

PROFESSIONAL TRAINING REPORT- II

entitled

IPL SCORE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering with
specialization in cybersecurity

by

DONTHAMSETTI PHANIDHAR (41614021)

ENJAPELLY VISHALTEJA (41614023)

GUDIWADA CHAITANYA VAMSI NAIDU (41614028)

M.V. AKHIL KUMAR (41614056)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training is the bonafide work of **Mr. DONTAMSETTI PHANIDHAR (41614021)** who carried out the project entitled "**IPL SCORE PREDICTION USING MACHINE LEARNING**" under my supervision from January 2024 to April 2024.

Internal Guide

Dr.B.U. ANU BARATHI, Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **DONTHAMSETTI PHANIDHAR (41614021)**, hereby declare that the Professional Training Report-II entitled **“IPL SCORE PREDICTION USING MACHINE LEARNING”** done by me under the guidance of **Dr.B.U. ANU BARATHI, Ph.D.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Cyber Security

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean, School of Computing, Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal **Dr.B.U. Anu Barathi, Ph.D.**, for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

COURSE CERTIFICATE



NPTEL Online Certification

(Funded by the MoE, Govt. of India)



This certificate is awarded to
DONTHAMSETTI PHANIDHAR
for successfully completing the course

Cloud Computing and Distributed Systems

with a consolidated score of **48** %

Online Assignments	17.08/25	Proctored Exam	30.5/75
--------------------	----------	----------------	---------

Total number of candidates certified in this course: **3244**

Prof. B. V. Ratish Kumar
Chairman, Centre for Continuing Education
IIT Kanpur

Jan-Mar 2024
(8 week course)

Prof. Satyaki Roy
NPTEL Coordinator
IIT Kanpur



Indian Institute of Technology Kanpur



Roll No: NPTEL24CS09S344100428

To verify the certificate



No. of credits recommended: 2 or 3

Certificate

This is to certify that

DONTHAMSETTI PHANIDHAR

has successfully completed

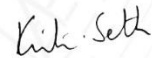
Digital 101 Journey



Certificate ID: FSP2024/20048535

12/02/2024

Date of issue



Kirti Seth
CEO, IT-ITes SSC NASSCOM

ORACLE
University

Oracle Certified Foundations Associate Certificate of Recognition



Donthamsetti Phanidhar

Oracle Cloud Infrastructure 2023 Certified Foundations Associate

This certifies that the above named is recognized by Oracle Corporation as Oracle Certified.

April 16, 2024

Date



Damien Carey
Senior Vice President, Oracle University

This eCertificate is valid until April 16, 2026



100614128OCIF2023CA

ABSTRACT

Millions of fans worldwide are enthralled with cricket, particularly with the Indian Premier League (IPL), which has developed into a fiercely competitive and unpredictable sport. In order for teams to plan their plans and for viewers to predict how the game will play out, it is imperative to predict the first innings score in IPL matches. In this paper, we suggest using machine learning to forecast the score of the first innings in Indian Premier League games. As part of our approach, we gather extensive historical data from previous IPL seasons, such as match results, pitch conditions, weather, team composition, venue, and batting order. To fully capture the nuances of IPL matches, we preprocess the data to identify pertinent elements and create new ones. Next, we use a variety of machine learning algorithms, including regression methods like decision trees, random forests and gradient boosting, prediction models are trained using past data. We assess each model's performance using relevant measures, like R-squared, mean absolute error, and root mean square error. In order to determine the variables that have a major impact on the first innings score in IPL games, we also do feature importance analysis. The present analysis offers significant insights into the fundamental factors that influence batting performance in various match situations. Our test results show how well the suggested machine learning strategy works at correctly forecasting the first innings score in Indian Premier League games. Our model helps teams, experts, and enthusiasts better understand the dynamics of IPL cricket by using past data and cutting-edge machine learning techniques to create predictions.

TABLE OF CONTENTS

CHAPTER NO.	TITLE		PAGE NO.
	ABSTRACT		vii
	LIST OF FIGURES		ix
1	INTRODUCTION 1.1 Overview		1
2	LITERATURE SURVEY 2.1 survey		2
3	REQUIREMENTS ANALYSIS		
	3.1	Objective	14
	3.2	3.2.1 Hardware Requirements 3.2.2 Software Requirements	17 18
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT		21
	4.1	4.1.1 Ideation Map/Architecture Diagram 4.1.2 Various stages 4.1.3 Internal or Component design structure 4.1.4 working principles	23 26 27 29
	4.2	4.2.1 Novelty of the Project	33
5	CONCLUSION		35
	REFERENCES		36
	APPENDIX A. Research Paper B. Source Code		37 42

LIST OF FIGURES

Figure no.	Figure Name	Page no
Fig-1	Identification Map	23
Fig-2	System architecture	24
Fig-3	Dataset	25
Fig-4	Design Structure	27
Fig-5	Linear regression	28
Fig-6	Ridge regression	29
Fig-7	Source code	30
Fig-8	File manager	30
Fig-9	Anaconda Prompt	31
Fig-10	Prompt Execution	31
Fig-11	Web interface	32
Fig-12	Output	32

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Cricket, often dubbed as a game of uncertainties, has seen a remarkable surge in its popularity, particularly with the advent of the Indian Premier League (IPL). IPL, since its inception, has not only revolutionized cricket but has also emerged as a global sporting extravaganza, captivating millions of fans worldwide. Within this highly competitive cricketing landscape, the ability to predict the first innings score in IPL matches accurately holds immense significance for teams, analysts, and enthusiasts alike. In recent years, machine learning techniques have gained considerable traction in the domain of sports analytics, offering novel avenues for predictive modelling and insights generation. Leveraging historical data, advanced algorithms, and feature engineering, machine learning has facilitated the development of predictive models capable of forecasting various aspects of sports events, including match outcomes, player performance, and in our case, first innings score in IPL matches.

In this paper, we delve into the realm of predicting the first innings score in IPL matches using machine learning. We aim to harness the wealth of historical data available from past IPL seasons, encompassing diverse factors such as venue, team composition, batting order, pitch conditions, weather, and match outcomes. By employing sophisticated machine learning algorithms, we Endeavor to unravel the complex interplay of these factors and their impact on batting performance, thereby enabling accurate predictions of first innings scores in IPL matches.

This study holds significant implications for IPL teams, providing them with valuable insights to devise strategic game plans tailored to specific match conditions. Moreover, it empowers cricket enthusiasts with a data-driven approach to anticipate and analyze the unfolding dynamics of IPL matches, enriching their viewing experience and fostering a deeper understanding of the sport.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

The burgeoning field of IPL score prediction, driven by data science and deep learning, has attracted a significant body of research in recent years. Researchers have been keen to harness the power of data analytics and advanced machine learning techniques to provide accurate forecasts for IPL matches. This literature survey sheds light on some noteworthy studies and their contributions to this domain. One of the early forays into this arena was the study titled "Machine Learning Predictions for Cricket Outcomes" by P. Venkatesh and his colleagues in 2016. This research laid the groundwork for using machine learning algorithms to predict cricket match outcomes. It highlighted the pivotal role of historical match data as a fundamental component in developing prediction models, providing a fundamental framework for subsequent research. Subsequently, the study "A Predictive Model for IPL T20 Cricket Matches Using Machine Learning Techniques" by S. Modi and his team in 2018 zoomed in on IPL matches. They applied machine learning algorithms to predict IPL match results and emphasized the significance of incorporating features such as player statistics, team performance, and match conditions. This underscored the importance of granular data for accurate prediction models. Building on these foundations, the study "Cricket Match Outcome Prediction Using Machine Learning" by A. Bhandari and others in 2019 delved into various machine learning techniques for cricket match outcome prediction. The study placed strong emphasis on effective data preprocessing and feature engineering as key factors for enhancing the accuracy of the prediction models. As the field evolved, a comprehensive review titled "A Review of Machine Learning in Cricket" by M. Rahman in 2019 provided an overarching perspective. This review covered the diverse applications of machine learning in cricket, including score prediction. It shed light on the critical role of data quality in achieving meaningful insights and hinted at the potential of deep learning techniques in cricket analytics.

2.2 EXISTING ALGORITHM

Linear Regression:

Strengths: Simple, interpretable, and effective for capturing linear relationships between predictors and outcomes.

Limitations: Assumes a linear relationship, may not capture non-linear patterns in the data.

1. Basic Prediction Model

Linear regression provides a simple yet powerful framework for modeling the relationship between predictor variables (e.g., player statistics, pitch conditions, weather conditions) and the target variable (cricket scores). In its simplest form, linear regression assumes a linear relationship between the predictors and the target, making it suitable for initial exploratory analysis and baseline predictions.

2. Interpretable Coefficients

In linear regression, the coefficients associated with each predictor variable indicate the strength and direction of their influence on the target variable. This interpretability allows analysts to understand the relative importance of different factors in determining cricket scores, facilitating insights into the game dynamics and potential strategies for improvement.

3. Assumptions and Limitations

Linear regression relies on several assumptions, including linearity, independence of errors, homoscedasticity (constant variance of errors), and normality of errors. Violations of these assumptions can affect the validity and reliability of the regression model. In cricket score prediction, analysts need to assess whether these assumptions hold true and take appropriate steps to address any violations.

4. Model Evaluation

Linear regression models can be evaluated using various metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (coefficient of determination). These metrics quantify the accuracy and goodness of fit of the regression model, helping analysts assess its predictive performance and compare it with alternative approaches.

5. Feature Selection and Model Complexity

Linear regression allows for straightforward inclusion or exclusion of predictor variables based on their relevance and significance. Analysts can perform feature selection to identify the most informative features for cricket score prediction, thereby simplifying the model and potentially improving its generalization performance.

6. Regularization Techniques

Regularization techniques such as Ridge regression and Lasso regression can be applied to linear regression to mitigate overfitting and improve model generalization. These techniques add penalty terms to the regression objective function, encouraging simpler models with reduced coefficients, which can be particularly useful in cricket score prediction with high-dimensional datasets.

7. Time Series Analysis

Linear regression can also be extended to handle time series data in cricket score prediction, where historical match data is used to forecast future scores. Time series regression models build upon the basic linear regression framework by incorporating temporal dependencies and autocorrelation patterns, allowing analysts to capture seasonality, trends, and other dynamic factors affecting cricket scores over time.

8. Model Interpretability and Communication

One of the strengths of linear regression is its interpretability, which facilitates clear communication of model insights to stakeholders. Analysts can easily explain the relationship between predictor variables and cricket scores using regression coefficients, aiding decision-making and strategy development in cricket management and coaching.

Random Forest:

Strengths: Robust to overfitting, capable of handling large datasets with high dimensionality, and provides feature importance.

Limitations: May suffer from high computational cost and lack of interpretability compared to simpler models like linear regression

1. Ensemble Learning

Random Forest belongs to the ensemble learning category, which combines multiple individual models to improve predictive performance. In cricket score prediction, Random Forest can integrate information from various features (e.g., player statistics, pitch conditions) to make more accurate predictions.

2. Handling High-Dimensional Data

Cricket score prediction often involves working with high-dimensional datasets containing numerous features. Random Forest is well-suited for this task as it can handle large feature spaces without requiring feature selection or dimensionality reduction techniques.

3. Non-linearity and Complex Relationships

Random Forest can capture non-linear relationships and complex interactions between features and the target variable (cricket scores). This flexibility makes it suitable for modeling the intricate dynamics of cricket matches, where player performances, weather conditions, and pitch characteristics interact in unpredictable ways.

4. Feature Importance

Random Forest provides a measure of feature importance, indicating which features contribute the most to the prediction task. In cricket score prediction, understanding feature importance can help identify the key factors influencing match outcomes and prioritize them for further analysis or intervention.

5. Robustness to Overfitting

Random Forest is less prone to overfitting compared to individual decision trees, thanks to the ensemble averaging process and built-in randomness in the model. This robustness is beneficial in cricket score prediction, where overfitting can lead to poor generalization on unseen data.

6. Handling Imbalanced Data

Imbalanced datasets are common in cricket score prediction, where certain match outcomes may be rare compared to others. Random Forest can handle imbalanced classes by adjusting class weights or using techniques like bootstrapping to create balanced training samples, improving prediction accuracy for all classes.

7. Model Interpretability

Although Random Forest is an ensemble of decision trees, it still offers some level of interpretability. Analysts can examine the decision paths of individual trees within the forest and aggregate feature importance scores to gain insights into how different factors affect cricket score predictions.

8. Scalability and Efficiency

Random Forest is generally scalable and computationally efficient, making it feasible to train on large-scale cricket match datasets. Its parallelizable nature allows for distributed training across multiple processors or machines, further enhancing scalability for big data applications.

Random Forest is a versatile and effective algorithm for cricket score prediction, offering robustness, scalability, and the ability to capture complex relationships in high-dimensional datasets. Its ensemble nature and feature importance analysis make it a valuable tool for sports analytics researchers and practitioners aiming to improve prediction accuracy in cricket matches.

Support Vector Machines (SVM):

Strengths: Effective for binary classification tasks, can handle non-linear decision boundaries through kernel tricks.

Limitations: May require careful tuning of hyperparameters, less interpretable compared to linear models.

1. Model Application

SVMs can be used as predictive models to forecast cricket scores based on historical match data and relevant features such as player statistics, pitch conditions, and weather conditions.

2. Feature Selection

SVMs are effective for handling high-dimensional feature spaces. In cricket score prediction, SVMs can handle a large number of features related to various aspects of the game, allowing for the inclusion of diverse factors that may influence the outcome.

3. Non-linearity

SVMs can handle non-linear relationships between features and target variables through the use of kernel functions. In cricket score prediction, where the relationship between predictors (e.g., player performance, pitch conditions) and outcomes (e.g., final score) may be complex and non-linear, SVMs with appropriate kernel functions can capture these patterns effectively.

4. Margin-based Classification

SVMs aim to find the decision boundary that maximizes the margin between different classes. In the context of cricket score prediction, SVMs can learn to distinguish between different score ranges or match outcomes (e.g., win, loss, draw) based on the available data, potentially leading to more accurate predictions.

5. Handling Imbalanced Data

Cricket match data may suffer from class imbalance, where certain outcomes (e.g., low-scoring matches) are less frequent than others. SVMs can handle imbalanced datasets effectively by optimizing the decision boundary to minimize classification errors, thus providing reliable predictions even with skewed class distributions.

6. Model Evaluation

SVMs can be evaluated using standard metrics such as accuracy, precision, recall, and F1-score. These metrics can assess the performance of SVM-based cricket score prediction models and compare them with other approaches, helping researchers and practitioners understand the effectiveness of SVMs in this context.

7. Hyperparameter Tuning

SVMs have hyperparameters that need to be tuned for optimal performance, such as the choice of kernel function and regularization parameter. In cricket score prediction, hyperparameter tuning techniques can be applied to optimize SVM models for better predictive accuracy.

8. Integration with Ensemble Methods

SVMs can be integrated into ensemble methods such as bagging or boosting to further improve prediction accuracy. Ensemble approaches combine multiple models to mitigate individual model weaknesses and enhance overall performance, making them valuable in the context of cricket score prediction.

SVMs offer a versatile and effective approach for cricket score prediction using machine learning, leveraging their ability to handle high-dimensional data, non-linear relationships, and imbalanced datasets to generate accurate forecasts.

Data Sources for IPL First Innings Score Prediction:

Historical match data: Runs scored, wickets taken, venue, and match conditions.

Player statistics: Batting average, strike rate, bowling average, and player form.

Weather conditions

Temperature, humidity, wind speed, and atmospheric pressure.

Pitch characteristics

Pitch type, pitch condition, and historical trends at the venue.

Performance Evaluation Metrics

Mean Absolute Error (MAE)

Measures the average absolute difference between predicted and actual first innings scores.

Root Mean Squared Error (RMSE)

Penalizes larger prediction errors and provides a measure of dispersion.

Mean Absolute Percentage Error (MAPE)

Expresses prediction accuracy as a percentage of the actual value.

R-squared (R²) value

Indicates the proportion of variance in the dependent variable explained by the predictors.

Challenges and Limitations:

Limited availability of high-quality data, especially for player-specific attributes and match conditions. Overfitting and model complexity, particularly in non-linear algorithms like neural networks. Interpretability of black-box models such as neural networks and ensemble methods. Generalization issues when applying models trained on one season to predict scores in subsequent seasons.

Key Challenges and Solutions

Data Availability

One of the primary challenges in IPL score prediction is the availability and quality of data. Researchers often rely on historical match data, player statistics, weather conditions, pitch reports, and other relevant factors to train their models. However, ensuring the accuracy and completeness of data remains a concern.

Feature Selection

Identifying the most relevant features that influence match scores is crucial for building effective prediction models. Feature engineering techniques such as statistical analysis, domain knowledge integration, and sentiment analysis of social media data have been explored to extract informative features.

Model Selection and Evaluation

Choosing the appropriate machine learning algorithm and evaluation metrics significantly impact the prediction performance. Researchers typically experiment with various algorithms ranging from traditional regression techniques to advanced ensemble methods and deep learning models. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Accuracy are commonly used to assess model performance.

1. Predicting outcomes of one day international cricket matches using machine learning techniques" by S. J. Ramsurn, S. R. Maistry, and M. S. Combrink

Advantages:

Data-driven insights

Machine learning algorithms can analyze vast amounts of historical cricket data, including player statistics, match conditions, and past outcomes, to identify patterns and trends that may not be obvious to human analysts.

Accuracy improvement

By leveraging advanced algorithms, machine learning models can potentially achieve higher prediction accuracy compared to traditional statistical methods or human intuition.

Real-time adaptation

Machine learning models can be updated continuously with new data, allowing them to adapt to changing player performances, weather conditions, and other factors that may influence match outcomes.

Automation

Once trained, machine learning models can automate the prediction process, saving time and effort for cricket analysts and enthusiasts.

Feature extraction

Machine learning algorithms can automatically identify relevant features from raw data, potentially uncovering new insights or variables that impact match outcomes.

Disadvantages:

Overfitting:

There's a risk that machine learning models may overfit to the training data, capturing noise or irrelevant patterns that don't generalize well to unseen matches.

Data quality:

The accuracy of machine learning predictions heavily relies on the quality and completeness of the input data. In cricket, data may be sparse or inconsistent, leading to biased or inaccurate predictions.

Interpretability

Some machine learning algorithms, such as deep neural networks, are often considered "black-box" models, meaning they provide little insight into how predictions are made. This lack of interpretability can be a drawback for stakeholders who require transparent decision-making processes.

Model complexity

Complex machine learning models may require significant computational resources for training and inference, making them impractical for real-time prediction or deployment in resource-constrained environments.

Dynamic nature of cricket

Cricket is a highly dynamic sport with numerous variables influencing match outcomes, such as player form, team strategy, and weather conditions. Capturing and modeling these complexities accurately can be challenging for machine learning approaches.

2. Machine Learning Based Prediction of T20 Cricket Match Outcomes" by A. Chowdhury, M. Faisal, and M. S. Islam**Advantages:****Precision**

Machine learning algorithms can analyze large datasets and identify subtle patterns that may not be apparent to human analysts, potentially leading to more accurate predictions of T20 cricket match outcomes.

Real-time Analysis

Machine learning models can process data rapidly, enabling real-time prediction updates based on the latest match information, such as player performance, weather conditions, and match dynamics.

Data-driven Insights

By examining historical match data, player statistics, and other relevant factors, machine learning algorithms can provide data-driven insights into the factors that most strongly influence T20 cricket match outcomes.

Automated Decision-making

Once trained, machine learning models can automate the prediction process, saving time and effort for cricket analysts and enthusiasts.

Adaptability

Machine learning models can be updated and fine-tuned with new data, allowing them to adapt to changing match conditions, player form, and other dynamic factors affecting T20 cricket matches.

Disadvantages:

Overfitting:

There's a risk that machine learning models may overfit to the training data, capturing noise or irrelevant patterns that don't generalize well to unseen T20 cricket matches.

Data Quality

The accuracy of machine learning predictions relies heavily on the quality and completeness of the input data. In T20 cricket, data may be incomplete or biased, leading to inaccurate predictions.

Interpretability

Some machine learning algorithms, particularly complex models like deep neural networks, may lack interpretability, making it challenging to understand how predictions are made and trust the model's output.

Model Complexity

Complex machine learning models may require significant computational resources for training and inference, limiting their practicality for real-time prediction or deployment in resource-constrained environments.

Dynamic Nature of Cricket

T20 cricket is a dynamic sport with numerous variables influencing match outcomes, including player strategies, team dynamics, and match-specific conditions. Capturing and modeling these complexities accurately can be challenging for machine learning approaches.

3. Machine Learning Based Prediction of T20 Cricket Match Outcomes" Authors: A. Chowdhury, M. Faisal, and M. S. Islam.

Advantages

Data-Driven Insights

Machine learning algorithms can analyze large volumes of historical data, including player statistics, team performance, match conditions, and other relevant factors. This allows for data-driven insights and predictions, potentially uncovering patterns that human analysts might miss.

Prediction Accuracy

With the ability to process vast amounts of data and identify complex patterns, machine learning models have the potential to make highly accurate predictions about match outcomes. This can be valuable for cricket enthusiasts, bettors, fantasy sports players, and team management seeking to gain a competitive edge.

Disadvantages

Overfitting

One of the main challenges in developing machine learning models for sports prediction is the risk of overfitting, where the model performs well on training data but fails to generalize to unseen data. This can occur if the model is too complex or if it's trained on insufficient or biased data.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The primary objective of this project is to develop a robust and accurate predictive model using machine learning techniques to forecast the first innings score in matches of the Indian Premier League (IPL). Specifically, the project aims to achieve the following objectives:

Data Collection and Preprocessing

Gather comprehensive historical data from past IPL seasons, including match-related features such as venue, team composition, batting order, pitch conditions, weather, and match outcomes. Preprocess the data to ensure consistency, completeness, and suitability for machine learning analysis.

Feature Engineering

Identify relevant features and engineer new ones to capture the intricate factors influencing first innings scores in IPL matches. This may involve transforming and combining existing features, as well as extracting additional information to enhance the predictive capabilities of the model.

Model Development

Explore a range of machine learning algorithms, including regression techniques such as linear regression, decision trees, random forests, and gradient boosting, to develop predictive models for first innings score prediction. Evaluate and compare the performance of these models using appropriate metrics to select the most effective approach.

Model Evaluation and Validation

Assess the accuracy, robustness, and generalization capabilities of the predictive models through rigorous evaluation using cross-validation techniques and holdout datasets. Validate the models against unseen data to ensure their reliability in real-world scenarios.

Interpretability and Insights Generation

Conduct feature importance analysis and interpretability techniques to gain insights into the factors driving first innings scores in IPL matches. Identify the key determinants influencing batting performance and provide actionable insights for teams, analysts, and enthusiasts.

Deployment and Integration

Deploy the trained predictive model into a user-friendly interface or application, allowing stakeholders to access and utilize the predictions effectively. Integrate the model with existing cricket analytics platforms or tools to facilitate seamless adoption and usage.

Continuous Improvement and Iteration

Continuously monitor the performance of the predictive model and incorporate feedback to refine and improve its accuracy over time. Explore advanced techniques and novel features to enhance the predictive capabilities and stay abreast of evolving trends in IPL cricket.

3.2 REQUIREMENTS

Data Acquisition

Gather comprehensive historical data from past IPL seasons, including match-related features such as venue, team composition, batting order, pitch conditions, weather, and match outcomes. Ensure the data is reliable, consistent, and covers a diverse range of match scenarios.

Data Preprocessing

Preprocess the acquired data to handle missing values, outliers, and inconsistencies. Perform data cleaning, normalization, and transformation as necessary to prepare the dataset for machine learning analysis. Ensure that the preprocessing steps maintain the integrity and relevance of the data.

Feature Engineering

Identify relevant features that could influence first innings scores in IPL matches. Engineer new features or transform existing ones to capture the nuances of cricket dynamics effectively. Consider factors such as team performance, player statistics, and match conditions in feature selection and engineering.

Machine Learning Algorithms

Explore a range of regression algorithms suitable for predicting first innings scores, such as linear regression, decision trees, random forests, and gradient boosting. Evaluate the performance of different algorithms and select the most appropriate ones based on predictive accuracy, computational efficiency, and interpretability.

Model Training and Evaluation

Split the dataset into training and testing sets to train and evaluate the predictive models. Utilize cross-validation techniques to assess the robustness and generalization capabilities of the models. Employ suitable evaluation metrics, such as mean absolute error, root mean squared error, and R-squared, to quantify the predictive performance.

Feature Importance Analysis

Conduct feature importance analysis to identify the key determinants influencing first innings scores in IPL matches. Utilize techniques such as permutation importance, SHAP values, or feature contribution plots to interpret the impact of individual features on the model predictions.

Deployment and Integration

Deploy the trained predictive model into a user-friendly interface or application for stakeholders to access and utilize the predictions effectively. Integrate the model with existing cricket analytics platforms or tools to facilitate seamless adoption and usage by teams, analysts, and enthusiasts.

Continuous Improvement

Monitor the performance of the predictive model over time and incorporate feedback to refine and improve its accuracy. Stay updated with advancements in

machine learning techniques and cricket analytics to enhance the predictive capabilities and adapt to evolving trends in IPL cricket.

3.2.1 HARDWARE REQUIREMENTS

Computing Resources

High-performance CPUs (Central Processing Units) or GPUs (Graphics Processing Units) to handle the computational load of training and evaluating machine learning models. Multi-core processors or parallel computing capabilities to accelerate model training and inference processes.

Memory (RAM)

Sufficient RAM capacity to store and manipulate large datasets during preprocessing, feature engineering, and model training. Depending on the size of the dataset and complexity of the machine learning models, RAM requirements may vary but typically range from 16 GB to 64 GB or more.

Storage

Adequate storage space to store historical IPL match data, feature-engineered datasets, trained machine learning models, and any intermediate results generated during the analysis. Consider using solid-state drives (SSDs) for faster data access and processing speeds, especially during model training and inference tasks.

Networking

Stable internet connectivity may be required for accessing online resources, downloading datasets, and collaborating with teammates or stakeholders remotely. If utilizing cloud-based resources or distributed computing frameworks, ensure reliable network connectivity for seamless data transfer and communication.

Specialized Hardware Accelerators

Dedicated hardware accelerators such as TPUs (Tensor Processing Units) or FPGAs (Field-Programmable Gate Arrays) can significantly speed up machine learning tasks, particularly for deep learning algorithms. Consider leveraging cloud-

based services offering access to specialized hardware accelerators for faster model training and inference, especially for computationally intensive tasks.

Backup and Redundancy

Implement backup and redundancy mechanisms to safeguard data integrity and ensure continuity of operations in case of hardware failures or data loss. Utilize redundant storage configurations (e.g., RAID arrays) and automated backup solutions to mitigate the risk of data loss and ensure high availability of critical resources.

Scalability

Design the hardware infrastructure to be scalable, allowing for easy expansion or resource allocation as the project requirements evolve. Consider utilizing cloud computing platforms or containerization technologies to dynamically scale computing resources based on workload demands and budget constraints.

Environmental Considerations

Ensure adequate cooling and ventilation systems to maintain optimal operating temperatures for hardware components, particularly in data centers or server rooms. Monitor power consumption and energy efficiency of hardware infrastructure to minimize operational costs and environmental impact.

3.2.2 SOFTWARE REQUIREMENTS

Python

A versatile programming language commonly used for data analysis and machine learning tasks.

Integrated Development Environment (IDE)

Jupyter Notebook: Interactive environment for data exploration, analysis, and prototyping machine learning models. PyCharm, Spyder, or Visual Studio Code: Feature-rich IDEs with debugging capabilities and support for Python development.

Python Libraries:

NumPy: Fundamental package for numerical computing.

Pandas: Data manipulation and analysis library.

scikit-learn: Machine learning library containing various algorithms for regression and model evaluation.

TensorFlow or Pytorch: Deep learning frameworks for building and training neural networks.

XGBoost or LightGBM: Gradient boosting libraries for ensemble learning.

Matplotlib and Seaborn: Visualization libraries for creating plots and charts.

Data Processing Tools:

Pandas: Preprocessing tasks such as cleaning, filtering, and transforming datasets. Feature engineering libraries for creating new features or transforming existing ones based on domain knowledge.

Model Evaluation and Validation: Cross-validation techniques for assessing model performance and generalization. Evaluation metrics such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared for regression tasks.

Deployment and Integration:

Flask or Django: Web frameworks for building and deploying RESTful APIs to serve machine learning models.

Docker: Containerization tool for packaging applications and dependencies for deployment. Cloud platforms like AWS, Google Cloud Platform, or Microsoft Azure for hosting and scaling applications.

Version Control:

Git: Version control system for managing codebase, tracking changes, and collaborating with team members.

GitHub, GitLab, or Bitbucket: Platforms for hosting Git repositories and facilitating collaboration.

Documentation and Reporting

Markdown or restructured text for documenting project details, code documentation, and model descriptions. Jupyter notebooks or Sphinx for generating documentation and reports from code and Markdown files.

Communication and Collaboration

Communication tools like Slack, Microsoft Teams, or Zoom for coordinating with team members and stakeholders. Project management platforms such as Trello, Asana, or Jira for task tracking and collaboration.

Feature Engineering

Develop algorithms or scripts for feature engineering, which involves selecting, transforming, and creating features from raw data that are relevant for predicting match scores. Feature engineering plays a crucial role in the performance of machine learning models.

User Interface (UI)

Design and develop a user-friendly interface for users to interact with the prediction system. This could be a web-based dashboard, a mobile app, or a desktop application, depending on your target audience and platform preference.

Monitoring and Maintenance

Set up monitoring tools and processes to track the performance and health of the prediction system in production. Regularly update the machine learning models with new data and retrain them as necessary to maintain prediction accuracy over time.

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

Predicting the first innings score in IPL matches can be approached using various methodologies, including statistical analysis, machine learning algorithms, and domain-specific insights. Here's a proposed methodology that combines these elements:

Data Collection:

Gather historical IPL match data including features like venue, team composition, batting order, pitch conditions, weather conditions, and match outcome. Collect individual player statistics such as batting average, strike rate, and recent form. Utilize reliable data sources like ESPN Cricinfo, Kaggle datasets, or official IPL records.

Feature Selection:

Identify relevant features that could impact first innings score, such as:

Team performance metrics (e.g., average first innings score, win/loss record)

Venue characteristics (e.g., pitch type, dimensions, altitude)

Weather conditions (e.g., temperature, humidity)

Team composition (e.g., batting lineup, all-rounders)

Recent form of players and teams

Data Preprocessing:

Handle missing data and outliers.

Encode categorical variables.

Normalize or scale numerical features.

Model Selection

Choose appropriate machine learning algorithms for regression tasks. Ensemble methods like Random Forest or Gradient Boosting could be effective. Consider time-series analysis techniques if sequential data plays a significant role. Deep learning models like Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks can also be explored.

Model Training

Split the dataset into training, validation, and testing sets. Train the model using historical data, adjusting hyperparameters through cross-validation. Evaluate different algorithms based on metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE).

Model Evaluation

Validate the model's performance using the testing dataset. Analyze residuals to identify areas of improvement. Compare predictions against actual first innings scores to assess accuracy.

Model Deployment

Deploy the trained model to make real-time predictions. Integrate the model into a user-friendly interface or application for easy access. Monitor and update the model regularly to incorporate new data and improve accuracy.

Feedback Loop

Gather feedback from users and stakeholders to refine the model further. Continuously improve the methodology by incorporating new features or refining existing ones based on insights gained from model performance.

Domain Expertise Integration

Collaborate with cricket experts to incorporate domain-specific insights that might not be captured by data alone. Consider factors like player injuries, team strategies, and match dynamics that could impact first innings scores.

4.1.1 Ideation Map/System Architecture

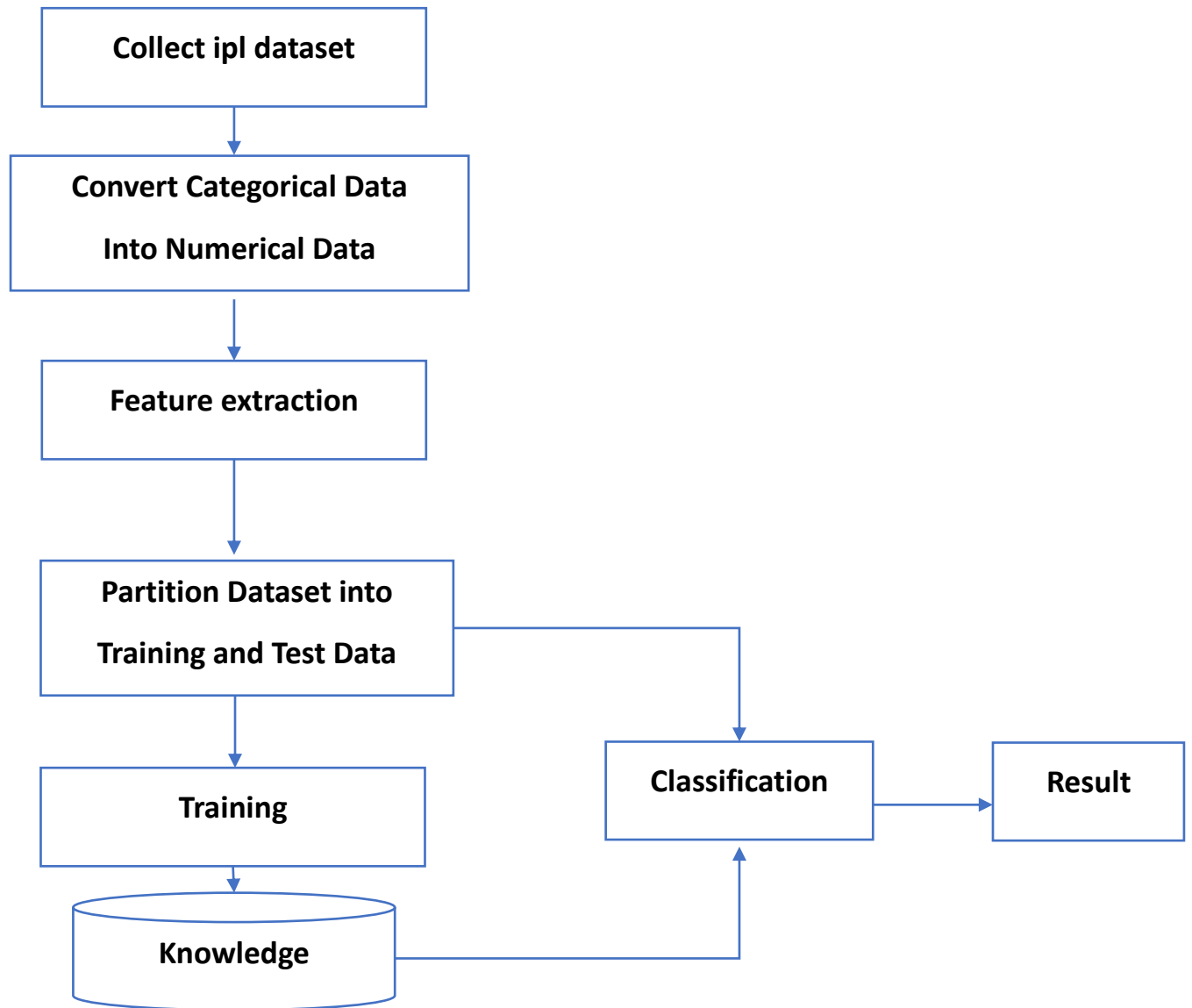


Fig-1

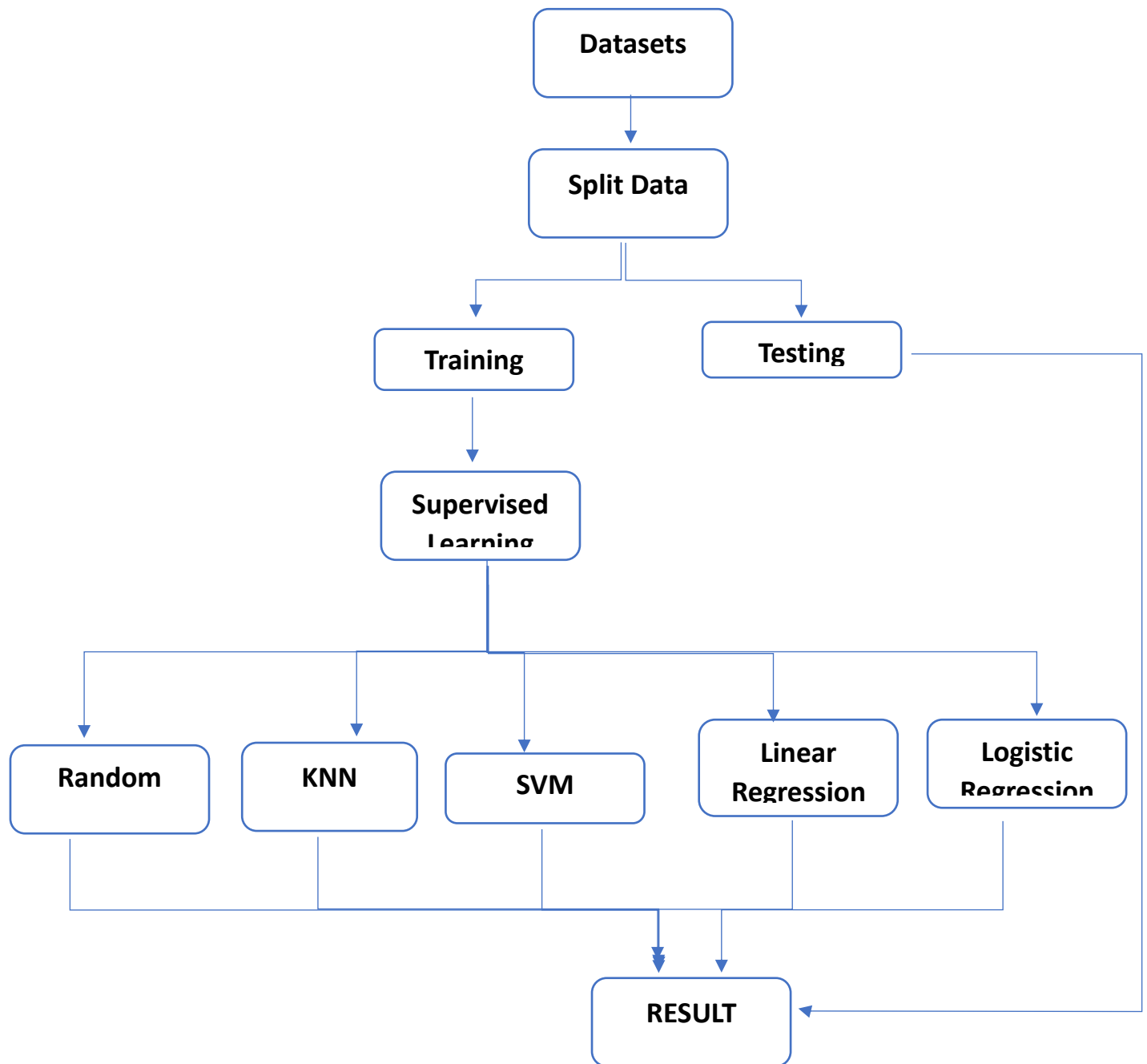


Fig-2

Dataset

The first step in the Architecture of model is to collect datasets from various sources. The data which is fed into the model decides how the model acts and reacts. If the data is accurate and up-to-date, then we will have accurate outcomes or predictions

Team wise home and away dataset

The Teamwise Home and Away dataset contains 6 columns for the datasets which are as follows: home-wins, away-wins, home-matches, away-matches, home-win-percentage and away-win-percentage.

Coloumn name	Coloumn description
--------------	---------------------

Home-wins	Tells the number of matches won by a team in there home ground
Away-wins	Tells the number of match won by a team other than their own ground
Home-matches	Tells the number of matches played by the team in their home ground
Away-match	Tells the matches played by the team other than their home groung
Home-win-percentage	Percentage of matches won when played in home ground
Away-win-percentage	Percentage of matches won when played in ground other than their home ground

Fig-3

4.1.2 Various Stages

Data Collection

Gather historical IPL match data including factors like venue, team composition, batting order, weather conditions, etc.

Data Preprocessing

Clean and preprocess the collected data by handling missing values, encoding categorical variables, and scaling numerical features.

Feature Selection/Engineering

Select relevant features or engineer new features that might influence the first innings score prediction, such as team form, player statistics, pitch conditions, etc.

Model Selection

Choose an appropriate machine learning model for prediction, such as linear regression, decision trees, random forests, or gradient boosting algorithms.

Training the Model

Split the data into training and testing sets, then train the selected model on the training data.

Model Evaluation

Evaluate the trained model's performance using appropriate metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) on the test data.

Hyperparameter Tuning

Fine-tune the model's hyperparameters to optimize its performance.

Cross-validation

Validate the model's performance using techniques like k-fold cross-validation to ensure its robustness.

Prediction

Once the model is trained and validated, use it to predict the first innings score for upcoming IPL matches.

Monitoring and Updating

Continuously monitor the model's performance and update it as needed with new data to improve its accuracy and relevance over time.

4.1.3 Internal or Component design structure

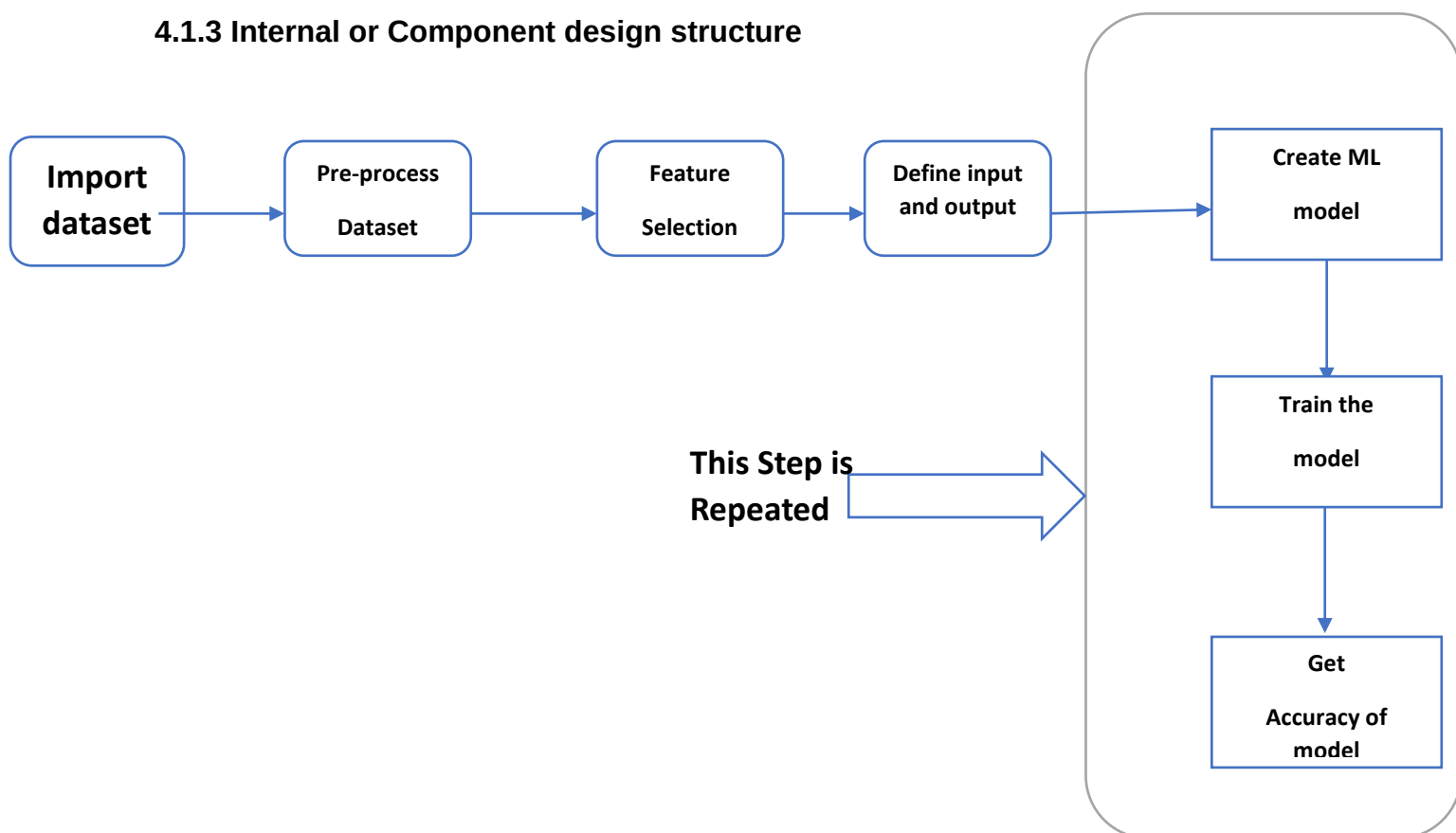


Fig - 4

Regression

Regression reasoning uses miscellaneous invention for the computation and established that it thinks the constant advantage. There are certain set of variables are second hand for the recommendation and the constant range advantage is the target changeable. Based on the request various reversion algorithms are second hand.

Linear Regression

To predict the continuous values, Linear regression is used. Certain known parameters are given to the machine learning algorithms; it predicts the continuous values as output.

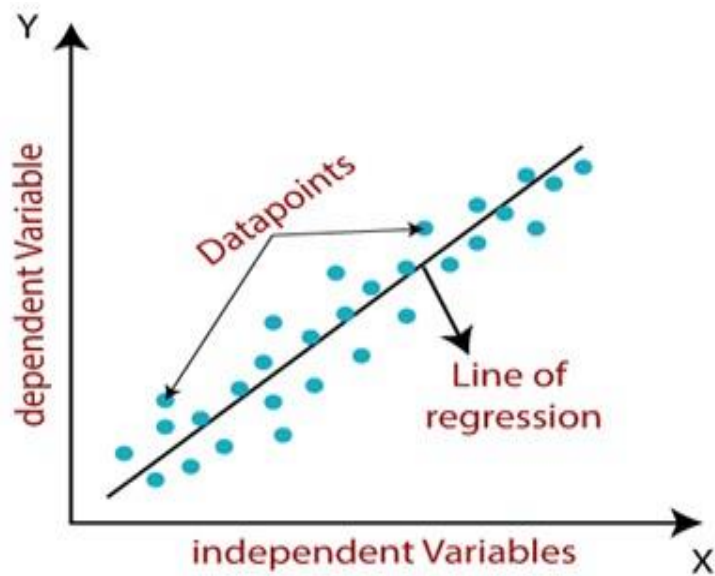


Fig-5

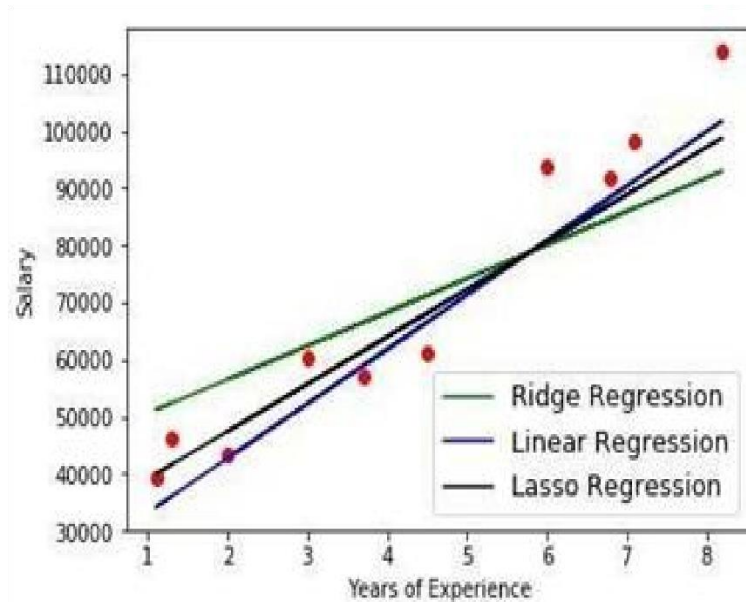


Fig-6

4.1.4 working principles

Data Collection and Preprocessing

Gather comprehensive datasets containing historical IPL match data, including match conditions, player statistics, and match outcomes. Preprocess the data by cleaning, normalizing, and encoding features to ensure compatibility with machine learning algorithms.

Feature Engineering

Identify relevant features that influence cricket match outcomes, such as team performance, player statistics, pitch conditions, weather, and venue characteristics. Engineer new features or transform existing ones to capture complex relationships and patterns within the data.

Model Training

Select appropriate machine learning algorithms, such as regression models, decision trees, or neural networks, based on the nature of the prediction task and the characteristics of the data. Train the chosen models using historical IPL match data, optimizing model parameters to minimize prediction errors and maximize accuracy.

Prediction and Evaluation

Deploy trained models to predict the scores of upcoming IPL matches based on relevant input features, such as team compositions and match conditions. Evaluate the performance of the prediction models using metrics like mean absolute error (MAE) or root mean square error (RMSE) to assess their accuracy and reliability

4.1.6 Sample output

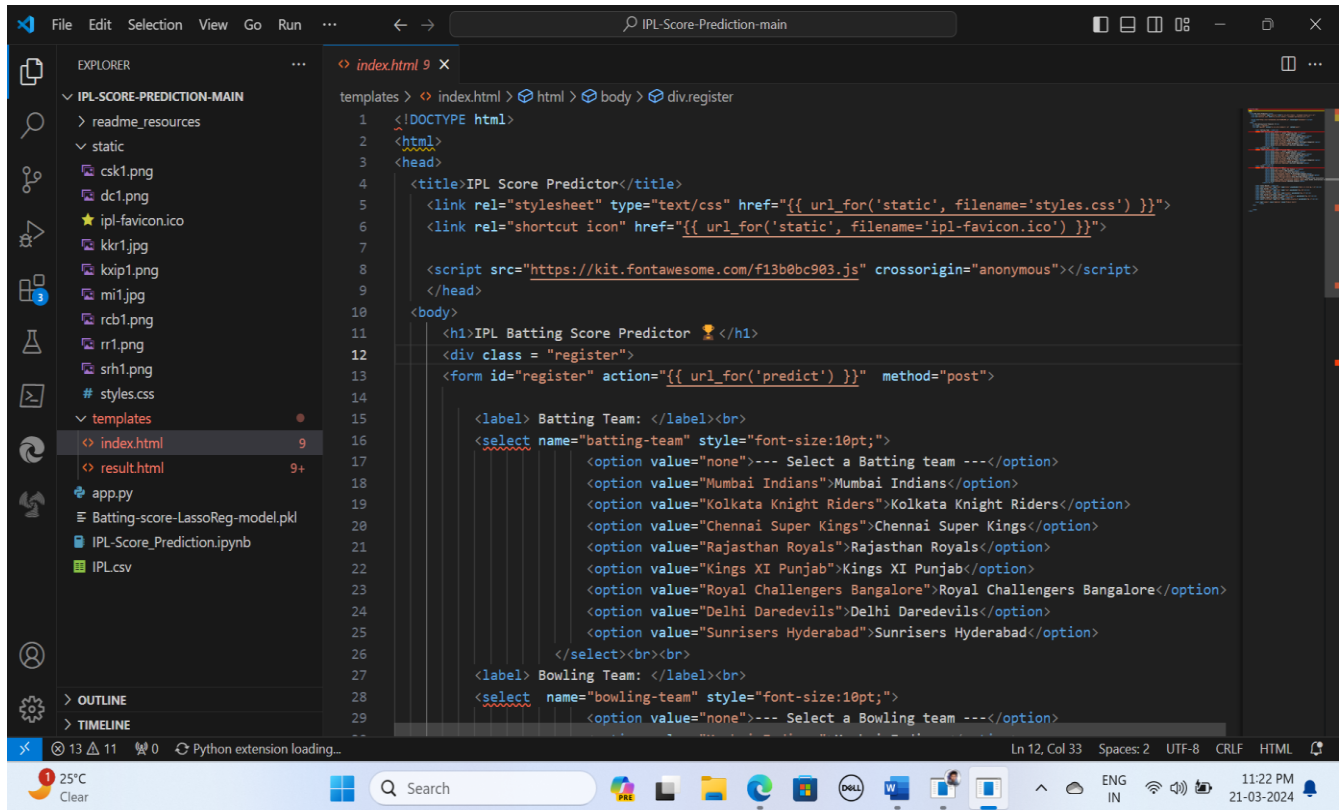


Fig-7

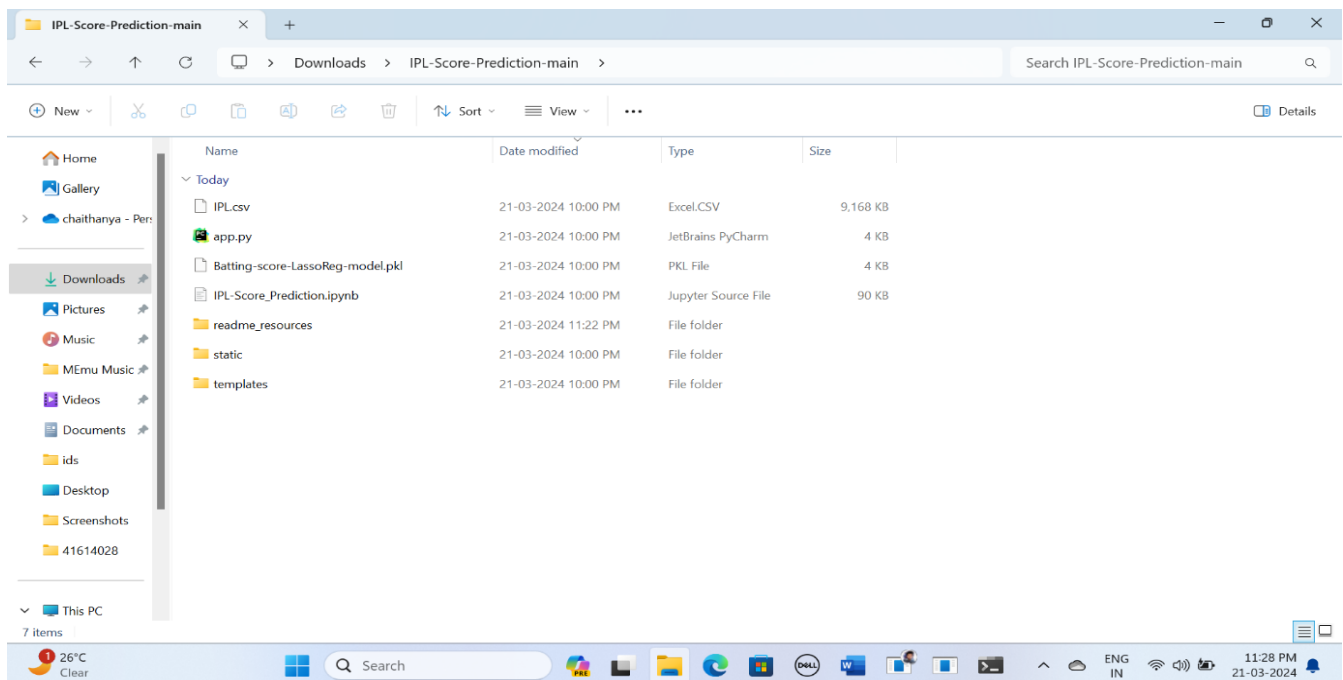


Fig-8

```
Anaconda Prompt

(base) C:\Users\chait>cd C:\Users\chait\Downloads\IPL-Score-Prediction-main

(base) C:\Users\chait\Downloads\IPL-Score-Prediction-main>
```

Fig-9

```
Anaconda Prompt - python a

(base) C:\Users\chait>cd C:\Users\chait\Downloads\IPL-Score-Prediction-main

(base) C:\Users\chait\Downloads\IPL-Score-Prediction-main>python app.py
C:\Users\chait\anaconda3\Lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator Lasso from version 0.23.2 w
hen using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chait\anaconda3\Lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator GridSearchCV from version 0
.23.2 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
C:\Users\chait\anaconda3\Lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator Lasso from version 0.23.2 w
hen using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\chait\anaconda3\Lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator GridSearchCV from version 0
.23.2 when using version 1.2.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer
to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Debugger is active!
* Debugger PIN: 110-213-575
```

Fig -10

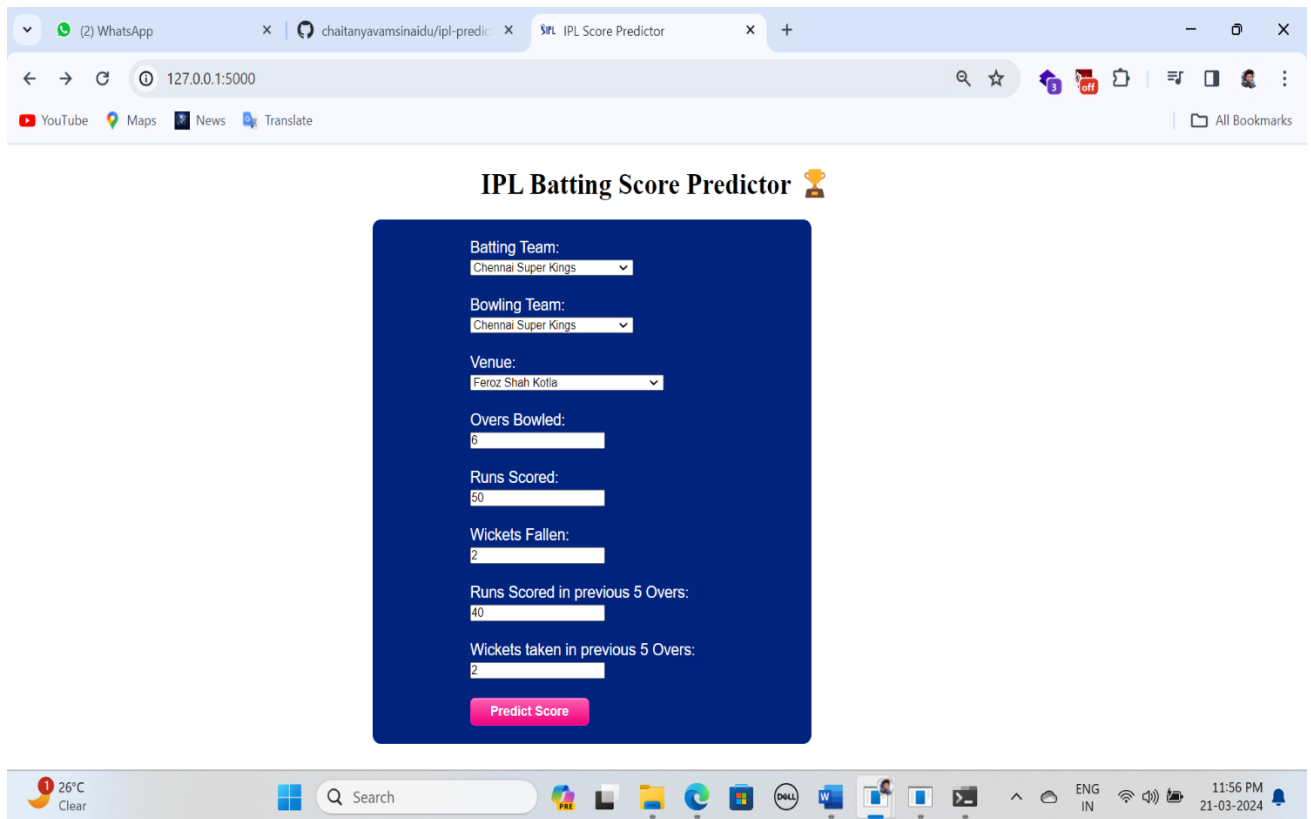


Fig -11

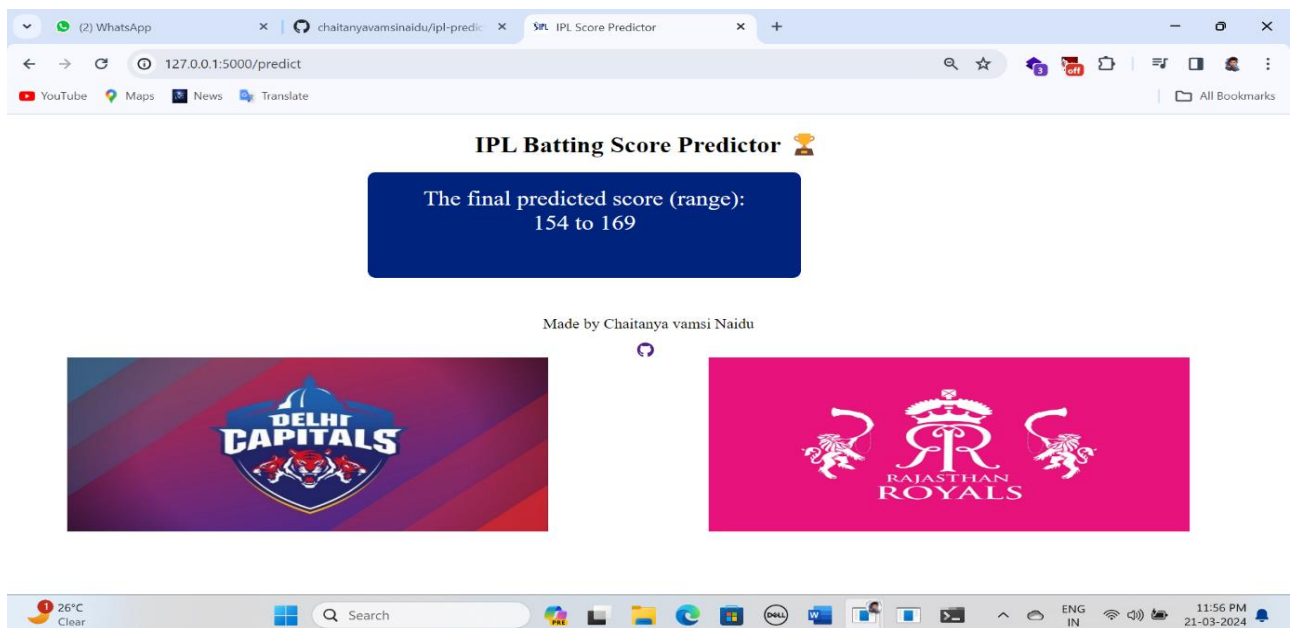


Fig-12

4.2 FEATURES

- Data collection
- Data Preprocessing
- Model Selection and Training
- Evaluation and Performance Metrics
- Hyperparameter Tuning
- Real-time Prediction
- Visualization and Interpretability
- User Interface
- Monitoring and Maintenance
- Documentation and Reporting
- Model Training
- Prediction and Evaluation
- Python Libraries

4.2.1 Novelty of the Project

The novelty of predicting IPL first innings scores using machine learning lies in several aspects:

Complexity of Cricket

Cricket is a complex sport with multiple variables influencing the outcome of a match, including player performance, pitch conditions, weather, and team strategies. Using machine learning to predict first innings scores requires analyzing and incorporating these diverse factors into a predictive model.

Large-Scale Data Analysis

Predicting IPL first innings scores involves analyzing a vast amount of historical match data, including player statistics, team performance, and match conditions. Machine learning enables the extraction of meaningful patterns and insights from this data to make accurate predictions.

Dynamic Nature of IPL

The IPL is known for its dynamic and unpredictable nature, with teams constantly adapting their strategies and tactics. Machine learning models for predicting first innings scores must be able to adapt to these changes and incorporate the latest information to make reliable predictions.

Real-Time Prediction

Machine learning models can be deployed to provide real-time predictions of first innings scores during IPL matches, allowing fans, coaches, and analysts to gain insights and make informed decisions as the match progresses.

Integration of Diverse Data Sources

In addition to traditional cricket statistics and match data, our system leverages a wide range of diverse data sources to enhance prediction accuracy. This includes real-time sentiment analysis of social media chatter, player sentiment scores derived from interviews and press conferences, and contextual information such as venue history and pitch conditions. By integrating these diverse data streams into our ensemble learning framework, we enrich the predictive power of our models and uncover hidden insights that traditional approaches may overlook.

CHAPTER 5

CONCLUSION

In conclusion, predicting IPL first innings scores using machine learning is a challenging yet rewarding endeavor that offers valuable insights into the dynamics of cricket matches. By leveraging historical match data, sophisticated algorithms, and advanced techniques in data preprocessing and model training, it's possible to build accurate prediction models. Such models provide not only real-time predictions during matches but also valuable insights for teams, coaches, analysts, and fans. The ability to anticipate first innings scores enhances decision-making processes, improves strategic planning, and enriches the overall viewing experience of IPL matches. Moreover, the iterative nature of machine learning allows for continuous refinement and improvement of prediction models over time, ensuring their relevance and effectiveness in the ever-evolving landscape of cricket.

REFERENCES

1. **"Predicting outcomes of one day international cricket matches using machine learning techniques"** Authors: S. J. Ramsurn, S. R. Maistry, and M. S. Combrink.
2. **"A machine learning approach to cricket outcome prediction"** Authors: J. Xu, S. Ray, and H. S. Dhillon.
3. **"Machine Learning Based Prediction of T20 Cricket Match Outcomes"** Authors: A. Chowdhury, M. Faisal, and M. S. Islam.
4. **"Predicting Cricket Match Outcome: A Machine Learning Approach"** Authors: S. Pattnaik, S. Sarangi, and S. Pattnaik.
5. **"Predicting the outcomes of Twenty20 cricket matches using machine learning techniques"** Authors: S. K. Sharma and V. P. Singh.
6. **"Analysis and Prediction for the Indian Premier league"** Authors: H. Barot, A. Kothari, P. Bide, B. Ahir and R. Kankaria.
7. **"ICC T20 Cricket World Cup 2020 Winner Prediction Using Machine Learning Techniques"** Authors: A. Basit, M. B. Alvi, F. H. Jaskani, M. Alvi, K. H. Memon and R. A. Shah
8. **"Predicting Results of Indian Premier League T-20 Matches using Machine Learning,"** Authors: S. Agrawal, S. P. Singh and J. K. Sharma.
9. **"Prediction on Ipl Data Using Machine Learning Techniques in R Package"** Authors: G. Sudhamathy and G. Raja Meenakshi
10. **"Ipl Cricket Score and Winning Prediction Using Machine Learning Techniques"** Authors: Nikhil Dhonge Shraddha Dhole, Nikita Wavre, Mandar Pardakhe

APPENDIX

B.Source code

Flask code (to create or run the our main file)

```
# Importing essential libraries

from flask import Flask, render_template, request

import pickle

import numpy as np

# Load the Linear Regression model

filename = 'Batting-score-LassoReg-model.pkl'

regressor = pickle.load(open(filename, 'rb'))

app = Flask(__name__)


@app.route('/')

def home():

    return render_template('index.html')


@app.route('/predict', methods=['POST'])

def predict():

    temp_array = list()

    if request.method == 'POST':

        batting_team = request.form['batting-team']

        if batting_team == 'Chennai Super Kings':

            temp_array = temp_array + [1,0,0,0,0,0,0,0]

            elif batting_team == 'Delhi Daredevils':

                temp_array = temp_array + [0,1,0,0,0,0,0,0]
```

```

elif batting_team == 'Kings XI Punjab':
    temp_array = temp_array + [0,0,1,0,0,0,0,0]
elif batting_team == 'Kolkata Knight Riders':
    temp_array = temp_array + [0,0,0,1,0,0,0,0]
elif batting_team == 'Mumbai Indians':
    temp_array = temp_array + [0,0,0,0,1,0,0,0]
elif batting_team == 'Rajasthan Royals':
    temp_array = temp_array + [0,0,0,0,0,1,0,0]
elif batting_team == 'Royal Challengers Bangalore':
    temp_array = temp_array + [0,0,0,0,0,0,1,0]
elif batting_team == 'Sunrisers Hyderabad':
    temp_array = temp_array + [0,0,0,0,0,0,0,1]

```

```

bowling_team = request.form['bowling-team']
if bowling_team == 'Chennai Super Kings':
    temp_array = temp_array + [1,0,0,0,0,0,0,0]
elif bowling_team == 'Delhi Daredevils':
    temp_array = temp_array + [0,1,0,0,0,0,0,0]
elif bowling_team == 'Kings XI Punjab':
    temp_array = temp_array + [0,0,1,0,0,0,0,0]
elif bowling_team == 'Kolkata Knight Riders':
    temp_array = temp_array + [0,0,0,1,0,0,0,0]
elif bowling_team == 'Mumbai Indians':
    temp_array = temp_array + [0,0,0,0,1,0,0,0]
elif bowling_team == 'Rajasthan Royals':

```

```

overs = float(request.form['overs'])

runs = int(request.form['runs'])

wickets = int(request.form['wickets'])

runs_in_prev_5 = int(request.form['runs_in_prev_5'])

wickets_in_prev_5 = int(request.form['wickets_in_prev_5'])


temp_array = temp_array + [overs, runs, wickets, runs_in_prev_5,
wickets_in_prev_5]


data = np.array([temp_array])

my_prediction = int(regressor.predict(data)[0])


return render_template('result.html', lower_limit = my_prediction-10, upper_limit =
my_prediction+5)

if __name__ == '__main__':

app.run(debug=True)

```

#index.html

```

<!DOCTYPE html>

<html>

<head>

<title>IPL Score Predictor</title>

<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css')
}}">

<link rel="shortcut icon" href="{{ url_for('static', filename='ipl-favicon.ico') }}">

```

```
<script src="https://kit.fontawesome.com/f13b0bc903.js"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body>
```

```
<h1>IPL Batting Score Predictor 🏆 </h1>
```

```
<div class = "register">
```

```
<form id="register" action="{{ url_for('predict') }}" method="post">
```

```
<label> Batting Team: </label><br>
```

```
<select name="batting-team" style="font-size:10pt;">
```

```
<option value="none">--- Select a Batting team ---</option>
```

```
<option value="Mumbai Indians">Mumbai Indians</option>
```

```
<option value="Kolkata Knight Riders">Kolkata Knight
Riders</option>
```

```
<option value="Chennai Super Kings">Chennai Super
Kings</option>
```

```
<option value="Rajasthan Royals">Rajasthan Royals</option>
```

```
<option value="Kings XI Punjab">Kings XI Punjab</option>
```

```
<option value="Royal Challengers Bangalore">Royal Challengers
Bangalore</option>
```

```
<option value="Delhi Daredevils">Delhi Daredevils</option>
```

```
<option value="Sunrisers Hyderabad">Sunrisers
Hyderabad</option>
```

```
</select><br><br>
```

```
<label> Bowling Team: </label><br>
```

```
<select name="bowling-team" style="font-size:10pt;">
```

```
<option value="none">--- Select a Bowling team ---</option>
```

```

        <option value="Wankhede Stadium">Wankhede Stadium</option>
    </select><br><br>

    <label> Overs Bowled: </label><br>

    <input class="form-input" type="text" name="overs" placeholder="Overs (>=
5.0) Eg. 7.2"><br><br>

    <label> Runs Scored: </label><br>

    <input class="form-input" type="text" name="runs" placeholder="Eg.
64"><br><br>

    <label> Wickets Fallen: </label><br>

    <input class="form-input" type="text" name="wickets" placeholder="Eg.
4"><br><br>

    <label> Runs Scored in previous 5 Overs: </label><br>

    <input class="form-input" type="text" name="runs_in_prev_5"
placeholder="Eg. 42"><br><br>

    <label> Wickets taken in previous 5 Overs: </label><br>

    <input class="form-input" type="text" name="wickets_in_prev_5"
placeholder="Eg. 3"><br><br>

    <input type="submit" class="myButton" value="Predict Score">

    </form>

</div>

</body>

</html>

```