Table of Contents

Chapter 1 Introduction

1.1 Project Description

This project seeks to enhance the analytical experience of live ongoing/cricket matches by providing up-to-the-minute scores, ever-changing statistics, and insights for fans, analysts, and aficionados of the game. By means of gaming engagement through the display of live match updates alongside changing metrics, the platform augments engagement of the sport. Information shared provides strategic counsel such as batting or bowling first based on historical venue statistics, weather conditions, and past performance records. The information attempts to render teams and supporters a more enhanced understanding prior to the first ball being bowled from a data-driven approach. Besides the pre-match analysis, the project puts forth a machine learning model aimed at predicting the likelihood of completing a successful chase during the second innings of a T20 match. This model perpetually updates after each ball, adjusting win possibilities to reflect the current match status. With the aid of Machine Learning, the platform not only improves viewers' experience, but also provides analysts with more predictive insights of the match changing the way T20 cricket is consumed and understood.

1.2 Problem Definition

Cricket teams frequently confront problems regarding whether to bat or field after winning a toss, as a number of conditions like ground situation, weather, and past performances need to be considered. Also, Fans and teams do not have well-calibrated, near-instantaneous forecasts of the game results that change with the progress of the game or with the players' and teams' statistics and performance indicators.

1.3 Proposed Solution

A Toss Decision Support Tool recommends when to bowl, while also analyzing pitch and weather conditions to recommend which team should bat first. The tool suggests that in overcast conditions, teams are more likely to bowl due to increased swing, thus enabling teams to capitalize swing-based data-informed decision making. A machine learning model in a Real-Time Match Outcome Predictor analyses live game data and combines it with historical and performance records to predict game outcomes alongside current match evaluations. Research proves that models built using Random Forest Classifier have high accuracy in match prediction, underscoring their value for teams and fans alike.

1.4 Purpose

The goal of this project is to develop a platform for real-time cricket scoring and prediction that engages fans by providing instant updates on matches and AI-assisted win probability metrics. Fans are able to monitor up-to-the-minute scores, match changes, and even interact with the game in new ways. Analysts can utilize advanced performance metrics accurately tracked and provided through external data sources.

1.5 Scope

Even though the project offers a strong way to predict and analyse ICC world cup matches, it still has some drawbacks. Nowadays, predictions are made solely on the needs of the second innings, where the chasing team's chances are examined using the run rate, wickets left, and how the game is going. In addition, toss recommendations depend on history, so they might miss certain quick changes in the environment. Adding the predictions for the first innings as well as new adaptive factors to the model might make it more useful and precise.

Chapter 2 Literature Survey

2.1 Domain Study

To use CricketVerse well, we must examine the domain by studying cricket analytics, using active match prediction platforms, checking for solutions, and figuring out users' needs. They must check the model's results along with its techniques and monitor changes in the AI sport analytics field and the sports industry. In this part, the focus is on tools such as machine learning as well as programs designed for web scraping and data processing via the cloud. It introduces several ways that people can be active in the sports industry as well as the business strategies and possible relationships involved in sports-related activities. The aim is to find new approaches to differentiation and at the same time give analytics-based services to satisfy those who follow and analyze cricket.

2.2 Related Work

This involves the study of research papers and journals. Literature survey is completed by considering following research papers.

- Winner Prediction in One-Day International Cricket Matches Using Machine Learning Framework: An Ensemble Approach
  Author :- Manoj Ishi, Dr. Jayantrao Patil et al.
  Publication :- Indian Journal of Computer Science and Engineering | e-ISSN: 0976-5166 | Volume:13/Issue:03/June-2022
  Summary :- The research focuses on creating machine learning tools that forecast the team to win an ODI match even before the game gets underway, allowing team officials and experts to use this data. The study takes historical data from 1693 ODI matches (2006–2019) to measure batting and bowling capacity, the ways teams score, and different team stats. Experiments were carried out using Logistic Regression, SVM, and methods such as ensemble methods, along with feature selection to boost the results' correctness. Models with the highest score of 96.31% revealed that machine learning is capable of managing crickets unpredictable nature and supports better strategies. More improvements will address including extra features and using this approach in other types of cricket formats or sports analysis.[1]

- Prediction of IPL Match Outcome Using Machine Learning Techniques

  Author :- Srikantaiah K C, Aryan Khetan et al.

  Publication :- 3rd International Conference on Integrated Intelligent Computing, Communication & Security (2019)

  Summary :- Machine learning models are used in this study to foretell how IPL matches will turn out using information from nine years in the past. After reviewing data that records the team's performances, match facts, players' performances, and each ball delivery, details like win percentages, the effects of the toss, and statistics for each player were extracted. Models such as Random Forest, SVM, Logistic Regression, and K-Nearest Neighbour were implemented, and Random Forest turned out to have the highest accuracy of 88.10%. Out of all the data, 70% was used for training and 30% for testing, and cross-validation helped make the results reliable. The findings here benefit team managers, analysts, and fans who can make smart decisions concerning game strategies and predictions. It will be important to analyze each player's performance and also add extra features to boost the prediction accuracy and usefulness of the tool.[2]

- Cricket Match Analytics and Prediction Using Machine Learning

  Author :- Param Dalal, Hirak Shah, Tej Kanjariya, Dhananjay Joshi

  Publication :- International Journal of Computer Applications (2006) | ISSN: 0975-8887 | Volume 186/Issue:26/June-2024

  Summary :- The research concentrates on cricket match prediction by applying machine learning to outcomes in the second innings, analysing the target needed, the number of runs still needed, wickets lost, and players' statistics. The introduction of a "Player Consistency" formula makes the predictions more precise because it mixes traditional numbers with up-to-date statistics. Models reviewed in the study included Random Forest, SVM, Logistic Regression, and Naive Bayes; Random Forest scored a testing accuracy of 89.82%. Thanks to feature extraction and deep learning for weight optimization, the authors highlight many improvements in cricket analytics that help countless players, teams, and fans with useful insights and show areas where future prediction models can be improved.[3]

- Predicting IPL Victories: An Ensemble Modelling Approach Using Comprehensive Dataset Analysis

  Author :- Pritpal Singh, Dr. Jatinder Kaur, Lovedeep Singh

  Publication :- 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (2024) | ISSN: 3503-4922 | Issue: 1, 24

  Summary :- Using statistics of players, teams, and venue along with weather, this research tries to predict the outcome of IPL matches using machine learning. Tests were done with Decision Trees, Random Forest, and the ensemble model; the ensemble model turned out to be 99% accurate and surpassed the others. Ensemble approaches are seen to increase accuracy, such findings suggest that it could be useful to add more specific types of information in future applications.[4]

- Outcome Prediction of ODI Cricket Matches Using Decision Trees and MLP Networks

  Author :- Jalaz Kumar, Rajeev Kumar, Pushpender Kumar

  Publication :- First International Conference on Secure Cyber Computing and Communication (2018)

  Summary :- In the research, Decision Trees and MLP Networks are applied to predict results of ODI cricket matches depending on past results, choice of venue, in which inning the teams play, and whether it is a home match. With data from 3933 ODI matches, the study built models using pre-match details and came up with a tool that can reliably predict the result of an ODI match. The MLP Classifier model managed to reach an accuracy of 57.4%, which was slightly better than the 55.1% of Decision Trees. It is evident from the results that such classifiers are effective in learning from the past and point out improvements that could be made by studying how teams are formed and using them in various sports.[5]

2.3 Existing System

Table 2.1: Comparison Table

Features	CricViz	ESPNcricInfo	Smartcric	CricketVerse
Live Score and Match Details	Y	Y	Y	Y
Toss Decision Prediction	N	N	N	Y
Second Innings Ball-by-Ball Victor Prediction	Y	Y	N	Y
Venue-Specific Insights	N	Y	N	Y
Prediction for all formats	Y	N	N	N

2.4 Technology Survey

This technology survey includes following technologies and how they are used in the developing CricketVerse project: -

- **Node JS: -**

  CricketVerse relies on Node.js to take care of building the server-side logic, integrating APIs, and processing data in real time. Requests and fake_useragent Python functions help to update and collect cricket scores that are immediately stored in MongoDB. RESTful APIs from the Node.js backend help the MERN client in showing updates on live scores in real time. Also, the system remains stable as it keeps only the most recent five content items in a priority queue within MongoDB. In addition, Node.js features AI that predicts the probability of winning, so that the victory probability bar will appear only for international T20 matches in the second innings. Due to its cellular architecture and ability to handle several tasks at the same time, CricketVerse is designed to handle many requests and quickly process them.

- **React JS: -**

  React.js supports CricketVerse by giving a dynamic and reliable interface forien scores and predictions made by AI. Thanks to its component model, the platform can quickly load and change match data that comes from the Node.js backend through RESTful APIs. Both the state system and local hooks in React help make updates to the user interface without rendering unnecessary data. In addition to other functions, the user interface has match cards, a score ticker, and a 'win chance' bar that becomes visible only when an international T20 match is in its second inning. React makes sure that the latest data is always shown on the screen, because it efficiently manages asynchronous requests. CricketVerse's design makes it easy for features to be added and maintained, for example, with filtering of more matches, respected user preferences, and advanced statistics tools.

- **CSS :-**

  By using CSS in the development, CricketVerse is able to provide a pleasing, responsive, and user-friendly way for showing users live cricket updates and AI-based predictions. Flexbox, Grid, and media queries are used in the modern CSS approach to ensure that the platform looks clear and adjusts to any screen size. With custom design and visual effects, the user finds match cards, the live scoreboard, and the win probability bar more interesting. A dark color scheme enhances the experience of reading, and the website's style is improved further by clear and professional colour options, typography, and hover effects.

- **JavaScript: -**

  JavaScript is key to CricketVerse because it provides engaging and lively features to enhance the user's experience. It makes sure that live cricket scores are up-to-date, even without a need to refresh your web page. Queries from APIs are handled by JavaScript, and the results are processed quickly, and displayed on the user interface. It also oversees users' actions, such as switching between matches, selecting who to match with, and animating the gameplay. The software makes sure the win probability meter only appears on the screen once play has restarted in the second innings. Besides, JavaScript plays a role in adding external libraries for displaying data and makes the platform work more smoothly and efficiently.

- **MongoDB: -**

  Live cricket match information is managed and kept in CricketVerse through the main database known as MongoDB. This is valuable because data is can be structured and processed easily right as it comes, which is needed for real-time use of applications. Records, teams' scores, players' statistics, and historical information are organized in a way that makes it easy to access and change them. Because MongoDB can deal with plenty of reads and writes at once, match updates are handled and displayed up-to-date. Only the five newest records are saved in the database to ensure the database works well and to avoid storing too much data. Because of its relationship with Node.js, CricketVerse can always keep its database and the website in sync with live updates.

Chapter 3 Hardware and Software Requirements

3.1 Hardware Requirements

Hardware Specification	
Specification	**Desired Value**
Processor	Intel i3 and above
Memory (RAM)	8GB Minimum
Hard Disk	10GB Minimum

3.2 Software Requirements

Software Specification	
Specification	**Desired Value**
Operating System	Web OS/ Mobile OS
Language	Python 3.7, JavaScript
Front end Tool	ReactJS (v18.2.0 or higher)
Backend Tool	NodeJS (v18.16.0 or higher)
Database	MongoDB (v5.0.2 or higher)
Web Server	ExpressJs (v4.18.2 or higher)
Web Browser	Any
Development Tool	Visual Studio Code
API Testing Tool	Postman
Library	NumPy 1.14.3, Matplotlib 2.1.2 Scikit-learn, Pandas, Seaborn, requests, BeautifulSoup

Chapter 4 Software Requirement Specifications

4.1 System Users

- There are following type of system users :-

- **User (Fans/Cricket Analyst):** Accesses live match updates, toss decision, prediction insights, player statistics and match result statistics.

4.2 Functional Requirements

- There are following functional requirements :-

- **Live Score and Statistics Display**

  CricketVerse gives its users quick access to live scores and plenty of useful information about the match and its players. This feature supplies key stats that join run rates, needed run rates, and the prediction of the winning score, making it possible for followers and other people to follow each game's development. The intuitive design prevents drops in the connection of matches so that fans can watch them without interruptions.

- **Second Inning Win Prediction**

  In real-time, the model uses machine learning to see if a team meets their target by calculating several possible outcomes of the game's second inning. The evaluations run after every bowl to check variables such as the number of remaining wickets, needed runs per ball, and statistics of every player. Horizon Match Preview gives accurate information on matches as they are played so that fans and analysts can follow the match more closely.

- **Toss Decision Recommendation**

  If they win the toss, CricketVerse gives a strategy suggestion by considering the team's history, the details of the place, and weather conditions, helping them choose whether to bat or bowl first. The tool gives teams certain ideas to follow that help them perform their best under today's circumstances.

- **Generate Match Result Statistics**

  CricketVerse runs its own statistics after each match comes to an end automatically. Information available on CricketVerse includes key scores, player details, and match summary so that you can always find out everything that happened in a game. Users use

this feature to view final matches so they can study the performance of teams and players for better preparation incoming matches.

4.3 Non-Functional Requirements

- There are following functional requirements :-

- **Scalability**

  The architecture used in CricketVerse can manage a high number of users at the same time, mainly when lots of people use the platform during matches. Due to effective load balancing and use of distributed computing, the platform can handle multiple cricketing events and go on to provide real-time information and predictions without dropping in performance.

- **Performance**

  CricketVerse prioritize performance optimization to deliver fast loading times and responsiveness across the platform. Through efficient code optimization, we enhance the platform's efficiency and usability, providing users with a seamless and enjoyable experience.

- **Usability and Accessibility**

  CricketVerse offers sophisticated cricket analytics in forms that are simple to understand thanks to its user-friendly interface. Regardless of technical expertise, customers can easily access match statistics and forecast insights thanks to carefully crafted dashboards and interactive visualizations. Fans of all skill levels can access extensive cricket analytics thanks to the platform's adherence to contemporary design concepts and user experience criteria.

Chapter 5 System Design

5.1 Architecture Diagram

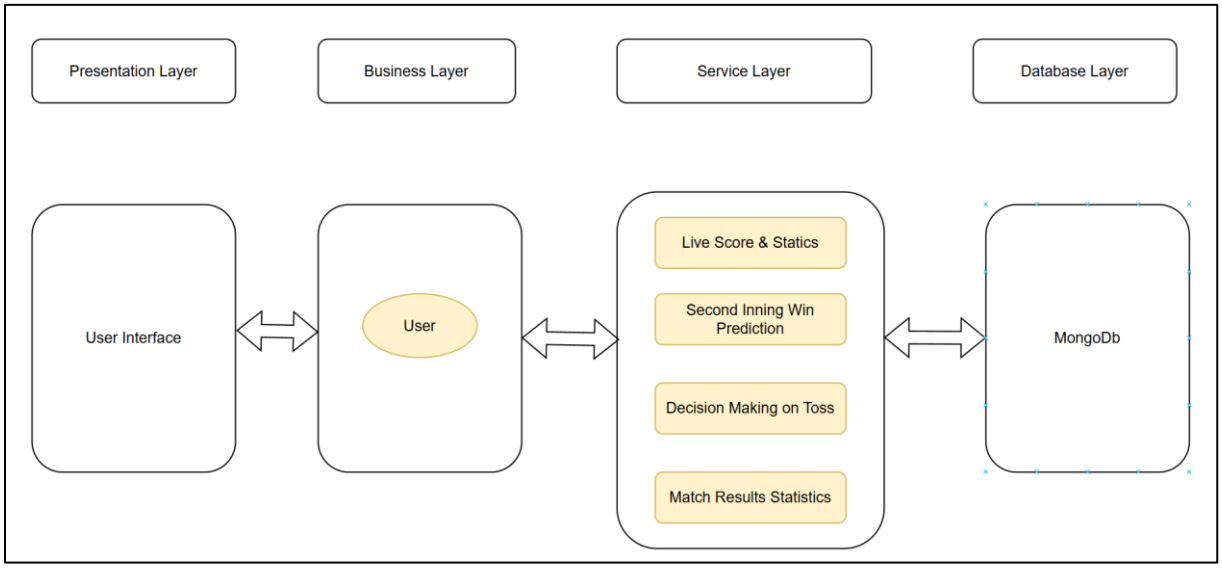

Figure 5.1: Architecture Diagram

The architecture diagram of our system comprises several distinct layers, including the Presentation Layer, Business Layer, Service Layer, and Database Layer, are depicted in the following image, which shows a layered architectural diagram that shows the layout of a software program. The database technology utilized is MongoDB. The User Interface, which enables communication with the User, is fundamental to the diagram. The Business Layer is in charge of processing essential features including match results data, second inning win prediction, live score and statistics, and toss decision-making.

In the context of athletic events, these components highlight the application's function in managing real-time data and statistics. The two-way arrows represent the data flow between the User, the Service Layer, and the Database, emphasizing the dynamic information exchange necessary to provide the end-user with current and pertinent content. All things considered, the design provides a smooth user experience by clearly illustrating how data is handled and processed within the application's ecosystem.

5.2 Use Case Diagram

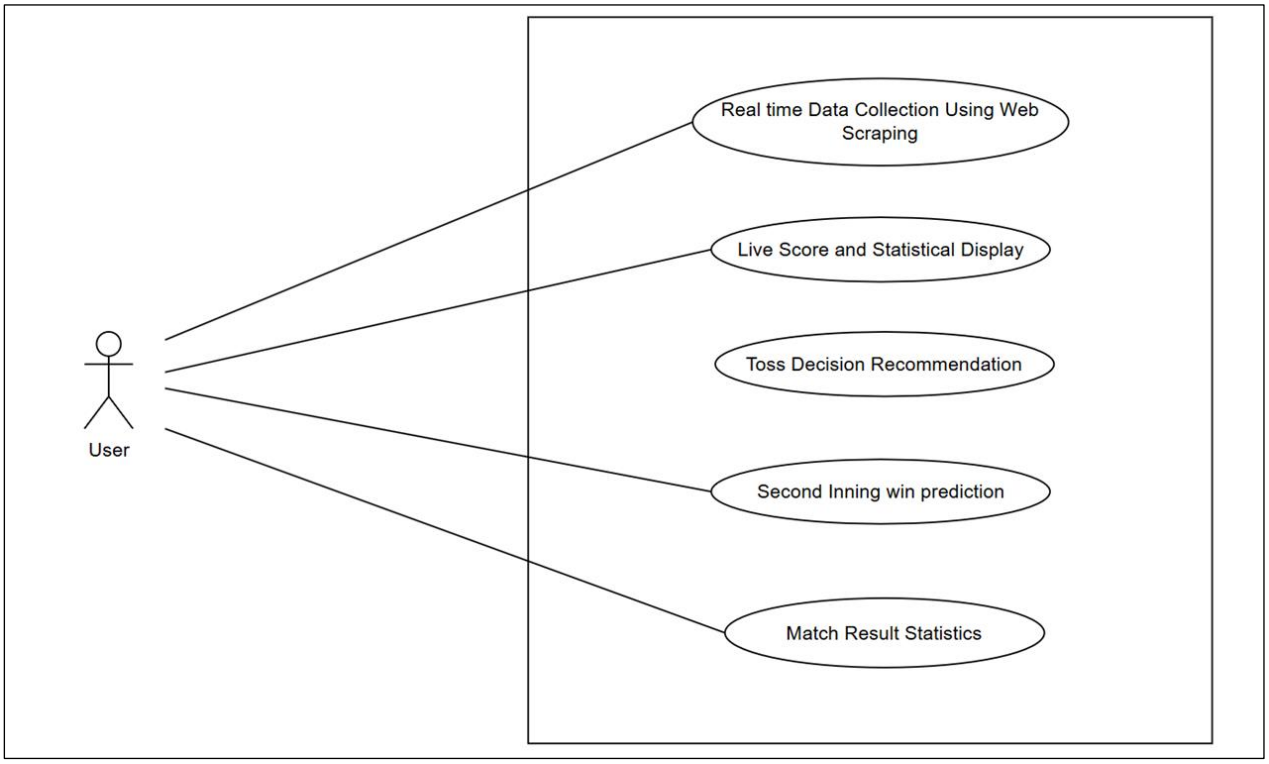

Figure 6.1: Use Case Diagram

CricketVerse's use case diagram (Figure 6.1) shows that the User acts as the main participant and cooperates with important components of the system. Using web scraping, the platform is able to display match updates as they happen, available to the user. In addition, customers are able to check performance stats, estimations for the final scores, and every ball's score by using features such as Live Score and Statistical Display. Using results from past matches and the pitch's circumstances,

CricketVerse recommendations help users select if a team should start by batting or bowling. This feature also benefits the user by giving the probability that the chasing team can make the required runs considering the existing conditions and player's performances. Besides, users may look at Match Result Statistics, which gives useful performance information and final outcomes of matches. This diagram shows how the user uses various features and how the platform makes cricket viewing more enjoyable using data.

Chapter 6 Detailed Design

6.1 Process Flow Diagram

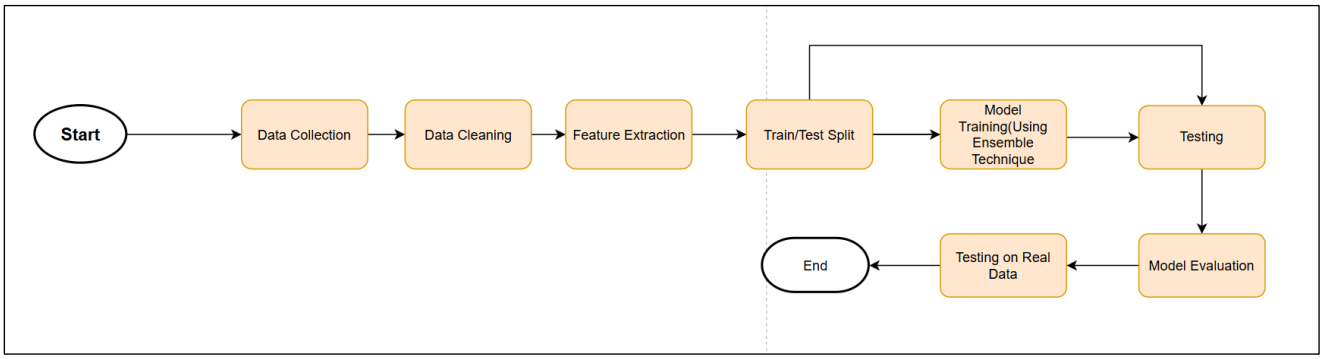

Figure 6.1: Process Flow Diagram

The process flow diagram for CricketVerse's Machine Learning Model Development depicts the methodical methodology utilized to create and evaluate the model.

- **Start** - The procedure begins with data collecting.
- **Data Collection** - Relevant cricket match data is collected from multiple sources, including live match statistics, historical records, and player performance measures.
- **Data Cleaning** – The collected data undergoes preprocessing, including handling missing values, removing inconsistencies, generating required columns out of existing one, and formatting for consistency.
- **Feature Extraction** – Key features like as batting performance, bowling impact, and run rates are extracted to boost model accuracy.
- **Train/Test Split** – The dataset is divided into training and testing subsets to ensure the model is trained effectively and evaluated properly.
- **Model Training Using Ensemble Techniques** – The model is trained using an ensemble approach, incorporating multiple algorithms to improve predictive accuracy.
- **Testing** — The trained model is tested with unseen data to measure its performance.
- **Model Evaluation** – Key evaluation metrics such as accuracy, precision, recall, and F1-score are analysed to determine the model's effectiveness.
- **Testing on Real Data** — The model is further verified using real-world match scenarios to assess its predicted dependability.

- **End** – The procedure closes, guaranteeing that the model is ready for deployment in CricketVerse for real-time match predictions and analytics.

6.2 Database Design

Document Structure

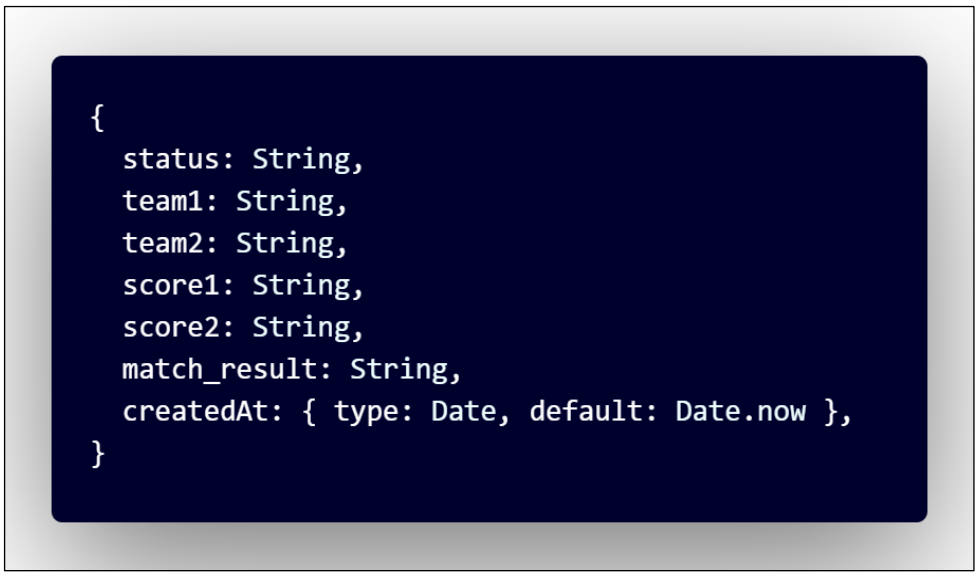

Figure 6.2: Live Match Data Document Structure


```
{
  "_id": {…},
  "match_id": "4115f02ef4a80178759ca08ac26c0a80",
  "status": "STUMPS",
  "team1": "India",
  "team2": "England",
  "score1": "471",
  "score2": "(49 ov) 209/3",
  "match_result": "Day 2 - England trail by 262 runs.",
  "match_url": "https://www.espncricinfo.com/series/india-in-england-2025-1445348/england-vs-in
  "match_format": "1st Test",
  "venue": "Leeds",
  "date": "June 20 - 24",
  "toss": "England, elected to field first",
  "player_of_the_match": "N/A",
  "current_run_rate": "CRR: 4.26 • Last 10 ov (RR): 52/1 (5.20)",
  "inning_1": {…},
  "inning_2": {…},
  "createdAt": {…},
  "__v": 0
}
```


Figure 6.2: Live Match Data Document Example

The first image (Figure 6.1 represents the schema definition for storing cricket match data in a MongoDB collection. It defines the structure of documents that will be inserted into the database. The schema includes:

- **status (String):** The current status of the match (e.g., “LIVE”, "ONGOING", "RESULT", "CANCELLED").
- **team1 (String):** The name of the first team.
- **team2 (String):** The name of the second team.
- **score1 (String):** The score of the first team.
- **score2 (String):** The score of the second team.
- **match_result (String):** The final result of the match (e.g., "India won by 142 runs").
- **createdAt (Date):** The timestamp when the document was created, with a default value of the current date and time.

The second image(Figure 6.2) displays a sample document stored in MongoDB using the defined schema. This document contains real match data, demonstrating how information is stored:

- The match has **status** "RESULT", indicating it has concluded.
- **team1** is "IND" (India), and **team2** is "ENG" (England).
- The **score1** is "356" (India's total runs).
- The **score2** is "(34.2/50 ov, T:357) 214", representing England’s score with overs played and the target.
- The **match_result** states "India won by 142 runs".
- **createdAt** shows a timestamp ("2025-02-13T11:03:05.933Z") indicating when the document was stored.

Chapter 7 Implementation

7.1 Pseudo Code

```
1. Load the trained regression model
   MODEL = load_model("path_to_saved_model")

2. Collect input data (minimal + optional fields)
   INPUT_1 = get_user_input("Field 1")   // e.g., Target Score
   INPUT_2 = get_user_input("Field 2")   // e.g., Current Score
   ...
   INPUT_N = get_user_input("Field N")

3. Calculate derived features if needed
   DERIVED_1 = compute_derived_feature(INPUT_1, INPUT_2)
   ...
   DERIVED_M = compute_derived_feature(...)

4. Create the final feature vector
   FEATURES = [INPUT_1, INPUT_2, ..., DERIVED_1, ..., DERIVED_M]

5. Pass features to model for prediction
   PREDICTION = MODEL.predict(FEATURES)

6. Display or return prediction result
   print("Predicted Output:", PREDICTION)

END
```

7.2 Screen Shot

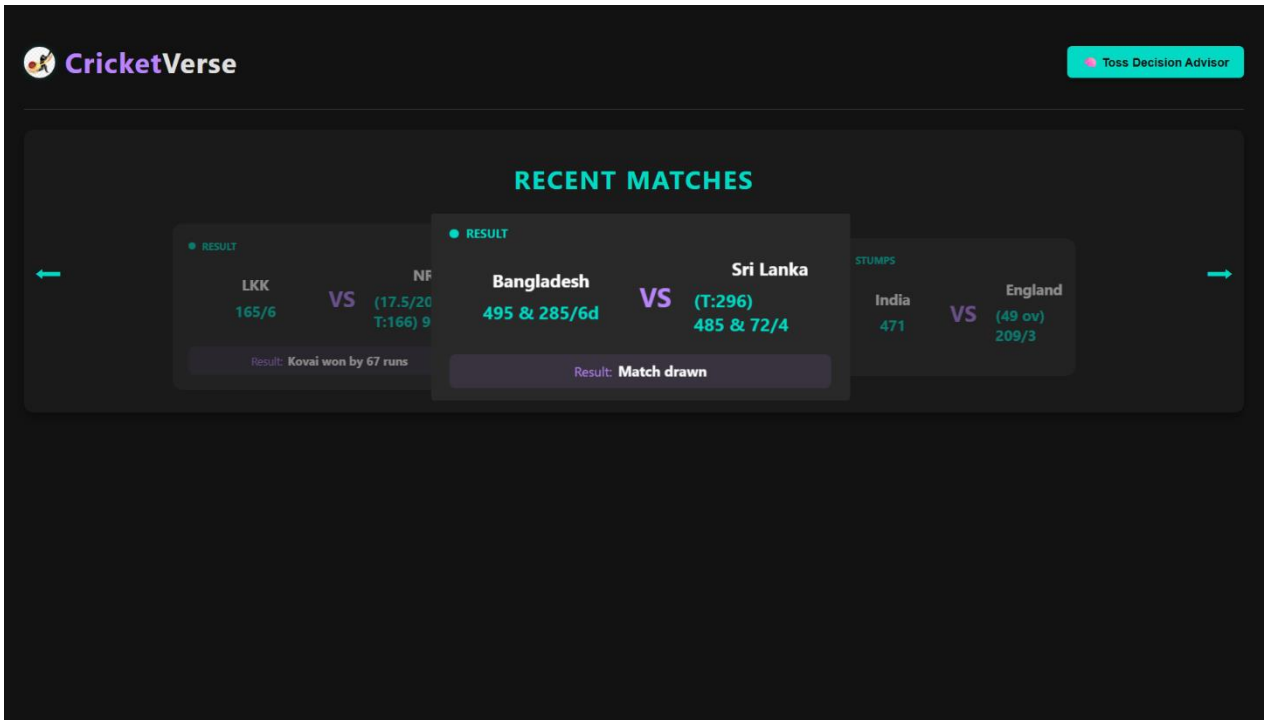

Figure 7.1: Landing Page

The landing page comes up the first time a user searches for CricketVerse. Here, the site offers clear and simple content made using ReactJS. The image is showing the screen of the CricketVerse live score. The user interface is dark with purple and cyan parts. The first thing viewers notice is “CRICKET VERSE”, and then there is a LIVE MATCHES section with the real-time results of ongoing games. Three games are being displayed at the moment, centered mainly on the Queensland vs. The game between New South Wales and another team. One can find the run total, number of wickets, the number of overs, and the target set for the chasing team in match details. The other matches are a bit visible, and moving the arrows allows users to see more games. Under the live match cards, there is a victory probability bar that shows a 65% chance. For now, the bar is stable since the AI-powered model is not in place. The bar shows the possibility that a team could win during the happening of the game.

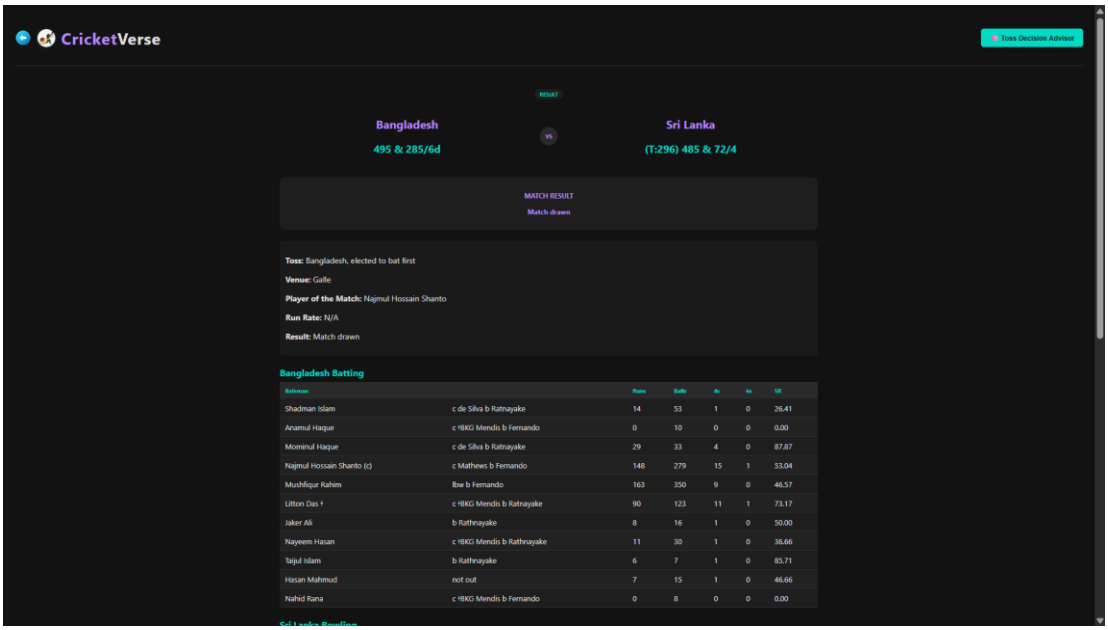

Figure 7.2: Data Set Header

On this page, you can find the scores, statistics and detailed description of a cricket match between Texas Super Kings and MI New York with such important data as the toss (MI New York won the toss and decided to field), venue (Oakland), Player of the Match (Calvin Savage) and the result (Texas Super Kings won by 3 runs). It also displays the final scores, Texas Super Kings 185/6 and MI New York 182/8 in 20 overs as well as a comprehensive batting scorecard of Texas Super Kings, indicating the runs, balls faced, boundaries and strike rates of the individual batsmen. In case of T20 ICC matches, the page would show a prediction of the match, however this is not a T20 ICC match hence it shows, prediction not available. Since the match has been played, on the page the status of the match is shown as "MATCH RESULT" and the winner is obvious.

	Unnamed: 0	Match ID	Date	Venue	Bat First	Bat Second	Innings	Over	Ball	Batter	...	Winner	Chased Successfully	Total Batter Runs	Total Non Striker Runs
0	0	1339605	2023-03-26	SuperSport Park	West Indies	South Africa	1	1	1	BA King	...	South Africa	1	1	0
1	1	1339605	2023-03-26	SuperSport Park	West Indies	South Africa	1	1	2	KR Mayers	...	South Africa	1	1	1
2	2	1339605	2023-03-26	SuperSport Park	West Indies	South Africa	1	1	3	BA King	...	South Africa	1	0	1
3	3	1339605	2023-03-26	SuperSport Park	West Indies	South Africa	1	1	4	J Charles	...	South Africa	1	0	1
4	4	1339605	2023-03-26	SuperSport Park	West Indies	South Africa	1	1	5	J Charles	...	South Africa	1	4	1

5 rows × 35 columns

Figure 7.3: Data Set Header

Figure 7.3 presents the dataset, which indicates what information is included in each column, but 5 Rows are presented not to confuse the analysis. figure 7.5 indicates that the dataset has 35 columns. There were however not enough space to display all the columns, thus a few more are not included in the figure. These columns include the following Non Striker, Bowler, Batter Runs, Extra Runs, Runs From Ball, Ball Rebowled, Extra Type, Wicket Method, Player Out, Innings Runs, Innings Wickets, Target Score, Runs to Get, Balls Remaining, Batter Balls Faced, Non Striker Balls Faced, Player Out Runs.

```
second_inning_data = data[data['Innings'] == 2]
second_inning_data = second_inning_data.drop(columns=['Date'], axis = 1)

# List of teams to keep
teams_to_keep = ["India", "Australia", "England","Pakistan","West Indies","South Africa","Sri Lanka","Bangladesh"]
# Replace with your desired teams

# Filter the dataset
filtered_data = second_inning_data[
    (second_inning_data['Bat First'].isin(teams_to_keep)) &
    (second_inning_data['Bat Second'].isin(teams_to_keep))
]

# Save the filtered dataset
filtered_file_path = 'second_inning_data.csv'
filtered_data.to_csv(filtered_file_path, index=False)

print(f"Filtered dataset saved to: {filtered_file_path}")
```

```
Filtered dataset saved to: second_inning_data.csv
```

Figure 7.4: Data Filter

The Figure 7.4 represents filtration of the preferred international teams to be used in the subsequent procedure. Other teams lack sufficient historical data to operate on since retaining all the teams might interfere with the accuracy of the model hence inaccuracy.

```
# Calculate additional features
second_inning_data['current_run_rate'] = second_inning_data['Innings Runs'] / second_inning_data['Over']
second_inning_data['required_run_rate'] = (second_inning_data['Target Score'] - second_inning_data['Innings Runs']) / (20 - second_inning_data['Over'])
second_inning_data['wickets_remaining'] = 10 - second_inning_data['Innings Wickets']

# Verify the new features
print(second_inning_data[['current_run_rate', 'required_run_rate', 'wickets_remaining']].head())
```

```
     current_run_rate  required_run_rate  wickets_remaining
124               4.0          13.421053                 10
125               8.0          13.210526                 10
126               8.0          13.210526                 10
127               8.0          13.210526                 10
128               9.0          13.157895                 10
```

Figure 7.5: Generating Columns

The Figure 7.5 shows creation of new features that is necessary in the prediction of CricketVerse. With the help of existing columns in the dataset creating new columns in the dataset and storing this new dataset in a different XML file.

```python
from sklearn.preprocessing import LabelEncoder

# Initialize label encoder
label_encoder = LabelEncoder()

# Encode categorical columns
for col in categorical_columns:
    second_inning_data[col] = label_encoder.fit_transform(second_inning_data[col])

# transformed data
print(second_inning_data.head())
```

```
     Unnamed: 0  Match ID  Venue  Bat First  Bat Second  Innings  Over  Ball  \
124         124   1339605    151         94          79        2     1     1
125         125   1339605    151         94          79        2     1     2
126         126   1339605    151         94          79        2     1     3
127         127   1339605    151         94          79        2     1     4
128         128   1339605    151         94          79        2     1     5

     Batter  Non Striker  ...  Total Non Striker Runs  Batter Balls Faced  \
124    1736         1793  ...                       0                   1
```

Figure 7.6: Label Encoding

The Figure 7.6 illustrates label encoding in which the categorical columns gave each category in a column a distinct integer such that algorithm can process them effectively. It is mainly applied when preprocessing categorical features ML models.

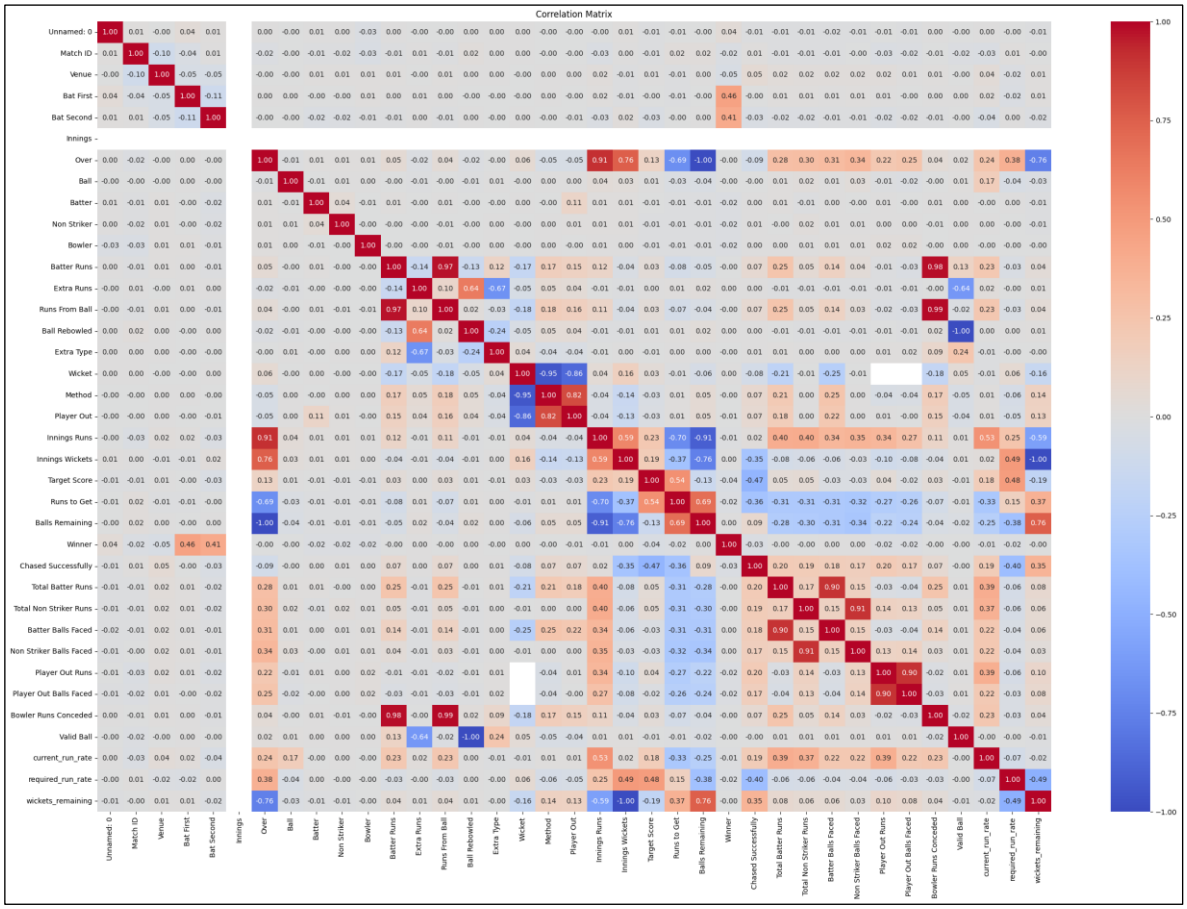

Figure 7.7: Corelation Table

The correlation matrix shown in the Figure 7.7 that gives an in-depth examination of the association between several numerical attributes in the cricket match data. The elements of the matrix are the correlation coefficients between two variables, with the values between -1 and +1. A positive correlation (shown in red shades) means that both the variables increase together and a negative correlation (shown in blue shades) implies that when one variable increases the other variable decreases. Values that are neutral or greyish in color indicate a low or no correlation amongst the variables. Based on the matrix, there are few important insights to note. Positive correlations with a strong value indicate that the more runs in an innings, the better the performances of individual batters ("Innings Runs" and "Total Batter Runs" 0.91).

On the same note, there is a very negative correlation between Balls Remaining and Runs to Get (-0.79), which basically indicates that the more the balls remaining, the more the runs are likely to be, especially at the start of an inning. One more valuable insight is the correlation between the Current Run Rate and Valid Ball (0.59) that indicates that the high amount of valid

deliveries influences the scoring rate in a positive way. Also among significant negative correlations there are

Wickets Remaining and Target Score (-0.76), meaning that the higher the target score is, the lesser number of wickets might be remaining, which symbolizes the batting side pressure.

The negative correlation between the number of runs to get and the current run rate (-0.69) means that when the required run rate is high, then the batting team is likely to be in trouble. Variables with moderate correlation, like, Winner and Chased Successfully (0.46), show that there is a plausible influence of successful chase on the outcome of matches. Likewise, the number of "Total Batter Runs" and the number of "Batter Balls Faced" (0.28) incline us to think that the more balls a person faces the higher his or her individual score is, but not necessarily with a very strong connection.

Correlation matrix is important in selecting features in predictive modelling. It assists in determining highly correlated variables, which could cause redundancy and multi collinearity in the machine learning models. Knowledge of these correlations can be used to perform feature engineering in a better manner; making sure that only highly relevant variables are considered to predict match outcomes. Particularly, the Current Run Rate and Required Run Rate parameters show considerable correlations with the elements of winning a match, thus being highly eligible to be used in predictive analysis in cricket analytics.

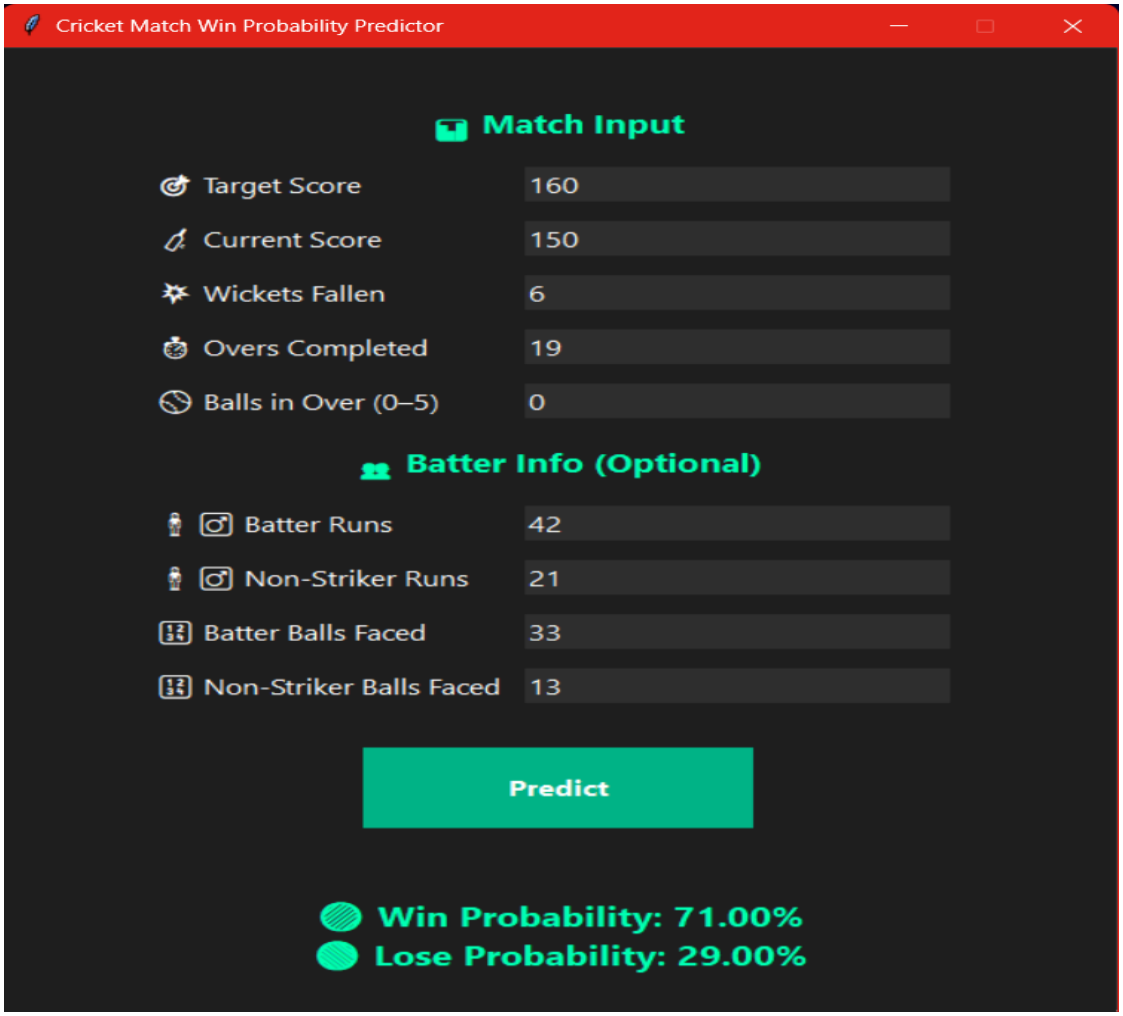

Figure 7.8: Prediction

This picture illustrates the interface of and the output of a Cricket Match Win Probability Predictor which is a tool that predicts the probability of a team winning or losing a live cricket match given the real-time data of the match. Input Features: As input variables, the predictor takes into consideration a few match and batter-specific features that are essential to define the win probability. These characteristics are: Target Score: The number of runs that have to be scored to win the match (in this case, 160). Current Score: The runs conceded by the batting team(in this case, 150). Wickets Fallen the number of wickets that have been lost (in this case 6). Overs Bowled: The number of overs that have been bowled (in this case 19). Balls in Over (05): Extra balls delivered in the over (in this case 0, since it is a new over). Batter Runs: Runs that have been scored by the striker at the moment (in this case 42). Non-Striker Runs: The runs that are scored by the non-striker (in this case 21). Batter Balls Faced: Number of balls which the striker has faced (in this case, 33). Non-Striker Balls Faced: Balls that have been faced by the non striker (in this case 13). These characteristics can be matched to the following variables in the machine learning model: required_run_rate Target Score Runs to Get current_run_rate Innings Wickets wickets_remaining Innings Runs Total Non Striker Runs Total Batter Runs Non Striker Balls Faced Balls Remaining Batter Balls Faced

Output: Once the match situation and the batter details are provided, the predictor computes and shows: Win Probability: The probability (in percentage) of the batting team to win the match (in this case, 71.00%). Lose Probability: The probability (in percentage) of the batting team to lose the match (in this case 29.00%). Such probabilities are useful to know the situation of the match in progress and the teams, analytical agents and fans can take advantage of such live data to assess the likelihood of winning.

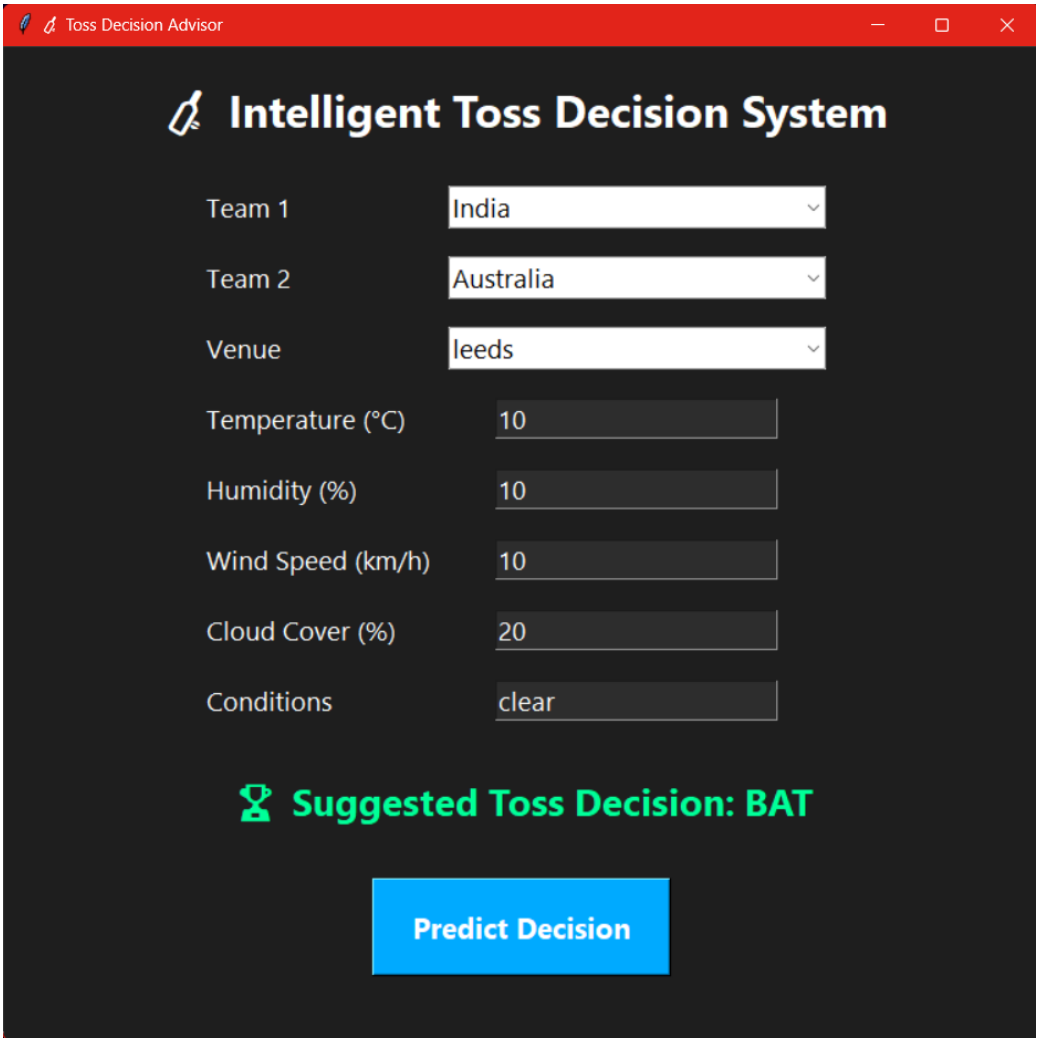

Figure 7.9: Toss Decision Recommendation

The picture represents the graphical user interface (GUI) of Intelligent Toss Decision System. It assists the users in making informed decision based on data at the toss of a cricket match. The interface requests important information which can influence a toss decision, i.e.:

• Team 1 and Team 2: International teams (e.g. India vs Australia).

• Venue: Where the match is going to take place (ex: Leeds).

Parameters of weather:

temperature (o C)

Humidity (%)

Wind Speed (km/h)-

Mean-cloud cover-percentage (%)

e.g. clear, overcast conditions.

Once these inputs have been made, the user presses on the Predict Decision button activating the background machine learning model that has been learnt using historical match and weather data. The system will then display a Suggested Toss Decision (BAT) or (BOWL) basing on both environmental and team factors.

In the case of using the system that India vs Australia match which took place at Leeds in mild and clear weather system, the system predicts that India should bat first. Such a decision is rounded off with the green colour to make it more noticeable.

The tool is developed user-friendly and readable. Choices are made through dropdown menu and numeric values are made through text fields. Large icons and high-contrast design can turn both desktop computers and the touch-based devices into easy-to-read and professional screen.

Chapter 8 Software Testing

8.1 Manual Testing Cases

Table 8.1: Second Inning Prediction

Test Scenario	**Second Inning Prediction**			
Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail/ Non-Executed/ Suspended
REG_01	First innings	Prediction not available	Prediction not available	Pass
REG_02	Invalid over (greater than 20)	Error or validation message shown	Error message	Pass
REG_03	All out before full overs	Wickets remaining = 0, model predict 0% win	Wickets remaining = 0 model predict win	Pass

Table 8.2: Live Match Scores

Test Scenario	**Live Match Scores**			
Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail/Non-Executed/ Suspended
REG_01	Click on Specific Match Card for Detail Information	Verify match details page loads properly	Match details page loads properly	Pass
REG_02	Layout on different screen sizes	Responsive layout, fields don't break	Responsive layout, fields don't break	Pass
REG_03	UI consistency	All UI elements follow consistent style guide	All UI elements following consistent style guide	Pass

Table 8.3: Match Details Page

Test Scenario	**Match Details Page**			
Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail/ Non-Executed/ Suspended
REG_01	Click on Specific Match Card for Detail Information	Verify match details page loads properly	Match details page loads properly	Pass
REG_02	Layout on different screen sizes	Responsive layout, fields don't break	Responsive layout, fields don't break	Pass
REG_03	UI consistency	All UI elements follow consistent style guide	All UI elements following consistent style guide	Pass

Table 8.4: Toss Decision Recommendation

Test Scenario	**Toss decision Recommendation**			
Test Case ID	Step Details	Expected Result	Actual Result	Pass/Fail/ Non-Executed/ Suspended
REG_01	Submit form with valid data for top teams (India vs Australia)	Toss decision is shown ("BAT" or "BOWL")	As expected	Pass
REG_02	API is down or unreachable	Show fallback error like “Server error: unable to compute decision.”	To be tested	Non-Executed
REG_03	UI consistency	All UI elements follow consistent style guide	All UI elements following consistent style guide	Pass

Chapter 9 Conclusion

The project Second Innings Prediction efficiently shows how machine learning methods can be applied to predict the probability of a team to win a T20 cricket match in the second innings. With the historic ball-by-ball data we were able to train and engineer features of the key attributes of a match including current score, target score, balls remaining, wickets lost, run rates etc. which enabled us to build a predictive model that could deliver probabilistic information in real-time.

It used a Random Forest Classifier to produce win/lose probabilities, which would give interpretable and trustworthy results. To make the tool user friendly to analysts, commentators and cricket lovers, the project was also improved with an intelligent command line interface and graphical user interface to eliminate input burden through automatic calculations of derived features.

Chapter 10 Future Enhancement

- **First Inning Prediction**

  What the present model is limited to is prediction of outcome only at the second innings. Among some of the improvements will be the development of a regression model that will predict the score in the fall of the first innings based on real time match features such run rate, wickets in hand, current match and form of the concerned players.

- **Support for all Formats**

  The model is of T20 format. Future releases will include support of ODIs (50 overs) and Test matches which go in different pace of the game, strategy and uncertain length of the match in each of the forms.

- **Contextual Feature Expansion**

  Extra details including pitch condition, weather, venue statistics, team powers and head-to-head statistics will be added to the model so that it is closer and sensible to the realities.

- **Mobile and Web Deployment**

  The system will be deployed not only as a responsive web application but also as a mobile application, which will enable people to interact with predictions even when they live elsewhere (on a live match).

- **Admin Dashboard and Data Management**

  A good administration interface will be integrated so that the information on a match can be administered, models updated and the performance of the prediction tracked. This will make it easy to maintain scale and mode.

Appendix A BIBLIOGRAPHY

[1] Manoj Ishi, Dr. Jayantrao Patil et al.” Winner Prediction in One-Day International Cricket Matches Using Machine Learning Framework: An Ensemble Approach” | Indian Journal of Computer Science and Engineering (2022)

[2] Srikantaiah K C, Aryan Khetan et al. “Prediction of IPL Match Outcome Using Machine Learning Techniques” | 3rd International Conference on Integrated Intelligent Computing, Communication & Security (2019)

[3] Param Dalal, Hirak Shah, Tej Kanjariya, Dhananjay Joshi. “Cricket Match Analytics and Prediction Using Machine Learning” | International Journal of Computer Applications (2024)

[4] Pritpal Singh, Dr. Jatinder Kaur, Lovedeep Singh “Predicting IPL Victories: An Ensemble Modelling Approach Using Comprehensive Dataset Analysis” | 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (2024)

[5] Jalaz Kumar, Rajeev Kumar, Pushpender Kumar “Outcome Prediction of ODI Cricket Matches Using Decision Trees and MLP Networks” | First International Conference on Secure Cyber Computing and Communication (2018)