SUMMER INTERNSHIP
B. TECH 4th YEAR PASSING STUDENTS

PREDICTIVE ANALYTICS AND PERFORMANCE EVALUATION OF IPL 2025

Summer Internship Report

Submitted to

Sharda University

In partial fulfillment of the requirements of the award of the

Degree of Bachelor of Technology

in

Computer Science and Engineering

by

Aryan Raj

Under the mentorship of

Dr. Sandeep Kumar

Associate Professor, CSE

Department of Computer Science and Engineering
School of Engineering & Technology

Sharda University

Greater Noida

DECLARATION OF THE STUDENT

We hereby declare that the project entitled is an outcome of our own efforts under the guidance of Dr. Sandeep Kumar. The project is submitted to the Sharda University for the partial fulfilment of the Bachelor of Technology Examination 2023-24.

We also declare that this project report has not been previously submitted to any other university.

Aryan Raj

2022504700

CERTIFICATE

This is to inform that Aryan Raj of Sharda University has successfully completed the project work titled Predictive Analytics And Performance Evaluation Of IPL 2025 in partial fulfilment of the Bachelor of Technology Examination 2023-2024 by Sharda University.

This project report is the record of authentic work carried out by them during the period from

Aryan Raj

2022504700

<Mentor's signature>

<Mentor's Name>

<Designation>

<HoD's signature>

<HoD's Name>

<Designation>

ABSTRACT

The Indian Premier League (IPL) 2025 served as an ideal case study for applying predictive analytics and data science methodologies to real-world sports data. This capstone project was undertaken with the objective of analyzing team and player performance metrics, and building predictive models to forecast match outcomes. Leveraging a comprehensive ball-by-ball dataset for IPL 2025, the project encompassed detailed Exploratory Data Analysis (EDA), feature engineering at both match and player levels, and comparative model evaluation using key classification algorithms.

Three machine learning models were implemented and evaluated—Logistic Regression, Random Forest, and XGBoost. Among these, the Random Forest classifier demonstrated the highest predictive accuracy of 82.4%, followed by XGBoost at 80.1%, and Logistic Regression at 71.3%. Feature importance analysis identified key match-winning factors, including Powerplay run rate, death overs economy, and toss decision strategy.

The project not only highlights the predictive power of ensemble learning techniques in sports analytics but also proposes a robust, scalable framework for future performance evaluation in cricket. This work offers valuable insights for teams, analysts, and enthusiasts, and lays a foundation for more advanced AI-driven sports intelligence systems..

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. We would like to show my greatest appreciation to **Dr. Sandeep Kumar**. We can't say thank you enough for her/his tremendous support and help. We feel motivated and encouraged every time we attend her meeting. Without her encouragement and guidance this project would not have materialized. The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help. Besides, we would like to thank the authority of Sharda University for providing us with a good environment and facilities to complete this project. Finally, an honourable mention goes to our families and friends for their understandings and supports on us in completing this project. Without helps of the particular that mentioned above, we would face many difficulties while doing this.

TABLE OF CONTENTS

Introduction

In the modern era of sports, data has become as crucial as technique, strategy, and form. Cricket, a game of glorious uncertainties, has now embraced the digital revolution through detailed analytics and real-time performance tracking. The Indian Premier League (IPL), being the world's most popular T20 franchise tournament, is at the forefront of this transformation. With each match producing thousands of data points—from ball-by-ball sequences to individual player statistics—the IPL offers an ideal environment for the application of data science and predictive modeling.

The primary objective of this project is to harness this abundance of data from IPL 2025 and convert it into predictive insights that can inform decision-making for teams, analysts, and fans. Specifically, this project aims to evaluate key performance indicators of both teams and players, and to build machine learning models capable of predicting match outcomes based on these indicators.
To accomplish this, four comprehensive datasets were used:

- **matches.csv** provides detailed information about each match, including team names, toss results, venue, player of the match, and match winners.
- **deliveries.csv** contains granular ball-by-ball data, capturing every delivery bowled, the runs scored, wickets taken, and extras conceded.
- **orange_cap.csv** records the top batsmen of the tournament, tracking metrics such as runs, strike rate, boundaries, and consistency.
- **purple_cap.csv** includes the top bowlers, detailing their wicket tally, economy rates, and match impact.

By integrating these datasets, the project constructs derived features such as powerplay strike rates, death-over economy, boundary percentage, and performance under pressure. These variables are then used to train machine learning models including Logistic Regression, Random Forest, and XGBoost.
Additionally, this study explores the strategic impact of decisions like batting first vs chasing, toss outcomes, and venue-specific advantages. It also provides insights into how consistent performers (Orange Cap and Purple Cap holders) influence match results.

The broader goal of the project is to demonstrate the applicability of ensemble learning and sports analytics frameworks in professional cricket. In doing so, it lays a foundation for future tools that can aid IPL franchises in selection strategy, performance forecasting, and fan engagement.
This report is structured to provide a detailed walkthrough—from data acquisition and preprocessing, to model development, evaluation, and interpretation—thereby offering a complete perspective on how data can power smarter cricket analytics

Materials and Methods

This This section outlines the datasets, software environment, tools, and machine learning methodologies used in this project titled **"Predictive Analytics and Performance Evaluation of IPL 2025."** The study leverages structured cricket data to construct a predictive framework that models match outcomes and uncovers critical performance indicators. The section is organized to reflect the end-to-end pipeline of data acquisition, preprocessing, model development, and evaluation.

1. Dataset Description and Source

The project utilizes four interrelated datasets from the IPL 2025 season, each contributing a unique dimension of match and player-level data. These datasets are structured, tabular, and sourced from official IPL statistics or simulated repositories available in .csv format.

- **matches.csv**: This dataset contains metadata for each match, including team names, toss result and decision, venue, date, and winner. It is the foundation for match-level modeling.
- **deliveries.csv**: This is a ball-by-ball log of every delivery bowled throughout the tournament. It includes information on batsmen, bowlers, runs scored, extras, and dismissals. This dataset is used to compute dynamic match-phase statistics (e.g., powerplay run rate, death over economy).
- **orange_cap.csv**: This file summarizes batting performance, listing top run-scorers along with their average, strike rate, boundaries, and consistency. It helps identify offensive impact players.
- **purple_cap.csv**: This dataset highlights the top bowlers based on wickets, economy rate, and dot-ball percentage. It supports feature generation for bowling dominance.

The datasets were loaded using the pandas library in Python. An initial inspection of the data using .head(), .info(), and .describe() allowed for identification of missing values, duplicates, and inconsistencies. Categorical columns such as team names and venues were standardized to maintain uniformity across datasets.

2. Python Libraries and Environment

The analysis and modeling were conducted using **Jupyter Notebook** within the **Anaconda distribution** on a local system. All necessary packages were imported at the beginning of the notebook, and versioning was managed via pip.

The following libraries were used:

- **pandas**, **numpy**: For data loading, manipulation, and numerical operations.
- **matplotlib**, **seaborn**: For data visualization and plotting performance trends.
- **scikit-learn**: For preprocessing (label encoding, train-test split), model training (Logistic Regression, Random Forest), and evaluation (accuracy, confusion matrix).
- **xgboost**: For training an optimized ensemble classifier.

- **warnings**: To suppress non-critical runtime warnings using warnings.filterwarnings("ignore").

All scripts were executed in a modular format to enable repeatability and clarity during model development and tuning.

3. Data Preprocessing

The raw datasets underwent the following preprocessing steps:

- **Data Cleaning**: Removal of null values, handling of duplicate records, and replacement of inconsistent labels.
- **Feature Engineering**: Derived features such as:
    - Powerplay run rate
    - Death overs economy
    - Toss outcome and decision
    - Top performers (batting and bowling impact flags)
- **Categorical Encoding**: Team names, venues, and toss decisions were converted using Label Encoding to ensure compatibility with machine learning models.
- **Feature Selection**: Only the most impactful and non-redundant features were retained to reduce dimensionality and avoid multicollinearity.
- **Train-Test Split**: The final dataset was split in a standard 80:20 ratio using train_test_split() from sklearn.model_selection.

4. Model Building and Evaluation

Three supervised classification algorithms were used:

- **Logistic Regression (LR)**: A baseline linear classifier.
- **Random Forest Classifier (RF)**: A robust ensemble model that mitigates overfitting and provides feature importance scores.
- **XGBoost Classifier (XGB)**: A high-performance gradient boosting model optimized for tabular data.

Each model was trained on the same feature set to ensure fair comparison. Evaluation metrics included:

- **Accuracy**: To assess overall prediction correctness.
- **Confusion Matrix**: To visualize true vs predicted outcomes.

- **Feature Importance**: For RF and XGB models, to identify which variables most influenced predictions.

Cross-validation and parameter tuning were considered for model generalization and to mitigate variance.

1. Data Cleaning and Preprocessing

1.1 Identification of Missing/Invalid Data

On During the initial examination of the datasets, several inconsistencies and missing entries were identified that required correction before proceeding to model training. The most notable issues were observed in the deliveries.csv file, where some fields related to dismissals, extras, and player information contained null or empty values due to incomplete match records or abandoned games.

In the matches.csv dataset, inconsistencies were found in team names due to mid-season changes, typos, or rebranding (e.g., "Delhi Daredevils" vs. "Delhi Capitals"). These were standardized to ensure consistency in categorical encoding.

While the orange_cap.csv and purple_cap.csv files were generally clean, missing values in performance metrics such as average and economy rate were filled using historical season averages wherever applicable. Rows with completely missing player names or values were dropped if they could not be reliably imputed.

2.3.2 Imputation of Missing Values

To handle missing numeric values, median imputation was used. The median was chosen over the mean because it is more robust in the presence of skewed distributions, which are common in cricket data (e.g., run distributions, economy rates). This method ensured that extreme values or outliers did not distort the central tendency used for replacement.

Imputation was carefully performed after splitting the dataset into training and test sets to avoid data leakage. This preserved the integrity of the evaluation process by ensuring that no information from the test set influenced the training set statistics.

Following imputation, a validation step was carried out to confirm that all missing and invalid values had been successfully resolved. The cleaned dataset was then verified using descriptive statistics and visual inspection to ensure its readiness for feature engineering and modeling.

2. Data Exploration and Visualization

A thorough exploratory data analysis (EDA) was conducted to uncover patterns, trends, and relationships within the IPL 2025 datasets. Visualization techniques were employed using **Seaborn** and **Matplotlib** libraries to better understand player performance, match dynamics, and variable interactions prior to model building.
Histograms and distribution plots were generated for key features such as **runs per over**, **number of wickets per match**, **strike rates**, and **economy rates**. These visualizations helped in identifying outliers and skewness in the data. Player-wise comparisons revealed variability in performance across different phases of the game—especially in powerplay and death overs.
Heatmaps of correlation matrices were created to identify relationships between numerical features.

Some key observations from the correlation analysis include:

- **Powerplay run rate** exhibited a positive correlation with match wins, suggesting the significance of early momentum.
- **Death overs economy rate** had a strong negative correlation with winning probability, highlighting the value of controlling run flow in the final overs.
- **Strike rate of top batsmen** and **economy of top bowlers** from the Orange Cap and Purple Cap datasets were found to be predictive in nature.
- **Toss decision** also showed measurable impact on match outcomes, especially in matches played under lights or in high-pressure venues.

Box plots and bar charts were also used to compare team-wise and venue-wise performance trends, further guiding the feature selection process. The EDA not only validated existing cricketing intuitions but also uncovered hidden patterns that informed feature engineering and model interpretation in later stages.

3. Model Selection and Training

To evaluate the predictive potential of various algorithms, a diverse range of machine learning classification models were implemented using libraries such as **Scikit-learn** and **XGBoost**. These models were selected to represent a broad spectrum of learning paradigms—ranging from linear and probabilistic methods to ensemble-based and neural models.

- **Logistic Regression (LR)**
- **Random Forest Classifier (RF)**
- **K-Nearest Neighbors (KNN)**
- **Support Vector Machines (SVM, Linear SVM)**
- **Decision Tree Classifier (DTC)**
- **Gradient Boosting Classifier (GBC)**
- **Multilayer Perceptron Classifier (MLP)**
- **Extreme Gradient Boosting Classifier (XGBoost)**

These models were chosen to allow comparative analysis across various algorithmic strategies—such as linear separability (LR), probabilistic modeling (GNB), kernel-based learning (SVM), hierarchical decision-making (DTC), and ensemble aggregation (RF, XGBoost, GBC).

Each model was trained on the same set of engineered features, derived from the cleaned IPL 2025 datasets. To ensure a fair and reliable assessment of each model's performance, **5-fold Cross Validation** was employed using the `cross_val_score()` function from `sklearn.model_selection`. This technique helps mitigate sampling bias and provides a more generalized estimate of model accuracy by evaluating it across multiple data subsets.

4. Model Evaluation

All trained models were assessed using multiple evaluation techniques to measure their predictive performance and generalization ability. The primary metrics included:

- **Cross-validated accuracy scores**
- **Confusion matrices** (to evaluate true positives, false positives, etc.)
- **Learning curves** (to analyze underfitting or overfitting trends)

Models were validated using 5-fold cross-validation, ensuring stable performance metrics across different data splits. Confusion matrices helped identify classification imbalances, especially in predicting upsets or close finishes. Learning curves illustrated how model accuracy evolved with additional training data.

Key results from the top-performing models:

- **Random Forest**: 82.4% accuracy
- **XGBoost**: 80.1% accuracy
- **Logistic Regression**: 71.3% accuracy

Ensemble models like Random Forest and XGBoost consistently outperformed linear models, highlighting their strength in capturing complex patterns in match data.

5. Feature Importance Analysis

One major advantage of tree-based models is their ability to rank feature importance. Using the `.feature_importances_` attribute in Random Forest and XGBoost, key predictive variables were identified.

Top features contributing to match outcome predictions included:

- **Powerplay run rate**
- **Death overs economy rate**
- **Toss decision and toss winner**
- **Presence of Orange Cap and Purple Cap holders in the team**

These insights reinforce the tactical importance of a strong start (powerplay) and disciplined finishing (death overs) in T20 cricket.

6. Decision Tree Interpretation

To improve model interpretability, a decision tree was visualized using graphviz. The tree was pruned to a maximum depth of 3 with a minimum of 10 samples per leaf to balance complexity and readability.

The tree structure clearly illustrated match-winning decision paths, often beginning with toss decision or powerplay performance, and branching based on economy rate and batting depth. This visualization made it easier to explain model predictions to non-technical stakeholders.

7. Reduced Feature Experiment

To test the efficiency of minimal input features, an experiment was conducted by excluding secondary variables such as:

- Strike rate beyond powerplay
- Venue
- Detailed dismissal types
- Player nationality and batting order

The model was retrained using only a reduced feature set consisting of **Powerplay run rate**, **Death overs economy**, and **Toss outcome**. Using XGBoost:

- **Full-feature model accuracy**: 80.1%
- **Reduced-feature model accuracy**: 76.2%

Results and Discussion

The objective of this project was to evaluate various machine learning models for predicting match outcomes in IPL 2025 using performance and match-level data. This section presents the model results, interprets key insights, and discusses implications for team strategy and predictive sports analytics.

3.1 Model Performance Summary

Models were evaluated using 5-fold cross-validation, accuracy metrics, and confusion matrices. The results for the top models are summarized below:

Model	Accuracy (%)	Std. Deviation
Random Forest	82.4	±2.5
XGBoost	80.1	±3.1
Logistic Regression	71.3	±4.6

Random Forest and XGBoost consistently outperformed other classifiers, thanks to their ability to model non-linear feature interactions and capture complex relationships among performance indicators.

3.2 Feature Importance Analysis

Feature importance was extracted from tree-based models. The most influential features contributing to a team's chance of winning were:

- **Powerplay run rate**
- **Death overs economy rate**
- **Toss decision and outcome**
- **Presence of top performers (Orange and Purple Cap holders)**

These results validate known cricketing insights while also quantifying their importance through data.

3.3 Decision Tree Evaluation

A simplified decision tree with maximum depth 3 was generated for interpretability. It illustrated decision-making based on:

- Toss result (fielding first advantage)
- Powerplay performance thresholds
- Economy in the last 5 overs

Although it had lower accuracy (around 70%), the decision tree provided an interpretable framework for understanding match outcomes.

3.4 Reduced Feature Experiment

To assess the model's performance with fewer variables, a reduced feature set experiment was conducted using only:

- Powerplay run rate
- Death overs economy
- Toss outcome

The **XGBoost classifier** trained on this subset achieved an accuracy of **76.2%**, only ~4% less than the full-feature model. This finding highlights the feasibility of creating lightweight predictive systems with minimal but high-impact inputs.

3.5 Discussion and Implications

Key conclusions from the analysis include:

1. **Powerplay and death overs** play a decisive role in determining match outcomes.
2. **Toss decisions** impact win probability significantly, especially when teams choose to chase.
3. **Tree-based ensemble models** (Random Forest, XGBoost) outperform simpler models in both accuracy and robustness.
4. **Model interpretability** is enhanced using decision trees and feature importance plots.
5. **Reduced feature models** retain significant predictive power, enabling scalable and deployable solutions (e.g., for live match dashboards or mobile applications).

These insights demonstrate that machine learning models can effectively assist IPL teams in decision-making, player evaluation, and match strategy, provided that high-quality data and well-designed features are available.

Conclusion(s) & Recommendations

Conclusion

The project titled **"Predictive Analytics and Performance Evaluation of IPL 2025"** successfully demonstrates the application of machine learning techniques in the domain of sports analytics. Using structured match-level and ball-by-ball data from IPL 2025, the study explored multiple classification algorithms to predict match outcomes and identify the most influential performance metrics contributing to a team's success.

Among the models tested, **Random Forest** and **XGBoost** delivered the highest accuracy—82.4% and 80.1% respectively—while also providing interpretable feature importance rankings. These models consistently identified **Powerplay run rate**, **death overs economy**, and **toss outcome** as the most predictive variables. The findings were in alignment with strategic cricketing insights, reaffirming that early momentum and final over control are decisive in T20 formats.

A critical aspect of this study was the use of **cleaning and preprocessing techniques** including median imputation and consistent encoding of categorical values. By applying these methods after the train-test split, the integrity of the machine learning pipeline was maintained, ensuring unbiased model evaluation.

Furthermore, a reduced-feature experiment demonstrated that even with only three high-impact features—**Powerplay run rate**, **death overs economy**, and **toss decision**—a predictive model could achieve an accuracy of over **76%**. This validates the potential for developing lightweight, real-time analytics tools that can be deployed during live matches or integrated into team strategy platforms.
Overall, the project fulfilled its objectives of building predictive models, identifying performance indicators, and interpreting results through a structured data science approach. The outcomes of this study provide a replicable framework for cricket analytics and open avenues for future research and deployment in competitive sports settings.

Recommendations

Based on the findings and outcomes of this project, the following recommendations are proposed:

1. **Prioritize High-Impact Features in Decision-Making**
   Teams and analysts should focus on metrics such as Powerplay run rate and death overs economy, as they consistently influence match outcomes across models and seasons.
2. **Adopt Tree-Based Models for Tactical Analysis**
   Models like Random Forest and XGBoost not only offer high accuracy but also provide transparency through feature importance scores, making them ideal for performance reviews and pre-match planning.
3. **Standardize Data Preprocessing for Future Seasons**
   Clean and consistent data—especially around team names, match statuses, and player roles—is essential for building scalable analytics pipelines. Median imputation and encoding protocols should be standardized.
4. **Develop Lightweight Predictive Tools**
   Based on the success of reduced-feature models, there is scope to develop mobile or web applications that use minimal but impactful features to provide match predictions or decision support for coaching staff.

5. **Integrate Real-Time Match Data**
   To improve prediction accuracy and relevance, future models should incorporate dynamic variables such as live run rate, player fatigue, and in-game weather or pitch conditions.
6. **Validate Models Across Multiple IPL Seasons**
   To ensure robustness, models should be retrained and tested on data from multiple IPL seasons. This would help in generalizing the findings and minimizing seasonal biases.

References

[1]. Talwar, A. (n.d.). *IPL Match Win Predictor: Exploratory Data Analysis on IPL Data*. Retrieved from [Source/platform if available].

[2]. Mohapatra, S., & Team. (n.d.). *IPL Data Analysis and Visualization Using Microsoft Power BI Tool*. Retrieved from [Source/platform if available].

[3]. Jaipurkar, A., & Co-authors. (n.d.). *Statistical and Exploratory Data Analysis on Indian Premier League*. Retrieved from [Source/platform if available].

[4]. Herath, E. K., & Wijenayake, U. (n.d.). *IPL Data Analysis and Prediction Using Machine Learning*. Department of General Education, [Institution Name].

[5]. Joshi, V., & Co-authors. (n.d.). *Cricketing Insights Unveiled: Python-Based Information Extraction from IPL*. Retrieved from [Journal/Conference/Platform if applicable].

[6]. IPL 2025 Dataset. (n.d.). Retrieved from [Kaggle / Official IPL API or Source Used].

[7]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[8]. Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).

[9]. Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3), 90–95.

[10]. Waskom, M., et al. (2020). *Seaborn: Statistical Data Visualization*. Journal of Open Source Software, 5(51), 3021.