Results

0%

of text is likely AI

AI-generated	0%
AI-generated & AI-refined	0%
Human-written & AI-refined	0%
Human-written	100%

Caution: Our AI Detector is advanced, but no detectors are 100% reliable, no matter what their accuracy scores claim. Never use AI detection alone to make decisions that could impact a person's career or academic standing.

report_for_ai_and_plagarism.pdf
7,503 Words 4% Matches 1Internet www.ijcse.com<1%
2Internet www.ijcaonline.org<1%
3Internet www.semanticscholar.org<1%
4Internet jisrc.szabist.edu.pk<1%
5Internet www.researchgate.net<1%
6Internet sjcit.ac.in<1%
7Internet ebin.pub<1%
8Internet tnsroindia.org.in<1%
9Internet www.oneturf.news<1%
10Internet docplayer.net<1% 11Internet ijcseonline.org<1%
12Internet www.newsbytesapp.com<1%
13Internet en.wikipedia.org<1%
14Internet link.springer.com<1%
15Internet ijsrem.com<1%
16Internet vdocuments.site<1%
17Internet pt.scribd.com<1%
18Internet ro.uow.edu.au<1%
19Internet www.frontiersin.org<1%

20Internet www.irjmets.com<1%

ver

Chapter 1

Introduction

1.1 Project Description

The aim of the present project is to provide a better experience of analysis of the ongoing/cricket live matches with real-time scores, constantly updating stats, and expert information to fans, commentators, and lovers of the sport. Through gaming interaction by the way of broadcasting live matches plus shifting statistics, the platform increases interest in the sport. The shared information will give strategic advice like; who bats first in terms of venue statistics in the past, a weather condition, and previous performance records. This is actually an effort of the information to provide a more improved perception to teams and followers before the initial delivery of a data-driven approach. In addition to the pre-match analysis, the project proposes a machine learning model that is supposed to determine the odds of succeeding in an attempt to complete a successful chase in the second innings of a T20 match. The formula always re-calculates after every ball making changes to the winning chances based on the situation at hand. The platform, with the help of Machine Learning allows improving not only the experience of the viewers, but also gives the analysts additional predictive insights about what will change in the match and transforms how T20 cricket can be consumed and interpreted.

1.2 Problem Definition

Whether to bat or to field are issues that challenge cricket teams after winning a toss, given that there must be many conditions to be considered such as ground situation, whether etc. The forecasts of final game results that vary with the course of the game, or with the statistics and performance indicators of the players and the teams, are also not well-calibrated, close to real-time predictions that Fans and teams could have.

ver

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 buiusing

Random Forest Classifier have high accuracy in match prediction, underscoring their value for

teams and fans alike.

1.4 Purpose

This project aims at creating the system of live scoring and prediction in cricket, which will involve fans and give them immediate information about the games and AI-based probability of win levels. The fans can follow minute-by-minute scores, adjustments in the game and even experience a novel relationship with the game. The external data sources will have advanced performance metrics which can be used accurately with the help of analysts.

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.

ver

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 includes research paper and journal study. The literature survey is done by taking into consideration the 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 study aims at development of machine learning-based tools to predict whether the team will win an ODI match before the game actually starts so that the team officials and experts can make use of such information. The paper uses the historic data of 2006 2019 ODI games (1693) to quantify the batting and bowling capacity, how teams score and various team stats. Logistic Regression, SVM, ensemble methods and feature selection were implemented in experiments with the aim of increasing the accuracy of the results. The models that scored the highest of 96.31 percent indicated that machine learning can handle the unpredictability of crickets and favour superior strategies. Further

1
1
ver
enhancements will include the addition of additional features and the application of the same to other forms of cricketing, or sports analyses in general.[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]
2
2
5
5
7
11
ver
• 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]

3
4
4
7
14
ver

2.3 Existing System

Table 2.1: Comparison Table

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.

ver

• 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.

ver

• 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.

ver

Chapter 3

Hardware and Software Requirements

3.1 Hardware Requirements

Table 3.1: Hardware Table

Hardware Specification

Specification Desired Value

Processor Intel i3 and above

Memory (RAM) 8GB Minimum

Hard Disk 10GB Minimum

3.2 Software Requirements

Table 3.2: Software Table

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
10
17
ver
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.

ver

• 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 emphasizes the optimization of performances in order to optimize loading time and response across the platform. It allows the platform to be more efficient and easy to use through systematic code optimization thus giving the user a smooth experience on the platform.

• Usability and Accessibility

CricketVerse does this with an advanced set of cricket analysis and a user-friendly and familiar interface. Regardless of whether users are new to statistics and bring little technical background to the table or whether they are seasoned authors of statistical procedures, they can easily embark on the discovery of match statistics and glean insights in terms of smartly designed dashboards and the interactive graphical presentations therein. Following the modern principles of design and focusing on user experience, CricketVerse will allow cricket enthusiasts of any expertise to get full fledged cricket statistics.

ver

Chapter 5

System Design

5.1 Architecture Diagram

Figure 5.1: Architecture Diagram

The mechanics of system design have been planned as a series of layers namely; Presentation Layer, Business Layer, Service Layer and Database Layer as shown in the diagram below, which is a representation of a layered architecture model of a piece of software. The database engine used is MongoDB and User Interface is the main base of interaction with users. Business Layer consists of the major functionalities of the project that are related to the dissemination of data about the results of the current match, predict the results of the second innings, offer live scores and detailed statistics, and help in the decisions which refer to the toss. In the environment of a competitive athletic sport, the components illustrate that the application helps to administrate real-time data and statistical facts. The two-way arrows depict how the dynamic information exchange takes place between the User, the Service Layer, and the Database, hence highlighting information movement that is essential in providing the end-users with the timely and relevant information. Altogether, this architecture facilitates consistent user experience by allowing clarity in the way data is treated and processed in the ecosystem of this system.

6

8

ver

5.2 Use Case Diagram

Figure 5.2: Use Case Diagram

User, as one of central participants (Figure 5.2), also interacts with key objects in the system within CricketVerse use case diagram. The web scraping method makes the platform provide real-time updates of the matches, whose content is instantly available to the user. Options like Live Score and Statistical Display allow a person to follow up on the measures of performance, final scores that are expected, and the result of every ball. The CricketVerse crunches on past results and the current pitch condition to give suggestions on which team should either bat or bowl first.

This functionality provides the user with probability specifying how probable it is that the side to chase can get the required runs with available conditions and performance of the players. Moreover, Match Result Statistics provide the important post-game statistics about the playing and the game outcome thus helping the user in analyzing matches. Altogether, the diagram can be used to demonstrate the way several features can make the user strong and the way the platform uses

data to deliver the best overall cricket viewing experience.

6
ver
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 first step in the methodology is the data collection.
• Data Collection - The data that comes in form of pertinent cricket-match indicators are collected through various sources but most significantly in form of live-match data, past-records, and player-performance index.
• Data Cleaning – Missing values imputation, inconsistencies resolved, creating the necessary columns out of the available ones, and standardizing the string data are done as a part of preprocessing activities.
• Feature Extraction – Important features, including the batting performance, bowling impact, and run rate, are brought out to improve the accuracy of the model.
• Train/Test Split – The data is divided into a training and testing part, which allows a good training of models and providing a strict evaluation of performance.
• Model Training– The model is trained using an random forest approach, incorporating multiple algorithms to improve predictive accuracy.
• Testing — The model is tested on unseen data to calculate its performance.
• Model Evaluation – Measures that are used to confirm the effectiveness of the model include accuracy, precision, recall, and F1-score.
8
15
16
ver
• Testing on Real Data — At last, the model undergoes a prediction testing on actual cricket matches in order to verify it is a reliable model.
• End – The process closes, ensuring 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

The first image (Figure 6.2 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:

ver

• 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").
• match_url (String): The URL link to the match details.
• match_format (String): The format of the match (e.g., "Test", "ODI", "T20").
• venue (String): The location where the match is held.
• date (String): The date of the match.
• toss (String): The result of the toss (e.g., team name that won the toss).
• player_of_the_match (String): The player awarded as the best performer of the match.
• current_run_rate (String): The current run rate during the match.
• inning_1 (Object): Details of the first innings, including:
• batting (Array[BatterSchema]): An array of batting statistics based on the BatterSchema.
• bowling (Array[BowlerSchema]): An array of bowling statistics based on the BowlerSchema.
• inning_2 (Object): Details of the second innings, including:
• batting (Array[BatterSchema]): An array of batting statistics based on the BatterSchema.
• bowling (Array[BowlerSchema]): An array of bowling statistics based on the BowlerSchema.
• createdAt (Date): The timestamp when the document was created, with a default value of

the current date and time.

ver

Figure 6.3: Live Match Data Document Example

The second image(Figure 6.3) 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".
• match_url:“https://www.espncricinfo.com/series/india-in-england-2025-1445348/england-vs-india-1st-test", the URL link to the match details.
• match_format: "1st Test", the format of the match.
• venue: "Leeds", the location where the match is held.
• date: "June 20 - 24", the date range of the match.
• toss: "England elected to field first", the result of the toss and the decision made.
• player_of_the_match: "N/A", indicating no player has been awarded yet.
• current_run_rate: "CRR: 4.26 - Last 10 ov (RR): 52/1 (5.20)", the current run rate and recent over performance.
• inning_1 and inning_2: Objects containing detailed batting and bowling statistics (data not fully expanded in the image).
• createdAt shows a timestamp ("2025-02-13T11:03:05.933Z") indicating when the document was stored.

ver

Chapter 7

Implementation

7.1 Pseudo Code2

nd

Inning Prediction

START
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

ver
Toss Decision Recommendation:
If Team 1, Team 2, or Venue is empty,
    Display error message: “Please select both teams and venue”

If Team 1 is same as Team 2,
    Display error message: “Teams must be different”

If weather inputs are invalid,
    Display error message: “Please enter valid weather values”

Else
    Fetch team1’s bat-first win rate
    Fetch team2’s bat-first win rate
    Fetch venue’s bat-first win rate
    Fetch venue’s average score
    Initialize decision_score = 0
    If team1 bat-first win rate > 0.5, add 1 to score
    If team2 bat-first win rate < 0.5, add 1 to score
    If venue bat-first win rate > 0.5, add 1 to score
    If venue average score > 160, add 1 to score
    If humidity > 70 or weather is Rainy, subtract 1 from score
    If wind speed > 20, add 0.5 to score
    If cloud cover > 60, subtract 0.5 from score

    If decision_score ≥ 2
        Display “Suggested Toss Decision: BAT”
    Else
        Display “Suggested Toss Decision: BOWL”
ver
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.

ver

Figure 7.2: Match Result

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.

9
13
ver

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.

ver

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.

ver

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.
ver

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.

ver

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 ver
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.

ver

Figure 7.8: Model Training

The image represents the Python code where authors use the library scikit-learn to create a

machine
learning model that could predict whether a chase in cricket games was successful. The required modules, namely, `train_test_split`, `RandomForestClassifier` and the evaluation metrics, are all imported in the code before processing is done on the data. Features are invoked in the matrix `X` and the target variable, `Chased Successfully`, is invoked in `y`. The data are then partitioned between training and test (80 % train, 20 % test) and the Pi-random state is fixed (42). A Random Forest Classifier (having 100 estimators) is then instantiated, and the training data is fed to it by calling the fit() const. One of the comments states that more samples will be necessary to improve further splitting.

Figure 7.9: Prediction

The image represents both the interface and the output of a system called Cricket Match Win Predictor a Computation system used to predict probability of a cricket team winning or losing a match, which is in progress, live, and hence the data available in real time is used in calculating the probability using this computation system. The predictor utilizes various match level and batter variables that are considered to be essential in computing the probability of winning. They are:
1. Target Score: runs that are needed to win (160 in this case).
2. Current Score: the runs scored by the batting team at the present point (in this case 150).
3. Wickets Fallen: how many wickets are lost by the concerned team (in this case, six).
4. Overs Bowled: the number of over which bowling has already been done (in this case 19).
5. Balls in Over (05): how many extra deliveries there are during the innings that are not part of a regular six-ball over (in this case zero because this is the first ball of an over).

6. Batter Runs: the scores of a team within the current strike batsman (in this case, 42).
7. The runs which the batsman is presently on the non-striker end (here 21) put together are called Non-Striker Runs.
8. Batter Balls Faced: how many deliveries did the striker face (33 in our case).

9. Non-Striker Balls Faced: the number of deliveries that the non-striker has received (in this case, 13).

In sum, these inputs can be directly linked to the following variables that are saved in the predictive model:

• required_run_rate: the run rate that one has to achieve in order to win or the required run rate = (160 runs / run we have to score per over / runs per over which is calculated by Overs Bowled and Balls in Over).

• current_run_rate: the real runs per over at the moment (150 scored runs divided by the Overs Bowled).

• wickets_remaining: the number of wickets remaining (each of the wickets left minus the Wickets Fallen).

• the runs that a batter has made, the runs that the non striker has made, how many balls have a batter faced, how many balls have the non striker faced, ball remains.

When the encoding of these variables has been achieved, it is at this time that the random forest regression algorithms are executed by the predictor to yield the ultimate winning probability.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.

ver

Figure 7.10: 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 be played (ex: Leeds).

Parameters of weather:

temperature (o C)

Humidity (%)

18

ver

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.

ver

Figure 7.11: Toss Decision Recommendation UI

The picture presented is a screenshot of CricketVerse application and covers the self-standing tool Toss Decision Advisor. The module will help users come up with evidence-based decisions regarding the adequacy of the flip in Twenty20 Internationals between teams constituting nation-level representation. The interface allows the users to:

1. Select a Match

Make a selection of a list of available or upcoming T20-matches. The current case in point is the India versus England game that would be played 20 24 June at Headingley in Leeds.

2. Automatic Weather Information Download

Once a fixture is defined, the system makes an API hit to access the current weather information of the venue and the playing date. The algorithm then displays a contextual weather report which is saying, currently, Rain, Partly cloudy, 22 (Celsius) and humidity level is 61.4 percent.

3. Recommendation-Toss Decision

After the identification of a fixture, an application provides statistical analysis and combines a

variety of factors including meteorological data, past precedents and conditions of the pitch. The system will then suggest a best tip to make, which is most often shown as either, batting first or fielding first. This recommendation is prompted by a button labeled as Suggest Decision.

4. UI Display

The recommended tossing option is then revealed on the user interface, where a set of summary will appear of the recommended course of action.

ver

Figure 7.12: Toss Decision Recommendation UI

The webpage CricketVerse uses the tool of Toss Decision Advisor that also chooses the matches based on T20 format as well as the only criteria is that the major international team should be from the following team: India, Australia, England, Pakistan, West Indies, South Africa, Sri Lanka and Bangladesh. No results are found, which suggests that currently T20 games between the teams are not available.

ver

Chapter 8

Software Testing

8.1 Manual Testing Cases

Table 8.1: Second Inning Prediction

ver

Table 8.2: Live Match Scores

ver

Table 8.3: Match Details Page

ver

Table 8.4: Toss Decision Recommendation

ver

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.

ver

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.

ver

Appendix A BIBLIOGRAPHY

RESEARCH PAPER REFERENCES

[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)

ver

BOOK REFERENCES

• “Broader Sports Analytics Books (Include Cricket)”

Practical sources of learning sports and therefore cricket data analysis are the book: Machine Learning
and Data Mining for Sports Analytics (Springer CCIS) by Ted Kwartler and the book: Sports Analytics in Practice with R, also by Kwartler. The Springer book presents methods and algorithms of machine learning to forecast the result and measure the play of athletes in several sports, whereas
the Kwartler book teaches practical analytics in R and contains real-life examples. Both of them are easy to follow and rather useful to those going into the field and to researchers as well, presenting clear illustration, case studies, and examples to employ the concept of data science to practice in sports.

WEB REFERENCES

• “Cricket Match Prediction Datasets”

Kaggle is a contemporary and free site that allows one to navigate sports analytics data and develop predictive models like the match results of a cricket game. It provides practical data, contests, and communal notebooks in either Python or R. Kaggle is excellent practice that allows learners to use machine learning to make the predictions on winners and performance analysis of players.
Kaggle: Your Machine Learning and Data Science Community