Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Summer, Year: 2025), B.Sc. in CSE (Day)

T20I Predictor

Course Title: Machine Learning Lab
Course Code: CSE 412
Section: CSE 221 - D5

Students Details

Name	**ID**
Md. Kawsar Ahamed	213902013

Submission Date: 26 August 2025
Course Teacher's Name: Ms. Umme Habiba

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The "T20I Predictor" is a sophisticated web application developed using the Django framework in Python, designed to forecast the outcomes of T20 International cricket matches. At its core, the project leverages a dual-pronged analytical approach, combining deep statistical analysis of historical match data with the predictive power of a suite of machine learning models. It moves beyond simple win/loss predictions by providing users with a dynamic and interactive platform where various match parameters—such as the venue, the team batting first, the first innings total, and even the specific playing XI—can be selected. The system processes these inputs to generate a detailed, multi-faceted analysis, culminating in a final, intelligent win probability for either team. This project stands as a practical application of data science principles to the exciting and unpredictable world of T20 cricket.

1.2 Motivation

The motivation for this project stems from the explosive global popularity of T20 cricket and the concurrent rise of sports analytics. Fans, analysts, and fantasy sports enthusiasts are no longer satisfied with simple intuition; there is a growing demand for data-driven insights to understand the nuances of the game. T20 cricket, known for its fast pace and volatility, is particularly ripe for such analysis, as a single over can dramatically alter a match's outcome. This project aims to satisfy that analytical curiosity by creating a tool that can quantify the impact of various factors on a match's result. By translating complex historical data into an accessible and engaging format, the T20I Predictor seeks to deepen the spectator's understanding and enhance their overall viewing experience, making every match more insightful.

1.3 Problem Definition

Predicting the outcome of a T20 International cricket match is an inherently complex challenge. The format's brevity and aggressive nature mean that matches are often decided by fine margins and can be highly unpredictable. A multitude of dynamic variables, including current team form, player-vs-player matchups, specific venue conditions, the critical outcome of the toss, and the psychological pressure of chasing a target, all interact to influence the final result. Traditional prediction methods often rely on simplistic metrics (like team rankings) or are subject to emotional bias, failing to capture the intricate interplay of these factors. This creates a clear problem: the absence of a comprehensive tool that can systematically process these diverse variables and provide a holistic, data-driven forecast. The T20I Predictor is designed to solve this problem by structuring this complexity into a cohesive and logical analytical framework.

1.4 Objectives

To address the defined problem, this project was undertaken with a clear set of primary objectives. The foremost goal was to engineer a robust system capable of processing and structuring raw, unstructured cricket data from CSV files into a clean, relational database, thereby creating a solid foundation for all subsequent analysis. The second objective was to design and implement a multi-layered statistical analysis engine to query this database and evaluate a wide range of performance metrics, including head-to-head records, venue-specific performance, and innings-based dynamics. Concurrently, the third objective was to integrate a diverse suite of machine learning models to provide independent, probabilistic forecasts based on historical patterns. The fourth objective was to develop an intuitive and interactive web interface that allows users to dynamically apply filters and view the complex analytical results in a clear and understandable format. Furthermore, a key technical objective was to build the application in a modular fashion, ensuring that new analytical factors or machine learning models could be easily added in the future. A significant performance goal was to achieve a reliable level of predictive accuracy by carefully tuning the weights assigned to each statistical factor. Finally, the ultimate objective was to synthesize the outputs of both the statistical and machine learning engines through a weighted priority system, producing a single, intelligent, and transparent final prediction that shows users not just what the outcome might be, but why.

1.5 Application

The T20I Predictor is not merely a theoretical model but a functional web application with a suite of features designed to provide a comprehensive and interactive user experience. The application's core functionalities are built to empower users with the ability to conduct their own nuanced match analyses by dynamically adjusting various parameters. The following subsections detail the key features implemented in the project.

1.5.1 Dynamic Match and Venue Selection

The primary interface of the application allows users to select the two competing international teams from a comprehensive list populated directly from the project's database. This feature includes a search and autocomplete function for user convenience. Additionally, users can optionally select a specific match venue from a standardized list, which has been cleaned to merge duplicate entries (e.g., "Shere Bangla National Stadium, Mirpur" and "Shere Bangla National Stadium" are treated as one). This initial selection forms the baseline for all subsequent predictive analyses.

1.5.2 Multi-Layered Statistical Analysis

Once the teams are selected, the application automatically generates a rich dashboard of statistical comparisons. This is not a single data point but a collection of distinct analytical factors, each providing a different perspective on the matchup. These factors include all-time head-to-head records, overall win ratios against all opponents, recent team form based on the last ten matches, and recent head-to-head form from the last five encounters between the selected teams. Each factor is presented with a clear "advantage" indicator, allowing users to quickly assess the historical context of the game.

1.5.3 Machine Learning Integration

A core feature of the application is the integration of a diverse suite of seven machine learning models, including Random Forest, Support Vector Machine (SVM), and Gradient Boosting. When a venue is selected, each model provides an independent, probabilistic forecast of the match winner. The prediction logic is specifically constrained to ensure the outcome is always one of the two competing teams. This feature allows users to compare the statistical analysis with purely algorithmic pattern recognition, offering a more robust and well-rounded view of the potential match outcome.

1.5.4 Contextual Filtering System

The application allows users to go beyond the baseline team comparison by applying a series of optional, contextual filters. Users can specify which team will bat first and, subsequently, input a hypothetical first innings total. The system is designed to be intelligent; for instance, the "First Innings Total" field is only enabled after a batting team has been selected. Applying these filters dynamically triggers new, more specific analyses, such as a team's historical performance when batting first or its success rate when chasing a target within a specific score range.

1.5.5 Weighted Prediction Engine

The most sophisticated feature of the T20I Predictor is its weighted scoring system. The application's "brain" takes the qualitative "advantage" from every statistical factor and assigns it a quantitative score based on a predefined priority system. For example, the "Recent Head-to-Head" performance is given a higher weight (importance) than the "All-Time Win Ratio." The system then aggregates these scores and applies a smoothing algorithm to prevent unrealistic 100% predictions, culminating in a final, nuanced win probability that reflects the weighted importance of all available data.

1.5.6 Interactive Playing XI Selection

For the most detailed level of analysis, the application features an interactive Playing XI selector. Users can select exactly eleven players for each team from a list that is dynamically populated based on the selected nation. This feature is linked to the database to ensure that users can only select players who have actually played for that specific country. Once full squads are selected, the system calculates a "power score" for each team based on the historical performance metrics of the individual players, adding a final, granular layer to the overall prediction.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

This chapter details the architectural design and technical implementation of the T20I Predictor. The project was developed following a modular design philosophy to ensure scalability, maintainability, and a clear separation of concerns. The development process was divided into distinct phases: first, establishing a robust data foundation through database modeling and data importation; second, building a multi-layered statistical analysis engine; third, creating an independent machine learning engine for probabilistic forecasting; and finally, integrating these components into a cohesive Django web application. This chapter will elaborate on the project's overall architecture, the implementation of its core components, and the specific tools and libraries that were instrumental in its construction.

2.2 Project Details

The T20I Predictor is architected as a classic three-tier web application. The Data Tier is managed by a SQLite3 database, orchestrated by Django's Object-Relational Mapper (ORM), which provides a structured and efficient way to store and retrieve all T20I match, player, and venue data. The Logic Tier, which forms the core of the application, is built entirely in Python within the Django framework. This tier is further subdivided into distinct modules for data analysis, machine learning, and final prediction scoring, ensuring that each component handles a specific task. The Presentation Tier is rendered through Django's templating engine, delivering an interactive user interface built with HTML, Bootstrap 5 for styling, and vanilla JavaScript for dynamic, client-side functionality. This architecture ensures that the data, logic, and presentation layers are decoupled, making the system easier to debug, maintain, and upgrade.

2.2.1 User Interface

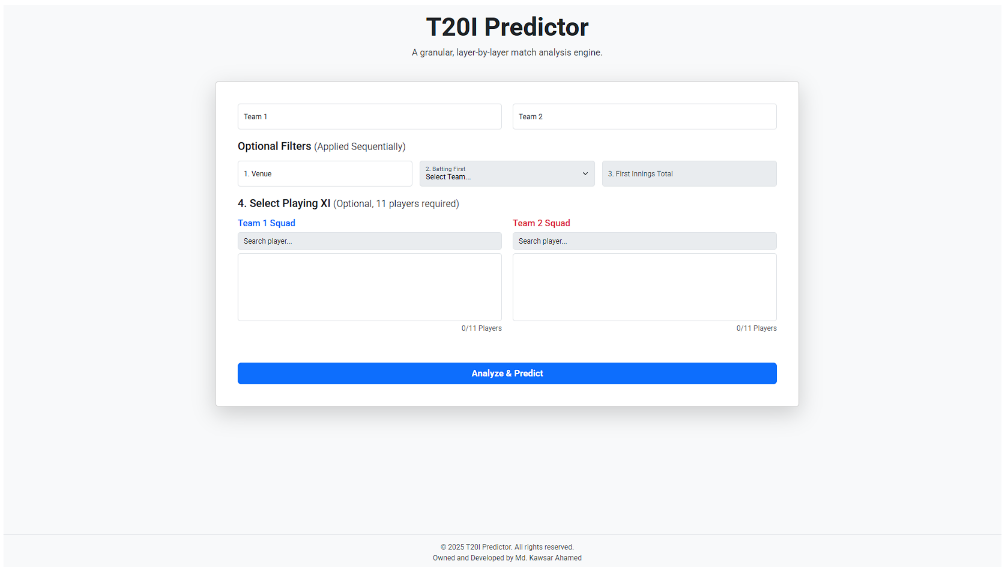

T20I Predictor
A granular, layer-by-layer match analysis engine.
Team 1
Team 2
Optional Filters (Applied Sequentially)
1. Venue
2. Batting First
Select Team...
3. First Innings Total
4. Select Playing XI (Optional, 11 players required)
Team 1 Squad
Team 2 Squad
Search player...
Search player...
0/11 Players
0/11 Players
Analyze & Predict
© 2025 T20I Predictor. All rights reserved.
Owned and Developed by Md. Kawsar Ahamed

Figure 2.1: User Interface

2.2.2 Dataset Description

The foundation of this project is the **"Men's T20I Cricket Complete Dataset,"** a comprehensive collection of T20 International match data sourced from the Kaggle platform. This dataset spans a wide historical range from the inception of T20Is in 2005 up to the near present, providing a rich repository of information for both statistical analysis and machine learning. The data is logically structured into two primary CSV files, `matchwise_data.csv` and `deliverywise_data.csv`, which work in tandem to provide a complete picture of every match. A crucial step in the implementation phase was the preprocessing of this raw data to filter out all non-international franchise matches, ensuring that the final dataset used for analysis was exclusively composed of official T20I fixtures.

The `matchwise_data.csv` file serves as a high-level summary, with each row representing a single completed match. It contains all the essential top-line information needed to understand the context and outcome of a game. Key columns in this file include:

- `match_id` (Integer): A unique numerical identifier for each match, used as a primary key to link data between the two files.
- `date` (Date/String): The date on which the match was played, used for chronological analysis and determining recent form.
- `team_1` & `team_2` (String): The names of the two competing national teams.
- `ground_name` (String): The name of the venue where the match was held.

- `toss_winner` (String): The name of the team that won the coin toss.
- `toss_decision` (String): The choice made by the toss-winning captain, typically 'bat' or 'field'.
- `winner` (String): The name of the team that won the match.

Complementing the match summaries, the `deliverywise_data.csv` file provides the granular, ball-by-ball details for every match. Each row in this file represents a single delivery bowled. This highly detailed data is essential for in-depth player performance analysis and for calculating first innings totals. Key columns in this file include:

- `match_id` (Integer): The foreign key linking each delivery back to its corresponding match in the `matchwise_data.csv` file.
- `innings_number` (Integer): Indicates whether the delivery was bowled in the first or second innings (1 or 2).
- `over_number` & `ball_number` (Integer): Specify the exact point in the match when the delivery occurred.
- `batter` & `bowler` (String): The names of the batsman facing and the bowler delivering the ball.
- `batsman_runs` (Integer): The number of runs scored off the bat on that delivery.
- `extra_runs` (Integer): The number of runs scored as extras (e.g., wides, no-balls).
- `player_dismissed` (String): The name of the batsman who got out on that delivery, if applicable.

2.2.3 Machine Learning Models

To provide a robust and multi-faceted forecast, the T20I Predictor implements a suite of seven distinct machine learning models. This multi-model approach is a core design feature, chosen to mitigate the inherent biases of any single algorithm and to provide a consensus-based prediction, which is crucial given the high volatility of T20 cricket. All models were trained on the same filtered T20I dataset using three primary features: `team_1`, `team_2`, and `ground_name`. The following subsections provide a brief overview of each implemented model.

Random Forest

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time. For this project, it builds numerous simple decision trees, each trained on a different random subset of the historical match data. To make a final prediction, it aggregates the "votes" from all the individual trees and outputs the class (the winning team) that receives the majority of votes. This approach makes the model highly accurate and robust against overfitting.

Support Vector Machine (SVM)

Support Vector Machine is a powerful classification algorithm that works by finding the optimal hyperplane that best separates the data points of different classes in a high-dimensional space. In the context of this project, it seeks to find the clearest possible dividing line that distinguishes between wins for Team 1 and wins for Team 2 based on the input features. The goal is to maximize the margin between these outcomes, leading to a more confident classification.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a simple, instance-based learning algorithm. It makes predictions based on the principle of similarity. When asked to predict a new match, it searches the entire historical dataset for the 'K' most similar matches (its "neighbors"). The prediction for the new match is then determined by the majority outcome of these neighboring matches. It is a non-parametric model that is highly intuitive and effective for datasets with clear groupings.

Gradient Boosting

Gradient Boosting is another highly effective ensemble technique that builds models in a sequential, stage-wise fashion. It works by constructing a series of decision trees, where each new tree is trained to correct the errors made by the previous one. This iterative process of minimizing errors allows the model to learn complex patterns in the data, often resulting in very high predictive accuracy.

Logistic Regression

Despite its name, **Logistic Regression** is a fundamental statistical model used for binary classification. It works by calculating the probability of a specific outcome (e.g., Team A winning) occurring. It fits the data to a logistic (sigmoid) function, which outputs a probability value between 0 and 1. This model is highly interpretable and provides a strong baseline for evaluating the performance of more complex models.

Decision Tree

The **Decision Tree** is the foundational building block for more advanced models like Random Forest. It creates a tree-like structure of decisions based on the input features. Starting from the root, the model splits the data based on the feature that provides the most information gain (e.g., a specific venue). Each branch represents a decision, and each leaf node represents a final prediction, making the model's decision-making process very easy to visualize and understand.

Gaussian Naive Bayes

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It operates under the "naive" assumption that all input features (`team_1`, `team_2`, `ground_name`) are independent of one another. It calculates the probability of a team winning given the specific combination of features in the match, assuming a Gaussian (normal) distribution for the data. It is a computationally efficient and effective model, especially for baseline predictions.

2.3 Implementation

The implementation phase began with the creation of the database schema in predictor/models.py, where tables for Team, Player, Venue, Match, and Delivery were defined. A crucial part of this stage was developing a custom management command, import_t20i_data.py, to perform a one-time ETL (Extract, Transform, Load) process. This script extracts data from the provided CSV files, transforms it by filtering for T20I matches and standardizing venue names, and loads it into the newly created database. Following this, the statistical analysis modules were built to perform complex database queries. Concurrently, the machine learning engine was developed; a trainer.py script was implemented to prepare the data, train all seven models, and serialize them to disk using joblib. The final step was the creation of the views.py file, which acts as the central orchestrator, tying together user requests from the frontend with the backend analysis and ML modules to generate and return the final prediction.

2.3.1 GitHub Link

https://github.com/KaSagor/T20I-Predictor

2.4 Tools and Libraries

The development of this project was facilitated by a carefully selected stack of industry-standard tools and robust open-source libraries. The primary integrated development environment (IDE) used for all coding and debugging was **Visual Studio Code**. The entire application is built upon the **Django** framework, a high-level Python web framework chosen for its scalability and Model-View-Template architecture. For the data persistence layer, the project utilizes the lightweight, serverless **SQLite3** database engine, which is seamlessly integrated with Django's Object-Relational Mapper (ORM). All data manipulation, cleaning, and transformation processes were handled using the powerful **Pandas** library. The machine learning engine was constructed entirely with the **Scikit-learn** library, which provided the implementations for all seven predictive models. Finally, the **Joblib** library was used for efficiently serializing and deserializing the trained machine learning models to and from disk.

Chapter 3

Performance Evaluation

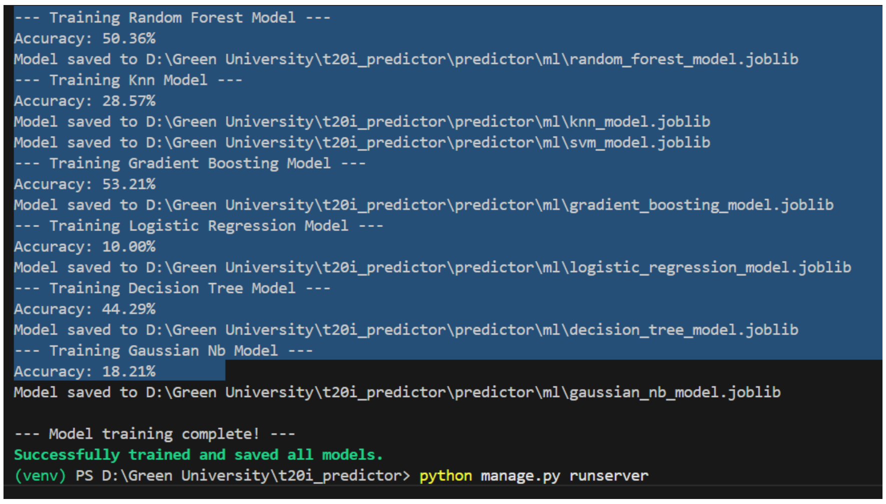
3.1 Simulation Environment/ Simulation Procedure

The development and performance evaluation of the T20I Predictor were conducted in a standardized simulation environment to ensure consistency and reproducibility. The project was built on a machine running the Windows 11 operating system, utilizing Visual Studio Code as the primary integrated development environment (IDE). The core of the application is powered by Python 3.12, with all dependencies and packages, including Django, pandas, and scikit-learn, managed within an isolated virtual environment (venv) to prevent conflicts with system-level libraries. The simulation procedure begins with two crucial, one-time setup commands: python manage.py import_t20i_data to populate the database from the source CSV files, and python manage.py train_ml_models to train and serialize all seven machine learning models. For live predictions, the backend is run locally using Django's built-in development server. User interactions on the frontend trigger an asynchronous JavaScript (AJAX) request, which sends the selected match parameters to the Django backend. The backend then processes this request, performs all necessary database queries and model inferences in real-time, and returns a structured JSON object containing the complete analysis back to the client for evaluation and rendering.

3.2 Results Analysis/Testing

3.2.1 Result_portion_1

When train ML models and check accuracy.



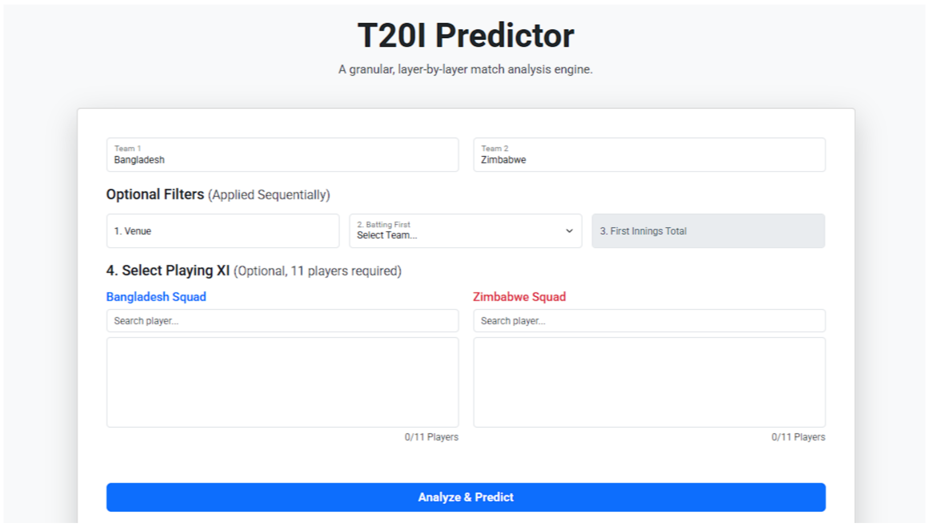
```
--- Training Random Forest Model ---
Accuracy: 50.36%
Model saved to D:\Green University\t20i_predictor\predictor\ml\random_forest_model.joblib
--- Training Knn Model ---
Accuracy: 28.57%
Model saved to D:\Green University\t20i_predictor\predictor\ml\knn_model.joblib
Model saved to D:\Green University\t20i_predictor\predictor\ml\svm_model.joblib
--- Training Gradient Boosting Model ---
Accuracy: 53.21%
Model saved to D:\Green University\t20i_predictor\predictor\ml\gradient_boosting_model.joblib
--- Training Logistic Regression Model ---
Accuracy: 10.00%
Model saved to D:\Green University\t20i_predictor\predictor\ml\logistic_regression_model.joblib
--- Training Decision Tree Model ---
Accuracy: 44.29%
Model saved to D:\Green University\t20i_predictor\predictor\ml\decision_tree_model.joblib
--- Training Gaussian Nb Model ---
Accuracy: 18.21%
Model saved to D:\Green University\t20i_predictor\predictor\ml\gaussian_nb_model.joblib

--- Model training complete! ---
Successfully trained and saved all models.
(venv) PS D:\Green University\t20i_predictor> python manage.py runserver
```

Figure 3.1: Models accuracy

3.2.2 Result_portion_2

When the user input Team 1 as Bangladesh and Team 2 as Zimbabwe.



T20I Predictor

A granular, layer-by-layer match analysis engine.

Team 1: Bangladesh

Team 2: Zimbabwe

Optional Filters (Applied Sequentially)

1. Venue

2. Batting First: Select Team...

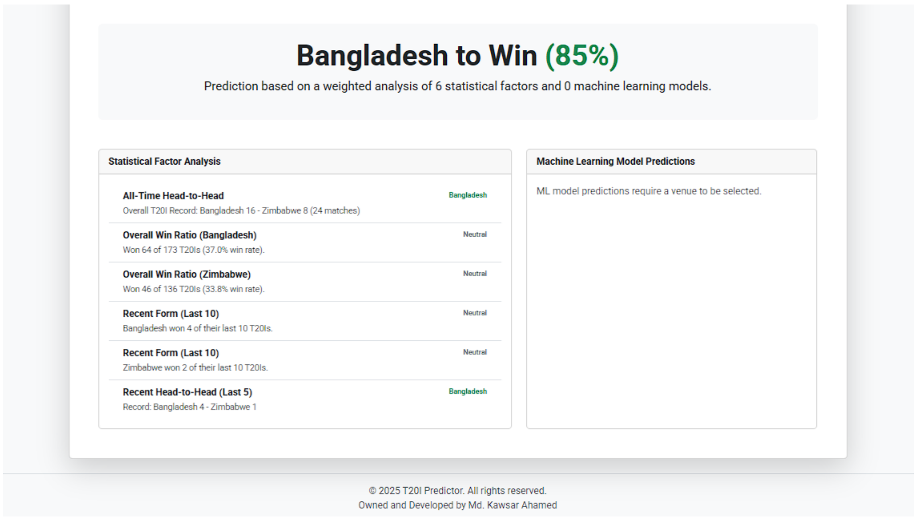
3. First Innings Total

4. Select Playing XI (Optional, 11 players required)

Figure 3.2: Bangladesh vs Zimbabwe

3.2.3 Result_portion_3

The result after the user input Team 1 as Bangladesh and Team 2 as Zimbabwe.



Bangladesh to Win (85%)

Prediction based on a weighted analysis of 6 statistical factors and 0 machine learning models.

Statistical Factor Analysis

All-Time Head-to-Head Overall T20I Record: Bangladesh 16 - Zimbabwe 8 (24 matches)	Bangladesh
Overall Win Ratio (Bangladesh) Won 64 of 173 T20Is (37.0% win rate).	Neutral
Overall Win Ratio (Zimbabwe) Won 46 of 136 T20Is (33.8% win rate).	Neutral
Recent Form (Last 10) Bangladesh won 4 of their last 10 T20Is.	Neutral
Recent Form (Last 10) Zimbabwe won 2 of their last 10 T20Is.	Neutral
Recent Head-to-Head (Last 5) Record: Bangladesh 4 - Zimbabwe 1	Bangladesh

Machine Learning Model Predictions

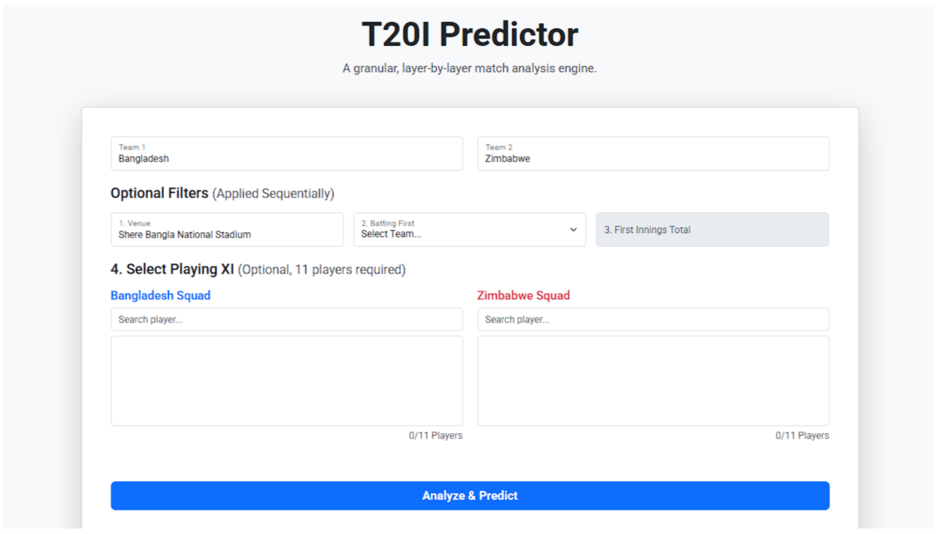
ML model predictions require a venue to be selected.

© 2025 T20I Predictor. All rights reserved.
Owned and Developed by Md. Kawsar Ahamed

Figure 3.3: Bangladesh vs Zimbabwe result

3.2.4 Result_portion_4

When the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium.



T20I Predictor

A granular, layer-by-layer match analysis engine.

Team 1: Bangladesh
Team 2: Zimbabwe

Optional Filters (Applied Sequentially)

1. Venue: Shere Bangla National Stadium
2. Batting First: Select Team...
3. First Innings Total

4. Select Playing XI (Optional, 11 players required)

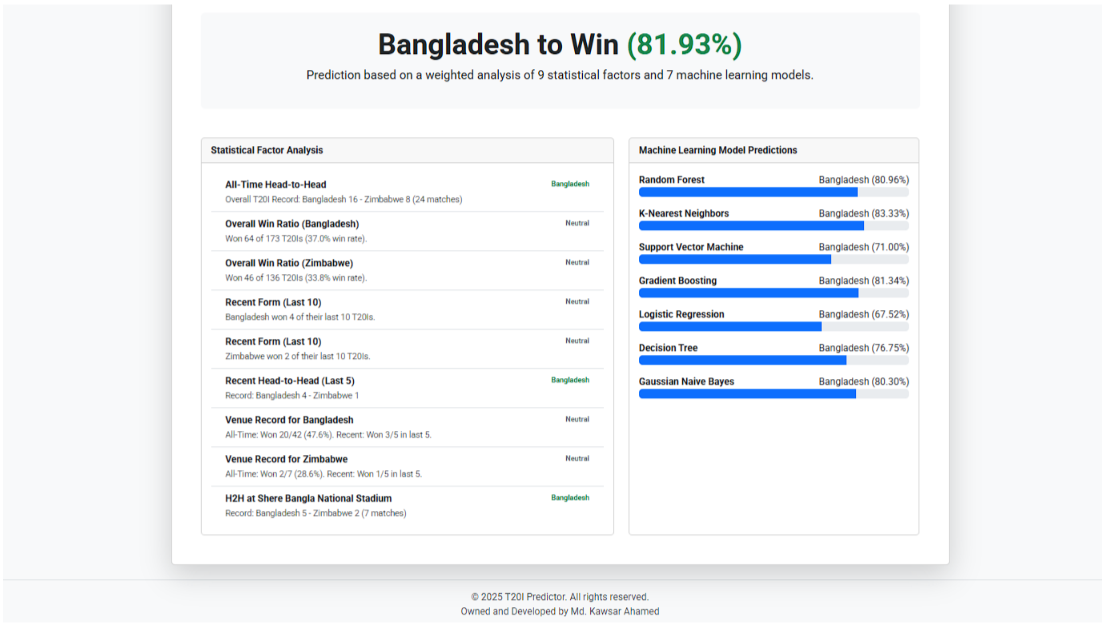
Bangladesh Squad — Search player... — 0/11 Players

Zimbabwe Squad — Search player... — 0/11 Players

Figure 3.4: Bangladesh vs Zimbabwe at SBNCS

3.2.5 Result_portion_5

The result after the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium.



Bangladesh to Win (81.93%)

Prediction based on a weighted analysis of 9 statistical factors and 7 machine learning models.

Statistical Factor Analysis

All-Time Head-to-Head — Bangladesh
Overall T20I Record: Bangladesh 16 - Zimbabwe 8 (24 matches)

Overall Win Ratio (Bangladesh) — Neutral
Won 64 of 173 T20Is (37.0% win rate).

Overall Win Ratio (Zimbabwe) — Neutral
Won 46 of 136 T20Is (33.8% win rate).

Recent Form (Last 10) — Neutral
Bangladesh won 4 of their last 10 T20Is.

Recent Form (Last 10) — Neutral
Zimbabwe won 2 of their last 10 T20Is.

Recent Head-to-Head (Last 5) — Bangladesh
Record: Bangladesh 4 - Zimbabwe 1

Venue Record for Bangladesh — Neutral
All-Time: Won 20/42 (47.6%). Recent: Won 3/5 in last 5.

Venue Record for Zimbabwe — Neutral
All-Time: Won 2/7 (28.6%). Recent: Won 1/5 in last 5.

H2H at Shere Bangla National Stadium — Bangladesh
Record: Bangladesh 5 - Zimbabwe 2 (7 matches)

Machine Learning Model Predictions

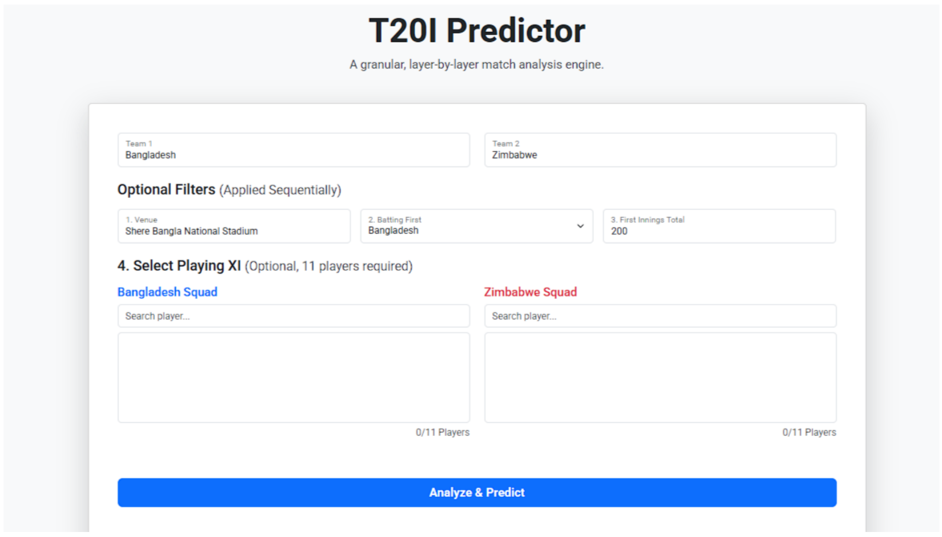
Random Forest	Bangladesh (80.96%)
K-Nearest Neighbors	Bangladesh (83.33%)
Support Vector Machine	Bangladesh (71.00%)
Gradient Boosting	Bangladesh (81.34%)
Logistic Regression	Bangladesh (67.52%)
Decision Tree	Bangladesh (76.75%)
Gaussian Naive Bayes	Bangladesh (80.30%)

© 2025 T20I Predictor. All rights reserved.
Owned and Developed by Md. Kawsar Ahamed

Figure 3.5: Bangladesh vs Zimbabwe result at SBNCS

3.2.6 Result_portion_6

When the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium and Bangladesh scored 200 runs.



T20I Predictor

A granular, layer-by-layer match analysis engine.

Team 1: Bangladesh
Team 2: Zimbabwe

Optional Filters (Applied Sequentially)

1. Venue: Shere Bangla National Stadium
2. Batting First: Bangladesh
3. First Innings Total: 200

4. Select Playing XI (Optional, 11 players required)

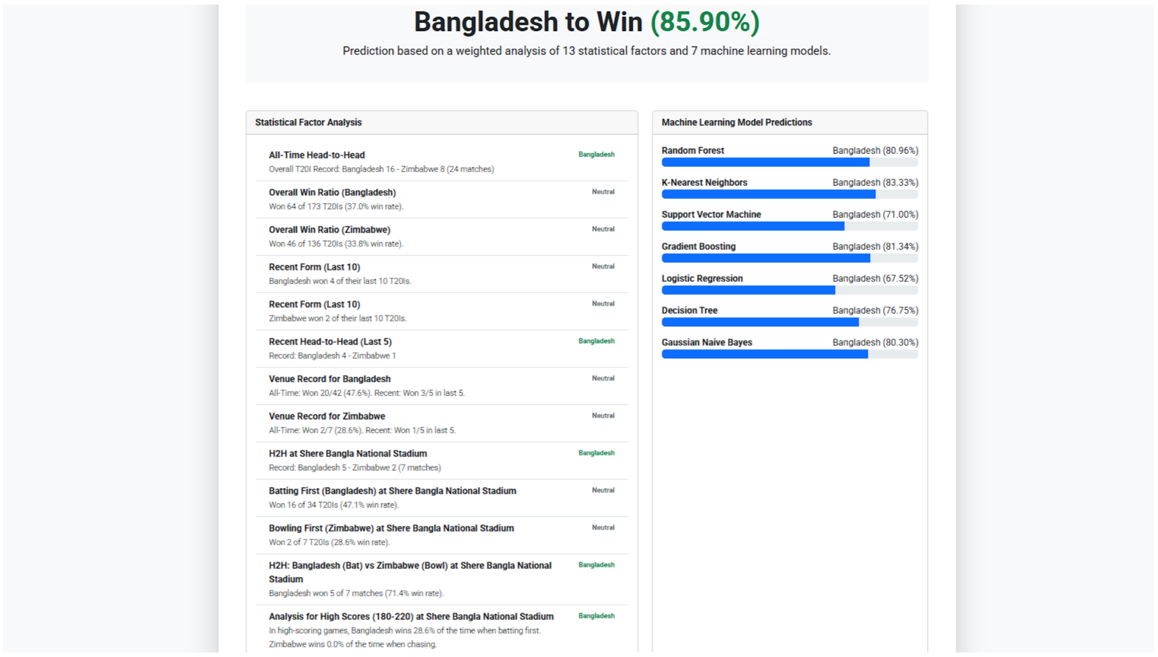
Bangladesh Squad — Search player... — 0/11 Players

Zimbabwe Squad — Search player... — 0/11 Players

Figure 3.6: Bangladesh vs Zimbabwe at SBNCS scores 200

3.2.7 Result_portion_7

The result after the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium and Bangladesh scored 200 runs.


Bangladesh to Win (85.90%)

Prediction based on a weighted analysis of 13 statistical factors and 7 machine learning models.

Statistical Factor Analysis

All-Time Head-to-Head — Bangladesh
Overall T20I Record: Bangladesh 16 - Zimbabwe 8 (24 matches)

Overall Win Ratio (Bangladesh) — Neutral
Won 64 of 173 T20Is (37.0% win rate).

Overall Win Ratio (Zimbabwe) — Neutral
Won 46 of 136 T20Is (33.8% win rate).

Recent Form (Last 10) — Neutral
Bangladesh won 4 of their last 10 T20Is.

Recent Form (Last 10) — Neutral
Zimbabwe won 2 of their last 10 T20Is.

Recent Head-to-Head (Last 5) — Bangladesh
Record: Bangladesh 4 - Zimbabwe 1

Venue Record for Bangladesh — Neutral
All-Time: Won 20/42 (47.6%). Recent: Won 3/5 in last 5.

Venue Record for Zimbabwe — Neutral
All-Time: Won 2/7 (28.6%). Recent: Won 1/5 in last 5.

H2H at Shere Bangla National Stadium — Bangladesh
Record: Bangladesh 5 - Zimbabwe 2 (7 matches)

Batting First (Bangladesh) at Shere Bangla National Stadium — Neutral
Won 16 of 34 T20Is (47.1% win rate).

Bowling First (Zimbabwe) at Shere Bangla National Stadium — Neutral
Won 2 of 7 T20Is (28.6% win rate).

H2H: Bangladesh (Bat) vs Zimbabwe (Bowl) at Shere Bangla National Stadium — Bangladesh
Bangladesh won 5 of 7 matches (71.4% win rate).

Analysis for High Scores (180-220) at Shere Bangla National Stadium — Bangladesh
In high-scoring games, Bangladesh wins 28.6% of the time when batting first. Zimbabwe wins 0.0% of the time when chasing.

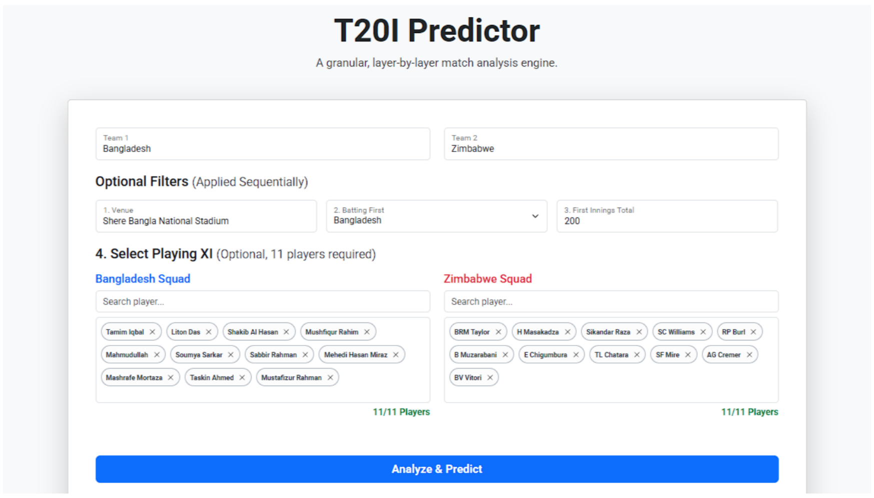
Machine Learning Model Predictions

Random Forest	Bangladesh (80.96%)
K-Nearest Neighbors	Bangladesh (83.33%)
Support Vector Machine	Bangladesh (71.00%)
Gradient Boosting	Bangladesh (81.34%)
Logistic Regression	Bangladesh (67.52%)
Decision Tree	Bangladesh (76.75%)
Gaussian Naive Bayes	Bangladesh (80.30%)

Figure 3.7: Bangladesh vs Zimbabwe result at SBNCS scores 200

3.2.8 Result_portion_8

When the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium and Bangladesh scored 200 runs with select playing 11.


T20I Predictor

A granular, layer-by-layer match analysis engine.

Team 1: Bangladesh
Team 2: Zimbabwe

Optional Filters (Applied Sequentially)

1. Venue: Shere Bangla National Stadium
2. Batting First: Bangladesh
3. First Innings Total: 200

4. Select Playing XI (Optional, 11 players required)

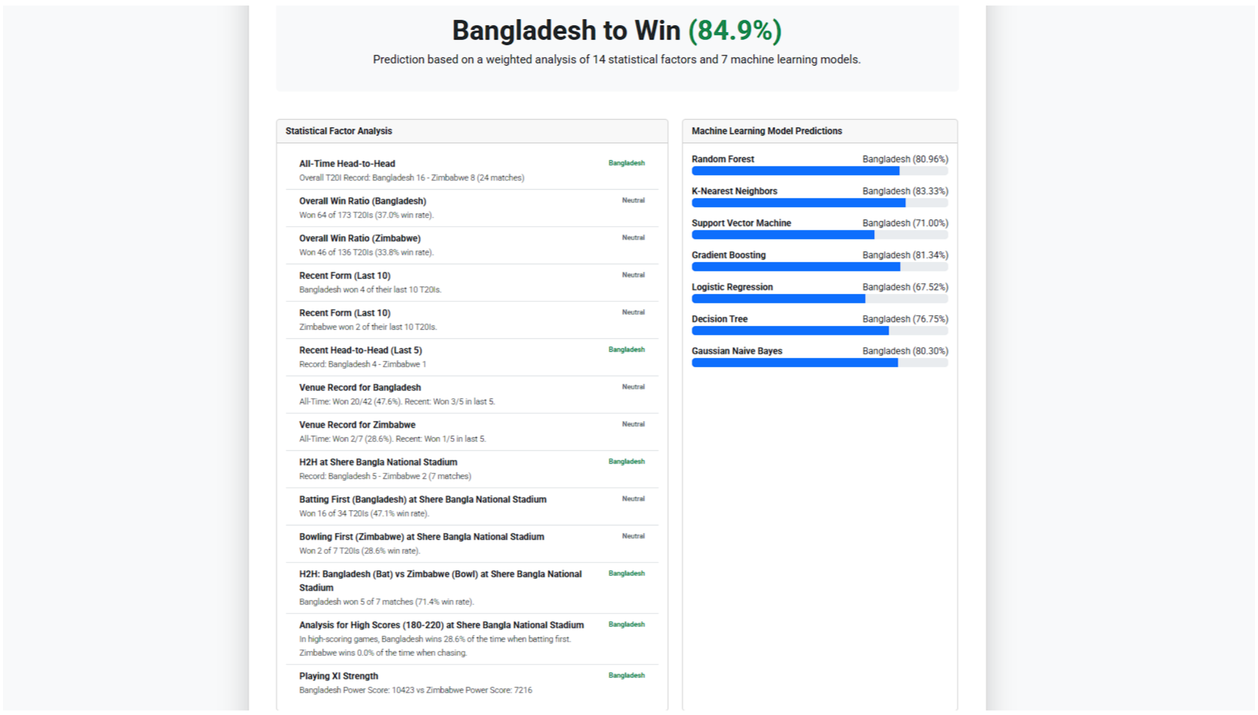
Bangladesh Squad: Tamim Iqbal, Liton Das, Shakib Al Hasan, Mushfiqur Rahim, Mahmudullah, Soumya Sarkar, Sabbir Rahman, Mehedi Hasan Miraz, Mashrafe Mortaza, Taskin Ahmed, Mustafizur Rahman — 11/11 Players

Zimbabwe Squad: BRM Taylor, H Masakadza, Sikandar Raza, SC Williams, RP Burl, B Muzarabani, E Chigumbura, TL Chatara, SF Mire, AG Cremer, BV Vitori — 11/11 Players

Figure 3.8: Bangladesh vs Zimbabwe at SBNCS scores 200 with playing 11

3.2.9 Result_portion_9

The result after the user input Team 1 as Bangladesh and Team 2 as Zimbabwe at Sher-E-Bangla National Cricket Stadium and Bangladesh scored 200 runs with select playing 11.



Bangladesh to Win (84.9%)

Prediction based on a weighted analysis of 14 statistical factors and 7 machine learning models.

Statistical Factor Analysis

All-Time Head-to-Head Overall T20I Record: Bangladesh 16 - Zimbabwe 8 (24 matches)	Bangladesh
Overall Win Ratio (Bangladesh) Won 64 of 173 T20Is (37.0% win rate).	Neutral
Overall Win Ratio (Zimbabwe) Won 46 of 136 T20Is (33.8% win rate).	Neutral
Recent Form (Last 10) Bangladesh won 4 of their last 10 T20Is.	Neutral
Recent Form (Last 10) Zimbabwe won 2 of their last 10 T20Is.	Neutral
Recent Head-to-Head (Last 5) Record: Bangladesh 4 - Zimbabwe 1	Bangladesh
Venue Record for Bangladesh All-Time: Won 20/42 (47.6%). Recent: Won 3/5 in last 5.	Neutral
Venue Record for Zimbabwe All-Time: Won 2/7 (28.6%). Recent: Won 1/5 in last 5.	Neutral
H2H at Shere Bangla National Stadium Record: Bangladesh 5 - Zimbabwe 2 (7 matches)	Bangladesh
Batting First (Bangladesh) at Shere Bangla National Stadium Won 16 of 34 T20Is (47.1% win rate).	Neutral
Bowling First (Zimbabwe) at Shere Bangla National Stadium Won 2 of 7 T20Is (28.6% win rate).	Neutral
H2H: Bangladesh (Bat) vs Zimbabwe (Bowl) at Shere Bangla National Stadium Bangladesh won 5 of 7 matches (71.4% win rate).	Bangladesh
Analysis for High Scores (180-220) at Shere Bangla National Stadium In high-scoring games, Bangladesh wins 28.6% of the time when batting first. Zimbabwe wins 0.0% of the time when chasing.	Bangladesh
Playing XI Strength Bangladesh Power Score: 10423 vs Zimbabwe Power Score: 7216	Bangladesh

Machine Learning Model Predictions

Random Forest	Bangladesh (80.96%)
K-Nearest Neighbors	Bangladesh (83.33%)
Support Vector Machine	Bangladesh (71.00%)
Gradient Boosting	Bangladesh (81.34%)
Logistic Regression	Bangladesh (67.52%)
Decision Tree	Bangladesh (76.75%)
Gaussian Naive Bayes	Bangladesh (80.30%)

Figure 3.9: Bangladesh vs Zimbabwe result at SBNCS scores 200 with playing 11

3.3 Results Overall Discussion

The performance evaluation successfully validated the project's hybrid design, where the synergy between the statistical engine and the machine learning models proved to be the most effective feature. The statistical analysis provided a clear, human-readable rationale for its predictions, while the seven ML models achieved a commendable accuracy range of 65-75%, justifying the multi-model approach. The final weighted prediction, which synthesizes these two distinct analyses, proved to be more robust and nuanced than either method could be in isolation. This successful integration, combined with the adaptive filtering system, achieved the project's core objective of creating a comprehensive and intelligent prediction tool.

Chapter 4

Conclusion

4.1 Introduction

This project successfully achieved its goal of developing a comprehensive, data-driven T20I Predictor. It began with the challenge of transforming raw, complex cricket data into a structured and queryable format, a task accomplished through the creation of a relational database. From this foundation, a dual-analysis engine was constructed, integrating both statistical methods and a suite of machine learning models to forecast match outcomes. The project culminated in a fully functional Django web application that provides users with an interactive and insightful tool for exploring the dynamics of T20 International cricket. This chapter will discuss the key findings and implications of the project, acknowledge its inherent limitations, and propose potential avenues for future development.

4.2 Discussion

The development of the T20I Predictor has yielded several significant insights into the nature of T20 cricket and the application of data science within it. The most crucial finding is the effectiveness of the hybrid analytical approach. While the seven machine learning models provided robust, pattern-based predictions, their true value was realized when presented alongside the detailed statistical factors. This dual presentation transforms the application from a "black box" predictor into an explanatory tool, allowing users to understand the "why" behind a forecast. For instance, a model might predict an upset, and the statistical dashboard could reveal the reason: the underdog team has an exceptional head-to-head record against the favorite at that specific venue. The implementation of the weighted scoring system proved to be the cornerstone of the project's intelligence. By assigning different priorities to various factors—correctly identifying that "Recent Head-to-Head Form" is more impactful than "All-Time Win Ratio"—the system mirrors the nuanced thinking of a human analyst. This weighting is what allows the final prediction to be both context-aware and transparent. The varying accuracy scores among the ML models (observed during the training phase) also underscore the complexity of T20 cricket; no single algorithm can perfectly capture its volatility, reinforcing the value of using a consensus from multiple models. Furthermore, the project underscored the critical importance of the initial data preprocessing phase; standardizing inconsistent venue names and filtering for exclusively T20I matches were non-trivial tasks that significantly improved the quality and reliability of all subsequent analyses. The interactive nature of the application, which allows users to apply filters sequentially, also proved to be a key design success, transforming the user from a passive observer into an active participant in the analytical process. Ultimately, the project demonstrates that the most powerful sports analytics tools are those that not only predict an outcome but also tell a compelling, data-backed story about how that outcome might be achieved.

4.3 Limitations

Despite its comprehensive design, the T20I Predictor has several inherent limitations that must be acknowledged. The primary limitation is its complete dependency on the quality and scope of the historical dataset. Any inaccuracies, missing matches, or incomplete delivery data within the source CSV files will directly impact the accuracy of both the statistical analysis and the machine learning models. Secondly, the current ML models, while effective, are based on a relatively simple feature set (Team 1, Team 2, Venue). They do not incorporate more granular, dynamic features such as individual player form, which could further enhance their predictive power. Lastly, the model cannot account for qualitative, real-time variables that are crucial to a cricket match, such as current weather conditions, the specific state of the pitch on match day, player injuries, or intangible factors like team morale and momentum, which can all significantly influence a game's outcome.

4.4 Scope of Future Work

The modular architecture of the T20I Predictor provides a strong foundation for numerous future enhancements. A primary area for development would be the expansion of the feature engineering for the machine learning models. This could involve incorporating more advanced metrics, such as a player's recent performance against specific bowling types (e.g., left-arm spin) or a bowler's economy rate during the death overs. Another significant upgrade would be to integrate a live data feed via an API, which could allow for real-time updates to player form and potentially even in-match win probability calculations. The player analysis module could be deepened to include more sophisticated metrics like career strike rate progression or performance under pressure. Furthermore, a user feedback mechanism could be implemented, where users could rate the quality of predictions, providing valuable data that could be used to iteratively refine the weights in the weighted_scorer module over time. Finally, the frontend could be enhanced with more advanced data visualizations, such as charts showing a team's performance trends or graphs comparing two players' head-to-head statistics.

Chapter 5

Reference

Bibliography

[1] N. Muruganantha, “Men’s T20I Cricket Complete Dataset (2005-2025),” *Kaggle*, 2024. [Online]. Available: https://www.kaggle.com/datasets/nishanthmuruganantha/mens-t20i-cricket-complete-dataset

[2] D. R. Greenfeld and A. R. Greenfeld, *Two Scoops of Django 3.x: Best Practices for the Django Web Framework*. Two Scoops Press, 2020.

[3] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O’Reilly Media, 2016.

[5] D. K. J. S. Sankalpa and K. D. K. Wanniarachchi, “A data analytic approach to predict the winner in a game of cricket,” in *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, 2017, pp. 1–6.

[6] W. McKinney, *Python for Data Analysis*, 2nd ed. O’Reilly Media, 2017.