Power System Technology
ISSN:1000-3673

Received: 06-06-2024 *Revised: 15-07-2024* *Accepted: 28-08-2024*

Prediction of Heteroscedastic Sports Data By using Regression and Various Machine Learning Models

Muhammad Waqas[1], Qamruz Zaman[2], Danish waseem[3],Sofia[4], Bushra Haider[5], Gohar Ayub[*6] Shahid Iqbal[7]

[1,2,5]Department of Statistics, University of Peshawar, Pakistan
[3]Department of Management Sciences, Abasyn University Peshawar, Pakistan
[4]College of Home Economics, University of Peshawar, Pakistan
[6]Department of Mathematics and Statistics, University of Swat, Swat, Pakistan
[7]Institute of Education & Research, University of Peshawar
Corresponding author email: Gohar Ayub, Department of Mathematics and Statistics, University of Swat, Swat, Pakistan
Emails : cricsportsresearchgroup@gmail.com, goharayub@uswat.edu.pk

Abstract

This study investigates the impact of extra deliveries—wide's, no balls, byes, and leg byes—on T20I cricket match outcomes. It aims to find out whether these extras have a significant effect on the result and which type contributes the most runs. Using heteroscedastic data from the previous three T20I World Cups (20212024), the analysis underlines the importance of reducing extras to boost a team's chances of winning. OLS assumes that the error component has constant variance; when this assumption is violated, heteroscedasticity arises, resulting in incorrect predictions and conclusions. This study employs several types of machine learning techniques to increase prediction accuracy for heteroscedastic data. We applied machine learning models such as Decision Tree, Random Forest, Gradient Boosting, and K Nearest Neighbor, as well as Linear Regression, to predict values in test datasets. Machine learning models typically outperformed linear regression, resulting in decreased sum of squares errors, with some models producing predictions that were very close to observed values. This study shows that machine learning works well with heteroscedastic data.

Keywords: T20, Extra deliveries, Neural Networks, KNN,

1. Introduction

Every ball and run are crucial in T20 International cricket, which adds to the excitement of the game. The 20over format encourages bowlers and batsmen to be aggressive, innovative, and full of energy. T20 Internationals provide some of the most thrilling moments in cricket, with teams having to score quickly in order to chase down difficult opposition totals. Predicting

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

the most successful run chases in T20 cricket is difficult and critical since it influences a team's strategy and decision making.[1] and [2]. Extras are important in cricket since they can have a substantial impact on a game's outcome. Extras increase a team's overall score, which increases its chances of winning. A single score, whether obtained by batsmen or bowlers, can change the outcome of a match due to extras, such as wide balls and no balls [3].

Types of extra deliveries:

Wide: A delivery is considered wide if it is out of the batter's reach and travels beyond the tram marks of the popping crease without striking the bat. Balls that wander down the leg side in limited overs matches are also immediately considered wide.

NO ball: no ball delivery is one in which the bowler's front foot crosses the crease or if the full toss is above waist height. Wides and no balls result in one run for the batting team and need the ball to be bowled again. Unlike byes and leg byes, runs from wide's and no balls count against the bowler's statistics.

Byes: Byes are awarded when the ball does not strike the bat or the hitter. The batting team can run and score runs, but the hitter does not receive credit for them.

Leg byes: are awarded when the ball strikes any area of the batter's body other than his hands. Runs from leg byes are included in the bowler's scorecard [4].

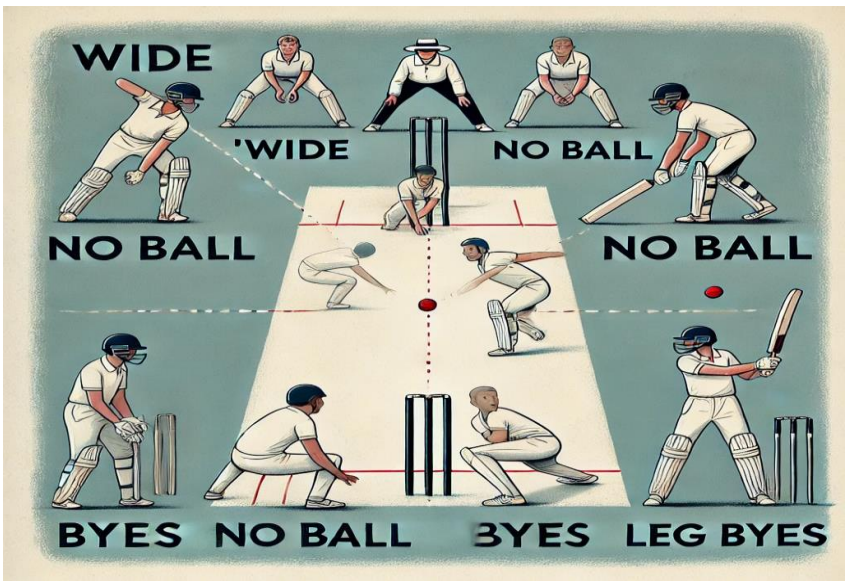


Figure-1: Pictorial Overview of Extras

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

Bowlers who concede too many extras may get demotivated, losing focus and confidence. This might lead to additional runs for the opposition, worsening the bowler's condition. Excessive extras can turn a potential triumph into a bitter defeat by interrupting the bowler's effectiveness and giving the opposition team more scoring opportunities. This emphasizes the significance of bowlers remaining accurate and focused throughout the match. [5-6]. Statistics based prediction and machine learning are two distinct ways to projecting future events or outcomes. When working with a heteroscedastic dataset, choosing the right technique is critical for making correct prediction. Linear regression is a statistical technique that models the relationship between a dependent variable and one or more independent variables [7]. The method implies that residuals (the differences between observed and predicted values) are homoscedastic, which means their variance remains constant across all levels of the independent variables [8]. Heteroscedasticity refers to when the variance of residuals varies with the level of independent factors. Linear regression can produce erroneous and biased coefficient estimates when heteroscedasticity is present [9]. To resolve heteroscedasticity in linear regression, various strategies can be utilized, including weighted least squares, dependent variable transformations, or a separate model that compensates for nonconstant variance. Weighted least squares is a method that assigns weights to each observation according on the independent variable's level. This strategy prioritizes data with larger variance, leading to more accurate forecasts. Statistical tests such as the White test, Breusch Pagan test, and Cook Weisberg test can identify heteroscedasticity. Machine learning prediction is a branch of artificial intelligence (AI) that uses computer algorithms to learn from data and generate predictions. Machine learning algorithms may learn and identify patterns in data without explicit programming. These models can adapt to heteroscedasticity by learning from data and adjusting predictions as needed. Machine learning prediction models include decision trees, random forests, and neural networks. Heteroscedastic datasets can be predicted using both linear regression and machine learning. Consider the chosen model's assumptions and its resistance to heteroscedasticity. Linear regression models need to account for heteroscedasticity, but machine learning models can learn from data. Check the residuals and assumptions of the chosen model to ensure accurate predictions. [10-13].

3. Methodology

In this study, the methodology was designed to rigorously evaluate the predictive performance of an Artificial Neural Network (ANN) model in forecasting cricket match outcomes.

3.1 Data Collection

The dataset for this study was compiled from the official Cricinfo website, a reputable source for comprehensive cricket match data. A total of 100 cricket matches were selected, ensuring a diverse representation of teams, match conditions, and outcomes. The data included various

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

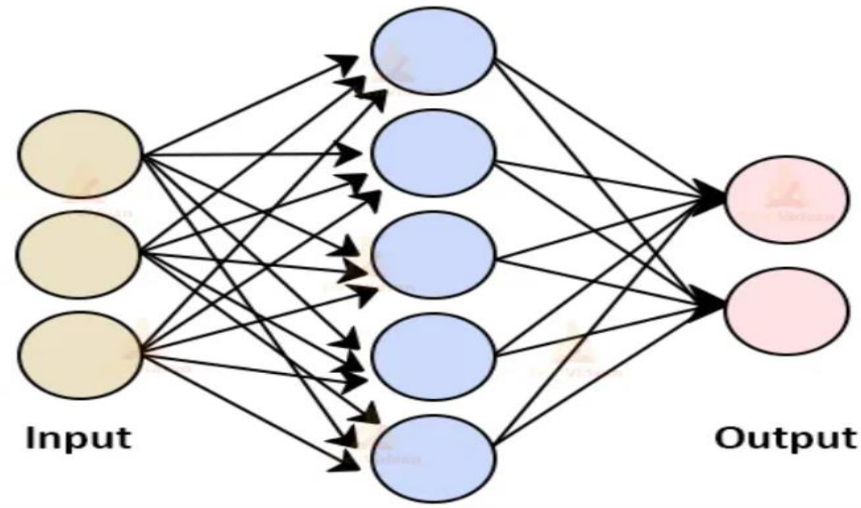
match specific variables such as first and second innings scores, wickets taken, extras, and other relevant factors that could influence the outcome of a match.

3.2 Model Development

The ANN model was developed using SPSS, a powerful tool for statistical analysis and machine learning. The architecture of the neural network was carefully designed, with input layers corresponding to the independent variables, hidden layers to capture complex relationships, and an output layer to predict the match outcome (win/loss).

3.3 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural networks, designed to recognize patterns, classify data, and predict outcomes through a learning process. ANNs are a subset of machine learning algorithms and have become increasingly popular due to their ability to model complex, nonlinear relationships in data. They are widely used in various fields, including finance, healthcare, image and speech recognition, and, as in this study, sports analytics.



3.4. Extreme Gradient Boosting (XG Boost)

XG Boost (Extreme Gradient Boosting) is an advanced machine learning algorithm that has gained significant popularity for its exceptional performance in various predictive modeling tasks. It is an implementation of gradient boosted decision trees designed for speed and performance, often outperforming other algorithms in terms of accuracy and efficiency. XG Boost is based on the gradient boosting framework, where multiple weak learners (typically decision trees) are combined to form a strong learner. Each tree in the sequence corrects the

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

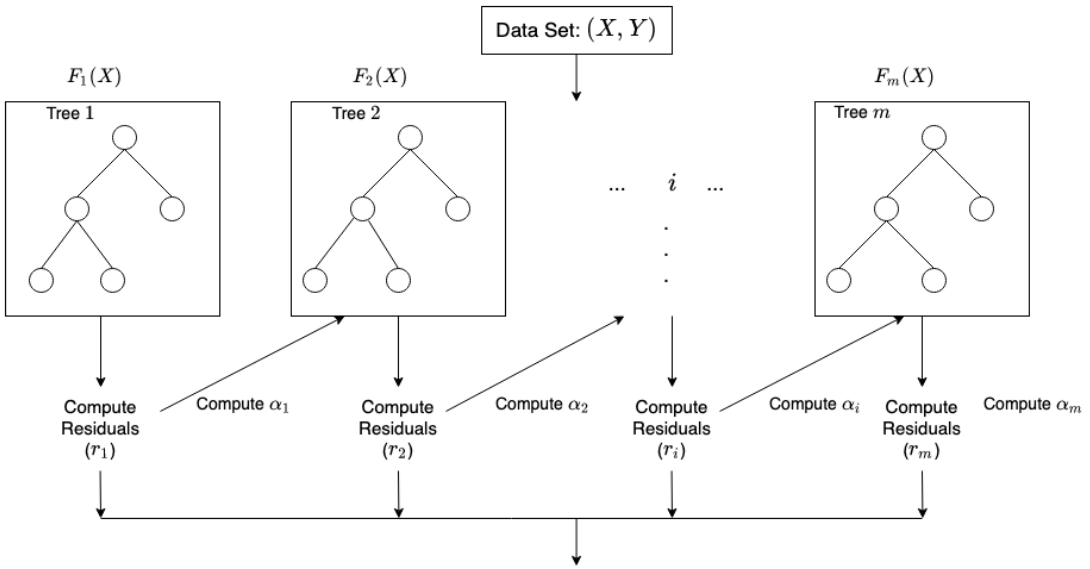
errors made by the previous trees, which leads to a model that is highly accurate. One of the standout features of XG Boost is its ability to include regularization (both L1 and L2), which helps prevent overfitting. Regularization is a technique that penalizes complex models, thereby encouraging the model to generalize better on unseen data. XG Boost can handle missing data inherently, allowing it to be more flexible and robust in real-world applications where missing values are common. It uses a sophisticated method to handle missing data, often leading to better performance than other algorithms [14].

XG Boost is optimized for both parallel and distributed computing. This makes it incredibly fast, even on large datasets, as it can efficiently utilize multiple cores of a processor or run on distributed systems. Unlike other boosting algorithms, XG Boost employs a technique known as "maxdepth" pruning, where trees are pruned backward. This technique prunes branches that do not improve the model, making it more efficient and less prone to overfitting. XG Boost allows the use of custom loss functions, providing flexibility to optimize the model for specific needs. This is particularly useful in cases where the standard loss functions (like mean squared error or logistic loss) do not align well with the problem at hand. XG Boost supports a wide range of objective functions, including regression, classification, and ranking tasks. This versatility allows it to be applied to many different types of machine learning problem. XG Boost is widely used for binary and multiclass classification problems. It is particularly effective in scenarios with imbalanced datasets, where it can still identify the minority class with high accuracy [15].

Regression: XG Boost is also commonly used in regression tasks, where it predicts continuous outcomes. Its ability to handle nonlinear relationships makes it superior to linear models in many cases. Ranking: In ranking tasks, such as search engine results or recommendation systems, XG Boost can be used to order items according to relevance or importance. Time Series Forecasting: XG Boost can be adapted for time series forecasting by engineering features related to time and using the model to predict future values based on past data. High Accuracy: Due to its ability to build strong models by correcting errors iteratively, XG Boost often achieves higher accuracy compared to other algorithms like logistic regression, decision trees, or even random forests. Speed and Efficiency: XG Boost is known for its computational efficiency, handling large datasets and high dimensional data quickly.

XG Boost is a powerful and versatile machine learning algorithm that excels in many predictive modeling tasks. Its combination of speed, accuracy, and flexibility makes it a popular choice in competitions, research, and industry applications. Despite its complexity and resource requirements, XG Boost remains a go to algorithm for many data scientists and machine learning practitioners when high performance is critical [16].

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024



3.5. Logistic regression

Logistic regression is a widely used statistical method for modeling the relationship between a dependent variable and one or more independent variables when the dependent variable is binary. Despite its name, logistic regression is a classification algorithm rather than a regression model. It predicts the probability that a given input belongs to a particular category, typically labeled as 0 or 1.

Logistic regression is used when the outcome variable is binary, meaning it has two possible outcomes (e.g., success/failure, yes/no, win/lose). The model estimates the probability that a given input falls into one of these categories [17].

The core of logistic regression is the logit function, which links the linear combination of the independent variables to the probability of the dependent variable being 1. The logit function is the natural logarithm of the odds of the dependent variable:

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_{(1X_1)} + \beta_{(2X_2)} + \ldots + \beta_{(nX_n)}$$

Here, p is the probability of the outcome being 1, $X_1, X_2, \ldots, X_n$ are the independent variables, and $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients.

The odds are the ratio of the probability of the event occurring to the probability of it not occurring. Logistic regression models the log odds as a linear function of the independent variables, and this can be transformed back to a probability:

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}$$

This logistic function ensures that the output of the model is always between 0 and 1, which can be interpreted as a probability [18].

Logistic regression uses maximum likelihood estimation (MLE) to estimate the coefficients of the model. MLE finds the values of the coefficients that maximize the likelihood of observing the given data.

The coefficients in logistic regression represent the change in the log odds of the outcome for a one unit change in the corresponding independent variable, holding all other variables constant. Exponentiating the coefficients gives the odds ratio, which indicates how the odds of the outcome change with a one unit increase in the predictor.

Binary Classification: Logistic regression is commonly used in scenarios where the goal is to classify observations into one of two categories, such as predicting whether a customer will purchase a product (yes/no) or whether a patient has a disease (positive/negative).

Credit Scoring: Logistic regression is used in finance to predict the probability of default on loans. The model helps assess the risk associated with lending.

Medical Research: In healthcare, logistic regression is often used to predict the presence or absence of a disease based on various risk factors.

Marketing: Companies use logistic regression to predict customer responses to marketing campaigns, such as whether a customer will click on an advertisement or make a purchase.

Simplicity: Logistic regression is easy to implement and interpret, making it a popular choice for binary classification problems. The coefficients can be easily understood in terms of odds ratios, which are intuitive to many decisionmakers.

Efficiency: Logistic regression can handle large datasets efficiently, and it performs well when the relationship between the independent variables and the dependent variable is approximately linear.

Probabilistic Output: Unlike many classification algorithms, logistic regression provides a probabilistic output, giving not just a class prediction but also the probability of that prediction. This can be useful in applications where uncertainty quantification is important [19].

No Assumptions on Distributions: Logistic regression does not assume that the predictors are normally distributed, making it flexible for various types of data.

Linearity in Log Odds: Logistic regression assumes a linear relationship between the log odds of the dependent variable and the independent variables. This can be a limitation when the true relationship is nonlinear.

Limited to Binary Outcomes: Standard logistic regression is designed for binary outcomes. Extensions like multinomial logistic regression are required for multiclass classification problems.

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

Sensitive to Outliers: Logistic regression can be sensitive to outliers, particularly when using continuous variables. Outliers can disproportionately affect the estimated coefficients.

Multicollinearity Issues: Logistic regression assumes that the independent variables are not highly correlated with each other. Multicollinearity can lead to unstable estimates of the coefficients and reduce the model's interpretability.

Logistic regression is a fundamental and widely used technique in binary classification problems. Its simplicity, efficiency, and probabilistic output make it an excellent starting point for many predictive modeling tasks. However, its assumptions and limitations should be carefully considered when applying it to real-world data. Despite these limitations, logistic regression remains a powerful tool, particularly when the relationship between the independent variables and the outcome is approximately linear, and interpretability is crucial.

3.6 Decision trees

Decision trees are a popular and intuitive machine learning algorithm used for both classification and regression tasks. A decision tree model works by recursively splitting the data into subsets based on the value of input features, leading to a treelike structure where each node represents a decision point and each branch represents an outcome [20].

A decision tree consists of nodes, branches, and leaves:

Root Node: The topmost node of the tree that represents the entire dataset. It is the starting point where the first split occurs.

Decision Nodes: Intermediate nodes that represent decisions or tests on the attributes. Each node splits the data into two or more branches based on a feature's value.

Leaves: The terminal nodes of the tree, representing the final decision or outcome. In a classification tree, leaves represent class labels, while in a regression tree, they represent continuous values.

The process of splitting nodes in a decision tree is guided by criteria that aim to maximize the separation of the data. Common splitting criteria include:

Gini Impurity: Measures the likelihood of incorrectly classifying a randomly chosen element if it were randomly labeled according to the distribution of labels in the dataset. It is used in classification tasks.

Information Gain: Based on the concept of entropy, it measures how much uncertainty is reduced after a split. The goal is to choose splits that provide the most information.

Variance Reduction: Used in regression tasks, it measures how much the variance is reduced after a split. The goal is to minimize the variance within each of the resulting subsets.

Decision trees use a process called recursive partitioning, where the data is split into subsets based on feature values repeatedly, creating a tree structure. The process continues until a stopping criterion is met, such as a maximum depth, minimum number of samples per leaf, or

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

no further improvement in the split.

Pruning is the process of removing branches from the tree that have little importance, in order to reduce complexity and improve generalization. Pruning can be done preemptively (prepruning) by setting constraints like maximum depth, or after the tree has been fully grown (postpruning) by removing branches that contribute little to predictive power.

Decision trees can handle both categorical and numerical data. For categorical data, the tree splits based on the different categories, while for numerical data, it splits based on threshold values.

Classification: Decision trees are commonly used for classification tasks where the goal is to categorize instances into discrete classes, such as spam detection, disease diagnosis, or customer segmentation.

Regression: In regression tasks, decision trees predict continuous values, such as predicting house prices, stock prices, or temperature.

Feature Importance: Decision trees can be used to rank the importance of features in a dataset. Features that are used near the root of the tree are generally more important for prediction.

Interpretability: One of the biggest advantages of decision trees is their interpretability. The tree structure is easy to visualize and understand, making it clear how decisions are made.

No Need for Feature Scaling: Decision trees do not require feature scaling (e.g., normalization or standardization), as the algorithm is based on splitting values rather than distances between points.

Handles Both Types of Data: Decision trees can handle both categorical and numerical data without the need for onehot encoding or other transformations.

Nonlinear Relationships: Decision trees are capable of capturing nonlinear relationships between features and the target variable.

Robust to Outliers: Unlike some other models, decision trees are relatively robust to outliers, as splits are based on the majority of the data.

Overfitting: Decision trees are prone to overfitting, especially when they are allowed to grow too deep. Overfitting occurs when the tree captures noise in the data rather than the underlying pattern, leading to poor performance on unseen data.

Instability: Small changes in the data can result in very different tree structures, as the algorithm is sensitive to the specifics of the dataset.

Bias Toward Dominant Classes: In imbalanced datasets, decision trees may become biased toward the majority class, leading to poor performance on minority classes.

Lack of Smooth Predictions: For regression tasks, decision trees produce piecewise constant predictions, which may not be smooth and can lead to high variance.

Random Forests: A popular ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It mitigates some of the weaknesses of individual decision trees by averaging their outputs.

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

Gradient Boosting Machines (GBMs): Another ensemble technique that builds decision trees sequentially, where each tree attempts to correct the errors of the previous ones. XG Boost, a variant of GBMs, is known for its high performance in competitions and real-world applications.

CART (Classification and Regression Trees): A specific type of decision tree algorithm that is commonly used. CART is the foundation for many tree-based methods, including random forests and boosting methods [21].

Decision trees are a fundamental and versatile machine learning technique with a wide range of applications in classification and regression tasks. Their simplicity, interpretability, and ability to handle various types of data make them a popular choice in many scenarios. However, care must be taken to avoid overfitting, and in some cases, ensemble methods like random forests or boosting may be preferred to enhance performance. Despite their limitations, decision trees remain a powerful tool in the machine learning arsenal, especially when transparency and interpretability are key considerations.

3.7. Random Forest.

Random Forest is a powerful ensemble learning algorithm that builds multiple decision trees and combines their predictions to improve accuracy and reduce the likelihood of overfitting. It is widely used for both classification and regression tasks and is particularly well-regarded for its robustness, ease of use, and ability to handle a variety of data types.

Random Forest belongs to a class of machine learning methods known as ensemble learning, where multiple models (in this case, decision trees) are combined to produce a more accurate and stable prediction than any individual model could achieve alone.

A key component of Random Forest is the bagging technique, which involves creating multiple subsets of the original dataset by sampling with replacement. Each decision tree in the forest is trained on a different subset, which helps to reduce variance and prevent overfitting. Because each tree is trained on different data, they become slightly different from one another.

Unlike traditional decision trees, where all features are considered for splitting at each node, Random Forest selects a random subset of features for each split. This randomness further reduces correlation among the trees, leading to a more diverse ensemble and improving overall model performance. For classification tasks, each tree in the forest makes a prediction, and the final output is determined by majority voting, where the class with the most votes is selected. For regression tasks, the predictions from each tree are averaged to produce the final output. This aggregation process helps to smooth out predictions and reduce the model's variance [22].

Random Forest has an inherent mechanism to estimate the model's performance without the need for a separate validation set. Since each tree is trained on a different bootstrap sample, about one-third of the data is left out of each sample. This data, known as outofbag (OOB)

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

data, can be used to evaluate the model's accuracy, providing an unbiased estimate of its performance.

Random Forest provides a measure of feature importance, which indicates how much each feature contributes to the model's predictions. This is calculated by assessing the impact of each feature on the accuracy of the model when it is permuted (randomly shuffled). Features that result in a significant drop in accuracy when permuted are considered important.

Classification: Random Forest is commonly used in classification tasks where it excels in handling large datasets with many features and in dealing with imbalanced data. Applications include spam detection, medical diagnosis, customer segmentation, and more.

Regression: Random Forest is also effective in regression tasks, such as predicting prices, stock market trends, or environmental factors. Its ability to model complex relationships without overfitting makes it a popular choice for these applications [23].

Feature Selection: Due to its built-in feature importance measure, Random Forest is often used to identify the most important features in a dataset, which can be valuable for reducing dimensionality and improving model performance.

Anomaly Detection: Random Forest can be used in anomaly detection, particularly when dealing with high dimensional data. By analyzing the decision paths taken by each tree, the model can identify instances that deviate significantly from the norm.

High Accuracy: Random Forest generally provides high predictive accuracy, particularly when dealing with complex datasets with numerous features. The ensemble approach ensures that the model captures a wide range of patterns in the data.

Robustness to Overfitting: The use of multiple decision trees trained on different subsets of data reduces the likelihood of overfitting, making Random Forest a robust choice for many applications.

Handles Missing Data: Random Forest can handle missing data by using the information from other trees or by imputing missing values based on the proximity of the data points in the feature space.

Resistant to Noisy Data: Due to its ensemble nature, Random Forest is less sensitive to noise in the data compared to individual decision trees.

Scalability: Random Forest can be easily parallelized, making it suitable for large datasets and scalable across distributed computing environments.

Complexity: While Random Forest models are more accurate than individual decision trees, they are also more complex and harder to interpret. The ensemble of trees can be seen as a "black box," which may be a drawback in applications where model interpretability is crucial.

Slower Prediction: Because Random Forest makes predictions by averaging or voting across many trees, it can be slower in generating predictions compared to simpler models like individual decision trees or logistic regression.

Memory Intensive: Random Forest models can be memory intensive, especially when dealing

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

with large datasets and a high number of trees. This can be a limitation in resource constrained environments.

Random Forest is a versatile and powerful machine learning algorithm that balances accuracy, robustness, and ease of use. Its ability to handle a wide variety of data types, combined with its resistance to overfitting and noise, makes it a go to choice for many classification and regression tasks. While it can be computationally intensive and less interpretable than simpler models, its advantages often outweigh these drawbacks, particularly in applications where predictive accuracy is paramount. Whether used for classification, regression, or feature selection, Random Forest continues to be a cornerstone of modern machine learning [24].

4. Results and Discussion

4.1. Summary Statistics

Variables	Min	Mean	Median	Max
Score 1st	44	144	151	218
Wickets 1st	2	7	7	10
Over played 1st	10	20	20	20
Total extras in 1st	2	7	7	14
Score 2nd	39	125	125	197
Wicket 2nd	0	6	5	20
Over played 2nd	3	17	18	20
Total extras in 2nd inning	1	6	6	13
Total runs by wide	3	7	7	13
By No balls	0	6	6	19
Byes	0	2	1	7
Leg byes	0	2	1	7
Run difference	1	21	4	134

4.2. Model Summary

Based on the testing errors obtained from applying various models to predict match winning outcomes, here's an interpretation. The logistic regression model has the highest testing error (0.471) among all the models tested. This suggests that logistic regression is not very effective in predicting match outcomes based on the independent variables provided. The high error indicates that the model frequently makes incorrect predictions. The decision tree model performs significantly better than logistic regression, with a much lower testing error of 0.0714. This indicates that the decision tree is more accurate in predicting match outcomes, though it

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

is still less accurate compared to the other advanced models like Random Forest, XG Boost, and KNN. The Random Forest model further improves the prediction accuracy, with a testing error of 0.031. This low error rate suggests that Random Forest is a strong performer, likely due to its ability to reduce overfitting and handle the complexity of the data better than a single decision tree. XG Boost achieves the lowest testing error (0.012) among all models, indicating that it is the most accurate model for predicting match outcomes. XG Boost's success can be attributed to its advanced boosting techniques, which combine multiple weak learners into a strong learner and effectively capture the relationships in the data. The KNN model also performs well with a testing error of 0.0571, but it is slightly less accurate than Random Forest and XG Boost and KNN's performance suggests that the proximity of data points (in terms of similarity in features) is somewhat useful in predicting match outcomes, but it is not as powerful as the ensemble methods. XG Boost is the most effective model for predicting match outcomes based on the independent variables provided, with the lowest testing error. This suggests that it captures the underlying patterns in the data better than the other models. Random Forest also performs well and is a strong alternative to XG Boost, offering low error and robustness to overfitting. KNN and Decision Tree models provide moderate accuracy, with KNN performing better than the Decision Tree. Logistic Regression shows the highest testing error, indicating that it may not be well-suited for this particular prediction task, potentially due to the linear nature of the model or the complexity of the relationships among the variables. For predicting match winning outcomes with the given independent variables, XG Boost is the most recommended model, followed by Random Forest. Both models significantly outperform the others in terms of accuracy. The performance difference between these models highlights the importance of using advanced machine learning techniques for complex predictive tasks.

Model	**Testing error**
Logistic Regression	0.471
Decision tree	0.0714
Random Forest	0.031
XG Boost	**0.012**
KNN	0.0571

The summary of the logistic model shows in the following tables. The intercept has a large negative estimate (8.3230), but it is not statistically significant (p = 0.7370). This indicates that the baseline log odds of the outcome when all predictors are zero is very low, but this estimate is highly uncertain.

The coefficient for the score in the first innings is negative (0.4783) but not statistically significant (p = 0.1565). This suggests that higher scores in the first innings are associated with slightly lower odds of winning, though this relationship is not strong enough to rule out random

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

chance. The positive coefficient (0.5841) for wickets lost in the first innings implies a potential increase in the odds of winning as more wickets are lost, but this effect is also not statistically significant (p = 0.2468). This result is counterintuitive and may indicate multicollinearity or other underlying complexities in the data. The coefficient (2.1726) for the number of overs played in the first innings is positive, suggesting that playing more overs is associated with higher odds of winning. However, this result is not statistically significant (p = 0.1486), indicating that the relationship could be due to chance. The negative coefficient (0.4466) for extras given in the first innings indicates that more extras might reduce the odds of winning, but the effect is not statistically significant (p = 0.2442). The positive coefficient (0.4862) for the score in the second innings suggests that higher scores in the second innings are associated with increased odds of winning, though the result is not statistically significant (p = 0.1349). The coefficient (0.9848) for wickets lost in the second innings is negative, indicating that losing more wickets in the second innings may decrease the odds of winning. This result is marginally significant (p = 0.0519), suggesting that this factor might have some predictive value, but it is close to the conventional threshold for significance The number of overs played in the second innings has a significant negative coefficient (1.5220, p = 0.0102). This suggests that playing more overs in the second innings is strongly associated with lower odds of winning, indicating that teams playing more overs might struggle to chase or defend their score. The coefficient for extras in the second innings is negative (0.6732) but not statistically significant (p = 0.3311), suggesting that more extras in the second innings could reduce the odds of winning, though the effect is not strong. The coefficient for runs given by wide's is negative (0.3806) but not statistically significant (p = 0.4621). This indicates that more runs given by wide's might reduce the odds of winning, but this effect is not robust. The coefficient for runs given by no balls is positive (0.2001) but not statistically significant (p = 0.5182), indicating a weak and uncertain relationship. The positive coefficient (0.9856) for runs given by byes suggests that more runs given by byes could increase the odds of winning, though the result is not statistically significant (p = 0.1639). The coefficient for runs given by leg byes is positive (1.2225) but not statistically significant (p = 0.1477). This indicates that more runs given by leg byes might increase the odds of winning, but the effect is not strong. The positive coefficient (0.4034) suggests that a higher difference in runs is associated with increased odds of winning, but the relationship is not statistically significant (p = 0.1988).

Most predictors in this model are not statistically significant, indicating that their association with the odds of winning may be due to chance. The only statistically significant predictor is the number of overs played in the second innings, which has a negative relationship with the odds of winning. This suggests that teams playing more overs in the second innings tend to have lower odds of winning, possibly reflecting difficulties in successfully chasing or defending scores in extended play. The marginal significance of the wickets lost in the second innings suggests that losing more wickets might reduce the odds of winning, but this result is

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

borderline.

The model overall appears to have a reasonable fit (Residual Deviance = 17.284, AIC = 45.284), but the lack of strong predictors might indicate that the model could be improved by including other relevant variables or considering interactions between the current variables.

Predictor Variable	B	SE	z	p	95% CI for B
Intercept	8.323	24.7846	0.336	0.737	[57.888, 41.242]
Score_1st	0.4783	0.3376	1.417	0.1565	[1.140, 0.184]
Wickets_1st	0.5841	0.5043	1.158	0.2468	[0.404, 1.572]
Overs_played_1st	2.1726	1.5041	1.444	0.1486	[0.775, 5.120]
Total_extras_in_1st_innings	0.4466	0.3835	1.165	0.2442	[1.198, 0.305]
Score_2nd	0.4862	0.3252	1.495	0.1349	[0.151, 1.124]
Wickets_2nd	0.9848	0.5065	1.944	0.0519	[1.978, 0.008]
Overs_played_2nd	1.522	0.5923	2.57	0.0102	[2.682, 0.362]
Total_Extras_in_2nd_innings	0.6732	0.6926	0.972	0.3311	[2.031, 0.685]
Total runs by Wides	0.3806	0.5175	0.735	0.4621	[1.395, 0.634]
By No balls	0.2001	0.3098	0.646	0.5182	[0.407, 0.807]
By Byes	0.9856	0.708	1.392	0.1639	[0.402, 2.373]
By Leg byes	1.2225	0.8445	1.448	0.1477	[0.433, 2.878]
Runs Difference	0.4034	0.3139	1.285	0.1988	[0.212, 1.019]

Table 4.1: Case Processing Summary

		N	Percent
Sample	Training	76	76.0%
	Testing	24	24.0%
Valid		100	100.0%
Excluded		0	
Total		100	

The Case Processing Summary indicates that in your Artificial Neural Network (ANN) model, the dataset was fully utilized, with 76% (76 cases) allocated to training the model and 24% (24 cases) reserved for testing its performance. All 100 cases were considered valid, with no exclusions, ensuring a comprehensive analysis of the data. This division allows the model's effectiveness to be evaluated based on its predictions on the testing set after being trained on the training set.

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

Table 4.2: Model Summary

Training	Cross Entropy Error	4.038
	Percent Incorrect Predictions	1.3%
Testing	Cross Entropy Error	8.620
	Percent Incorrect Predictions	4.2%

The Model Summary reveals key performance metrics of your Artificial Neural Network (ANN) during both the training and testing phases. During training, the model achieved a Cross Entropy Error of 4.038, with a low percentage of incorrect predictions at 1.3%, indicating that the model was able to learn the patterns in the data quite effectively. However, when the model was tested on the separate testing dataset, the Cross Entropy Error increased to 8.620, and the percentage of incorrect predictions rose to 4.2%. This suggests that while the model performed well during training, its predictive accuracy slightly decreased when applied to new, unseen data, though it still maintained a relatively low error rate. This discrepancy between training and testing errors might indicate a need to refine the model to improve its generalization to new data.

Table 4.3: Classification

Sample		Predicted		
		Team 1	Team 2	Percent Correct
Training	Team 1	33	0	100.0%
	Team 2	1	42	97.7%
	Overall Percent	44.7%	55.3%	98.7%
Testing	Team 1	11	1	91.7%
	Team 2	0	12	100.0%
	Overall Percent	45.8%	54.2%	95.8%

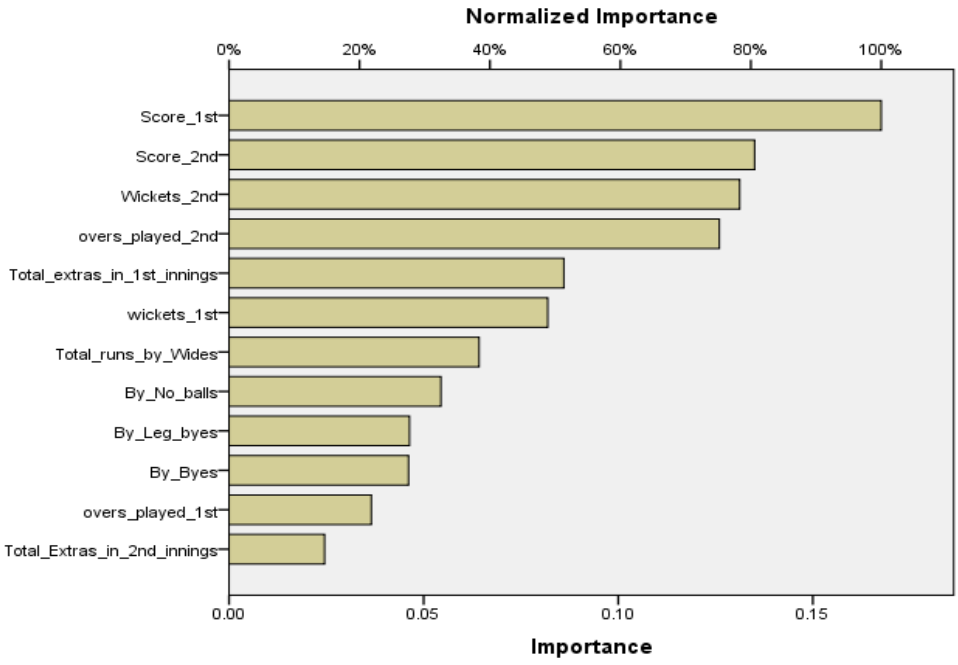
The Classification Summary provides insights into the predictive performance of your Artificial Neural Network (ANN) model for both the training and testing datasets. During training, the model accurately predicted Team 1 outcomes 100% of the time (33 correct predictions), while it correctly predicted Team 2 outcomes 97.7% of the time (42 correct out of 43). This results in an overall accuracy of 98.7% across the training dataset, with 44.7% of predictions corresponding to Team 1 and 55.3% to Team 2. In the testing phase, the model correctly predicted 91.7% of Team 1 outcomes (11 correct out of 12) and 100% of Team 2 outcomes (12 correct). The overall accuracy for the testing dataset was slightly lower at 95.8%, with 45.8% of the predictions for Team 1 and 54.2% for Team 2. This suggests that the model

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

is highly accurate, particularly in predicting Team 2 outcomes, with a small reduction in accuracy when applied to new data in the testing phase.

Table 4.4: Independent Variable Importance

	Importance	Normalized Importance
Score_1st	.168	100.0%
wickets_1st	.082	48.9%
overs_played_1st	.037	21.8%
Total_extras_in_1st_innings	.086	51.3%
Score_2nd	.135	80.6%
Wickets_2nd	.131	78.3%
overs_played_2nd	.126	75.2%
Total_Extras_in_2nd_innings	.025	14.7%
Total runs by Wides	.064	38.3%
By No balls	.054	32.5%
B _Byes	.046	27.5%
By Leg byes	.046	27.7%



Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

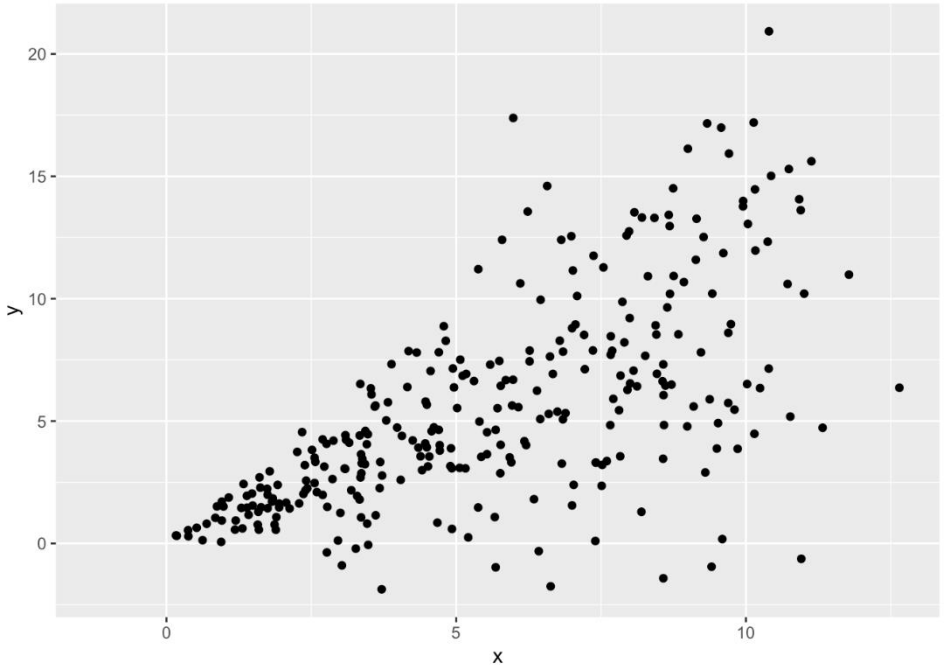


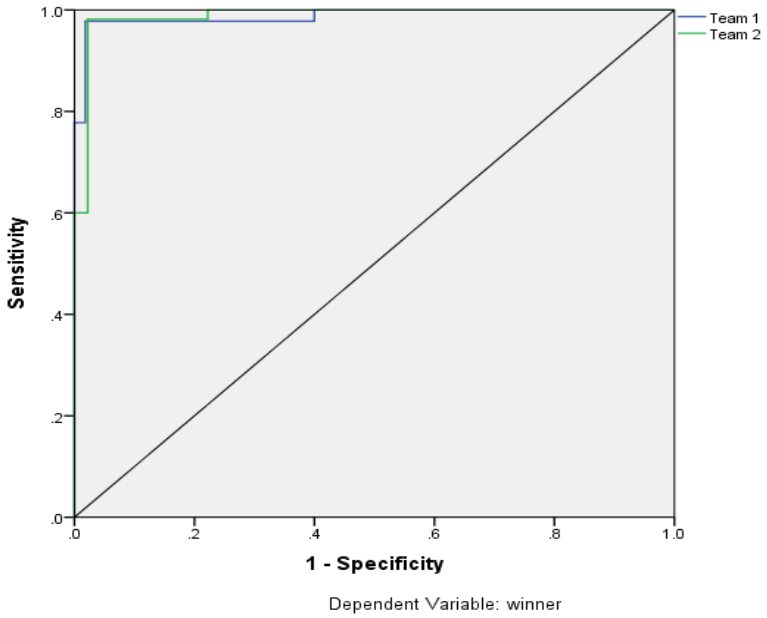
Figure 1 Residual plot of logistic regression shows heteroskedastic data

Pig 4.1: Independent Variable Importance

The Independent Variable Importance table provides insights into the relative contribution of each variable to the predictions made by your Artificial Neural Network (ANN) model. Among the variables, score_1st emerged as the most critical factor, with an importance score of 0.168, and is set as the baseline with a normalized importance of 100%. This means it plays a pivotal role in the model's predictions. Score_2nd and Wickets_2nd is also significant, with importance scores of 0.135 and 0.131, and normalized importances of 80.6% and 78.3%, respectively, indicating their strong influence on the outcome but slightly less than Score_1st.

Total_extras_in_1st_innings and overs_played_2nd follow with moderate importance (51.3% and 75.2% normalized), suggesting they also contribute meaningfully to the model's decisions, though to a lesser extent. On the lower end, variables like Total_Extras_in_2nd_innings and components such as _No balls, Byes, and _Leg byes have lower importance scores, indicating that while they do have some influence, their impact on the model's predictions is comparatively minor. This distribution of importance helps to understand which factors are driving the model's predictions and could guide further refinement of the model or focus areas for analysis.

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024



Pig 4.2: Sensitivity and Specificity

The ROC curve provided illustrates the performance of your Artificial Neural Network (ANN) model in predicting the winning team between Team 1 and Team 2. The curves for both teams (represented by blue and green lines) are close to the top left corner of the plot, indicating that the model has a high sensitivity (true positive rate) and a low false positive rate, which reflects strong discriminative power. The diagonal line represents a model that would classify outcomes randomly, and the fact that your model's curves are well above this line suggests that it significantly outperforms random guessing. The proximity of the ROC curves to the top left corner implies that the model is highly effective at correctly distinguishing between Team 1 and Team 2, demonstrating excellent predictive accuracy.

5. Conclusion

In conclusion, the Artificial Neural Network (ANN) model demonstrated superior performance in predicting cricket match outcomes, particularly when applied to heteroskedastic data, characterized by varying levels of variability across observations. By focusing on predicting outcomes based on the role of extras—such as wide balls, no balls, byes, and leg byes—in match results, the ANN model outperformed other models, including logistic regression, decision trees, random forest, and XGBoost.

The analysis revealed that extras such as wide's, no balls, byes, and leg byes have a negative effect on winning a match. Additionally, the number of wickets lost in the first innings also negatively impacts the probability of winning. Conversely, the number of overs played in the

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

first innings positively influences the match outcome, highlighting the strategic importance of preserving wickets and maximizing the use of available overs.

The ANN effectively captured these complex patterns in the data, leading to high accuracy during both the training and testing phases. Although there was a slight increase in error during testing, the model's overall predictive power remained robust, with minimal incorrect predictions and strong generalization to unseen data. The analysis of variable importance further reinforced the model's focus on critical aspects of the game, with the first innings score and second innings wickets also playing significant roles in determining match outcomes.

Additionally, the model's effectiveness was confirmed by the ROC curve, which indicated a strong ability to distinguish between winning and losing teams. While the ANN outperformed other models, the slight discrepancy between training and testing errors suggests that further refinement—such as additional tuning or incorporating more relevant variables—could enhance its generalization and predictive accuracy even further.

Overall, the application of ANN to heteroscedastic data, particularly in predicting match outcomes based on extras, underscores its effectiveness in capturing the nuanced dynamics of cricket matches, making it a valuable tool for sports analytics.

References

1. Irvine, S., & Kennedy, R. (2017). Analysis of performance indicators that most significantly affect International Twenty20 cricket. *International Journal of Performance Analysis in Sport*, *17*(3), 350359.
2. Moore, A., Turner, J. D., & Johnstone, A. J. (2012). A preliminary analysis of team performance in English firstclass TwentyTwenty (T20) cricket. *International Journal of Performance Analysis in Sport*, *12*(1), 188207.
3. Modekurti, D. P. V. (2020). Setting final target score in T20 cricket match by the team batting first. *Journal of Sports Analytics*, *6*(3), 205213.
4. https://www.sportsadda.com/cricket/features/whatareextrasincricket.
5. Anuraj, A., Boparai, G. S., Leung, C. K., Madill, E. W., Pandhi, D. A., Patel, A. D., & Vyas, R. K. (2023, March). Sports data mining for cricket match prediction. In *International Conference on Advanced Information Networking and Applications* (pp. 668680). Cham: Springer International Publishing.
6. Nimmagadda, A., Kalyan, N. V., Venkatesh, M., Teja, N. N. S., & Raju, C. G. (2018). Cricket score and winning prediction using data mining. *International Journal for Advance Research and Development*, *3*(3), 299302.

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

7. Astivia, O. L. O., & Zumbo, B. D. (2019). Heteroscedasticity in Multiple Regression Analysis: What it is, How to Detect it and How to Solve it with Applications in R and SPSS. *Practical Assessment, Research & Evaluation*, *24*(1), n1.
8. Wang, L., & Zhou, X. H. (2007). Assessing the adequacy of variance function in heteroscedastic regression models. *Biometrics*, *63*(4), 12181225.
9. J. S. Long and L. H. Ervin, "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model," Am. Stat., vol. 54, no. 3, pp. 217–224, Aug. 2000, doi: 10.1080/00031305.2000.10474549.
10. Midi, H., Rana, S., & Imon, A. H. M. R. (2009). The performance of robust weighted least squares in the presence of outliers and heteroscedastic errors. *WSEAS Transactions on Mathematics*, *8*(7), 351361.
11. Rahul, A. Gupta, A. Bansal, and K. Roy, "Solar energy prediction using decision tree regressor," Proc. 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021, no. Iciccs, pp. 489–495, 2021, doi: 10.1109/ICICCS51141.2021.9432322.
12. Karir, D., Ray, A., Bharati, A. K., Chaturvedi, U., Rai, R., & Khandelwal, M. (2022). Stability prediction of a natural and manmade slope using various machine learning algorithms. *Transportation Geotechnics*, *34*, 100745.
13. Klein, A. G., Gerhard, C., Büchner, R. D., Diestel, S., & SchermellehEngel, K. (2016). The detection of heteroscedasticity in regression models for psychological data. *Psychological Test and Assessment Modeling*, *58*(4), 567
14. Yigit, A. T., Samak, B., & Kaya, T. (2020). An XGBoost-lasso ensemble modeling approach to football player value assessment. *Journal of Intelligent & Fuzzy Systems*, *39*(5), 6303-6314.
15. Weeraddana, N. I. M. M. I., & Premaratne, S. A. M. I. N. D. A. (2021). Unique approach for cricket match outcome prediction using xgboost algorithms. *Journal of Theoretical and Applied Information Technology*, *99*(9), 2162-2173.
16. Husain, A., Salem, A., Jim, C., & Dimitoglou, G. (2019, December). Development of an efficient network intrusion detection model using extreme gradient boosting (XGBoost) on the UNSW-NB15 dataset. In *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (pp. 1-7). IEEE.
17. 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.
18. Chalitsios, C., Nikodelis, T., Panoutsakopoulos, V., Chassanidis, C., & Kollias, I. (2019). Classification of soccer and basketball players' jumping performance characteristics: a logistic regression approach. *Sports*, *7*(7), 163.
19. Prasetio, D. (2016, August). Predicting football match results with logistic regression. In *2016 International Conference On Advanced Informatics: Concepts, Theory And*

Power System Technology
ISSN:1000-3673
Received: 06-06-2024 Revised: 15-07-2024 Accepted: 28-08-2024

Application (ICAICTA) (pp. 1-5). IEEE.
20. Xu, S., Liang, L., & Ji, C. (2020). College public sports culture practice based on decision tree algorithm. *Personal and Ubiquitous Computing*, *24*, 207-221.
21. Kasera, M., & Johari, R. (2021). Prediction using machine learning in sports: a case study. In *Data Analytics and Management: Proceedings of ICDAM* (pp. 805-813). Springer Singapore.
22. Liu, J., & Carr, P. (2015). Detecting and tracking sports players with random forests and context-conditioned motion models. In *Computer Vision in Sports* (pp. 113-132). Cham: Springer International Publishing.
23. Weiwei, H. (2022). Classification of sport actions using principal component analysis and random forest based on three-dimensional data. *Displays*, *72*, 102135.
24. Gao, Z., & Kowalczyk, A. (2021). Random forest model identifies serve strength as a key predictor of tennis match outcome. *Journal of Sports Analytics*, *7*(4), 255-262.