# Intel Unnati Industrial Training Program 2024 - 2025

PS 12: Knowledge Representation and Insights Generation from Structured Datasets

IPL Data Analysis

*Submitted by:*

Rishita Garg

Pranav Kumar Gupta

Manipal Institute of Technology, Manipal

# Table of Contents


- Introduction
- Objectives
- Dataset Description
- Methodology
- Results and Discussions
  - Data Preprocessing
  - Knowledge Representation
  - Pattern Identification
  - Insight Generation
- Edge Case Handling
- Conclusion

## Introduction

The Indian Premier League (IPL), established in 2008, is a premier cricket league that features top talent from India and around the world. Every match, with its own set of deliveries brings out data that gives an overview of how the game has changed over the years. This project aims to leverage IPL data to derive meaningful insights and predictions.

With the help of various resources available, useful information can be generated that allows better training, conditioning for the players, optimization of the coaching and support staff along with difference between the various venues on which the IPL is hosted.

## Objectives

- To visualise data after extracting them from their respective dataset files
- To display key performance indicators for various teams and players
- To build a model to predict statistics such as player of the match, win percentage
- To improve decision making by various stakeholders
- To generate useful insights and information for analysis

## Dataset Description

In this project, we have used two datasets named matches.csv and deliveries.csv

- matches.csv contains the id, season, city, date, team1, team2, toss_winner, toss_decision, result, dl_applied, winner, win_by_runs, win_by_wickets, player_of_match, venue, umpire1, umpire2, umpire3

- deliveries.csv contains information about each ball delivered such as match_id, inning, batting_team, bowling_team, over, ball, batsman, non_striker, bowler, is_super_over, wide_runs, bye_runs, legbye_runs, noball_runs, penalty_runs, batsman_runs, extra_runs, total_runs, player_dismissed, dismissal_kind, fielder.

These column names are self-explanatory based on its contents. Some of the columns have 0/1 entries, for yes or no respectively and others have string entries as their content.

## Methodology

The dataset comprises two files: one detailing match outcomes, including winners and players of the match, and another recording every ball delivered and its outcome.

Pandas (pd)

Pandas is a powerful and flexible data manipulation and analysis library for Python.

**Usage in Project:**

- Reading and writing data from various file formats like CSV.
- Data cleaning, preprocessing, and transformation.
- Merging, joining, and aggregating data from different sources.

NumPy (np)

NumPy is the fundamental package for scientific computing with Python. It provides support for arrays, matrices, and high-level mathematical functions to operate on these arrays.

**Usage in Project:**

- Performing numerical operations efficiently.
- Handling and manipulating large datasets.
- Utilized in conjunction with Pandas for data manipulation.

Seaborn (sns)

Seaborn is a data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.

**Usage in Project:**

- Creating visualizations such as bar plots, line plots and histograms
- Providing insights into the distribution and relationships of the data.
- Enhancing the interpretability of data through well-designed visual representations.

Matplotlib (plt)

Matplotlib is a plotting library for Python that enables the creation of static, interactive, and animated visualizations.

**Usage in Project:**

- Plotting data to visualize trends and patterns.
- Customizing visualizations with labels, titles, and legends.
- Supplementing Seaborn visualizations with additional customization.

Scikit-learn

Scikit-learn is a machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It is built on NumPy, SciPy, and Matplotlib.

**Usage in Project:**

- Model Selection (train_test_split): Splitting the dataset into training and testing sets to evaluate the performance of machine learning models.
- Ensemble Methods (RandomForestClassifier): Implementing Random Forest, an ensemble learning method for classification, to predict match outcomes based on features.
- Metrics (accuracy_score): Evaluating the accuracy of the classification model to determine its performance.

Ipywidgets (widgets)

Ipywidgets is a library that provides interactive widgets for Jupyter Notebooks and the IPython kernel. It enables the creation of interactive and dynamic user interfaces.

**Usage in Project:**

- Building a graphical user interface (GUI) within the Jupyter Notebook.
- Allowing users to input parameters (e.g., team names, venue, first innings score) and predict match outcomes using the trained machine learning model.
- Displaying results interactively and enhancing user engagement.

IPython Display (display)

The display function from IPython.display is used to display widgets and other outputs in Jupyter Notebooks.

**Usage in Project:**

- Displaying the interactive widgets created using Ipywidgets.
- Ensuring that the GUI elements are properly rendered in the notebook interface.

## Results and Discussion

- Data Preprocessing

The datasets we took had minor entries that could have created anomalies in the solution, so we made the following changes in the dataset to clean it and structure it better:

- Change the win_by_runs to NaN instead of zero.
- Wherever the match is tied there can be no winner instead of a default name.
- The win_by_wickets and win_by_run column values cannot be float, because that is unnecessary.

- Knowledge Representation

The solution effectively represents the knowledge we derived from the dataset. Some of the implemented representations have been added below:

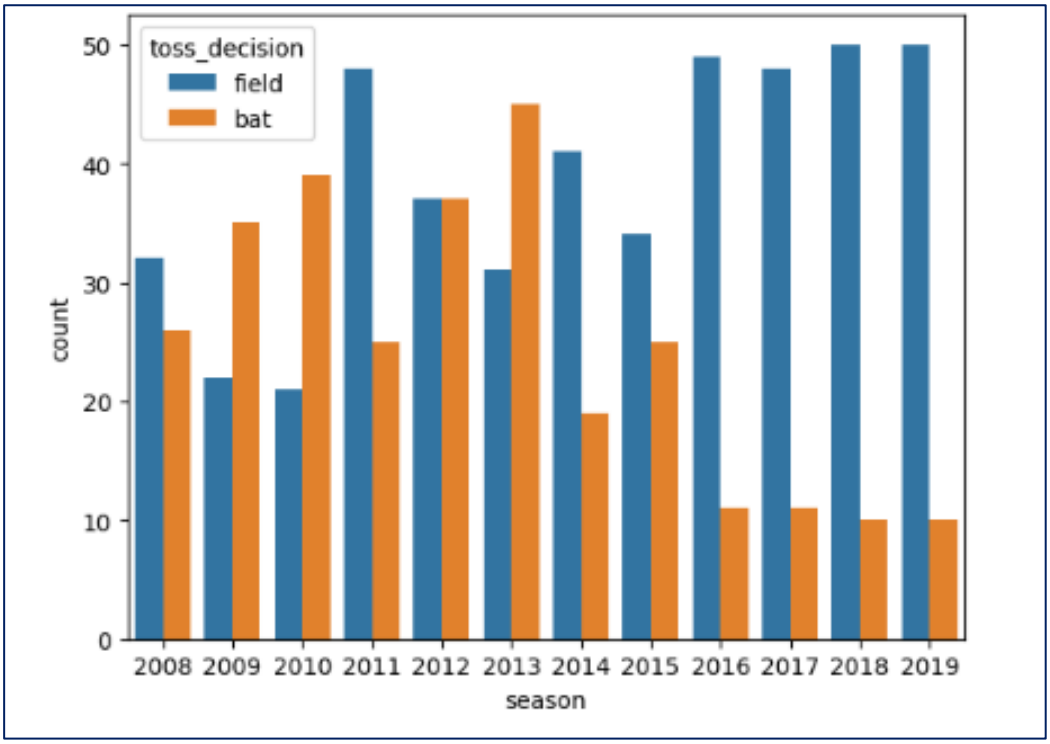



Fig: Visualize the toss decision across seasons

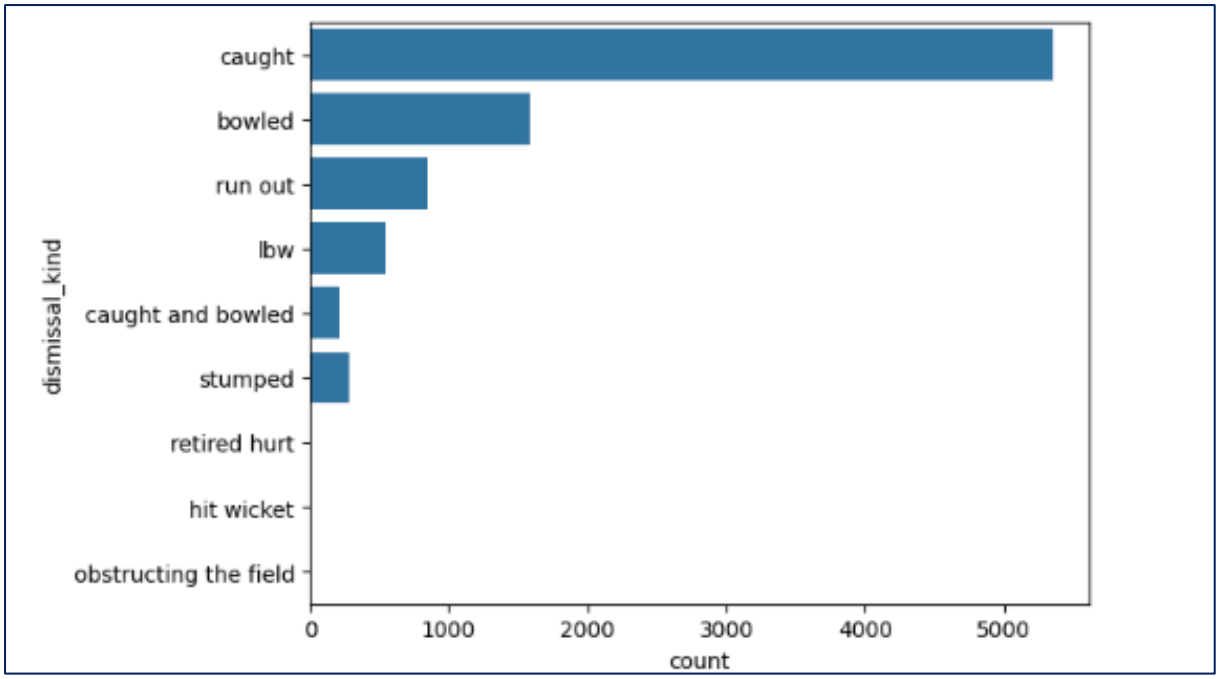



Fig: Find the dismissal kind and visualize using best fit graph.

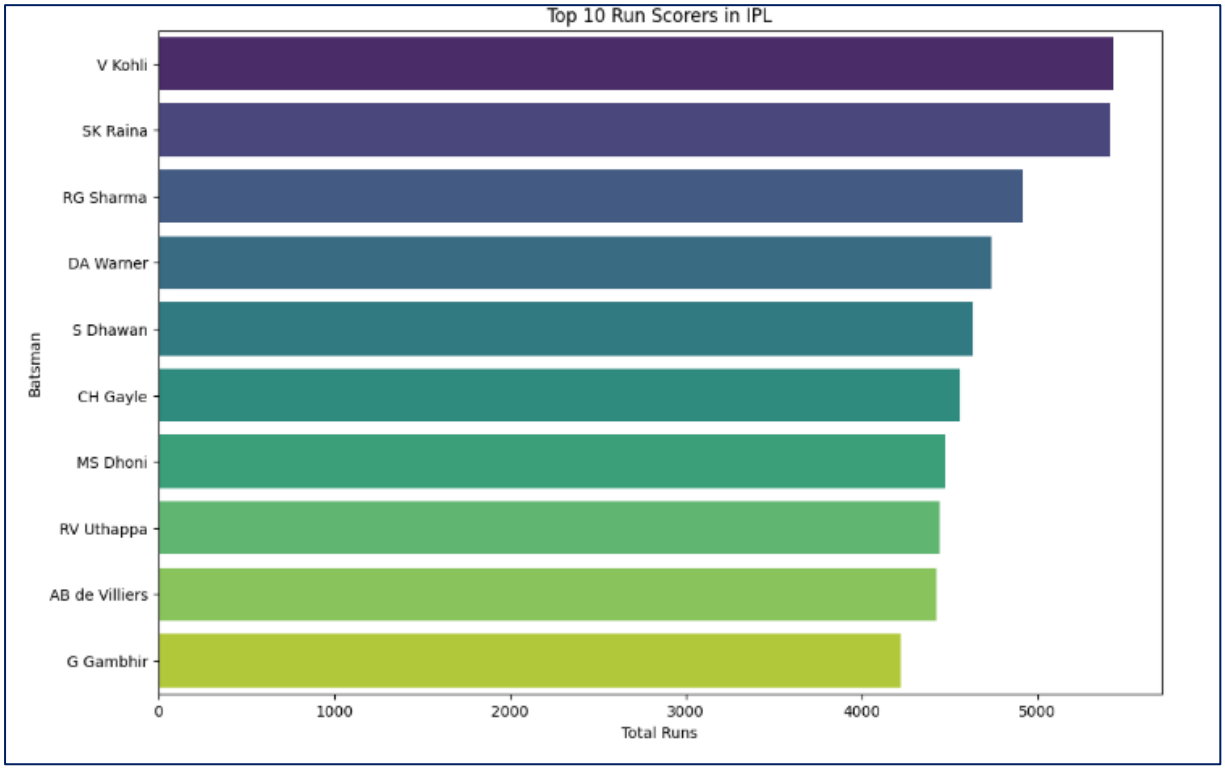



Fig: Visualising the top 10 run scorers in IPL over the years.

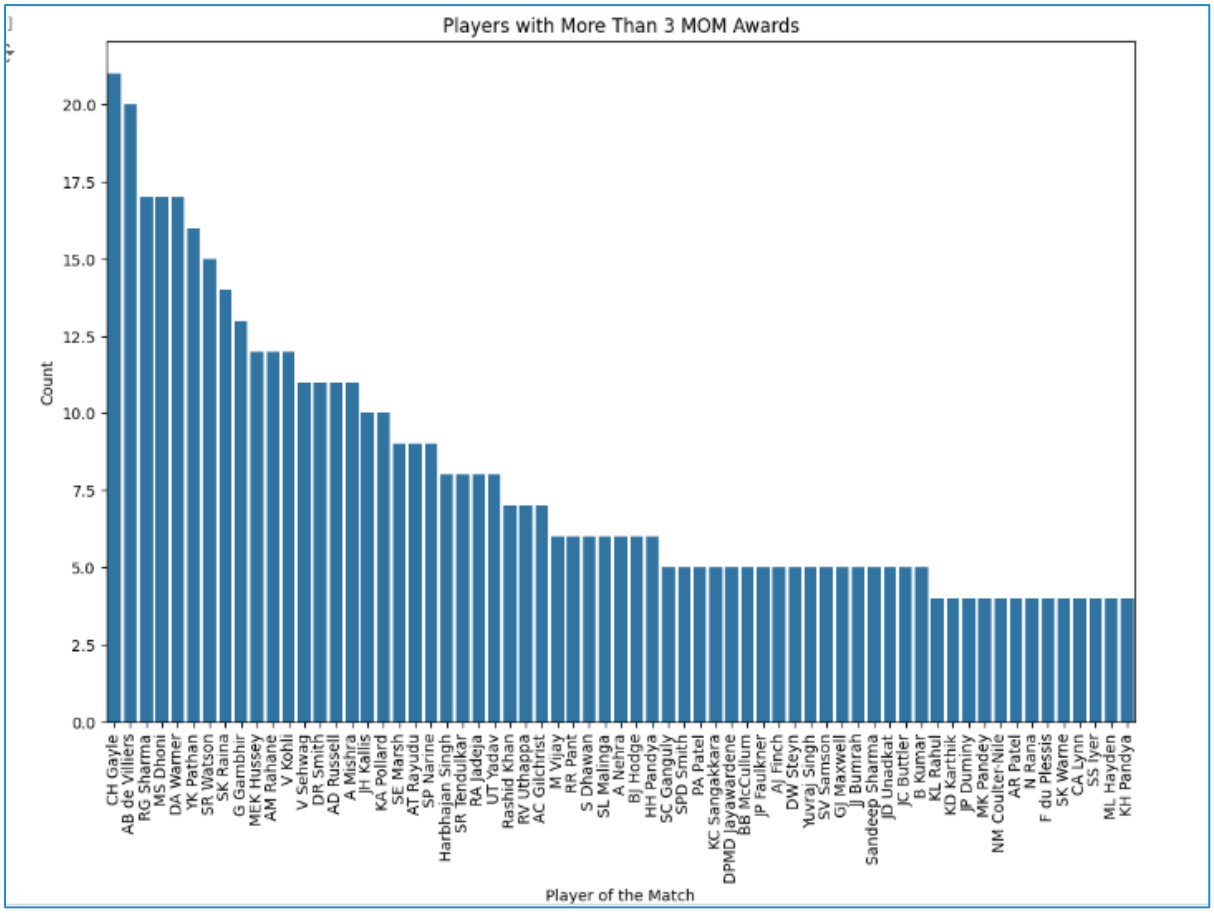



Fig: Visualising to represent the players who have won more than 3 Man of the Match titles.

| toss_winner | Chennai Super Kings | Deccan Chargers | Delhi Capitals | Delhi Daredevils | Gujarat Lions | Kings XI Punjab | Kochi Tuskers Kerala |
|---|---|---|---|---|---|---|---|
| toss_decision | | | | | | | |
| bat | 48 | 24 | 2 | 29 | 1 | 26 | 3 |
| field | 41 | 19 | 8 | 51 | 14 | 55 | 5 |

| Mumbai Indians | Pune Warriors | Rajasthan Royals | Rising Pune Supergiant | Rising Pune Supergiants | Royal Challengers Bangalore | Sunrisers Hyderabad |
|---|---|---|---|---|---|---|
| 44 | 11 | 32 | 0 | 3 | 20 | 20 |
| 54 | 9 | 48 | 6 | 4 | 61 | 26 |

Fig: Teams along with the number of times they opted for batting vs balling after winning toss.

| | toss_winner | Total Matches | won | lost | tie |
|---|---|---|---|---|---|
| 0 | Chennai Super Kings | 89 | 57 | 31 | 1 |
| 1 | Deccan Chargers | 43 | 19 | 24 | 0 |
| 2 | Delhi Capitals | 10 | 6 | 3 | 1 |
| 3 | Delhi Daredevils | 80 | 35 | 45 | 0 |
| 4 | Gujarat Lions | 15 | 10 | 4 | 1 |
| 5 | Kings XI Punjab | 81 | 34 | 46 | 1 |
| 6 | Kochi Tuskers Kerala | 8 | 4 | 4 | 0 |
| 7 | Kolkata Knight Riders | 92 | 53 | 38 | 1 |
| 8 | Mumbai Indians | 98 | 55 | 42 | 1 |
| 9 | Pune Warriors | 20 | 3 | 17 | 0 |
| 10 | Rajasthan Royals | 80 | 41 | 38 | 1 |
| 11 | Rising Pune Supergiant | 6 | 5 | 1 | 0 |
| 12 | Rising Pune Supergiants | 7 | 3 | 4 | 0 |
| 13 | Royal Challengers Bangalore | 81 | 40 | 39 | 2 |
| 14 | Sunrisers Hyderabad | 46 | 23 | 23 | 0 |

Fig: When the toss winner was also the match winner, grouped by teams.

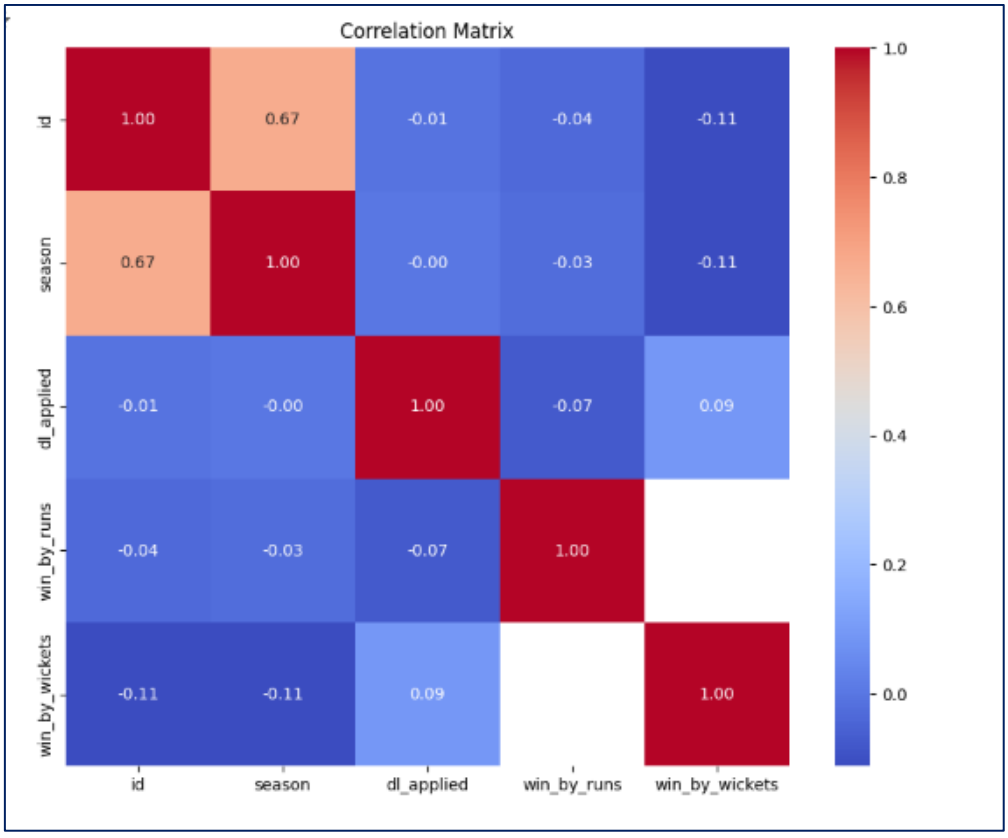
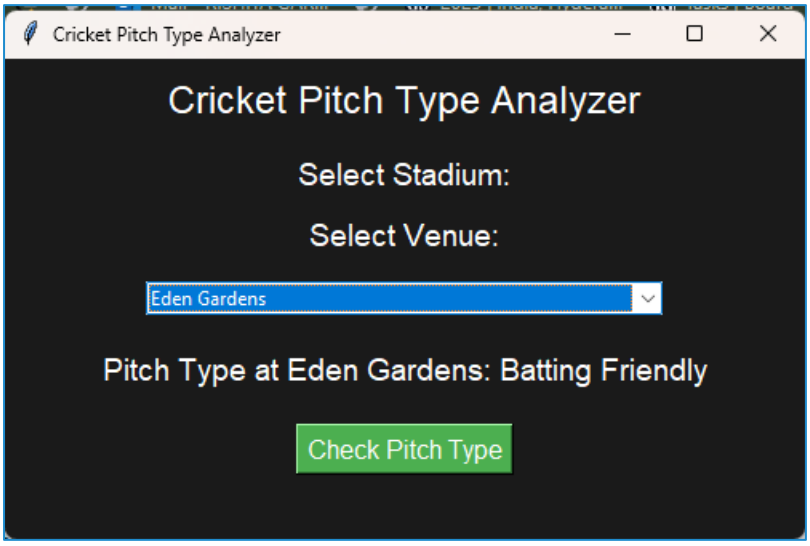



Fig: Correlation Matrix - Shows the correlation between two variables

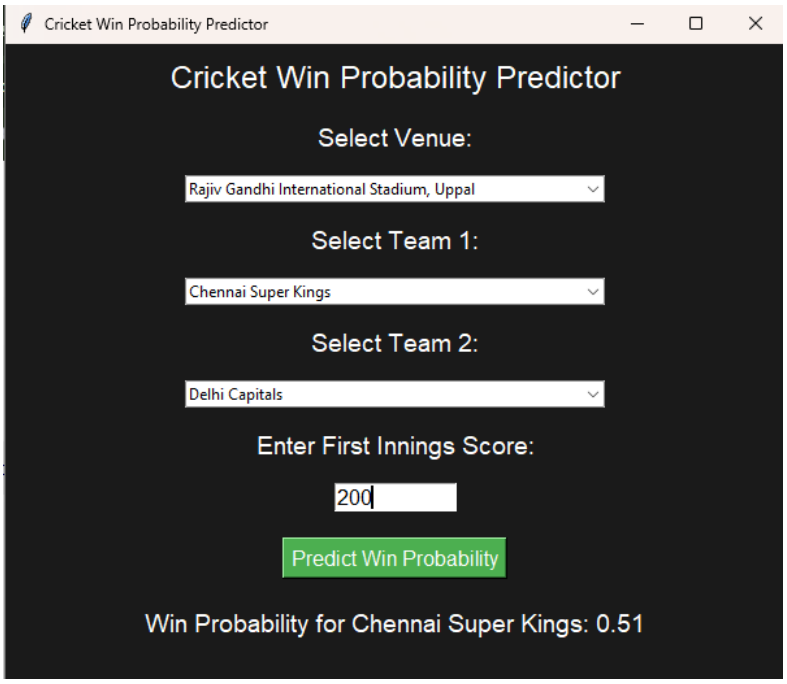
- Pattern Identification

For this we came up with an intriguing model that determines whether a pitch is better for bowling or batting, and the UI component for the same is uploaded in our git repository- Pattern Identification UI.



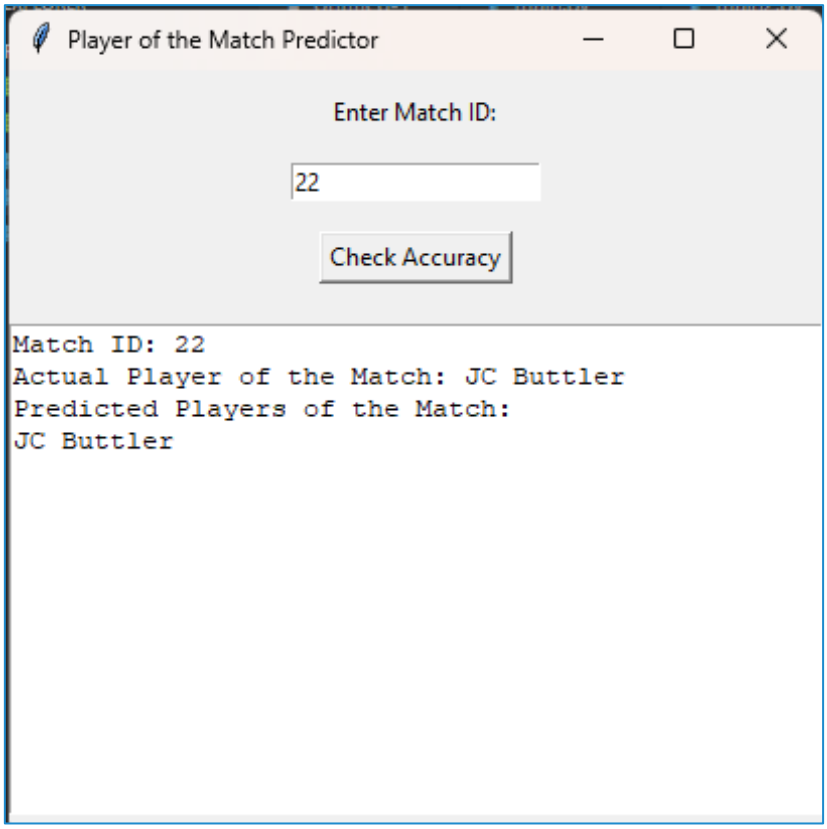

The deliveries data is aggregated to compute the total runs scored in the first innings of each match. This aggregated data is then merged with selected features from the matches dataset. The merged data is used to calculate average scores at each venue, classifying pitches as either 'batting' or 'bowling' based on whether the venue's average score exceeds the global average. The resulting data is split into features and target variables, with categorical variables being one-hot encoded. A Random Forest Classifier model is trained on this data to predict pitch type. The code includes a function to classify new matches based on input features. Additionally, a Tkinter-based GUI is provided to allow users to input match details and get the pitch classification, enhancing user interaction with the model.

- Insight Generation
  - Win Probability Predictor




The system integrates data from matches.csv and deliveries.csv, focusing on venue, team compositions, and match outcomes. Initial data preprocessing involved calculating first innings scores and encoding categorical variables such as venues and team names using Label Encoder. Utilizing a RandomForestClassifier optimized through GridSearchCV for ROC AUC scoring, the model predicts match outcomes based on input parameters including venue, team compositions, and first innings score. The tkinter-based GUI enables users to select venue and team combinations, input first innings scores, and receive real-time predictions on win probabilities.

- Player Of the Match Predictor




Leveraging pandas for data manipulation and sklearn for model implementation, the system merges match and delivery datasets, computes player statistics such as strike rate for batsmen and economy rate for bowlers, and labels players based on their likelihood of being the Player of the Match. Using a Random Forest Classifier, the model achieves a commendable accuracy score on test data, demonstrating its effectiveness in predicting key player performances. The project culminates in a user-friendly tkinter-based interface that allows users to input match IDs and receive predictions and actual outcomes, enhancing cricket match analysis capabilities.

## Edge-Case Handling

In developing these predictive models, we took steps to handle various real-world scenarios to ensure their accuracy and reliability. For instance, we adjusted the dataset to accurately reflect match outcomes: replacing zero values in the win_by_runs column with NaN where applicable and ensuring that tied matches correctly indicate no winner instead of default values. We ensured that columns like win_by_wickets and win_by_runs were appropriately represented as integers rather than unnecessarily as floats.

Despite these improvements, it's important to acknowledge that the models still have inherent limitations. They do not account for detailed player-specific statistics or external factors such as individual player form, match-day conditions like weather or pitch conditions post-rainfall, which can significantly impact match results but were not fully considered due to data availability and scope constraints.

Our objective was to develop models that provide meaningful insights and accurate predictions based on the data we could effectively incorporate.

## Conclusion

This project successfully developed robust models for pitch classification, win probability prediction, and player of the match identification in IPL cricket. By integrating and analysing comprehensive match and delivery data, we created insightful tools that aid strategic decision-making and enhance understanding of game dynamics. Our user-friendly interface ensures accessibility, making it a valuable asset for analysts, teams, and enthusiasts.