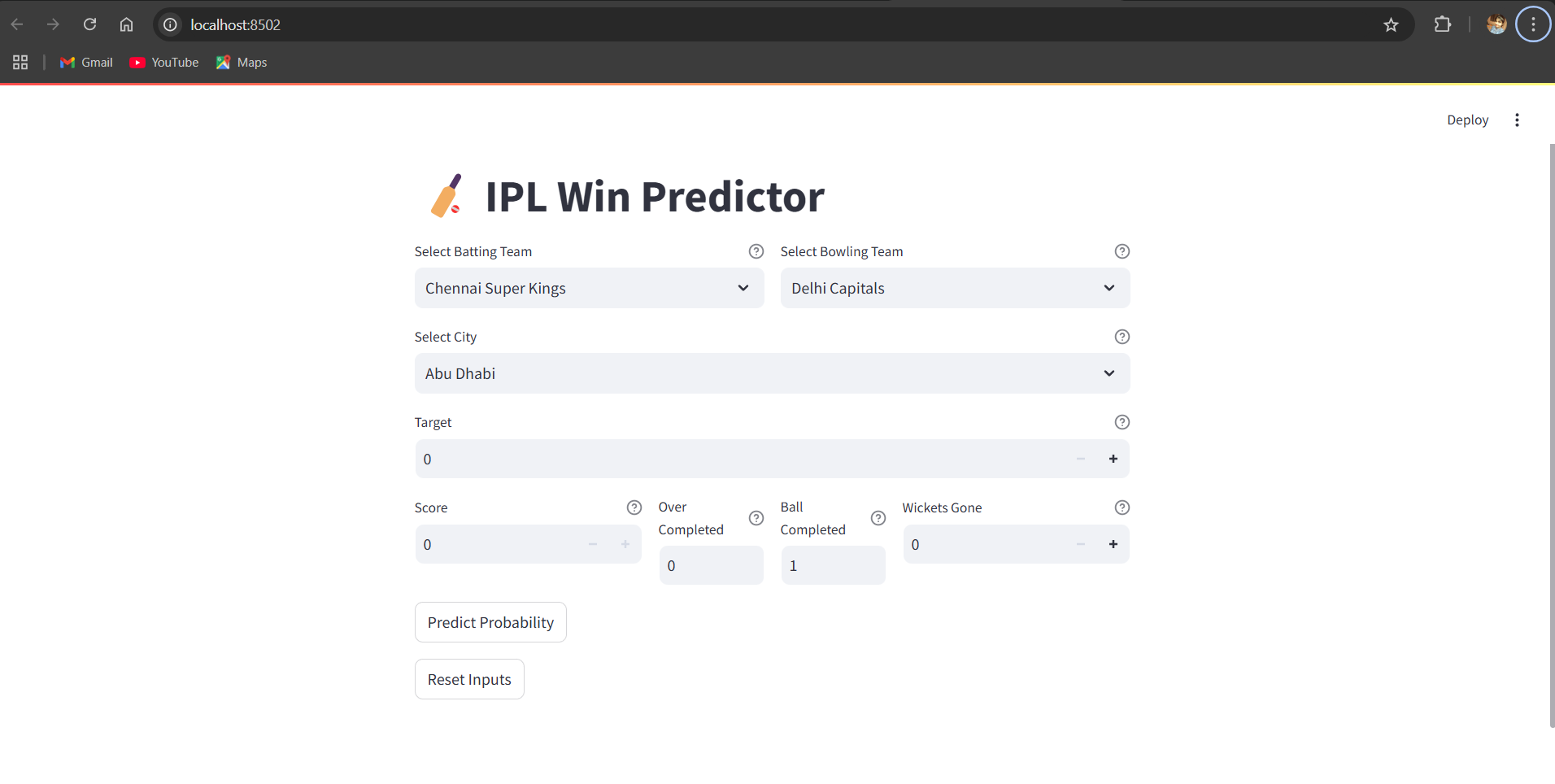


Fig 5.9: User Interface

**6.Tools and Technologies**

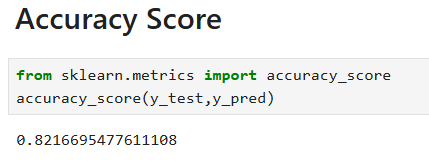
The following tools and technologies were used to implement the IPL winning prediction system:

* **Programming Languages**:
  + **Python**: The primary programming language used due to its rich ecosystem of libraries for data analysis and machine learning.
* **Libraries and Frameworks**:
  + **Streamlit**: Streamlit is an open-source library of Python used for building interactive web applications. It allows developers to easily create data applications minimal code. In this project, Streamlit is used to create the user interface, enabling users to input match details, view predictions, and visualize the results in real-time
  + **Pandas**: Pandas is a data manipulation and analysis library for Python. It provides data structures like Data Frames, which make it easy to handle and analyze structured data. In this project, Pandas is used to load, preprocess, and manage the IPL dataset, as well as to handle the input and output data for predictions.
  + **NumPy:** NumPy is a fundamental package used for computing in Python, providing support for large, multi-dimensional l arrays and matrices. It also offers a collection of mathematical functions to operate on the arrays. In the project, NumPy is used for numerical operations, such as calculating the required run rate (rrr), current run rate (crr), and other match-related calculations.
  + **Scikit-learn:** Scikit-learn is a popular machine learning library in Python, providing tools for building and evaluating machine learning models. It includes algorithms for classification, regression, clustering, and more. In this project, Scikit-learn is used to implement and evaluate the machine learning model (such as Logistic Regression or Random Forest) that predicts the match outcome based on the given features.
  + **Pickle:** Pickle is a Python library used for serializing and deserializing objects. It is commonly used to save and load machine learning models. In this project, Pickle is used to load the pre-trained IPL prediction model fr om a .pkl file, allowing the application to use it for making predictions without retraining the model each time**.**
  + **Matplotlib:** Matplotlib is a library for plotting in Python, used to create static, animationed, and interactive visualizations. In this project, Matplotlib is used to generate pie charts that visualize the probability distribution of winning for the batting and bowling teams, helping users easily understand the prediction results.
  + **Time**: The Time module in Python provides functions for working with time-related tasks. In this project, it is used to create delays in the progress bar, giving users a dynamic, real-time experience as they see the probability of winning change during the prediction process.
* **Development Environment**:
  + **Jupyter Notebook**: The IPL Win Predictor project was developed in a **Jupyter Notebook** environment, which provides an interactive interface for coding and running Python scripts.. In this project, Jupyter Notebook was used to:
    - **Prototype the Application**: The notebook allowed for quick testing and iteration of the code, including data loading, preprocessing, model training, and prediction.
    - **Data Analysis**: The dataset was analyzed and cleaned using Pandas and NumPy, with the results displayed directly in the notebook.
    - **Model Evaluation**: The machine learning model's performance was evaluated within the notebook, and various metrics were computed to understand its effectiveness before deploying it.
    - **Visualization**: The project made use of Matplotlib to generate visualizations of the match predictions, which were also displayed directly in the notebook.

**7.Results and Discussion**

The experiments are performed on a laptop powered by AMD Ryzen 5 5600H with Raedon as a graphics card for GPU support. As far as the software requirement is concerned, Python framework’ and ‘streamlit’ libraries are used. Program is written in Jupyter notebook environment.

The present section deals with the simulation results and discussion. In this work we estimate the chances of winning IPL using ML algorithms. The ML algorithm used in this study is Logistic Regression, which has an accuracy of 82%. Fig.5.8 shows screen shot of the obtained results from the simulation software.

****

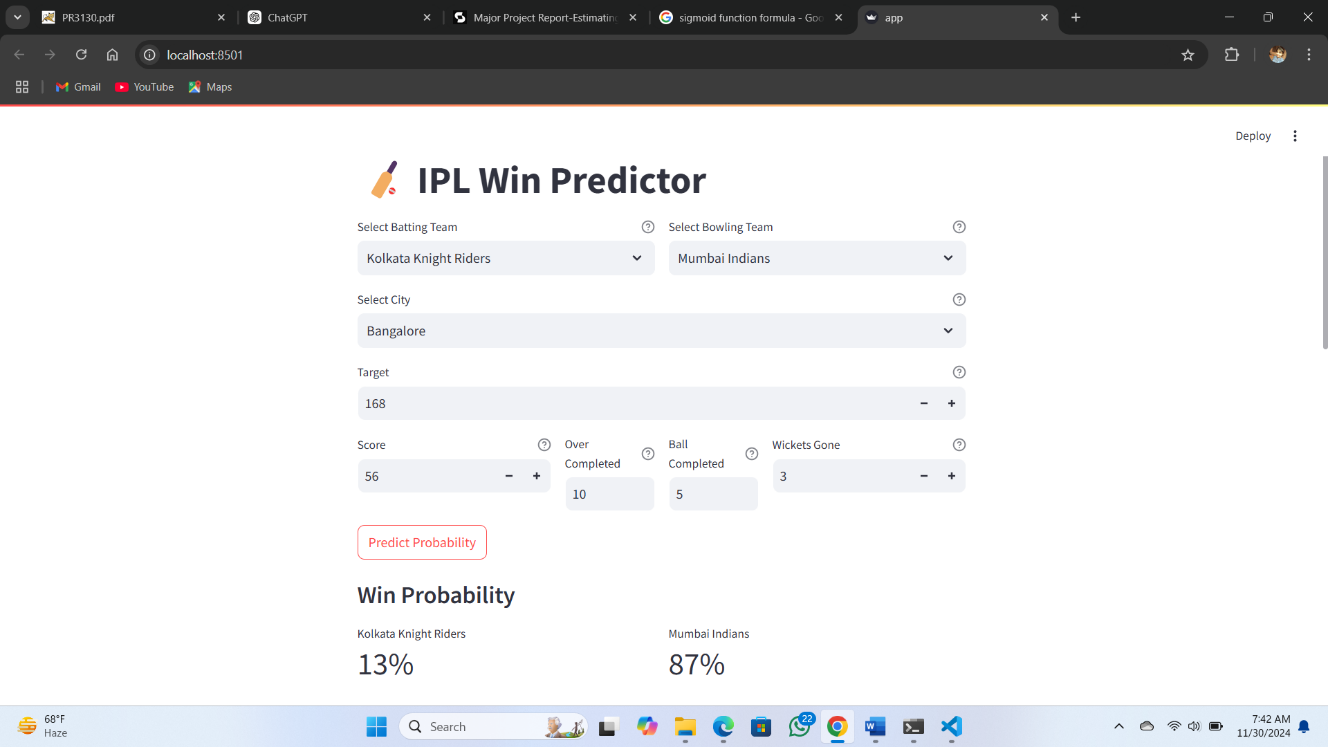
****

Fig 7.1: Final Prediction Model

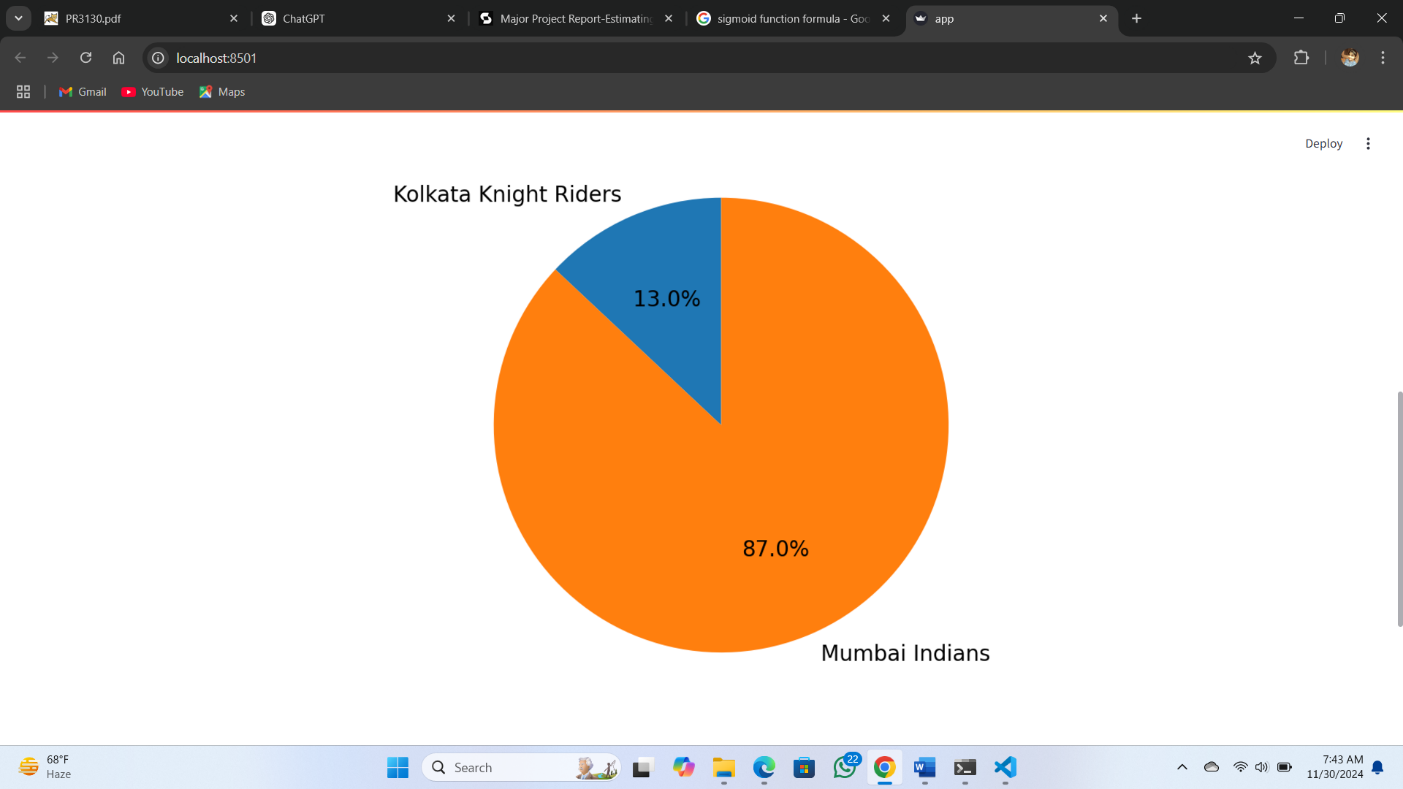
****

Fig 7.1: Pie-chart of the Possibility of Winning

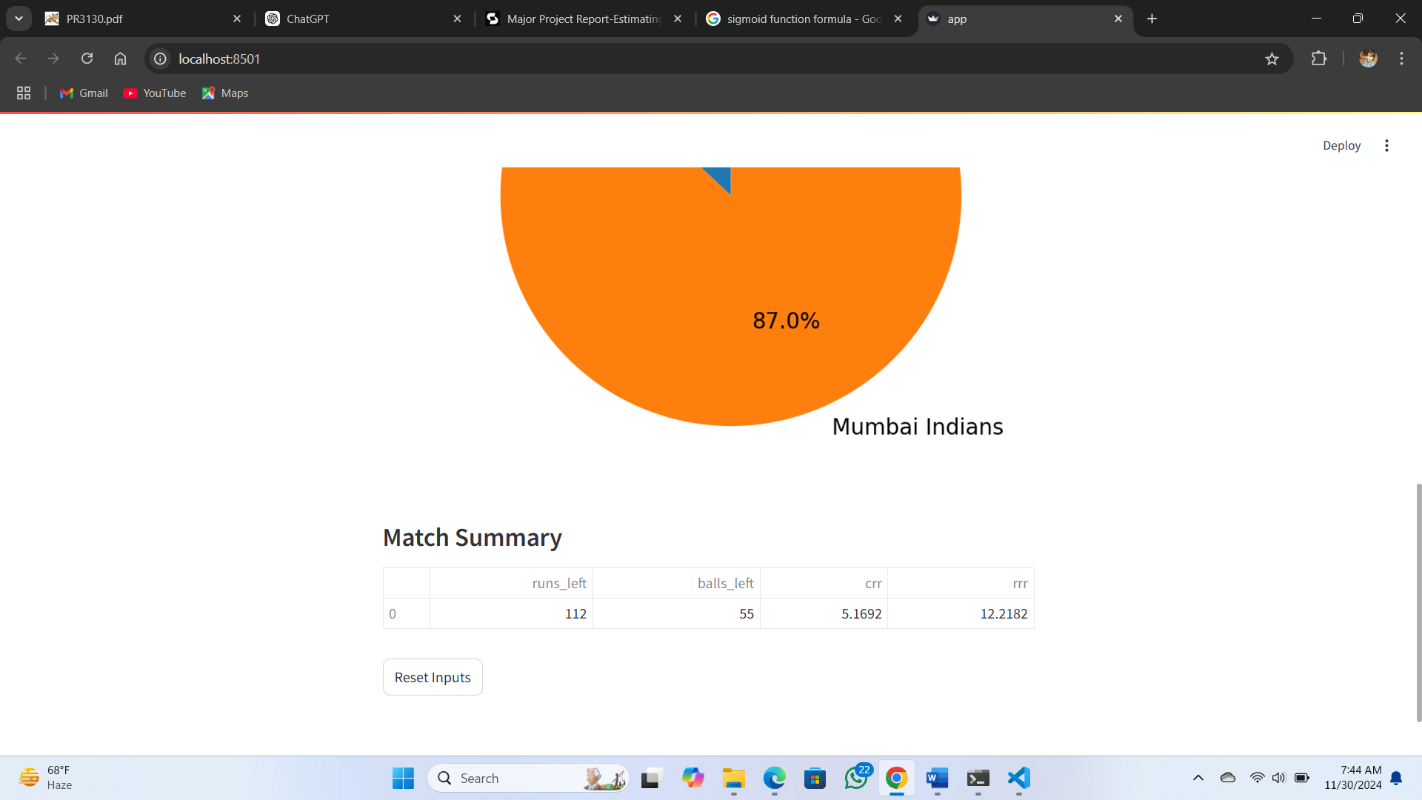
****

Fig 7.1: Showing the summary

**Future Scope**

Cricket is often referred to as a game of uncertainity. After studying the opponent team on many criteria such as the opponent team's players and their current form, climatic conditions, play grounds, and so on, the system may be enhanced with new capabilities such as player recommendation for strategic lead, Live Data Integration and Adding more visualizations like bar graphs or line charts to track performance metrics over time.. Another factor that can improve the model's performance is the addition of a larger sample set or new Indian Premier League season data.

**Conclusion**

The **IPL Win Predictor** project is a machine learning-based web application designed to predict the outcome of IPL matches in real-time. Leveraging historical IPL match data and key match parameters such as current score, overs completed, wickets lost, and required run rate, the project uses machine learning models like Logistic Regression or Random Forest to estimate the probability of a team's victory. Built using **Streamlit**, the application provides an interactive interface where users can input real-time match data and receive instant predictions visualized through metrics and charts. The tool combines sports analytics, real-time data processing, and user-friendly design, making it valuable for cricket enthusiasts, analysts, and anyone looking to understand the dynamics of IPL match and this provides more accurate prediction which is approximately 82% earlier in other models [1],[2] where accuracy was 68.33% and 63.89% respectively which was quite low and there predictions might be even less because of the reason that those projects were trying to predict the match result before the start of match which leads to automatic less accuracy but in our project this is more effective because we are predicting match winning percentage in the second inning.

**References**

1. Naik, A., Pawar, S., Naik, M., & Mulani, S. (2018). Winning Prediction Analysis in One-Day-International (ODI) Cricket Using Machine Learning Techniques. *International Journal of Emerging Technology and Computer Science, 3*(2), April 2018.
2. Singhvi, A., Shenoy, A., Racha, S., & Tunuguntla, S. (2015). Prediction of the Outcome of a Twenty-20 Cricket Match.
3. Nimmagadda, A., Kalyan, N. V., Venkatesh, M., Teja, N. N. S., & Raju, C. G. (2018). Cricket Score and Winning Prediction Using Data Mining. *International Journal of Advance Research and Development, 3*(3).
4. Lemmer, H. H. (2011). Team selection after a short cricket series. *South African Journal for Research in Sport, Physical Education and Recreation, 33*(3), 83–93.
5. Perera, H., & Davis, C. (2016). The effect of scoring rate on batting performance in one-day cricket. *Journal of Sports Analytics, 2*(1), 13–22.
6. Mukherjee, S. (2012). Quantifying individual performance in cricket – A network analysis of batsmen and bowlers. *Physica A: Statistical Mechanics and its Applications, 391*(23), 6066–6076.
7. Swartz, T. B., Gill, P. S., & Beaudoin, D. (2006). Optimal batting orders in one-day cricket. *Computers & Operations Research, 33*(7), 1939–1950.
8. Kapoor, A., & Bhatia, P. K. (2016). Analysis and Prediction of Indian Premier League (IPL) Matches Using Machine Learning Techniques. *Indian Journal of Science and Technology, 9*(38).
9. Saikia, H., & Bhuyan, B. (2019). Predicting Performance of Players in IPL Using Regression Models. *Journal of Emerging Technologies and Innovative Research, 6*(6).
10. Biswas, M., & Bhattacharjee, D. (2016). Cricket Match Outcome Prediction Using Machine Learning. *International Journal of Engineering Development and Research, 4*(2).
11. Rathod, S. G., & Aradhye, N. (2019). Analysis of Cricket Data Using Hadoop. *International Journal of Recent Technology and Engineering, 8*(1S4).
12. Asif, M., & McHale, I. (2016). Modelling cricket match outcomes using binary logistic regression. *International Journal of Performance Analysis in Sport, 16*(2), 687–697.
13. Choudhary, R., & Singh, G. (2017). IPL Match Result Prediction Using Decision Trees and Support Vector Machines. *International Journal of Computer Science and Mobile Computing, 6*(6), 157–165.
14. Saxena, S., & Sharma, R. (2019). Cricket Team Performance Analysis Using Data Visualization Techniques. *International Journal of Scientific Research in Computer Science and Engineering, 7*(1).
15. Silva, R., & Swartz, T. (2018). A probabilistic approach to evaluating bowling performance in cricket. *Journal of Sports Analytics, 4*(2), 123–132.
16. Lewis, A. J. (2008). Extending the range of the Duckworth-Lewis method. *Journal of the Operational Research Society, 59*(5), 729–742.
17. Clarke, S. R. (1998). Dynamic programming in one-day cricket—Optimal scoring rates. *Journal of the Operational Research Society, 49*(3), 230–239.
18. Bailey, M. J., & Clarke, S. R. (2006). Predicting the match outcome in one-day international cricket matches, while the game is in progress. *Journal of Sports Science & Medicine, 5*(4), 480–487.
19. Prakash, A., & Gupta, R. (2017). Decision Tree-Based Analysis of Cricket Data. *International Journal of Advance Research in Science and Engineering, 6*(4).
20. Jayakumar, V., & Usha, R. (2020). Prediction of Performance in IPL Using Ensemble Models. *International Journal of Recent Technology and Engineering, 9*(1S).
21. Tarapore, S., & Palshikar, G. K. (2012). Cricket Score Prediction Using Dynamic Bayesian Networks. *International Conference on Machine Learning Applications*.
22. Narayanan, R., & Goswami, A. (2018). Player Performance Analysis and Prediction Using Deep Learning in Cricket. *International Journal of Advanced Research in Artificial Intelligence, 7*(9).
23. Sharma, S., & Jain, A. (2020). Comparative Analysis of Various Algorithms for Predicting Cricket Matches Outcome. *International Journal of Innovative Technology and Exploring Engineering, 9*(4).
24. Turner, D. (2013). Utilizing Monte Carlo Simulation for Cricket Score Prediction. *Proceedings of the 12th Australasian Conference on Mathematics and Computers in Sport*.
25. Varghese, R., & Ravi, V. (2017). Analyzing Indian Premier League Team Performance Using Clustering Techniques. *International Journal of Computational Science and Engineering, 15*(2).
26. Kumar, V., & Reddy, V. (2019). Predicting the Outcome of T20 Cricket Matches Using Logistic Regression. *International Journal of Engineering and Advanced Technology, 8*(6S).
27. Gupta, R., & Rathore, S. S. (2019). Predicting Cricket Match Outcomes Using Statistical Models and Machine Learning. *International Journal of Computer Applications, 182*(34), 25–30.
28. Jain, K., & Gupta, P. (2018). Analysis of IPL Team Performances Using Data Mining Techniques. *International Journal of Applied Engineering Research, 13*(10), 7212–7217.
29. Agarwal, V., & Pandey, R. (2017). A Study on Decision-Making Models in Cricket Using Machine Learning Algorithms. *International Journal of Advanced Research in Computer Science, 8*(5), 1346–1352.
30. Khandelwal, R., & Singh, A. (2021). A Comparative Study of Prediction Models for Cricket