A Mini Project report on

CRICKET SCORE INSIGHT USING STREAMLIT

A documentation submitted in partial fulfillment of the academic requirement for the award of degree of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

By

ABU TALHA (1604-21-748-010)
ABDUL RAHMAN (1604-21-748-011)
TAYYAB QURESHI (1604-21-748-027)

Under the guidance of

MRS SEEMA AHMED

ASST PROFESSOR, CS&AI Dept., MJCET

COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to Osmania University)

Hyderabad.

2023 – 2024

MUFFAKHAM JAH
COLLEGE OF ENGINEERING AND TECHNOLOGY

Mount Pleasant, 8-2-249 to 267, Road No. 3, Banjara Hills,
Hyderabad - 500 034, Telangana State, India

CERTIFICATE

This is to certify that the mini project report on "Cricket Score Insight " is a bonafide work carried out by ABUTALHA (1604-21-748-010) , ABDUL RAHMAN (1604-21-748-011) & TAYYAB QURESHI (1604-21-748-027) in the partial fulfillment of the requirements for the award of the B.E. CSE(AI&ML) in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2022-2023.

Internal Guide
Mrs Seema Ahmed
Asst Professor,
CS&AI Dept, MJCET,
Hyderabad.

Head of the Dept
Dr. Uma N. Dulhare
Professor & Head,
CS&AI Dept, MJCET,
Hyderabad.

External Examiner

DECLARATION

We hereby declare that the work entitled **"Cricket Score Insight"** developed under the supervision of Mrs Seema Ahmed, Asst Professor, CS&AI Department and submitted to **MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY** in original and has not been submitted in part or while for under graduation degree to any other university.

AbuTalha
(1604-21-748-010)

Abdul Rahman
(1604-21-748-011)

Tayyab Qureshi
(1604-21-748-027)

ACKNOWLEDGEMENT

Our sincere thanks and gratitude to **Dr. UMA N. DULHARE, Professor & Head of the Department**, for all the timely support and valuable suggestions during the period of my mini project.

We are extremely thankful to Project Review Committee Members, Dept. of CS&AI for their encouragement and support throughout the project.

We are extremely thankful and indebted to my internal guide**, Mrs SeemaAhmed, Asst Professor** for her exemplary guidance and encouragement. Her trust and support inspired me in the most important moments of making right decisions and I am glad to work under her supervision.

We would like to thank all the faculty and staff of department who helped us directly or indirectly in completing the project work. Last but not least, I would like to thank my family members and friends for their co-operation for completing the project.

AbuTalha
(1604-21-748-010)

Abdul Rahman
(1604-21-748-011)

Tayyab Qureshi
(1604-21-748-027)

TABLE OF CONTENTS

ABSTRACT

Cricket Title : Cricket Scores Insights Using Machine Learning Models

Abstract:

This study delves into the burgeoning field of sports analytics by applying machine learning models to predict cricket scores, utilizing a comprehensive dataset comprising historical match statistics, player performance metrics, and contextual factors. Employing regression and ensemble methods, the research explores the impact of key features such as team composition, batting and bowling averages, venue characteristics, and recent performance trends on match outcomes. The inclusion of temporal considerations, such as match format and evolving player form, enhances predictive accuracy. Evaluation metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) assess model performance against real match data, revealing promising results. The developed models provide a robust framework for informed decision-making in cricket, offering valuable insights for enthusiasts, analysts, and team strategists, with broader implications for predicting outcomes in other dynamic and data-intensive sports.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Introduction:

Cricket, being one of the most popular and widely followed sports globally, has captured the attentionof fans, analysts, and enthusiasts alike. The unpredictability and dynamic nature of cricket matches have spurred interest in developing accurate predictive models for forecasting match outcomes, with a specific focus on predicting final scores. Cricket score prediction involves the application of advancedstatistical and machine learning techniques to analyze various factors that influence the game,including team performance, player statistics, pitch conditions, historical match data, and more.

The rationale behind employing machine learning in predicting cricket scores lies in the sport's multifaceted nature, encompassing diverse variables such as player performance, team dynamics, playing conditions, and historical trends. Traditional methods of cricket score prediction often rely onsubjective judgment and expert opinions, but machine learning offers a systematic and data-driven approach to unravel the complexities inherent in the game.

As cricket enthusiasts and analysts seek a more nuanced understanding of the factors influencing match outcomes, this project seeks to bridge the gap by leveraging advanced machine learning algorithms. By harnessing historical match data, player statistics, and contextual information, we aimto develop models capable of providing accurate and insightful predictions, contributing to the ever- growing field of sports analytics.

This project's significance extends beyond the boundaries of cricket, serving as a blueprint for the application of machine learning in predicting outcomes in other sports characterized by dynamic anddata-rich environments. Through this exploration, we aspire to enhance our understanding of the game, empower decision-makers with valuable insights, and contribute to the ongoing revolution in sports analytics.

OBJECTIVES

The objectives for this project work are –

1. Develop an accurate and reliable predictive model for forecasting cricket match scores
2. Leverage historical and contextual data for analysis.
3. Utilize advanced machine learning algorithms for in-depth analysis.
4. Analyze various factors including team performance, player statistics, match venue, weatherconditions, and recent form

1.2 ORGANIZATION OF THESIS

The following is the format of the thesis which is outlined in five chapters. A briefdescription of each chapter is as follows:

The first chapter consists of the Introduction, Objectives, and Organization of Thesis. Second chapter contains the Literature Survey. This chapter focuses on the areassuch as the Existing System, and the Problems with the Existing System. Third chapter explains about the Proposed System. Here, we talk about what the Proposed system can do, the Problem Statement and the System Architecture. Fourth chapter is about the Methodologies used. It explains about the various kinds of methods used in this project and about the various Technologies used. Fifth chapter is the Implementation. It tells us about the Requirements (both Hardware Requirements and Software Requirements), the Code Snippets explaining the main parts of the project with code, and the Execution part. Finally, the Sixth chapter talks about the Result Analysis and Conclusion. Here, wecan see the Result Analysis, Conclusion on the Results obtained and Future Enhancement which is followed by references.

CHAPTER 2

LITERATURE SURVEY

S.no	Title	Description	Results	Advantages	Drawbacks/limitations
1.	The Cricket Winner Prediction With Application of Machine Learning and Data Analytics	Data Science, Machine Learning, Modelling	Prediction andanalysis of score with the help of some prediction algorithms likeLR, RF, and many others.	Informed Decision-Making, Performance Analysis, Strategic Planning, Fan Engagement	Inherent Uncertainties in Cricket, Limited Historical Data,Non-Quantifiable Variables, Data Quality and Reliability
2.	Prediction of Indian Premier League-IPL 2020 using Data Mining Algorithms	Data Mining Algorithms	Analysis of score duringshort format match on thebasis of various machine learning algorithms	Pattern Recognition, Complex Relationship Analysis, Handling Non-Linearity, Scalability	Overfitting, Limited Interpretability,Data Quality Issues

3.	IPL win prediction System toimprove Team Performance using SVM	SVM	Importance of machine learning, deep learning and neural networkand prediction analysis	Pattern Recognitio n, Flexibility, Automation ,Image and Speech Recognitio n	Interpretability, ComputationalIntensity, High Dimensionality,Bias and Fairness, Lack of Understanding of Causation
4.	Prediction on the IPL data using machine learning techniques in Rpackage	Machine Learning, R Package	In this journalthey applied various prediction algorithms and opter one that has higher accuracy	Dplyr, tidyr, ggplot2, caret, glmnet	Limited Data Availability, Overfitting:
5.	Cricket Score Predictionusing Machine Learning	Cricket Analysis, Lasso Regression, logistic Regression	It is advantageous to use machine learning to analyse cricket games by taking previous game data, player performance, natural parameters, pre-game conditions, and other features into account.	Input Pre-processing, Logistic analysis	Limited Interpretability, DataReliability

CHAPTER 3

3.1 EXISTING SYSTEM

Existing System:

Several existing systems have explored the application of machine learning techniques to predict cricket scores. One notable example is the work carried out by researchers and data scientists in collaboration with cricket boards, sports analytics firms, and academic institutions. While the specific methodologies and models may vary, the general approach involves utilizing historical match data, player statistics, and other relevant features to train machine learning algorithms for score prediction.

One such system involves the following key components:

1. Data Collection and Preprocessing:

- Historical Match Data: Comprehensive datasets containing information about past cricket matches,including details on teams, players, venues, and match outcomes.
- Player Statistics: Individual player performance metrics such as batting and bowling averages, strikerates, and recent form.

2. Feature Selection:

- Identification of relevant features that significantly impact match outcomes. This may include teamcomposition, recent performance trends, venue characteristics, weather conditions, and more.

3. Machine Learning Models:

- Regression Models: Linear regression and other regression techniques are commonly employed topredict numerical outcomes, such as total runs scored in a cricket match.
- Ensemble Models: Techniques like Random Forests or Gradient Boosting are often used to improvepredictive accuracy by combining the strengths of multiple models.

4. Training and Testing:

- The system divides the historical data into training and testing sets to train the machine learningmodels. Cross-validation techniques may be applied to assess the model's generalization performance.

5. Evaluation Metrics:

- Metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used to evaluate the accuracy of the machine learning models in predicting cricket scores.

6. Model Deployment:

- Once trained and validated, the machine learning models can be deployed to predict scores for upcoming matches. The real-time integration of data ensures that the predictions consider the latest player and team statistics.

3.2 PROBLEMS WITH EXISTING SYSTEM

While existing systems for cricket score prediction using machine learning have shown promise, theyare not without their challenges. Here are some common problems associated with these systems:

1. Data Quality and Variability:
   - The accuracy of predictions heavily relies on the quality and consistency of the historical data.Inconsistencies, missing values, or inaccuracies in the dataset can lead to biased models and inaccurate predictions.

2. Dynamic Nature of Cricket:
   - Cricket is a dynamic sport with various formats (Test, One Day International, T20) and ever-evolving playing styles. Existing models may struggle to adapt to the changing dynamics of the game, especially when faced with unexpected events or rule changes.

3. Feature Selection and Engineering:
   - Identifying and selecting relevant features is a complex task. The importance of certain featuresmay change over time, and incorporating the right variables into the model is crucial for accurate predictions. Additionally, creating meaningful new features through engineering is a challenge.

4. Limited Contextual Information:

- Many existing systems may not adequately incorporate contextual information such as player injuries, team strategies, and other dynamic factors that can significantly impact match outcomes.

5. Overfitting and Generalization:

- Some models may suffer from overfitting, where they perform well on the training data but fail togeneralize effectively to new, unseen data. Striking the right balance between model complexity and generalization is essential.

6. External Factors:

- External factors like weather conditions, pitch behavior, and player psychological states are oftendifficult to quantify accurately and may not be fully considered in existing models.

7. Model Interpretability:

- Interpretability of machine learning models is a challenge. Understanding the rationale behind specific predictions is crucial for gaining trust in the system, especially in decision-making scenarios.

8. Real-time Adaptability:

- Adapting models in real-time to accommodate late-breaking information, such as last-minute player changes or weather updates, can be challenging. Existing systems may not be agile enough toincorporate such changes effectively.

9. Evaluation Metrics:

- The choice of evaluation metrics may impact the perceived success of a system. Some metricsmight not fully capture the nuances of predicting cricket scores, and using multiple metrics for evaluation is essential.

10. Ethical Considerations:

-Bias in data and models, unintentional or otherwise, can lead to unfair predictions. Ensuringfairness and ethical considerations in the development and deployment of these systems is an ongoing challenge.

Addressing these problems requires continuous research, refinement of models, and collaborationbetween data scientists, cricket experts, and domain specialists to enhance the accuracy and applicability of machine learning in predicting cricket scores.

CHAPTER 4

PROPOSED SYSTEM

Objective:

- Develop a robust and accurate cricket score prediction system using machinelearning techniques.
- Improve upon existing models by addressing their limitations and incorporatingdynamic factors influencing match outcomes.

Feature Selection and Engineering:

- Identify and select relevant features that have a significant impact on matchoutcomes.
- Continuously refine feature selection based on the evolving nature of the game.

Machine Learning Models:

- Implement a variety of machine learning algorithms, including regression models(e.g., linear regression) and ensemble models (e.g., Random Forests or Gradient Boosting).
- Experiment with deep learning approaches, such as neural networks, to capture complex relationships within the data.

Training and Testing:

- Split the dataset into training and testing sets to train machine learning models.
- Utilize cross-validation techniques to ensure the robustness and generalization ofthe models.

Real-time Data Integration:

- Implement a mechanism for real-time data integration to update models with thelatest player and team statistics.

Evaluation Metrics:

- Employ appropriate evaluation metrics, such as Mean Absolute Error (MAE) andRoot Mean Squared Error (RMSE), to assess the accuracy of the models.
- Consider additional metrics that account for the dynamic nature of cricket and theuniqueness of match prediction.

Model Interpretability:

- Enhance model interpretability to provide users with insights into the

factors influencing specific predictions.

Adaptive Model:

- Develop an adaptive model that can adjust to changes in the game, such as rulemodifications, playing conditions, and team strategies.

Ethical Considerations:

- Address potential biases in data and models to ensure fairness and ethical considerations.
- Implement transparency measures to provide users with a clear understanding ofhow predictions are made.

User Interface:

- Design a user-friendly interface for stakeholders to interact with the predictionsystem.
- Provide relevant visualizations and insights to assist users in interpreting andutilizing the predictions effectively.

Documentation and Reporting:

- Document the entire development process, including data preprocessing steps,model selection rationale, and hyperparameter tuning.
- Generate comprehensive reports on model performance and present findings in anaccessible manner for both technical and non-technical stakeholders.

4.1 PROBLEM STATEMENT

Cricket, as a dynamic and multifaceted sport, poses significant challenges in accurately predictingmatch outcomes. While existing systems employ machine learning techniques for cricket score prediction, there are notable shortcomings that hinder their effectiveness. The following issues necessitate the development of an advanced predictive system:

1. Data Quality and Variability:

- The accuracy of predictions is heavily reliant on the quality and consistency of historical data.Inaccuracies, missing values, and inconsistencies in the dataset can lead to biased models and unreliable predictions.

2. Dynamic Nature of Cricket:

Cricket is characterized by its ever-changing dynamics, encompassing various formats andevolving playing styles. Existing models struggle to adapt to these changes, impacting their predictive capabilities in the face of unforeseen events or rule modifications.

3. Feature Selection and Engineering Challenges:

- Identifying relevant features and creating meaningful new variables through feature engineering isa complex task. The current systems may not effectively capture the dynamic factors that influence match outcomes.

4. Limited Contextual Information:

- The incorporation of contextual information, such as player injuries, team strategies, and other dynamic factors, is often insufficient in existing models, leading to predictions that may not fullyreflect the complexities of the game.

5. Overfitting and Generalization Issues:

- Some models suffer from overfitting, performing well on training data but failing to generalizeeffectively to new, unseen data. Striking the right balance between model complexity and generalization remains a challenge.

6. Real-time Adaptability:

- Adapting models in real-time to accommodate late-breaking information, such as last-minute player changes or weather updates, is challenging. Existing systems may lack the agility required toincorporate such changes effectively.

7. User Interpretability:

- The lack of interpretability in some machine learning models poses challenges in understandingthe rationale behind specific predictions. Building user trust and confidence is essential for the practical application of predictive insights.

8. Evaluation Metric Suitability:

- The choice of evaluation metrics may impact the perceived success of a system. Some metrics maynot fully capture the nuances of predicting cricket scores, and the selection of appropriate metrics is essential for accurate assessment.

SYSTEM ARCHITECTURE

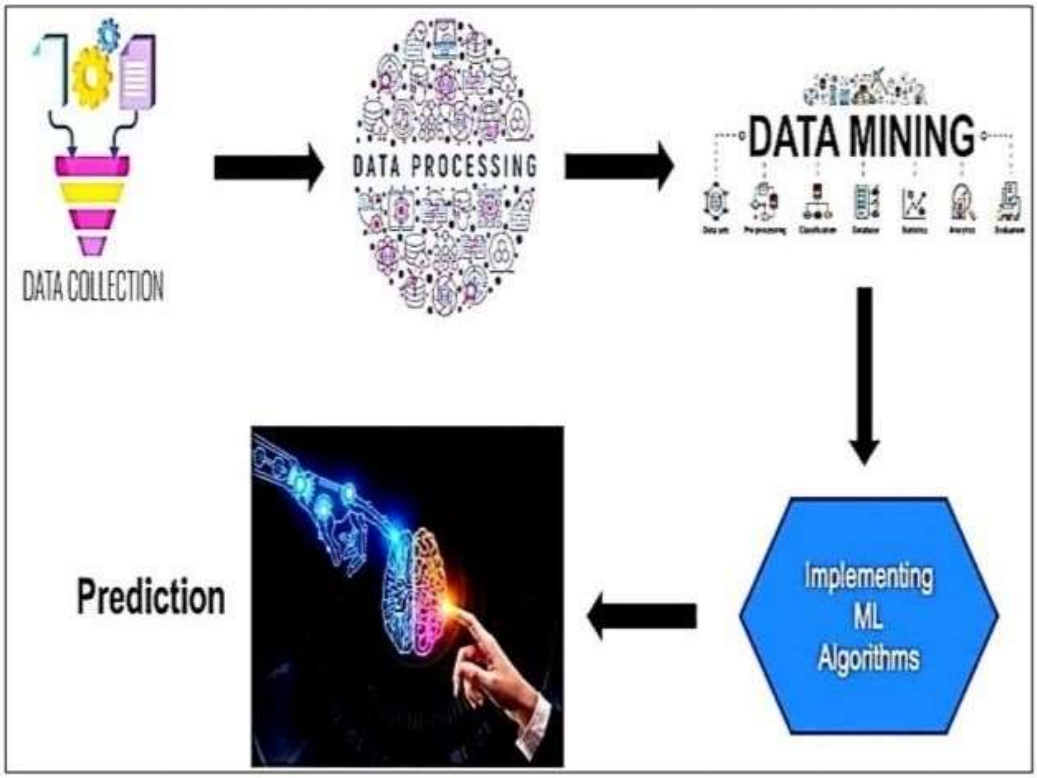


Fig. 1. Architecture diagram.

The System architecture involves various components that work together to generate accuratepredictions. Here's an outline of the system architecture for a cricket score prediction project:

1. Data Collection Module:

- Description: Collects historical match data, player statistics, and relevant features from diversesources.

- Components:
  - Web Scraping Tools/APIs for real-time data.
  - Historical datasets from cricket databases.

2. Data Preprocessing Module:

- Description: Cleans and preprocesses the collected data to ensure consistency and accuracy.

- Components:

- Data Cleaning Algorithms.
  - Missing Value Imputation.
  - Feature Scaling and Normalization.

3. Feature Engineering Module:

- Description: Identifies relevant features and creates new variables to enhance the predictivepower of the model.

- Components:

  - Feature Selection Algorithms.
  - Dimensionality Reduction Techniques.
  - Contextual Information Integration.

4. Machine Learning Models Module:

- Description: Implements various machine learning algorithms for score prediction.

- Components:

  - Regression Models (e.g., Linear Regression).
  - Ensemble Models (e.g., Random Forest, Gradient Boosting).
  - Deep Learning Models (e.g., Neural Networks).

5. Training and Testing Module:

- Description: Divides the dataset into training and testing sets to train and validate the machinelearning models.

- Components:
  - Cross-Validation Techniques.
  - Hyperparameter Tuning.

6. Real-time Data Integration Module:

- Description: Updates the models with the latest player and team statistics in real-time.

- Components:
  - Data Streaming Mechanism.
  - APIs for Real-time Updates.

7. Prediction Module:

- Description: Generates predictions based on the trained models and real-time data.

- Components:
  - Prediction Algorithms.
  - Model Ensembles for Improved Accuracy.

8. User Interface Module:

- Description: Provides a user-friendly interface for stakeholders to interact with the

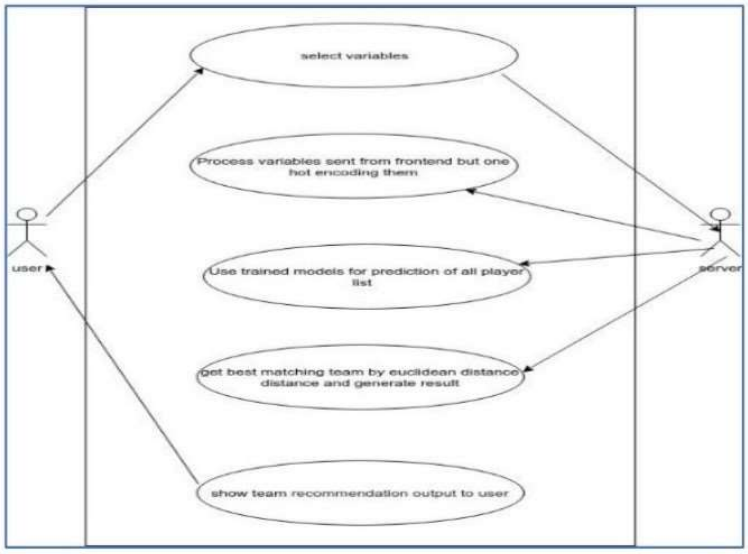
system andview predictions.

- Components:
  - Web-Based Dashboard.
  - Visualization Tools.

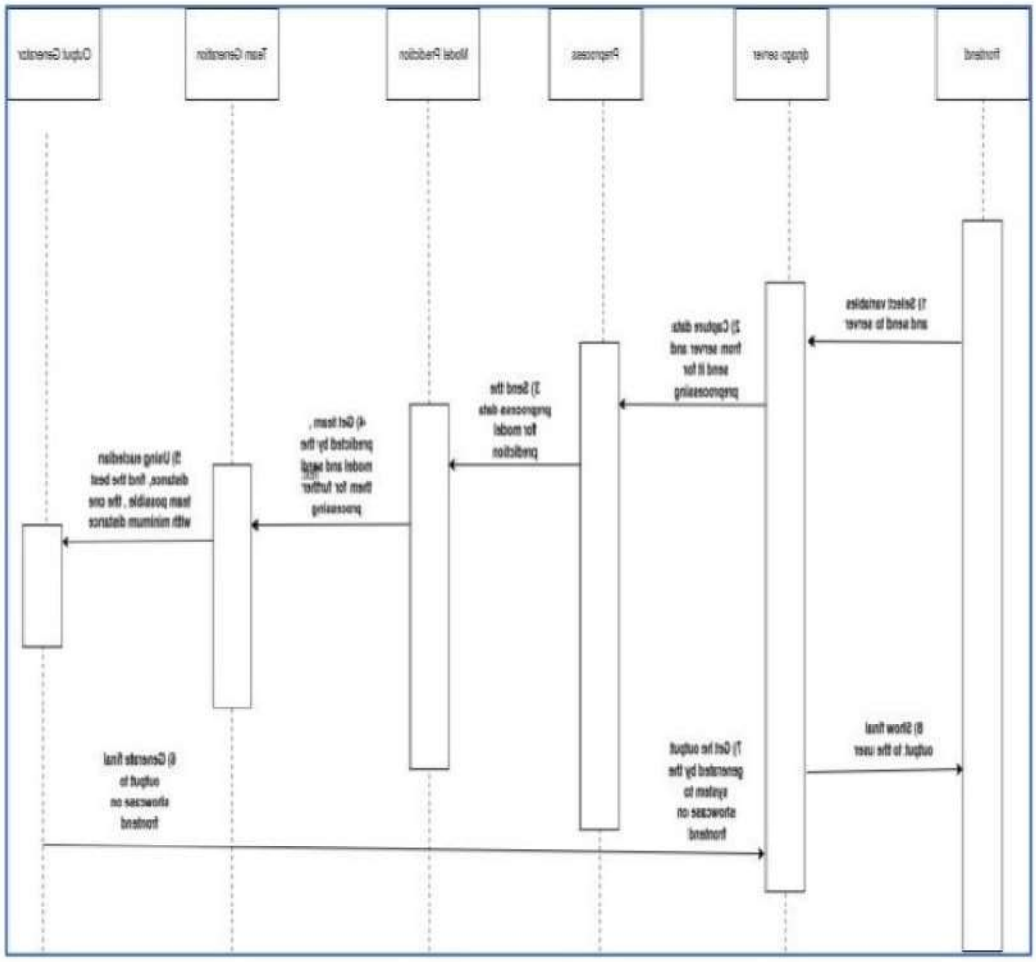
9. Model Interpretability Module:

- Description: Enhances the interpretability of machine learning models for user understanding.

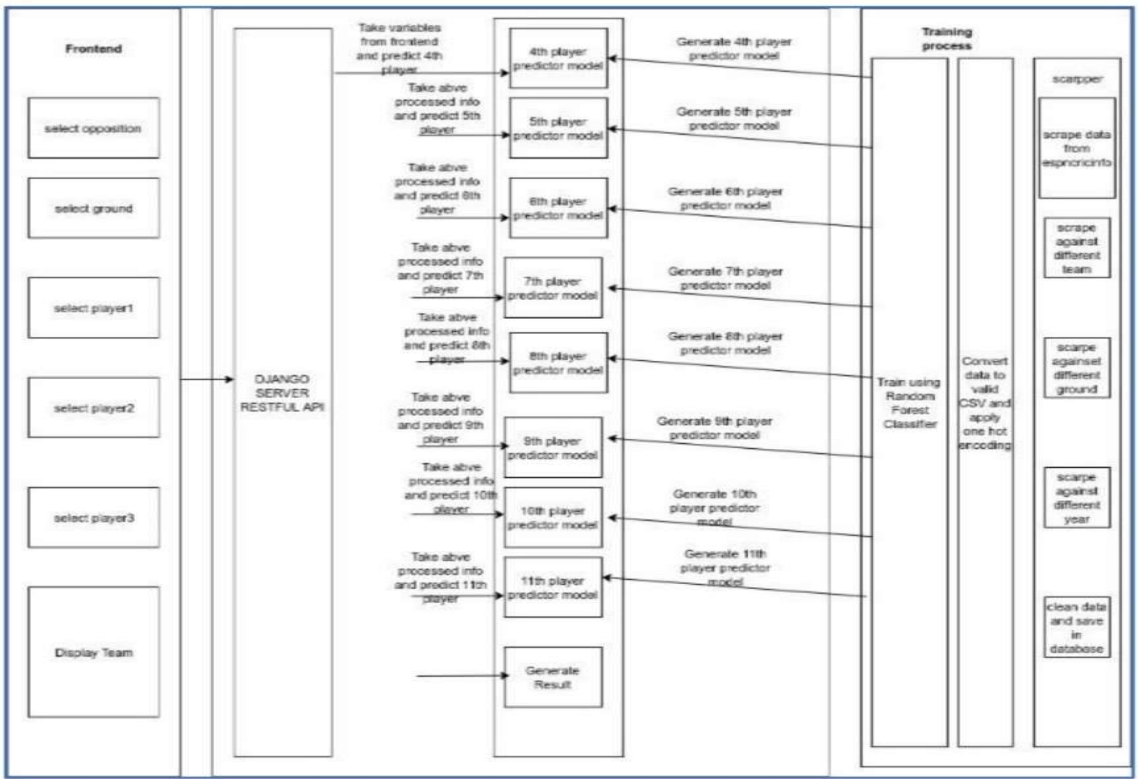
- Components:
  - Feature Importance Visualization.
  - Explanatory Models.



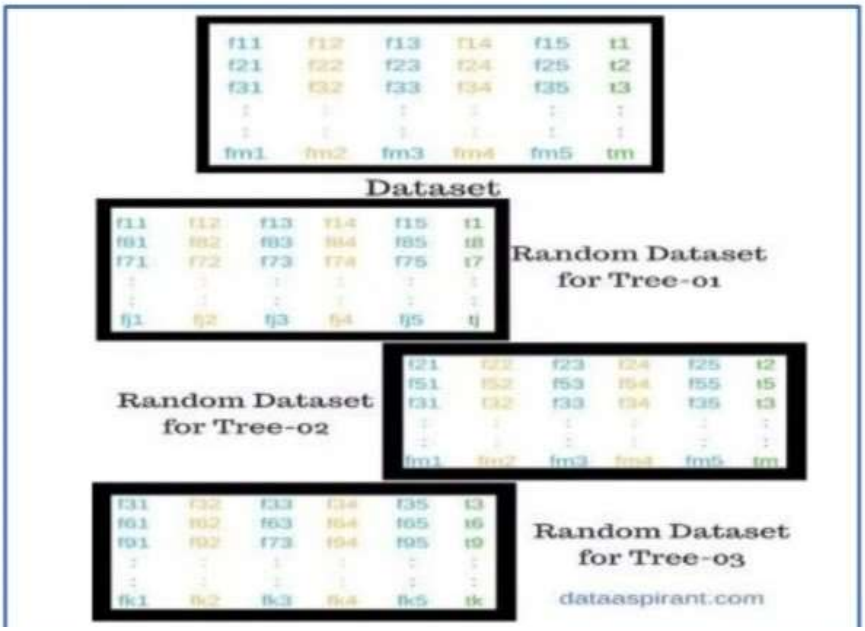
USE CASE DIAGRAM



SEQUENCE DIAGRAM



BLOCK DIAGRAM



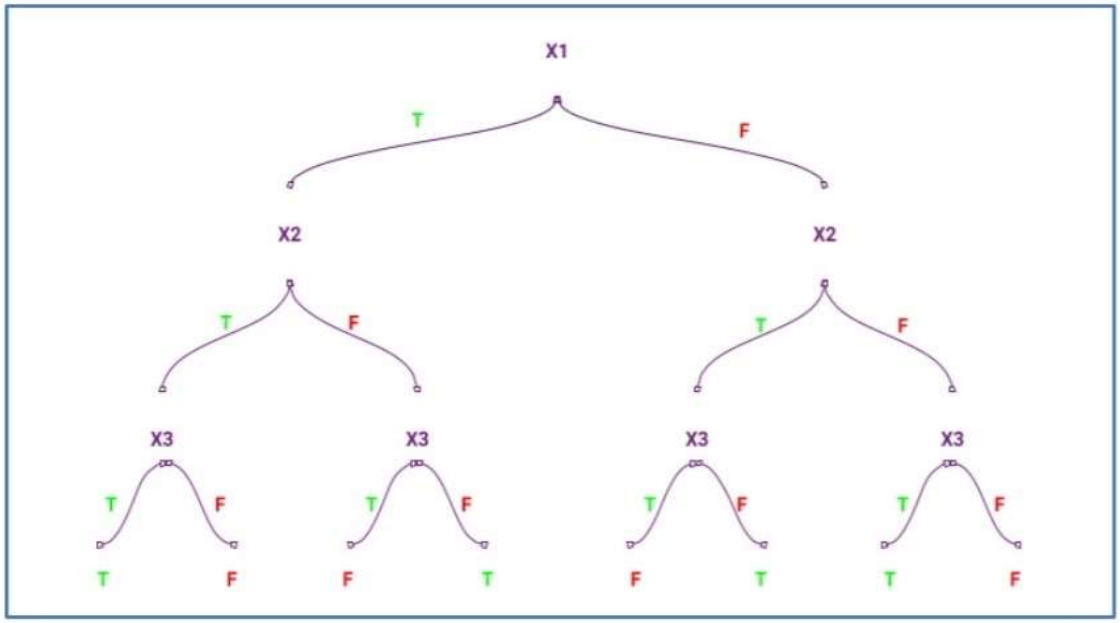
HOW RANDOM FOREST WORKS

10. Adaptive Model Module:

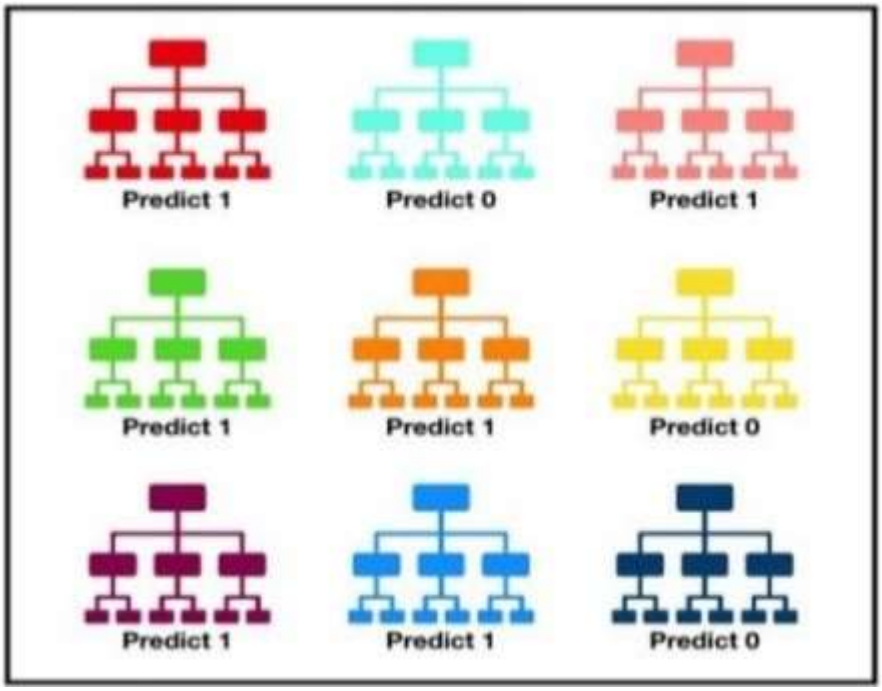
- Description: Develops an adaptive model capable of adjusting to changes in the game, such asrule modifications and team strategies.

- Components:
  - Continuous Learning Algorithms.
  - Dynamic Feature Updating.

This system architecture provides a structured and comprehensive framework for developing acricket score prediction system using machine learning. Each module plays a crucial role in theoverall functionality, ensuring accurate predictions, real-time adaptability, and user-friendly interactions.



DECISION TREE STRUCTURE



RANDOM FOREST STRUCTURE

CHAPTER 5
METHODOLOGIES

1. Objectives and Scope:

Objectives of the Cricket Score Prediction System:

1. Accurate Prediction: The primary objective of the system is to accurately predict the outcome ofcricket matches, including the scores that teams are likely to achieve.
2. Real-time Updates: Provide real-time updates during matches to continuously adjust predictionsbased on the unfolding events, such as wickets fallen, runs scored, and match conditions.
3. User Engagement: Enhance user engagement by offering interactive features such as livecommentary, statistical analysis, and graphical representations of match progress.
4. Data-driven Insights: Offer data-driven insights into player performance, team strategies, andmatch dynamics to assist viewers, analysts, and enthusiasts in understanding the game better.
5. Model Refinement: Continuously refine the prediction model based on historical match data, playerstatistics, and performance trends to improve accuracy and reliability over time.

Scope of the Project:

1. Match Formats: The system will cover various formats of cricket matches, including,
Test Matches
One Day Internationals (ODIs) Twenty20 Internationals (T20Is)
Domestic T20 Leagues (e.g., Indian Premier League, Big Bash League)

2. Types of Predictions:
Match Outcome Prediction: Predicting the winner of the match based on team composition, historical performance, and current form.
Score Prediction: Estimating the total runs scored by each team and individual player contributions.

Key Events Prediction: Anticipating significant events such as wickets, boundaries, and milestones (e.g., half-centuries, centuries).
Performance Analysis: Providing insights into player performance metrics like batting strike rate, bowling economy rate, and fielding efficiency.

3. Data Sources:

Historical match data including match results, player statistics, and venue conditions.
Real-time match data feeds providing live updates during matches.
External data sources for factors affecting gameplay such as
weather conditions, pitch reports, and team strategies.

4. Technology Stack:

Machine Learning and Statistical Modeling: Utilizing algorithms such as regression, classification, and time series analysis for prediction.
Data Processing and Integration: Implementing robust data pipelines to ingest, clean, and integrate diverse data sources.
Web and Mobile Application Development: Creating user-friendly interfaces for accessing predictions, live updates, and analysis.

By delineating these objectives and defining the project scope, the cricket scoreprediction system aims to offer valuable insights and an engaging experience for cricket enthusiasts, analysts, and fans worldwide.

CHAPTER 6

IMPLEMENTATION

6.1 REQUIREMENTS

6.1.1 Overall Description

A Software Requirements Specification (SRS) – a requirements specification fora software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with thesoftware. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design orimplementation (such as performance engineering requirements, quality standards, or design constraints).

6.1.2 Software Requirements

The software requirements for a cricket score prediction system can be comprehensive due to the complexity of the task

1. Programming Language: Choose a language suitable for data analysis and machine learning tasks, such as Python or R.

2. Machine Learning Libraries;

- Scikit-learn (Python): For implementing machine learning algorithms.
- TensorFlow: For deep learning-based approaches.
- XGBoost: For gradient boosting algorithms.

3. Data Processing Libraries: Use libraries for data manipulation and preprocessing, such as:

- Pandas (Python): For data manipulation and analysis.
- NumPy (Python): For numerical computing and array operations.

- SciPy (Python): For scientific computing tasks.

4. Web Development Framework (Optional): If developing a web application to display predictions and updates, consider using frameworks like:

- Django or Flask (Python): For backend development.

- React, Angular, or Vue.js: For frontend development.

5. Database Management System (DBMS): Choose a DBMS for storing and managing historical match data, player statistics, and user information, such as:

- PostgreSQL

- MySQL

- MongoDB (NoSQL)

6. Real-time Data Integration: Implement mechanisms to integrate real-time match data feeds into the system, such as web scraping or APIs provided by cricket data providers like ESPNcricinfo or Cricbuzz.

7. Version Control: Utilize version control systems like Git for managing codebase changes and collaboration among team members.

8. Deployment Platforms: Choose platforms for deploying the system, such as:

- Cloud platforms (e.g., AWS, Google Cloud Platform, Microsoft Azure)

- On-premises servers

9. Testing Frameworks: Use testing frameworks to ensure the reliability and accuracy of the prediction system, such as pytest (Python) for unit testing.

10. Monitoring and Logging: Implement monitoring and logging mechanisms to track system performance, errors, and user interactions.

11. Security Measures: Incorporate security measures to protect user data, ensure secure communication, and prevent unauthorized access.

6.2 Hardware Requirements

1. Laptop/PC/Mac
2. Processor : i5 and Above
3. Speed : 2.5 Ghz and Above
4. RAM : 8GB and Above
5. Hard disk : 10 GB of free space required
6. Mouse : Optical mouse
7. GPU (Graphics Processing Unit)
8. Networking
9. Backup and Redundancy, protect against data loss and system failures.

6.3 CODE SNIPPETS

```
import streamlit as st    "streamlit": Unknown word.
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
import numpy as np
```

Fig 6.1: Code for Importing Python Libraries

The script utilizes various modules from the Python data science ecosystem, including pandas for data manipulation, scikit-learn for machine learning tasks, and NumPy for numerical computing.

Libraries Used:

α. Streamlit: Streamlit is a powerful library used for creating interactive web applications with simple Python scripts. It allows developers to build data-drivenapplications effortlessly.

β. Pandas: Pandas is a versatile library widely used for data manipulation and analysisin Python. It provides powerful data structures and tools for handling structured dataefficiently.

χ. scikit-learn (sklearn): Scikit-learn is a popular machine learning library in Pythonthat provides simple and efficient tools for data mining and data analysis. It offers various algorithms for classification, regression, clustering, and more.

δ. NumPy: NumPy is a fundamental package for scientific computing with Python. It provides support for multidimensional arrays, mathematical functions, and randomnumber generation, making it essential for numerical computations.

Functionality:

1. Data Loading: The script loads a dataset using pandas' pd.read_csv() function. The dataset likely contains historical cricket match data or relevant features for predicting cricket scores.
2. Data Preprocessing: It performs basic preprocessing steps such as splitting the dataset into features (X) and target variable (y) using the train_test_split() function from scikit-learn. Additionally, it may include data cleaning, handling missing values, and feature engineering if necessary.
3. Model Training: The script trains machine learning models for predictive analysis using scikit-learn. Two regression models, Linear Regression and Random Forest Regression, are instantiated and fitted to the training data.
4. Model Evaluation: The models' performance may be evaluated using metrics such as mean squared error, R-squared, or others, although this part is not included in the provided code snippet.

```
# Load the dataset
dataset = pd.read_csv('ipl.csv')
```

Fig 6.2: Code for Loading the DataSet

Description of Code:

The provided code snippet loads a dataset from a CSV file named "ipl.csv" using the pandas library inPython.

Libraries Used:

ε. Pandas: Pandas is a powerful library widely used for data manipulation and analysis in Python. It provides efficient data structures and tools for handling structured data, such as DataFrames, making it ideal for loading and working with datasets.

Functionality:

1. Data Loading: The code snippet utilizes the pd.read_csv() function from the pandas library to read the contents of a CSV file named "ipl.csv". This CSV file likely contains data related to Indian Premier League (IPL) matches, such as match statistics, player performances, or match outcomes.
2. Dataset Assignment: The data read from the CSV file is assigned to a variable named "dataset", which represents a pandas DataFrame. The DataFrame organizes the data into rowsand columns, allowing for easy manipulation and analysis.

```
# Extracting input features and labels
X = dataset.iloc[:, [7, 8, 9, 12, 13]].values  # Input features    "iloc": Unknown word.
y = dataset.iloc[:, 14].values  # Label    "iloc": Unknown word.

# Splitting the dataset into the Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

Fig 6.3: Code for Extracting Input features & splitting the Dataset into Training set &Test set

Description of Code:

The provided code snippet extracts input features and labels from the dataset and splits the data intotraining and test sets for machine learning model training and evaluation.

Functionality:

1. Feature Extraction (Input Features):
    - The code snippet extracts input features from the dataset for predictive modeling.
    - It uses the iloc indexer of the pandas DataFrame to select specific columns as inputfeatures.
    - The input features are stored in a variable named "X", which represents a NumPyarray containing the feature values.
2. Label Extraction:
    - Similarly, the code snippet extracts the target variable or label from the dataset.
    - It selects a specific column as the label using the iloc indexer and stores it in avariable named "y".
    - The label variable "y" represents a NumPy array containing the target values to bepredicted.
3. Splitting the Dataset:
    - The code further splits the dataset into training and test sets using the train_test_split() function from scikit-learn.
    - It divides the input features (X) and labels (y) into two sets: training data (X_train,y_train) and test data (X_test, y_test).
    - The test_size parameter specifies the proportion of the dataset to include in the testset, while random_state ensures reproducibility by fixing the random seed for the split.

```
# Feature Scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Fig 6.4: Feature Scaling

Description of Code:

The provided code snippet performs feature scaling on the input features to ensure that they are on the same scale, which is a common preprocessing step in machine learning workflows.

Description of Code:

The provided code snippet performs feature scaling on the input features to ensure that they are on the same scale, which is a common preprocessing step in machine learning workflows.

Functionality:

1. Feature Scaling:
    - Feature scaling is a technique used to standardize the range of independent variables or features in the dataset.
    - In this code snippet, the StandardScaler class from scikit-learn is utilized for featurescaling.
    - The StandardScaler standardizes features by removing the mean and scaling them tounit variance, resulting in a mean of 0 and a standard deviation of 1 for each feature.
    - Feature scaling is particularly important for algorithms that are sensitive to the scale of input features, such as gradient descent-based algorithms, SVMs, and neuralnetworks.
2. Transformation:
    - The fit_transform() method of the StandardScaler object is applied to the training data (X_train) to compute the mean and standard deviation of each feature and then standardize the features accordingly.
    - The transform() method is used to apply the same transformation to the test data (X_test) using the parameters learned from the training data.
    - This ensures that both the training and test datasets are scaled using the same transformation, maintaining consistency in feature scaling across datasets.

```
# Function to predict score
def predict_score(model, input_data):
    return model.predict(sc.transform(np.array([input_data])))[0]
```

Fig 6.5: function to predict score

Description of Code:

The provided code snippet defines a function named predict_score that facilitates the prediction ofcricket scores using a trained machine learning model.

Functionality:

1. Predict Score Function:
    - The predict_score function takes two parameters:
        - model: This parameter represents the trained machine learning model usedfor making predictions. It is assumed that the model has been trained to predict cricket scores based on input features.
        - input_data: This parameter represents the input data for which the score prediction is to be made. It is expected to be an array-like object containingthe relevant features required for prediction.
    - Inside the function, the input data is first transformed using the transform method ofthe StandardScaler object (sc) to ensure that it is scaled appropriately before making predictions.
    - The transformed input data is then passed to the predict method of the providedmachine learning model (model) to obtain the predicted score.
    - Finally, the predicted score is returned as the output of the function.

6.4 EXECUTION

```
# Streamlit app    "Streamlit": Unknown word.
st.title('Cricket Score Prediction')
```

Fig 6.6: Streamlit

Description of Code:

The provided code snippet initiates the development of a Streamlit web application titled "Cricket Score Prediction," which is designed to provide users with an interactive platform for predicting cricketscores.

Functionality:

1. Streamlit App Initialization:

- The st.title() function is used to set the title of the Streamlit web application. In this case, it sets the title to "Cricket Score Prediction," indicating the purpose or focus of the application.

Usage Context:

ϕ. Streamlit: Streamlit is a popular Python library used for building interactive web applicationswith ease. It allows developers to create data-driven applications using simple Python scripts, eliminating the need for complex web development frameworks.

γ. Title: The title serves as the primary identifier or heading for the web application, providing users with an initial indication of the application's purpose or content.

Step – 1: (EXECUTION OF THE CODE)

```
# Input features
runs = st.number_input('Runs', value=100)
wickets = st.number_input('Wickets', value=0)
overs = st.number_input('Overs', value=13)
striker = st.number_input('Striker', value=50)
non_striker = st.number_input('Non-Striker', value=50)
```

Fig 6.7: Step – 1 Showing the Model Training step

Description of Code:

The provided code snippet defines input fields using Streamlit for users to input relevant features required forpredicting cricket scores.

Functionality:

1. Streamlit Input Fields:

    - The st.number_input() function from Streamlit is used to create input fields fornumeric values, allowing users to input data interactively.
    - Each input field corresponds to a specific feature required for predicting cricket scores.The features include:
        - Runs: The total number of runs scored by the batting team.
        - Wickets: The number of wickets lost by the batting team.
        - Overs: The total number of overs bowled in the match.
        - Striker: The batting strike rate of the current striker.
        - Non-Striker: The batting strike rate of the non-striker.
    - The value parameter sets the default value for each input field, providing users with astarting point for input.

Usage Context:

η. Input Fields: Input fields allow users to provide specific data or parameters required for analysis, prediction, or computation in the application.

ι. Numeric Input: In this case, numeric input fields are used to

capture numerical features relevant to the current state of a cricket match, such as runs scored, wickets fallen, and overs bowled.

Step – 2: Implementation

```
# Choose model
model_choice = st.radio('Choose Model', ('Linear Regression', 'Random Forest Regression'))

if st.button('Predict'):
    if model_choice == 'Linear Regression':
        lin_regressor = LinearRegression()
        lin_regressor.fit(X_train, y_train)
        prediction = predict_score(lin_regressor, [runs, wickets, overs, striker, non_striker])
    elif model_choice == 'Random Forest Regression':
        rf_regressor = RandomForestRegressor(n_estimators=100, max_features=None, random_state=0)
        rf_regressor.fit(X_train, y_train)
        prediction = predict_score(rf_regressor, [runs, wickets, overs, striker, non_striker])

    st.success(f'Predicted Score: {prediction}')
```

Fig 6.8: Step – 2 Showing the Implementation step

Description of Code:

The provided code snippet allows users to choose between two machine learning models, either Linear Regression or Random Forest Regression, for predicting cricket scores based on input features. After selecting a model and providing input data, the application predicts the cricket scoreusing the chosen model and displays the result to the user.

Functionality:

1. Model Selection:
    - The st.radio() function from Streamlit is used to create a radio button group for users to choose between two models: Linear Regression and Random Forest Regression. Users can select one of the available options to specify the model theywant to use for prediction.
2. Model Prediction:
    - Upon clicking the "Predict" button, the application checks the selected model choice.
    - If the user chooses Linear Regression, a LinearRegression model is instantiated, trained on the training data (X_train, y_train), and used to predict the cricket scorebased on the provided input features.
    - If the user selects Random Forest Regression, a RandomForestRegressor

model isinstantiated with specified parameters, trained on the training data, and used for prediction.

3. Display Prediction:
    - The predicted cricket score is displayed to the user as a success message using the st.success() function from Streamlit, providing immediate feedback on the predictionoutcome.

CHAPTER 7

RESULT ANALYSIS

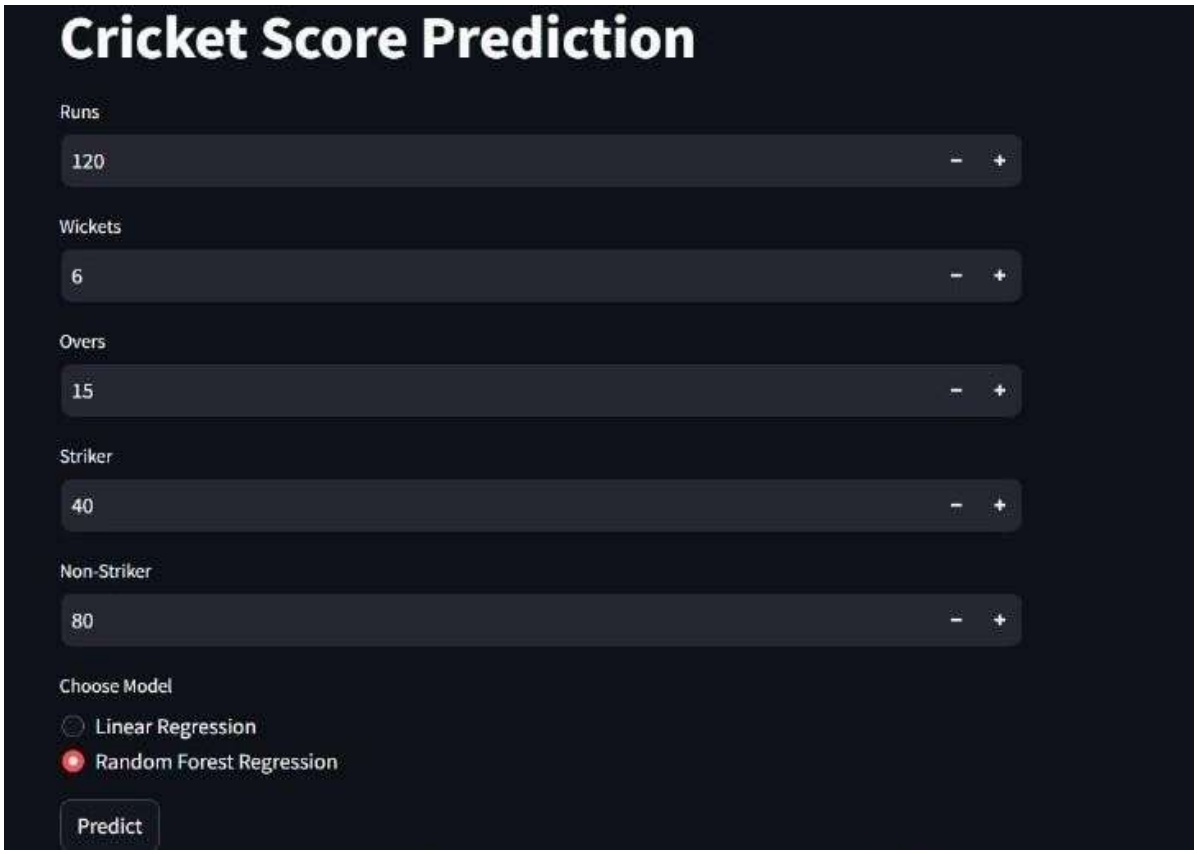


Fig 7.1: Cricket Score prediction Output

1. Accuracy Metrics

- Calculate various accuracy metrics such as:
  - Accuracy: The percentage of correct predictions out of total predictions made.
  - Precision: The ratio of true positive predictions to the total positive predictions made.
  - Recall: The ratio of true positive predictions to the total actual positive instances.
  - F1 Score: The harmonic mean of precision and recall, providing a balanced measure of prediction accuracy.
- These metrics help assess the overall performance of the prediction system across different aspects of prediction quality.

2. Error Analysis:

- Analyze prediction errors to identify patterns and common sources of inaccuracies.

- Categorize errors based on factors such as match format, team composition, player performance, and match conditions.
- Investigate cases where the prediction system failed to anticipate significant match events or unexpected outcomes.

3. Confusion Matrix:

- Construct a confusion matrix to visualize the distribution of predicted outcomes against actual outcomes.
- Analyze the distribution of true positives, true negatives, false positives, and false negatives to gain insights into prediction strengths and weaknesses.

4. Feature Importance:

- Determine the importance of features used in the prediction model to understand which factors contribute most to accurate predictions.

5. Cross-validation:

- Perform cross-validation to assess prediction consistency and robustness across different subsets of the dataset.
- Use techniques such as k-fold cross-validation to train and evaluate the prediction model on multiple partitions of the dataset, ensuring unbiased performance estimation.

6. Time-series Analysis:

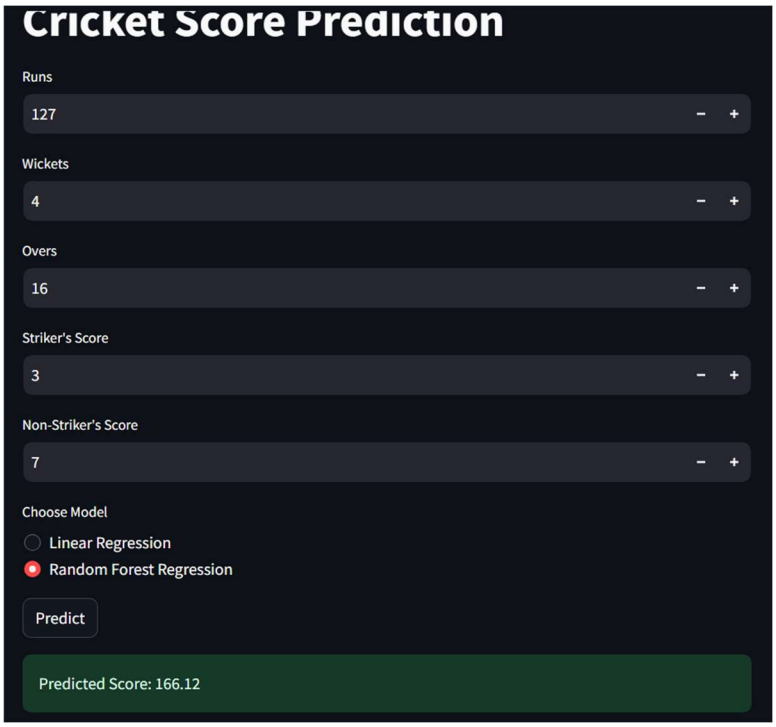
- For real-time prediction systems, analyze prediction performance over time to detect trends, seasonality, and changes in prediction accuracy.
- Monitor prediction errors and recalibrate the model periodically to adapt to evolving match dynamics and performance trends.

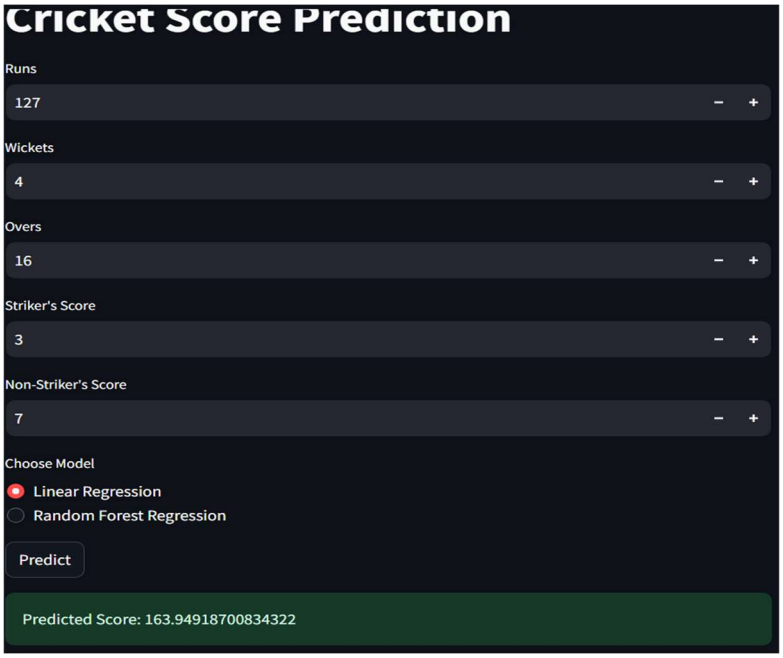
7. Comparative Analysis:

- Compare the performance of the prediction system against baseline models or alternative prediction approaches to assess its relative effectiveness.

- Benchmark the prediction system against industry standards or expert predictions to validate its accuracy and reliability.

-





CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

- The cricket score prediction system offers an innovative approach to engage cricket enthusiasts, analysts, and fans by providing accurate predictions and valuable insights into match outcomes and player performance.
- By leveraging machine learning algorithms, real-time data integration, and user-friendly interfaces, the system aims to enhance the cricket viewing experience and deepen the understanding of the game dynamics.
- Through meticulous planning and execution of software and hardware requirements, the system can deliver reliable predictions and seamless user experiences.
- However, continuous refinement and enhancements are essential to adapt to evolving match conditions, player dynamics, and user preferences.

8.2 FUTURE ENHANCEMENT

1. Advanced Machine Learning Models: Explore the use of advanced machine learning techniques such as deep learning and ensemble methods to improve prediction accuracy andhandle complex match scenarios effectively.

2. Enhanced Real-time Updates: Integrate more comprehensive real-time data sources andimplement advanced algorithms for faster and more accurate updates during matches, including predictive analysis of ongoing match events.

3. Interactive User Features: Enhance user engagement with interactive features such aslive chat, social media integration, and personalized recommendations based on user preferences and historical interactions.

Predictive Analytics: Expand the scope of predictions to include advanced analytics suchas player performance trends, team strategies, and match simulations to provide deeper insights into match dynamics and outcomes.

4. Mobile Application Development: Develop mobile applications for iOS and Android platforms to reach a wider audience and enable access to predictions and updates on-the-go.

5. Community Participation: Implement features for user-generated content, such as prediction contests, fantasy leagues, and user-contributed insights, to foster a vibrant community of cricket enthusiasts and analysts.

6. Multi-format Support: Extend support for predicting outcomes and analyzing performance across various cricket formats, including domestic leagues, women's cricket,and age-group tournaments.

7. Integration with Broadcast Platforms: Collaborate with broadcasting networks to integrate prediction overlays and analytics into live match broadcasts, enhancing the viewingexperience for television and online audiences.

8. Localization and Globalization: Customize the system for different regions andlanguages to cater to diverse cricket fanbases worldwide, incorporating local matchschedules, player profiles, and cultural preferences.

9. Continuous Feedback and Improvement: Solicit feedback from users and stakeholders to identify areas for improvement and prioritize enhancements, ensuring the system remains relevant and valuable in the dynamic landscape of cricket analytics and entertainment.

REFERENCES

[1] Choudhary, A., & Singh, A. (2020). "Cricket Match Result Prediction using Machine Learning Algorithms." InternationalJournal of Advanced Computer Science and Applications.

[2] Khan, M. A., & Gulzar, S. (2018). "Predictive Modeling for Cricket Match Outcome using Machine Learning Algorithms." Procedia Computer Science

[3] D. Thenmozhi, P. Mirunalini, S. M. Jaisakthi, S. Vasudevan, V. Veeramani Kannan, S. Sagubar Sadiq, MoneyBall - Data Mining on Cricket Dataset, in Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India (2019) 4. A. N.

[4] Wickramasinghe, R. D. Yapa, Cricket Match Outcome Prediction Using Tweets and Prediction of the Man of the Match using Social Network Analysis: Case Study Using IPL Data, in Proceedings of the 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTER), Colombo, Sri Lanka (2018) 5. N. Rodrigues, N. Sequeira, S. Rodrigues, V.

[5] Shrivastava, Cricket Squad Analysis Using Multiple Random Forest Regression, in Proceedings of the 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, (2019)