Predictive Modelling on IPL Cricket Dataset

Manthan Nager, Akshat Darji, Maulik Parmar, Shelvi Kamani, Astha Patel
Department of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmadabad, Gujarat-382481, India
23MCD011@nirmauni.ac.in,
23MCD001@nirmauni.ac.in,
23MCD012@nirmauni.ac.in,
23MCD006@nirmauni.ac.in,
23MCD013@nirmauni.ac.in

***Abstract*—Cricket in India (BCCI) has changed the game's dynamics. In recent years, machine learning (ML) has emerged as a powerful tool in sports analytics. By using Machine Learning and statastical modeling techniques it is able to capture the patterns. In this paper, we present a Random Forest regressor model for predicting the runs per innings in Indian Premiar league(ipl) . The model is based on a Regression that will used to predict the runs bases of current situation of match as per available data of match.**

***Index Terms*—Predictive Analysis, IPL, Decision tree, classification, regression, supervised leanrning**

I. Introduction

The Indian Premier League (IPL) is a major example of how entertainment and sports may coexist in the context of contemporary cricket. The Indian Premier League (IPL), which was founded in 2008 by the Board of Control for Cricket in India (BCCI), has changed the game's dynamics and grown to be a major international athletic event. The use of predictive analysis in cricket has become a fascinating way to understand the nuances of the game as the tournament's intense contests continue to exciting spectators.

T20 cricket is exciting because it is fast-paced and unpredictable, which presents obstacles for both teams and spectators. Predicting player performances and match outcomes by predictive analysis is becoming more and more popular as a way to stay competitive. The vast IPL dataset is the focus of this term paper's use of predictive analytics. The objective is to identify trends and patterns in the data that will aid in the understanding of the variables influencing match results.

The use of machine learning methods, such as Decision Trees, Naive Bayes, and K-Nearest Neighbour's, is fundamental to this investigation. These algorithms are used to examine past data that includes team dynamics, player statistics, and match circumstances. The main objective is to create models that can forecast match outcomes while also provide insightful analysis of the elements that influence an IPL team's success or failure.

In light of the IPL, this term paper attempts to explore the unexplored field of predictive analytics. The goal is to add to the growing field of sports analytics by using machine learning algorithms to provide a greater knowledge of the variables that influence the results of one of the most renowned and watched cricket competitions worldwide.

II. Machine Learning

A. *Fundamentals of Machine Learning:*

Definition:

The goal of machine learning, a branch of artificial intelligence, is to create statistical models and algorithms that let computers learn from data without explicit programming. The basic concept is to let machines to become more proficient at a certain activity over time as they are exposed to more data.

Key Concept:

1) Training Data:: Without being explicitly programmed, machine learning models derive their predictions and conclusions from past data, or "training data."

2) Features Engineering and Labels:: In supervised learning, the data is typically divided into features (input variables) and labels (output variables), with the model learning the mapping from features to labels.

3) Model Training:: During training, the model adjusts its parameters to minimize the difference between its predictions and the actual labels in the training data.

4) Prediction and Generalization:: Once trained, the model can make predictions on new, unseen data. The goal is for the model to generalize well, providing accurate predictions on data it hasn't encountered before.

B. *Supervised Learning:*

Definition:

An algorithm is trained on a labeled dataset in supervised learning, a kind of machine learning. In order to forecast new, unknown data, the algorithm must first learn the mapping between input attributes and matching output labels.

Key Concept:

1) Labeled Data:: The training dataset consists of examples where the input features are paired with the correct output labels.

2) Training Process:: During training, the algorithm adjusts its parameters to minimize the difference between its predictions and the actual labels in the training data.

Types of Supervised Learning:

3) Classification:: Predicting a categorical label (e.g., Whether the team will win the match or not).

4) Regression:: Predicting a continuous numerical value (e.g., predicting total runs per innings in cricket).

C. *Regression:*

Definition:

Predicting continuous numerical values is the main goal of regression, a kind of supervised learning. Regression may be used in your term paper to forecast the final number of runs at the conclusion of an IPL innings.

Key Concept:

1) Dependent and Independent Variables:: Dependent Variable (Y): The variable we want to predict (e.g., total runs). Independent Variables (X): The features used for prediction (e.g., batsman_run, extras_run).

2) Objective:: The objective in regression is to find the relationship between the independent variables and the dependent variable, allowing for accurate predictions on new data.

D. *Different Regression Algorithms:*

Overview:

Depending on the situation at hand and the type of data, a number of regression techniques can be utilized. Here are a few typical examples:

1) Linear Regression:: Assumes a linear relationship between the independent and dependent variables.

2) Decision Tree Regression:: Assumes a linear relationship between the independent and dependent variables.

3) Random Forest Regression:: Ensemble method using multiple decision trees for improved accuracy.

4) Gradient Boosting Regression:: Builds a series of weak learners (usually decision trees) to create a strong predictive model.

5) Support Vector Regression (SVR):: An extension of support vector machines for regression tasks.

An extension of support vector machines for regression tasks.

III. Literature Review

The Work done in the field of predictive analysis on the Datasets like IPL or cricket various algorithms such as Decision tree, KNN, Naive bayes, Regression algorithms etc, are applied on the IPL dataset for the analysis in this field. Here are some Literature that we review and used in this model for the analysis.

In [2] authors C. Srikantaiah, Aryan Khetan, Baibhav Kumar, Divy Tolani, and Harshal Patel, aims to comprehend the intricacies of the IPL dataset spanning the past 10 years. The focus lies on unraveling the operational principles of four distinct machine learning algorithms implemented in R. Through the establishment of a model and training dataset, the study facilitates predictions using the generated model. Classification of the data ensues, and a comparative analysis, incorporating accuracy, error rate, precision, recall, sensitivity, and specificity measures, guides the selection of the most effective algorithm—Random Forest. Emphasizing graphical representation and comparative analysis, this work not only explores IPL data insights but also provides a valuable decision-making tool for stakeholders, including the Indian Premier League and its fan base, enabling informed predictions and strategic choices for future success."

In [1] authors performed a thorough analysis of previous studies on the subject of employing machine learning algorithms to forecast the score of the first inning in Indian Premier League games. We used Ridge Regression and Linear Regression, compared the results, and determined which method performed best by using metrics like RMSE, MSE, and MAE to measure accuracy. Our model used Jupyter Notebook and Python to assess important variables including runs, overs, wicket fall, and batting and bowling teams. Our goal was to provide a predictive algorithm that could accurately forecast first-inning scores, based on subtle insights from the IPL dataset. We demonstrated the correlations between these aspects through graphical representations, giving cricket teams an invaluable tool to plan strategically based on well-informed projections.

In [3] The study suggested a multivariate regression model based on team performance indicators and found important elements impacting Indian Premier League (IPL) match results. For the 2018 IPL season, machine learning classifiers—the Multilayer Perceptron (MLP) in particular—showed excellent predicting accuracy. While acknowledging the inherent unpredictability of Twenty20 cricket, the study's encouraging findings demonstrate the potential of machine learning algorithms to forecast results in this dynamic and ever-evolving sport.

IV. Methodology

The suggested approach is divided into five smaller modules, loading the dataset, pre-processing, feature selection and construction , classification using different algorithms, and best model selection .

A. *Data-set Information*

Here we use two data-set CSV files named `'IPL_Ball_by_Ball_2008_2022.csv'` and `'IPL_Matches_2008_2022.csv'`. Size of datasets are 19,483 KB and 461 KB respectively. In `IPL_Ball_by_Ball_2008_2022.csv`, the total number of attributes is 17, and the number of records is 225,955. The features' names are 'ID', 'innings', 'overs', 'ballnumber', 'batter', 'bowler', 'non-striker', 'extra_type', 'batsman_run', 'extras_run', 'total_run', 'non_boundary', 'isWicketDelivery', 'player_out', 'kind', 'fielders_involved', 'BattingTeam'. In `IPL_Matches_2008_2022.csv`, the total number of attributes is 20, and the number of records is 951. These datasets are taken from the Kaggel repository. These Dataset Plays a crutial role in our analysis, helping us address specific questions related to predicting the total runs at the end of

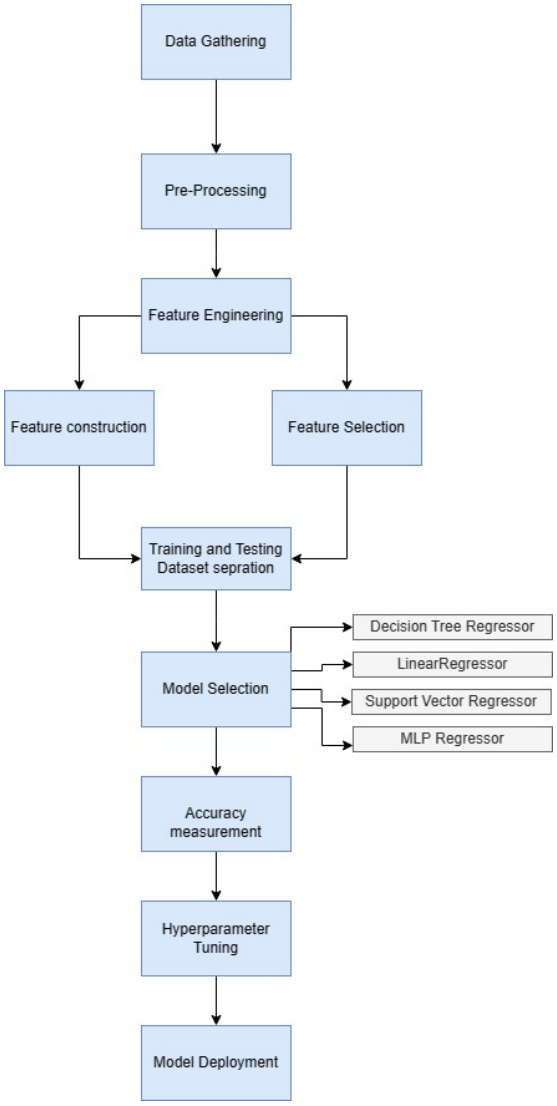

Fig. 1: ML lifecycle of the project

innings using machine learning algorithms. The features from `IPL_Ball_by_Ball_2008_2022.csv` such as 'batsman_run', 'extras_run', and 'total_run' are particularly important for building and training our predictive models.

1) ***Data Gathering****:* In this project, the data gathering process played a important role in gaining the foundation for predicting the runs by these IPL dataset at the end of innings in the context of IPL cricket matches. Two key datasets, namely `IPL_Ball_by_Ball_2008_2022.csv` and `'IPL_Matches_2008_2022.csv'` were employed to capture the essential details of ball-by-ball interactions and overall match information spanning from the year 2008 to 2022. The primary dataset, df, was loaded using the Pandas library with the command df = pd.read_csv(`IPL_Ball_by_Ball_2008_2022.csv`), while the secondary dataset, df1, was similarly imported with df1 = pd.read_csv(`'IPL_Matches_2008_2022.csv'`). These datasets encompass a wealth of information, ranging from player statistics to match outcomes, providing a comprehensive perspective on the IPL cricketing landscape.

The foundation for training and assessing predictive models, such as Decision Tree Regression, Linear Regression, Random Forest Regressor, Support Vector Machine (SVM), and Multilayer Perceptron (MLP), was created by integrating these datasets. The study's goal was to use these datasets to identify trends and connections in the data that would improve the precision of runs predicted at the end of innings.

B. Exploratory Data analysis

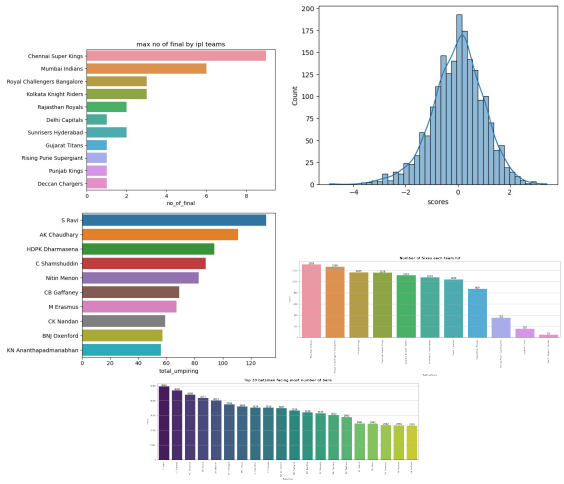

Fig. 2: EDA

C. Data Prepossessing and Feature Engineering

The suggested approach is divided into five smaller modules, loading the data-set(Data Gathering), feature Construction.

1) ***Feature Selection****:* To improve the models' capacity for prediction, much care was applied throughout the feature development procedure. An essential component of this stage was the DataFrame `df`, a representative subset of the "IPL_Ball_by_Ball_2008_2022.csv" dataset. The following crucial columns were carefully chosen: `'ID,' 'innings,' 'overs,' 'batter,' 'bowler,' 'extra_type,' 'total_run,' 'non_boundary,' 'isWicketDelivery,' 'player_out,'` and `'BattingTeam.'` These columns were picked to highlight important facets of the ball-by-ball action in Indian Premier League games. Nominal variables were encoded to guarantee compatibility with the techniques used, which facilitated model training. These encoded columns were used to generate the selected characteristics, which attempted to capture subtleties like player identities, over-specific information, additional run kinds, wicket occurrences, and team dynamics. This methodical approach to feature building

created the groundwork for a strong and thorough.

*2) **Feature Construction**:* In our Predictive model, we focused on making our dataset easy for the model so that the model can be trained easily using that dataset. We did this by turning certain columns like 'batter,' 'bowler,' 'extra_type,' 'player_out,' and 'batting_team.' into numerical form through a method called **One Hot Encoding**. This process is essential for creating new columns for each unique category in these features, changing our data into a format that the models can work with more effectively. Also constructed the label for this problem 'final_run,' 'curr_run,' 'curr_wk,' which are created by applying mathametical formulations.

A careful examination of ball-by-ball dynamics in each innings enhanced the feature creation process. A consecutive count of legitimate deliveries, called "ball_number," was calculated for each match in the dataset. This count included the removal of deliveries that were flagged as "Wides" or "noballs," guaranteeing precise information on the inning's development. The iterative calculation, which was performed for innings 0 and 1, resulted in an exhaustive list that contained the ball numbers that were successive during each game. Through the integration of this dynamic feature, the model was able to acquire a more sophisticated comprehension of changing the game condition, which improved its ability to forecast runs at the end of IPL innings.

*3) **Data Cleaning**:* First of all we have checked the null values and impute them in the data with appropriate imputer and Certain changes were performed to the encoded features in order to clean our dataset for the best possible model performance. Notabely in order to improve the redundancy and simplify the data, binary columns that represented specific category were eliminated. After one hot encoading the 'innings feature, the columns 'innings__2,' 'innings__3,', 'innings__4,' 'innings__5,' and 'innings__6' are removed since they no longer required in our process. By removing unnecessary information, the dataset became more accurate and effective, which help our model to concentrate on the key elements of the data that are necessary for accurate prediction.

One-hot encoding approaches were also used for categorical data such as 'extra_type,' 'player_out,' 'batter,' 'bowler,' and 'BattingTeam,' which improved the model's comprehension and allowed it to draw conclusions that were meaningful. The dataset was further refined by removing unnecessary one-hot encoded columns, making it more suitable for analysis later on. By ensuring that the dataset was well-prepared and devoid of irregularities, this meticulous data cleaning procedure laid the groundwork for strong and trustworthy predictive modeling.

*4) **Split The Data into Training and Testing**:* To enable model assessment and validation, the dataset is partitioned into training and testing sets throughout the crucial stage of model construction. The characteristics and labels were divided into several entities. Using the scikit-learn library's 'train_test_split' function, 80% of the data were assigned to the training set and the remaining 20% to the testing set. To prevent biases, the 'shuffle' option was set to 'True,' ensuring a representative and random distribution of cases between the two sets. The machine learning models were trained on one portion of the dataset, and their predictive power was evaluated on a separate subset. This methodical partitioning of the dataset allowed for the evaluation of the models' real-world performance and generalization skills.

V. REGRESSION ANALYSIS

Regression analysis in the context of run prediction in sports, particularly in cricket or baseball, involves using statistical methods to model the relationship between various factors (such as batting order, player statistics, pitch conditions, etc.) and the number of runs scored in an innings.

A. *Decision Tree Regressor*

A flexible machine learning approach for classification and regression applications is the decision tree. It makes judgments at each node as it recursively divides the dataset according to important attributes. Until certain requirements are satisfied, the process is repeated. Decision trees can overfit big datasets, yet they are easily understood and straightforward. To improve their performance, strategies like pruning and ensemble approaches like Random Forests are applied. Their ability to effectively capture relationships within data makes them useful in a variety of sectors and offers a straightforward visual picture of decision-making. This model will work fine in our data set because the data set has to much categorical features.

.

B. *Linear Regression*

A straightforward machine learning approach for forecasting continuous outcomes is called linear regression. In order to reduce prediction errors, it uses a linear equation to represent the connection between the variables and finds the best-fit line. Because of its simplicity, it is frequently used for tasks like forecasting housing values.

C. *Random Forest Regressor*

A popular ensemble learning technique for both classification and regression applications is Random Forest. During training, it builds a large number of decision trees, from which it produces the mode for classification tasks or the average prediction for regression tasks.

D. *Support Vector Regressor*

The Support Vector Regressor (SVR) is utilized for run prediction in sports datasets like cricket or baseball due to its ability to handle complex relationships between various input features and the continuous target variable of runs scored. SVR is particularly adept at capturing nonlinear relationships by transforming data into a higher-dimensional space and finding an optimal hyperplane that best fits the data points while maximizing the margin of error tolerance. However, in certain cases, SVR might not yield accurate predictions in run prediction datasets due to several reasons. One primary factor could be the dataset's characteristics, such as insufficient or noisy data, which might make it challenging for SVR to generalize and find a suitable hyperplane for accurate predictions. Additionally, if the features contributing to run prediction lack a clear linear or nonlinear relationship with the target variable, SVR's performance might be limited, resulting in less accurate predictions. In such scenarios, feature engineering, data preprocessing, or considering alternative regression techniques tailored to the dataset's characteristics might be necessary to improve prediction accuracy.

E. *Multi Layer Perceptron regressor*

An artificial neural network that is utilized for regression problems is called a Multi-Layer Perceptron (MLP) regressor. MLP regressors, which are composed of many layers of nodes (neurons) with non-linear activation functions, are capable of modeling intricate correlations between continuous output variables and input data. The network modifies weights during training in order to reduce the discrepancy between expected and actual results.

VI. Results and Outcomes

To get the best outcome, the optimal classification algorithm for a specific dataset must be chosen. It is difficult since it necessitates making a number of crucial methodological decisions. The metrics used to evaluate the algorithms' classification performance and rank are the main focus of this study. Here are the most often used measurements, along with a discussion of their characteristics. Over the years, many different policies have been put forward.

Model	MSE	MAE	RMSE
Decision Tree	89.33	2.76	9.45
Linear Regression	481.14	16.35	21.93
Random Forest	39.45	3.63	6.28
SVR	230	12	15.27
MLP Regressor	521.09	17.12	22.82

TABLE I: Comparative Analysis of Models

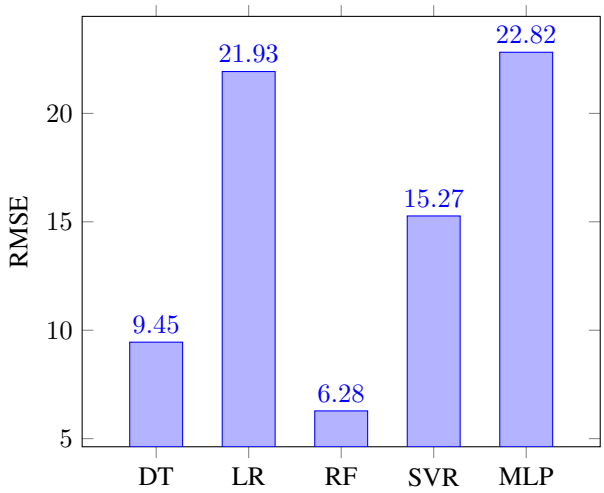

Fig. 3: Comparison of RMSE values for different regression models

Acknowledgment

We express our sincere gratitude to the Kaggle repository *'IPL_DATA_2008_2022' By 'AMOL KHAIR'* for providing the extensive and invaluable datasets—`IPL_Ball_by_Ball_2008_2022.csv` and `IPL_Matches_2008_2022.csv`. This term paper would not have been possible without the wealth of information encapsulated in these datasets, which enabled us to delve into the intricacies of ball-by-ball interactions and overall match dynamics. Our profound appreciation extends to the Kaggle repository for making these datasets accessible. Additionally, we acknowledge the foundational role of machine learning algorithms, including Decision Trees, Linear Regression, Random Forest, Support Vector Machine (SVM), and Multilayer Perceptron (MLP), in unraveling patterns and predicting the outcome of IPL innings. The collaborative efforts of the cricketing community, data providers, and machine learning advancements have significantly contributed to the depth and insights of this predictive modeling endeavor.

VII. Conclusion

In this comprehensive exploration of predictive modeling on IPL cricket datasets, our analysis incorporated diverse machine learning algorithms, including Decision Tree, Linear Regression, Random Forest, Support Vector Machine (SVM), and Multilayer Perceptron (MLP). The objective was to identify the most effective model for forecasting the total runs at the end of IPL innings. Following a rigorous comparative analysis based on Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), the Random Forest Regressor emerged as the standout performer, exhibiting the lowest RMSE value of 6.28.

The Random Forest Regressor's ability to capture the nuanced relationships within the dataset, coupled with its relatively lower prediction error, makes it the optimal

choice for our IPL runs prediction model. This algorithm's interpretability and simplicity further enhance its practicality for cricket analytics.

Therefore, for our subsequent predictions and analyses, we will employ the Random Forest Regressor as the primary model. Its proficiency in handling the dynamics of IPL innings positions us to make accurate and insightful forecasts, contributing to a deeper understanding of the game and aiding strategic decision-making for teams, players, and enthusiasts alike.

REFERENCES

[1] Raja Ahmed, Prince Sareen, Vikram Kumar, Rachna Jain, Preeti Nagrath, Ashish Gupta, and Sunil Kumar Chawla. First inning score prediction of an ipl match using machine learning. In *AIP Conference Proceedings*, volume 2555. AIP Publishing, 2022.

[2] Srikantaiah C, Aryan Khetan, Baibhav Kumar, Divy Tolani, and Harshal Patel. Prediction of ipl match outcome using machine learning techniques, 09 2021.

[3] Rabindra Lamsal and Ayesha Choudhary. Predicting outcome of indian premier league (ipl) matches using machine learning, 09 2018.