Kurdish Studies
September 2024
Volume: 12, No: 5, pp 1242-1250
ISSN: 2051-4883 (Print) | ISSN 2051-4891 (Online)
www.KurdishStudies.net

DOI: 10.53555/ks.v12i5.3457

A Comparative Study Of Supervised Machine Learning Models For Predicting Bowler Performance In T-20I Cricket

Abdurrahman Sabir[1], Qamruz Zaman[1], Muhammad Irfan uddin[2], Syed Habib Shah[2], Neelam[1], Gohar Ayub[3*]

[1]Department of Statistics, University of Peshawar, Pakistan.
[2]Institute of Numerical Sciences Kohat University of Science, Pakistan.
[3]Department of Mathematics and Statistics, University of Swat, Swat, Pakistan
Email: cricsportsresearchgroup@gmail.com , *goharayub@uswat.edu.pk

Abstract
This study investigates the performance of the top 100 T20 International (T20I) bowlers, utilizing data sourced from Cricinfo to analyze key performance metrics that influence bowlers' success in the format. The research employs various classification algorithms, including Decision Trees, Naïve Bayes, Logistic Regression, Support Vector Machines, Extreme Gradient Boosting, and Random Forests, to categorize bowlers based on attributes such as age, bowling style, and playing role. Data was collected through web scraping techniques, focusing on match statistics and performance-specific metrics. Results indicate that the Decision Tree classifier achieved the highest accuracy (85%) in classifying bowlers into spin and fast categories, while Random Forest exhibited lower performance (60%). The study highlights the significance of age, bowling style, and performance metrics in determining bowler classification and effectiveness, emphasizing the need for further optimization and feature engineering in the predictive modeling of bowler performance.

Key Note: Cricket, T-20I, Bowlers, Machine Learning Algorithm, Classification Model

Introduction

Cricket is the most widely viewed and played sport in the South Asian region, surpassing all other sports in terms of popularity and participation [1]. This team sport involves two sides, each comprising eleven players, striving for victory through a balanced mix of batsmen, bowlers, and all-rounders. Batsmen aim to score runs, while bowlers focus on taking wickets and restricting the opposition's score. All-rounders contribute in both areas various factors, including form, the opposing team, and the venue influence a player's performance. In team management, coaches and captains meticulously analyze player abilities and past performances to select the optimal lineup for each match, aiming to predict and maximize player performance [2].

Cricket is played in three formats: Test cricket, a five-day format; One Day International (ODI), which comprises 50 overs per team; and Twenty20 (T20), where each team plays 20 overs. T20, being the most popular and exciting format, has led countries like India, Pakistan, Bangladesh, and Sri Lanka to organize their own domestic T20 tournaments in recent years [3]. Given the fast-paced nature of T20, understanding bowler performance becomes crucial for teams aiming to succeed in this format.

In this context, this paper aims to predict the performance of bowlers in T-20 International (T-20I) matches by analyzing their characteristics and statistics through supervised machine-learning techniques. Specifically, we seek to predict the number of wickets a bowler be it a spin bowler or a fast bowler will take in a match. By examining performance metrics and historical data, we aim to develop models that account for the unique characteristics and playing styles of both types of bowlers, thereby enabling more accurate predictions of their wicket-taking potential in various match scenarios.

Numerous researchers have introduced machine learning-based models for predicting match outcomes in various sports, showcasing the potential of these methods in performance analysis. For instance, Hucaljuk and Rakipovik [4] applied machine learning techniques to forecast football match outcomes, demonstrating the effectiveness of these methods in sports prediction. Similarly, CupJhanwar and Paudi [5] predicted cricket match outcomes by comparing the strengths of competing teams, and assessing individual performances to develop algorithms for modeling batsmen and bowlers. This body of research highlights the versatility of machine learning approaches in sports analytics, setting the stage for our focus on bowlers in T20 cricket.

Our work represents a novel approach to predicting the number of wickets a bowler will take in a match. While Muthuswamy and Lam [11] conducted a similar study using neural networks for eight Indian bowlers, their model lacks generalizability. In contrast, we utilize various supervised machine-learning algorithms to create prediction models that can be applied to any bowler in a given match. This approach enhances the applicability of our findings across a broader spectrum of players and matches.

Machine learning-based classifiers have been extensively employed to predict outcomes in Indian Premier League (IPL) T20 matches, providing valuable insights into the effectiveness of different algorithms. Researchers like Kapadia et al. [12] analyzed data from 10 IPL seasons (2008-2017) using statistical and probabilistic classification algorithms. Lamsal and Choudhary [13] achieved a notable 71.66% accuracy with a Multilayer Perceptron (MLP) compared to other machine learning algorithms. Meanwhile, Tripathi et al. [14] focused solely on pre-match data, using player career history and team strength, and achieved the highest prediction accuracy of 60.043% with Random Forest. These studies illustrate the competitive landscape of machine learning applications in T20 cricket predictions and emphasize the need for continuous improvement in modeling techniques. Furthermore, various studies have employed distinct methodologies to analyze match outcomes and player performances. Pathak and Wadhwa [15] used only pre-match data, applying Naive Bayes, SVM, and Random Forest to predict ODI outcomes. Similarly, Naik et al. [16] used pre-match features, analyzing a single game based on player performance and order, which may not scale well. Kumar et al. [17] applied a Multilayer Perceptron (MLP) to pre-match data, achieving a 57.4% performance rate. Rahman et al. [18] analyzed matches between Bangladesh and other teams from 2005 to 2017, predicting outcomes before and after the first innings, with an SVM classifier yielding 63.63% accuracy. This range of approaches underlines the diverse landscape of predictive analytics in cricket, paving the way for our research.

In addition to these approaches, Mahmood et al. [19] utilized Pakistan Super League (PSL) data from four seasons (115 entries), which included post-match features such as team batting and bowling performances and match scores. Their proposed model achieved 82.00% classification accuracy, highlighting the impact of incorporating a wide range of data features. Finally, Mahtab et al. [20] used the Bengali ABSA dataset (five attributes, 1601 entries) for sentiment analysis of cricket-related public opinions sourced from social media and news outlets. Using a Support Vector Machine (SVM), they achieved 64.00% classification accuracy, further illustrating the broad applications of machine learning in analyzing cricket and related phenomena.

Research Methodology

Data Collection

Data was collected from Cricinfo to analyze the performance of the top hundred (100) T-20I bowlers, focusing on key performance variables that influence a bowler's success in the format. The variables collected included the bowler's age, bowling style (e.g., fast, spin, left-arm, right-arm), and playing role (e.g., specialist bowler or all-rounder). In addition, match-specific statistics were gathered, including the number of matches played (Mat), innings bowled (Inns), total balls delivered (Balls), runs conceded (Runs), and wickets taken (Wkts). Performance-specific metrics such as best bowling in an innings (BBI), best bowling in a match (BBM), bowling average (Ave), economy rate (Econ), and strike rate (SR) were also recorded. To gauge bowlers' impact during crucial moments, the number of 4-wicket hauls (4w), 5-wicket hauls (5w), and 10-wicket hauls (10w) were included.

The data collection process involved using web scraping techniques with Python libraries such as BeautifulSoup or Selenium to extract data directly from Cricinfo's statistics pages. After scraping, the data was cleaned and organized in a structured format, ready for further analysis and classification model training. The collected variables were critical in understanding and predicting bowlers' performance in T20Is based on their overall contribution to the game.

Classification

Classification is assigning a specific category to a query observation by training on a dataset of known data points [21]. Classification algorithms aim to predict the target class with the highest accuracy. These algorithms identify the relationship between input features and the target variable to construct predictive models.[22]

Decision Tree

A Decision Tree (DT) is a widely used tree-based supervised classification technique. It resembles a flowchart where internal nodes represent attributes, branches represent attribute tests, and leaf nodes denote class labels. The tree's branches split recursively until the final level is reached, with each split representing a test on data attributes. Through this hierarchical process of splitting, classification is achieved [23]

Naïve Bayes

It is a supervised probabilistic classification technique that determines the optimal mapping between a new data point and a set of classifications within a given problem domain. This is achieved by applying mathematical manipulations that transform joint probabilities into the product of prior and conditional probabilities [24].

Logistic Regression

Logistic regression is a commonly employed statistical method for modeling the relationship between a binary dependent variable and one or more independent variables. Although it carries the term "regression," it functions as a classification algorithm rather than a traditional regression model. It calculates the probability that a given input falls into a particular category, usually 0 or 1. Logistic regression is applied when the binary outcome variable indicates two possible outcomes (e.g., success/failure, yes/no, win/lose). The model estimates the probability that a given input belongs to one of these categories [25].

Support Vector Machine

Support Vector Machine (SVM) is a classification technique that utilizes a hyperplane to maximize the margin between two or more classes in the training dataset. The algorithm aims to find a maximum margin hyperplane that separates the positive and negative data points (e.g., Spin or Fast, win or loss). If no such hyperplane perfectly separates the classes, SVM chooses the one that divides the samples as strictly as possible [26].

Extreme Gradient Boosting

it is a powerful machine learning algorithm based on the gradient boosting framework, which builds models sequentially to correct the errors of previous models. It is known for its speed, scalability, and ability to handle large datasets efficiently using optimizations like tree pruning and parallelization. XGBoost also includes regularization techniques (L1 and L2) to prevent overfitting, making it more robust than traditional boosting methods. Additionally, it has built-in mechanisms for handling missing data and sparsity, further increasing its flexibility in real-world applications [27].

Random Forest

Random Forest is an ensemble machine learning algorithm that constructs multiple decision trees and aggregates their outputs to make a final prediction. Developed by Leo Breiman in 2001, it uses bagging (bootstrap aggregating), where each tree is trained on a random subset of data, and random feature selection, where only a random subset of features is considered at each split in the tree. These techniques help reduce overfitting and improve generalization by creating diverse, less correlated trees. By averaging the predictions of multiple trees, Random Forest provides a more stable and accurate prediction than individual decision trees [28].

Gradient Boosting

Gradient Boosting is a machine learning technique used for both classification and regression tasks that builds models sequentially, where each new model corrects the errors of the previous ones. It combines multiple weak learners, typically shallow decision trees, to form a strong predictive model. Unlike Random Forest, which builds trees independently, Gradient Boosting builds them sequentially, with each tree reducing the residual errors of prior models. The technique minimizes a loss function through gradient descent and uses hyperparameters like the learning rate to control how much each tree contributes to the final prediction. Regularization techniques such as shrinkage and subsampling help prevent overfitting. While Gradient Boosting is highly effective and flexible, it can be computationally intensive and requires careful tuning [29].

Light Gradient Boosting

It is a highly efficient gradient-boosting framework developed by Microsoft, optimized for speed and performance, especially on large-scale datasets. Unlike traditional gradient boosting, LightGBM grows trees in a leaf-wise manner, splitting the leaf with the highest loss to reduce error more efficiently. It also introduces techniques like Gradient-Based One-Side Sampling (GOSS), which focuses on informative data points to speed up training, and Exclusive Feature Bundling (EFB), which reduces the number of features by combining mutually exclusive ones. LightGBM supports categorical features directly, improving both accuracy and efficiency without the need for one-hot encoding. These optimizations make LightGBM particularly effective for handling large datasets and high-dimensional data, offering faster training times and better scalability [30, 31].

Results and Discussion

Performance Matrices

We have utilized supervised classification algorithms and computed the Recall, Precision, and F1 scores to evaluate their performance. A comparative analysis of the models was conducted using these metrics. The formulas for the evaluation metrics mentioned above are presented in the equations below.

$$Precision = \frac{True\ Positive(TP)}{True\ Positive\ (TP) + False\ Positivie(FP)} \quad (1)$$

$$Recall = \frac{True\ Positive(TP)}{True\ Positive\ (TP) + False\ Negative(FN)} \quad (2)$$

$$F1 - Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (3)$$

Table No # 1 Description of the Dataset and Frequency Distribution for each Attribute

Attribute	**Description**	**Frequency**
Age	Age of the Bowler	(19 to 25 Year) 21, (26 to 33 Year) 56, (34 to 41 Year) 23
Bowling	Bowling Style	Spin (35), Fast (65)
Playing Role	Playing the role of the Bowler	Bowler (73), Bowling Allrounder (8), Batting Allrounder (18), Middle Order (1)
Mat	Total Matches played by the bowler	(11 to 30) 23, (31 to 50) 23, (51 to 70) 35, (71 to 90) 7, (91 to 110) 7, (111 to 130) 5.
Inns	Best bowling figures in an innings.	(11 to 30) 25, (31 to 50) 23, (51 to 70) 35, (71 to 90) 6, (91 to 110) 7, (111 to 130) 4.
Balls	Total Delivered Balls	(359 to700) 19, (701 to 1042) 18, (1043 to 1384) 24, (2411 to 2753) 2, (1385 to 1726) 14, (1727 to 2068) 4, (2069 to 2410) 7, (2411 to 2753) 2
Runs	Total Runs given by the bowler to the opponent	(209 to 641) 18, (642 to 1074) 22, (1075 to 1507) 19, (1508 to 1940) 25, (1941 to 2373) 5, (2374 to 2806) 6, (2807 to 3229) 4, (3240 to 3672) 1

Wkts	Total Wickets taken by the Bowlers	(13 to 31) 21, (32 to 50) 23, (51 to 69) 23, (70 to 88) 14, (89 to 107) 9, (108 to 126) 5, (127 to 145) 2, (146 to 164) 3
BBI	Best Bowler Inning of the Bowlers	(0.09 to 0.20) 43, (0.21 to 0.30) 21, (0.31 to 0.40) 16, (0.41 to 0.50) 10, (0.51 to 2) 10
BBM	Best Bowler Matches of the Bowlers	(0.09 to 0.20) 43, (0.21 to 0.30) 21, (0.31 to 0.40) 16, (0.41 to 0.50) 10, (0.51 to 2) 10
Ave	The average rate of the Bowler	(9.5 to 14.5) 4, (14.6 to 19.5) 21, (19.6 to 23.5) 34, (23.6 to 28.5) 31, (28.6 to 32.5) 10
Econ	The economy rate of the Bowler	(4.5 to 6.0) 4, (6.1 to 8.0) 69, (8.1 to 10.0) 27
SR	The Strike rate of the bowler	(11 to 16) 28, (17 to 22) 65, (23 to 28) 7
4w	Four wickets were taken by the bowlers.	(0 to 2) 82, (3 to 5) 16, (6 to 8) 2
5w	Five wickets were taken by the bowlers.	(0) 75, (1) 19, (2) 6

Figure No # 1: Age and Playing Role of the Bowler

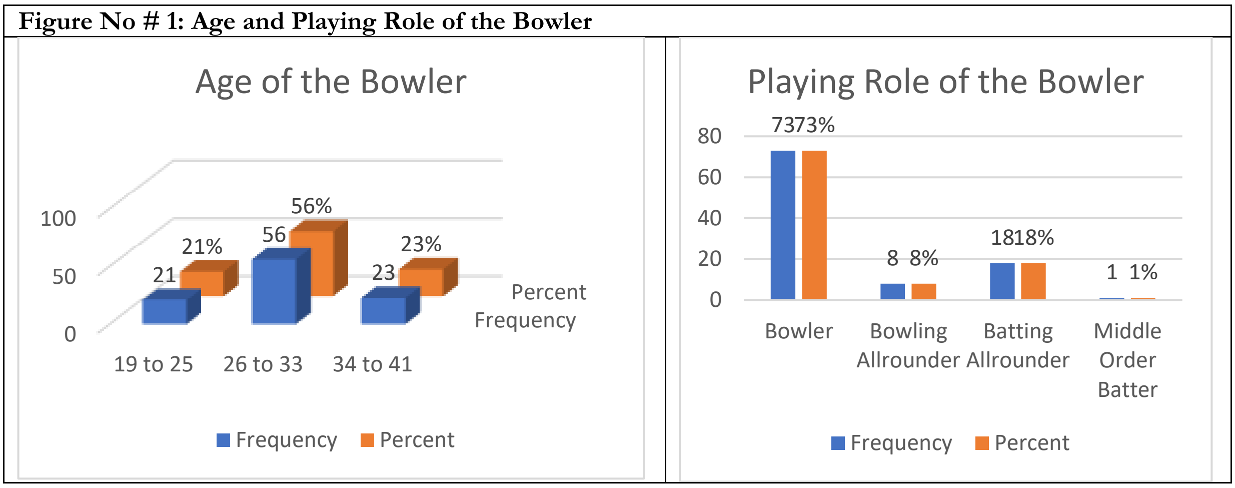


Figure No # 1 shows the age and playing roles among bowlers revealing that the majority of specialized bowlers (54.8%, or 40 out of 73) and bowling allrounders (62.5%, or 5 out of 8) are concentrated in the 26 to 33 age group, indicating that this is the peak performance period for most bowlers. The 19 to 25 age group shows moderate participation, with 20.5% (15 out of 73) of specialized bowlers and 37.5% (3 out of 8) of bowling allrounders, reflecting a phase of development and growth. The 34 to 41 age group contributes significantly to the batting allrounder category (33.3%, or 6 out of 18), but overall participation decreases with age. The minimal representation of middle-order batters (1%, or 1 out of 100) suggests a strong emphasis on bowling-focused roles across all age groups. This distribution highlights the dominance of the 26 to 33 age group, which comprises 56% of all participants, in both specialized and versatile playing roles.

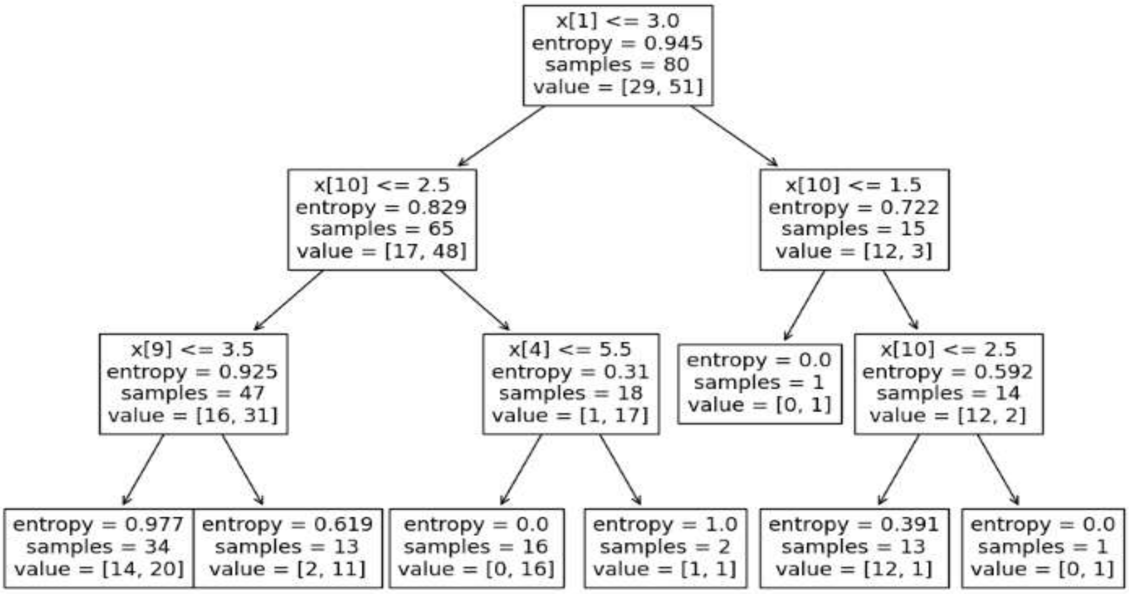


Figure No # 2 shows a decision on the performance of Bowlers

Figure No # 2 shows the decision on the bowlers' performances, dividing them based on Age, Bowling Style (spin vs. fast), and Playing Role. The root node (x[1] <= 3.0) likely represents Age, splitting bowlers into younger (19-25 years) and older (26-33, 34-41 years). From the 80 total samples, the left branch contains 65 bowlers with a lower entropy of 0.829, representing a more homogeneous group, possibly spin bowlers or those with fewer matches and innings. The right branch contains 15 bowlers, likely representing fast bowlers or those with different playing characteristics. This split highlights how factors like age and playing style impact a bowler's overall performance.

Key performance indicators like matches (Mat) played, innings (Inns) bowled, balls delivered, runs conceded, and wickets were taken further to refine the tree. Bowlers are classified into various ranges, with matches played between 11 and 130 and wickets taken between 13 and 164. For example, some bowlers have delivered between 359 to 2753 balls and given away 209 to 3672 runs. The model differentiates high-performing bowlers with strong Best Bowling Figures (BBI and BBM) and consistent averages (ranging from 9.5 to 32.5). Spin bowlers with more consistent Economy Rates (6.1 to 8.0) and better Strike Rates (11-22) are likely clustered together, while fast bowlers may show variability in these metrics.

In terms of standout performances, most bowlers (82) have taken 0 to 2 four-wicket hauls, and only a small portion have achieved multiple five-wicket hauls (19 bowlers with one 5-wicket haul and 6 with two). The tree ultimately reveals how different combinations of metrics such as age, experience, and bowling type influence performance. Fast bowlers might excel in one set of conditions (higher matches, more wickets), while spin bowlers could perform better in another set (better economy, strike rate), showing how these attributes play out in match scenarios.

Accuracy

Accuracy is a measure of how often a classification model correctly identifies both positive and negative classes. It is defined as the ratio of correct predictions (true positives and true negatives) to the total number of predictions made.

$$Accuracy = \frac{True\ Positives\ (TP) + True\ Neagtive(TN)}{Total\ Number\ of\ Predictions}$$

For the given confusion matrix:

$$Accuracy = \frac{13+1}{1+5+1+13} = 0.70$$

Thus, the model correctly classified 14 out of 20 instances, yielding an accuracy of 70%.

Sensitivity (Recall or True Positive Rate)

Sensitivity, also known as recall, measures the proportion of actual positives (in this case, "Spin") that the model correctly identifies. It is defined as:

$$Sensitivity\ (Recall) = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Neagtive(FN)}$$

For the confusion matrix:

$$Sensitivity\ (Recall) = \frac{13}{13+1} = 0.93$$

The model correctly identified 93% of the "Spin" instances.

Specificity (True Negative Rate)

Specificity measures the proportion of actual negatives (in this case, "Fast") that the model. It is defined as:

$$Specificity = \frac{True\ Neagtive(TN)}{True\ Neagtive(TN) + False\ Positivie(FP)}$$

For the confusion matrix:

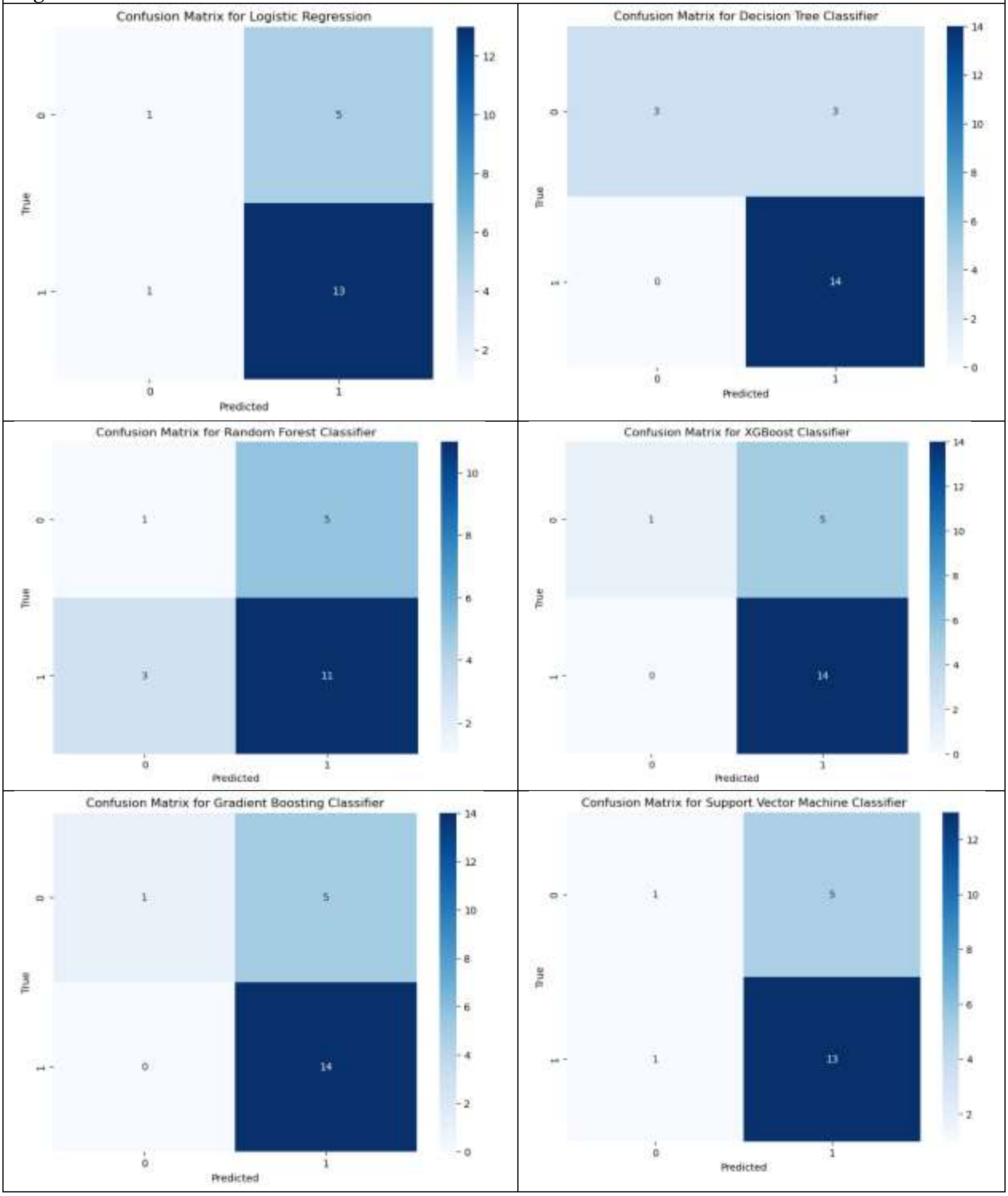
$$Specificity = \frac{13}{13+5} = 0.72$$

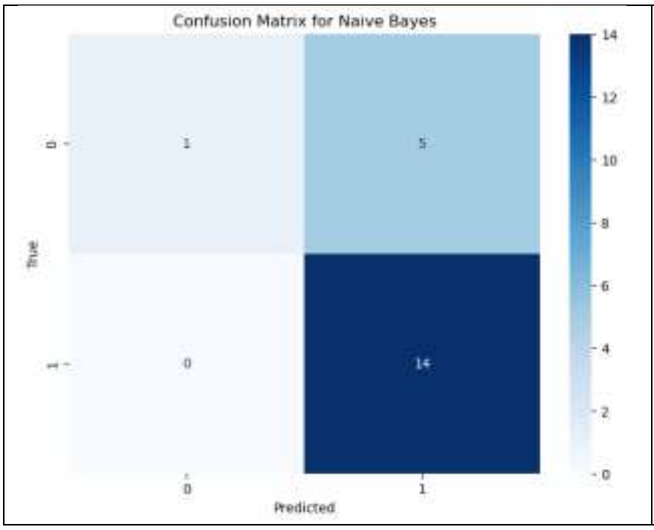
The model correctly identified 72% of the "Fast" instances.

Training Set Score (0.7750): This score indicates that the model performs well on the training data, achieving approximately 77.5% accuracy. A higher score here is often desired, but if it’s too high compared to the test score, it may suggest overfitting.

Test Set Score (0.7500): This score shows the model's performance on unseen data, achieving about 75% accuracy. A score lower than the training score can indicate that the model is not generalizing well to new data.

Figure No # 3: Confusion Matrices of Different Classification Models





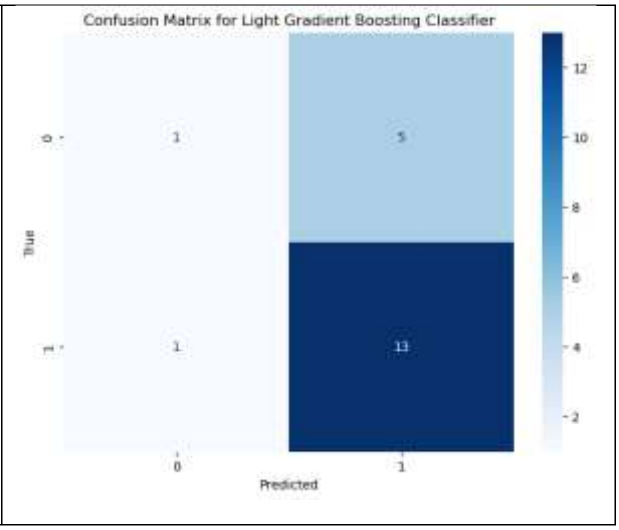


Figure No # 3 shows the "Confusion Matrices of Different Classifiers" which illustrate various classifiers' performance in predicting the bowlers' categories (Spin and Fast). Logistic Regression identifies all fast bowlers but struggles with spin bowlers, yielding low precision for that category. The Decision Tree classifier demonstrates high accuracy for fast bowlers but misclassifies several as spin bowlers, while Random Forest exhibits similar trends with a higher number of false positives. XGBoost and Gradient Boosting effectively classify fast bowlers, accurately identifying them with minimal errors but performing poorly with spin bowlers. Support Vector Machine also performs well for fast bowlers, and Naive Bayes reflects similar results. Overall, classifiers like Logistic Regression and Support Vector Machine excel at identifying fast bowlers, while improvements are needed across all classifiers to enhance accuracy for spin bowlers.

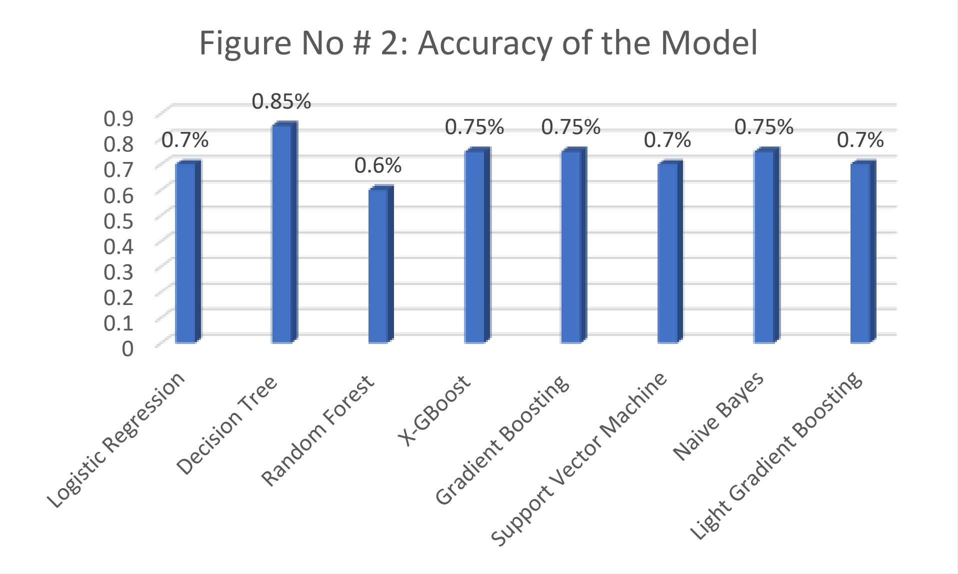


Figure No # 2: Accuracy of the Model

Figure No # 2 shows the model performance results indicating that the Decision Tree is the best performer, achieving an accuracy of 0.85. This suggests it effectively captures the relationships within the dataset, but its tendency to overfit necessitates careful tuning, such as pruning or setting a maximum depth. XGBoost, Gradient Boosting, and Naive Bayes follow closely with an accuracy of 0.75, demonstrating comparable effectiveness. These models can leverage non-linear relationships in the data, making them strong candidates for further exploration and hyperparameter tuning. Logistic Regression and Support Vector Machines, both at 0.70 accuracy, may not fully capture the complexities of the dataset, indicating a need for feature engineering and kernel experimentation for SVM. On the other hand, the Random Forest model performed the worst with an accuracy of 0.60, which is surprising given its ensemble nature. This may indicate issues with data representation or model configuration, suggesting a need to adjust parameters such as the number of trees or tree depth. Overall, the results emphasize the importance of hyperparameter tuning and feature analysis to enhance model performance.

Table No # 2: Comparison of different Classification Models

Models	**Confusion Matrix**	**Accuracy**	**Categories**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	$\begin{bmatrix} 1 & 5 \\ 1 & 13 \end{bmatrix}$	0.70	Spin	0.50	0.17	0.25
			Fast	0.72	0.93	0.81
Decision Tree		0.85	Spin	1.00	0.50	0.67

	$\begin{bmatrix}3 & 3\\0 & 14\end{bmatrix}$		Fast	0.82	1.00	0.90
Random Forest	$\begin{bmatrix}1 & 5\\3 & 11\end{bmatrix}$	0.60	Spin	0.25	0.17	0.20
			Fast	0.69	0.79	0.73
X-GBoost	$\begin{bmatrix}1 & 5\\0 & 14\end{bmatrix}$	0.75	Spin	1.00	0.17	0.29
			Fast	0.74	1.00	0.85
Gradient Boosting	$\begin{bmatrix}1 & 5\\0 & 14\end{bmatrix}$	0.75	Spin	1.00	0.17	0.29
			Fast	0.74	1.00	0.85
Support Vector Machine	$\begin{bmatrix}1 & 5\\1 & 13\end{bmatrix}$	0.70	Spin	0.50	0.17	0.25
			Fast	0.72	0.93	0.81
Naive Bayes	$\begin{bmatrix}1 & 5\\0 & 14\end{bmatrix}$	0.75	Spin	1.00	0.17	0.29
			Fast	0.74	1.00	0.85
Light Gradient Boosting	$\begin{bmatrix}1 & 5\\1 & 13\end{bmatrix}$	0.70	Spin	0.50	0.17	0.25
			Fast	0.72	0.93	0.81

Table No # 2 shows the comparison of different classification models, as detailed in the table, reveals varying performance metrics across categories, specifically for Spin and Fast bowlers. Logistic Regression achieves an overall accuracy of 0.70, demonstrating low precision (0.50) and recall (0.17) for spin bowlers, but better performance for fast bowlers, with precision at 0.93 and recall at 0.81. In contrast, the Decision Tree classifier stands out with the highest accuracy of 0.85, obtaining perfect precision (1.00) for spin bowlers, although its recall is only 0.50, while excelling for fast bowlers with a precision of 0.82 and perfect recall (1.00). The Random Forest classifier shows lower accuracy at 0.60, with poor precision (0.25) and recall (0.17) for spin bowlers but improved performance for fast bowlers (precision of 0.69 and recall of 0.79). Both X-GBoost and Gradient Boosting exhibit similar results with an accuracy of 0.75, achieving perfect precision for spin bowlers but low recall (0.17), while successfully identifying fast bowlers (precision of 0.74 and recall of 1.00). Support Vector Machine presents an accuracy of 0.70, mirroring Logistic Regression in performance for spin bowlers but performing better for fast bowlers (precision of 0.72 and recall of 0.93). Naive Bayes matches the results of XGBoost and Gradient Boosting with an accuracy of 0.75, achieving perfect precision for spin bowlers yet struggling with recall. Lastly, Light Gradient Boosting reflects the same performance metrics as Support Vector Machine and Logistic Regression, indicating an accuracy of 0.70 with low effectiveness in classifying spin bowlers but reasonable results for fast bowlers. Overall, the Decision Tree classifier emerges as the best performer in terms of overall accuracy and spin bowler classification, highlighting the need for further optimization in all classifiers to enhance their effectiveness in accurately identifying spin bowlers.

Conclusion

The analysis of the bowler dataset highlights significant insights into player demographics and performance metrics. The dataset presents a comprehensive view of bowlers, with attributes such as age, bowling style, playing role, and various performance statistics. A key finding is that the majority of specialized bowlers fall within the age group of 26 to 33 years, which is identified as a peak performance period. This demographic trend indicates that younger and mid-career bowlers are more prevalent, while older bowlers contribute predominantly to the batting allrounder category. The data suggests a strong emphasis on specialized bowling roles, reinforcing the importance of age and experience in cricket performance.

Various supervised classification algorithms were employed to distinguish between spin and fast bowlers, with their effectiveness evaluated through accuracy, precision, recall, and F1-score. The Decision Tree classifier emerged as the best-performing model, achieving an accuracy of 0.85 and perfect precision for identifying spin bowlers. However, its recall for the spin category was lower, suggesting that while it successfully identifies those classified as spin bowlers, it may overlook some instances. Other models, including XGBoost, Gradient Boosting, and Naive Bayes, showed comparable performance, with accuracies around 0.75 but varying effectiveness in classifying spin bowlers.

In contrast, Logistic Regression and Support Vector Machines demonstrated lower accuracy (0.70) and precision for spin bowlers, highlighting potential difficulties in capturing the complexities of the dataset. The Random Forest classifier's performance was unexpectedly low at 0.60 accuracy, suggesting issues with model configuration or data representation. Overall, these findings underscore the need for continued optimization of classification models through hyperparameter tuning and feature analysis to enhance their effectiveness in predicting bowler categories.

The results emphasize the intricate relationships between various attributes, performance metrics, and the classifications made by different algorithms. The study indicates that the Decision Tree model effectively captures these relationships but requires careful tuning to mitigate overfitting. The varying performances of other models point to the necessity for further exploration of their potential through additional training and validation techniques. This comprehensive evaluation serves as a foundation for developing more robust models in predicting bowler classifications, ultimately contributing to better strategic decisions in cricket.

References

1. Abid, A., Hassan, B., Abid, K., Farooq, U., Shoaib, M., & Naeem, M. A. (2018). Sports culture in south Asia: Effects of modern bowling action rules on cricket, an information technology perspective. *South Asian Studies*, *33*(01), 211-220.
2. Passi, K., & Pandey, N. (2018). Increased prediction accuracy in the game of cricket using machine learning. *arXiv preprint arXiv:1804.04226*.

3. Pramanik, M. A., Suzan, M. M. H., Biswas, A. A., Rahman, M. Z., & Kalaiarasi, A. (2022, January). Performance analysis of classification algorithms for outcome prediction of T20 cricket tournament matches. In *2022 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 01-07). IEEE
4. Hucaljuk and A. Rakipovik, "Predicting football scores using machine learning techniques," in International Convention MIPRO, Opatija, 2011.
5. M. G. Jhanwar and V. Pudi, "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach," in European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2016 2016), 2016.
6. J. McCullagh, "Data Mining in Sport: A Neural Network Approach," International Journal of Sports Science and Engineering, vol. 4, no. 3, pp. 131-138, 2012
7. H. H. Lemmer, "The combined bowling rate as a measure of bowling performance in cricket," South African Journal for Research in Sport, Physical Education and Recreation, vol. 24, no. 2, pp. 37-44, January 2002.
8. R. P. Schumaker, O. K. Solieman and H. Chen, "Predictive Modeling for Sports and Gaming," in Sports Data Mining, vol. 26, Boston, Massachusetts: Springer, 2010.
9. S. R. Iyer and R. Sharda, "Prediction of athletes performance using neural networks: An application in cricket team selection," Expert Systems with Applications, vol. 36, pp. 5510-5522, April 2009.
10. D. Bhattacharjee and D. G. Pahinkar, "Analysis of Performance of Bowlers using Combined Bowling Rate," International Journal of Sports Science and Engineering, vol. 6, no. 3, pp. 1750-9823, 2012.
11. S. Muthuswamy and S. S. Lam, "Bowler Performance Prediction for One-day International Cricket Using Neural Networks," in Industrial Engineering Research Conference, 2008.
12. Kapadia, K., Abdel-Jaber, H., Thabtah, F., & Hadi, W. (2022). Sport analytics for cricket game results using machine learning: An experimental study. *Applied Computing and Informatics*, *18*(3/4), 256-266.
13. Lamsal, R., & Choudhary, A. (2018). Predicting outcome of Indian premier league (IPL) matches using machine learning. *arXiv preprint arXiv:1809.09813*.
14. Tripathi, A., Islam, R., Khandor, V., & Murugan, V. (2020). Prediction of IPL matches using Machine Learning while tackling ambiguity in results. *Indian J. Sci. Technol*, *13*(38), 4013-4035.
15. Pathak, N., & Wadhwa, H. (2016). Applications of modern classification techniques to predict the outcome of ODI cricket. *Procedia Computer Science*, *87*, 55-60.
16. Pramanik, M. A., Suzan, M. M. H., Biswas, A. A., Rahman, M. Z., & Kalaiarasi, A. (2022, January). Performance analysis of classification algorithms for outcome prediction of T20 cricket tournament matches. In *2022 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 01-07). IEEE.
17. Kumar, J., Kumar, R., & Kumar, P. (2018, December). Outcome prediction of ODI cricket matches using decision trees and MLP networks. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 343-347). IEEE.
18. Rahman, M. M., Shamim, M. O. F., & Ismail, S. (2018, October). An analysis of Bangladesh one day international cricket data: a machine learning approach. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)* (pp. 190-194). IEEE.
19. Mahmood, T., Riaz, M., Nasir, M., Afzal, U., & Siddiqui, M. H. (2021). Psl eye: Predicting the winning team in pakistan super league (psl) matches. *KIET Journal of Computing and Information Sciences*, *4*(2), 13-13.
20. Mahtab, S. A., Islam, N., & Rahaman, M. M. (2018, September). Sentiment analysis on bangladesh cricket with support vector machine. In *2018 international conference on Bangla speech and language processing (ICBSLP)* (pp. 1-4). IEEE.
21. Ren, Y., Zhang, L., & Suganthan, P. N. (2016). Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational intelligence magazine*, *11*(1), 41-53.
22. Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20-28.
23. Liu, Y. Y., Yang, M., Ramsay, M., Li, X. S., & Coid, J. W. (2011). A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending. Journal of Quantitative Criminology, 27, 547-573.
24. Yang, F. J. (2018, December). An implementation of naive bayes classifier. In *2018 International conference on computational science and computational intelligence (CSCI)* (pp. 301-306). IEEE.
25. Zheng, S., & Man, X. (2022). An Improved Logistic Regression Method for Assessing the Performance of Track and Field Sports. *Computational Intelligence and Neuroscience*, *2022*(1), 6341495.
26. Mahtab, S. A., Islam, N., & Rahaman, M. M. (2018, September). Sentiment analysis on bangladesh cricket with support vector machine. In *2018 international conference on Bangla speech and language processing (ICBSLP)* (pp. 1-4). IEEE.
27. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
28. Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.
29. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
30. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.
31. Quinto, B. (2020). *Next-generation machine learning with spark: Covers XGBoost, LightGBM, Spark NLP, distributed deep learning with keras, and more.* Apress.