STM JOURNALS

Journal of
Computer Technology & Applications

ISSN: 2229-6964 (Online)
ISSN: 2347-7229 (Print)
Volume 17, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoCTA

https://journals.stmjournals.com/jocta

Research JoCTA

Comparative Analysis of Supervised Learning Algorithms

Kalpesh U. Gundigara[1,*], Jeet S. Pandya[2]

Abstract

Supervised learning is a fundamental and widely used branch of machine learning in which models are trained on labeled datasets, meaning that each input is associated with a known output. Supervised learning algorithms develop predictive capability by understanding the mapping between input variables and corresponding output labels, enabling them to accurately forecast outcomes for previously unseen data. Due to this capability, supervised learning has found extensive applications across diverse domains such as image and speech recognition, natural language processing, fraud detection in financial systems, spam filtering, recommendation systems, and medical diagnosis, where reliable and interpretable predictions are essential This study provides an in-depth comparative evaluation of widely adopted supervised learning techniques, including logistic regression, decision trees, support vector machines (SVM), k-nearest neighbors (KNN), and random forests. Each of these algorithms has distinct characteristics, strengths, and limitations in terms of performance, interpretability, scalability, and computational complexity. To achieve an objective and thorough comparison, the algorithms are assessed using standard evaluation measures, including accuracy, precision, recall, F1-score, and training duration. These metrics help assess not only the correctness of predictions but also the efficiency and robustness of the models under different conditions. For experimental validation, a dataset consisting of 65 one-day international cricket match records was collected and analyzed. The dataset includes relevant features that influence match outcomes, enabling effective supervised learning-based classification. By applying and evaluating the selected algorithms on this real-world dataset, the study highlights how algorithm performance varies depending on data characteristics and problem context. The results of this comparative study aim to provide practical guidance to researchers, data scientists, and practitioners in selecting the most suitable supervised learning algorithm for their specific application requirements and constraints.

Keywords: Logistic regression, machine learning, random forest, supervised learning, support vector machine learning

INTRODUCTION

Machine learning, an important subfield of artificial intelligence, is concerned with building systems that can autonomously learn from data and enhance performance over time. Among these techniques, supervised learning is the most widely applied, which uses labeled data to train models to generate accurate predictions for given inputs. This study evaluated the performance of different supervised learning algorithms on structured datasets, highlighting their advantages, limitations, and suitable application scenarios.

***Author for Correspondence**
Kalpesh U. Gundigara
E-mail: kugundigara@gmail.com

[1]Assistant Professor, Department of Computer Science, Shri Swaminarayan College of Computer Science-Bhavnagar, Gujarat, India
[2]Teaching Assistant, Department of Computer Science, Shri Swaminarayan College of Computer Science-Bhavnagar, Gujarat, India

Received Date: December 16, 2025
Accepted Date: December 25, 2025
Published Date: February 20, 2026

Citation: Kalpesh U. Gundigara, Jeet S. Pandya. Comparative Analysis of Supervised Learning Algorithms. Journal of Computer Technology & Applications. 2026; 17(1): 25–30p.

RELATED WORK

Numerous studies have analyzed the effectiveness of various supervised learning methods. Logistic regression is often appreciated for its straightforward nature and interpretability.

Decision trees and random forests offer robust capabilities for modeling complex, nonlinear patterns, whereas support vector machines (SVMs) are especially well-suited for managing high-dimensional datasets. Comparative analyses of these algorithms can assist in identifying the most appropriate model for a given application by considering factors such as accuracy, computational efficiency, and interpretability.

Several researchers have conducted comparative studies of supervised learning algorithms, similar to the approach adopted in this study. Acharya and Ghevariya performed a comparative analysis of k-nearest neighbors (KNN), SVM, decision trees, and logistic regression using 20 Newsgroups and Wine datasets. Their study evaluated basic performance metrics, such as accuracy, precision, recall, and F1-score, and found that SVM and logistic regression often performed better on high-dimensional text data, while simpler structures achieved similar results on less complex datasets [1].

Other studies have focused on practical classification problems in various domains. For example, studies comparing logistic regression, random forest, SVM, and KNN for water quality classification have shown that ensemble methods, such as random forest, can perform competitively, with stacking techniques sometimes outperforming traditional classifiers [2]. Similarly, machine learning research on human resources datasets has found that random forest often achieves the best overall performance across accuracy, precision, recall, and F1 metrics, whereas SVM and logistic regression exhibit varying results depending on the underlying data characteristics [3].

In sports analytics, machine learning has been applied to cricket match data and player performance. A comparative study of T20 International cricket bowlers classified players using decision trees, Naïve Bayes, logistic regression, SVM, extreme gradient boosting, and random forests, illustrating the usefulness of classification models in sports contexts [4]. Another research article evaluated match outcome prediction for white-ball cricket (ODI/T20) by comparing models including random forest, SVM, KNN, Naïve Bayes, and XGBoost with standard metrics such as accuracy and F1-score [5].

Additionally, in a broader machine learning context, studies on exoplanet detection using Kepler data [6] and other classification tasks have demonstrated that ensemble methods such as random forest often achieve high accuracy and balanced precision-recall trade-offs compared to simpler models such as logistic regression and decision trees [7]. Such works reinforce the importance of empirical evaluation across different datasets and metrics when comparing supervised learning algorithms.

ALGORITHMS OF SUPERVISED LEARNING

Logistic Regression

Logistic regression is a supervised learning method that is widely applied in data science for classification problems, where it estimates discrete or categorical outcomes using a set of input variables. For instance, a logistic regression model can be applied to determine whether a loan application should be approved using variables such as savings balance, income level, and credit score. The model produces a binary result, which is commonly expressed as yes or no, 0 or 1, or true or false.

This method uses a logistic function, also known as the sigmoid, to compute the likelihood of an event. The sigmoid is an S-shaped curve that maps real-valued inputs to a range between 0 and 1. Through this transformation, logistic regression expresses the inputs as probabilities, which are then assigned to one of the two possible classes.

The sigmoid function, used to compute these probability values, is defined as: Equation of logistic regression.

$$f(x) = \frac{1}{1 + e-^{x}}$$

where,

- e = base of natural logarithms
- value = the numerical value one wishes to transform.

The following equation represents logistic regression: (logistic regression–sigmoid function).

$$y = \frac{e^{(b_0+b_1x)}}{1 + e^{(b_0+b_1x)}}$$

here,

- x = input value.
- y = predicted output.
- b_0 = bias or intercept term.
- b_1 = coefficient for input (x).

This formulation resembles linear regression in that the input variables are linearly combined using weights or coefficients to produce an output (Figures 1 and 2). However, in contrast to the linear regression, the predicted result in this case is binary (0 or 1) instead of a continuous numerical value [8].

Decision Trees

A decision tree is a tree-structured predictive model that generates decisions by applying threshold-based splits to the input features [9]. It is widely appreciated for its simplicity and interpretability, though it can be susceptible to overfitting if not properly regulated, it is widely appreciated for its simplicity and interpretability.

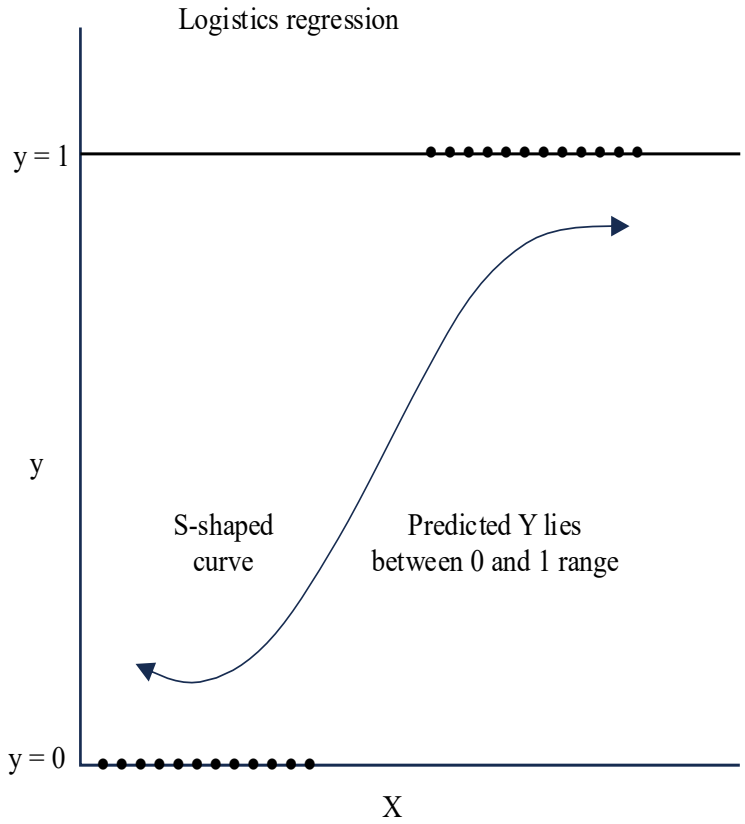

Figure 1. Logistics regression.

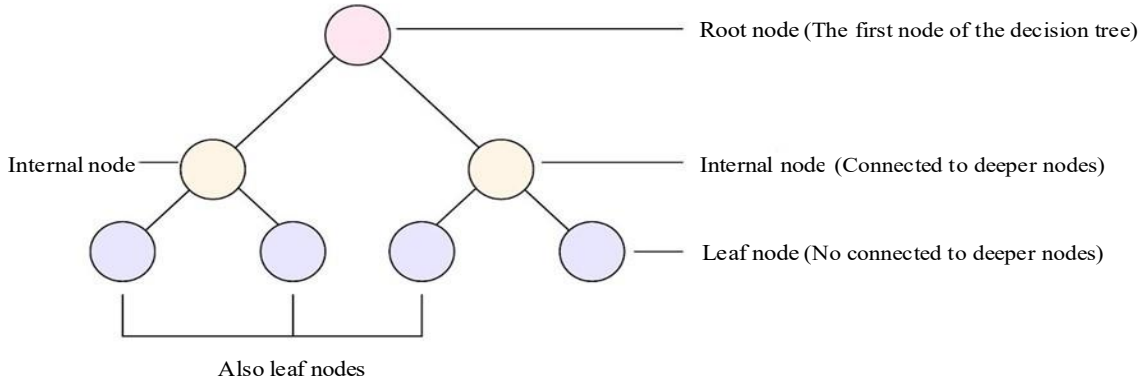

Figure 2. Decision tree.

As a supervised learning approach, decision trees are applicable to both classification and regression problems. In regression, it supports predictive modeling by estimating continuous outcomes, while in classification, it assigns data instances to predefined categories [10].

Structurally, the decision tree resembles a flowchart. It begins at the root node, where an initial question regarding the data is posed. Each possible answer leads to a branch that connects to subsequent decision (internal) nodes that present additional questions and further divisions of the data. This hierarchical process continues until the analysis reaches a terminal or "leaf" node, which provides the final predicted outcome.

Support Vector Machines

Support vector machines are supervised learning methods employed for both classification and regression purposes. They are widely used in various fields, including pattern recognition, image processing, and natural language processing [11]. The central idea behind SVMs is to identify the optimal hyperplane that can effectively separate the data points belonging to different classes.

A hyperplane acts as a decision boundary in high-dimensional feature space. In two-dimensional space, this boundary appears as a straight line, whereas in three-dimensional space, it is represented as a plane. In even higher dimensions, the hyperplane generalizes accordingly to separate classes [12].

The distance between a data point and a hyperplane is computed using the following expression:

$$\text{Distance} = \frac{w \cdot x + b}{\|w\|}$$

Where, w denotes the weight vector, b represents the bias term, and $\| w \|$ is the Euclidean norm of the weight vector. The vector w is perpendicular to the hyperplane and determines its orientation, whereas the bias b term shifts the hyperplane within the feature space.

K-Nearest Neighbors

The KNN algorithm is a non-parametric supervised learning approach that determines classifications or predictions by measuring the closeness of data points. It is commonly adopted because of its straightforward nature and effectiveness in handling both classification and regression problems.

In classification, KNN determines the label of a new instance by examining the class distribution among its k nearest neighbors. For example, if the five nearest neighbors of a data point include three belonging to Class A and two belonging to Class B, the model assigns the point to Class A based on majority voting.

In regression, the KNN generates predictions for continuous targets by computing the average value of the k-nearest neighbors. For instance, in a house-price prediction scenario, the algorithm estimates the price of a new property by averaging the prices of the k most similar houses.

Random Forest

Random forests is an ensemble-based learning approach that is commonly applied to classification, regression, and other prediction tasks. This technique works by constructing multiple decision trees during the training. In classification scenarios, the outcome is determined by majority voting across the trees, whereas in regression cases, the prediction is calculated as the average of the individual tree outputs.

Random forest models share most of their hyperparameters with the decision trees and bagging methods. However, it is unnecessary to manually combine a decision tree with a bagging classifier because the random forest algorithm inherently incorporates both concepts. Additionally, by using the corresponding regressor version of the algorithm, random forests can effectively handle regression-based problems.

DATASET AND EXPERIMENTAL SETUP

Dataset Used

A sample dataset for a one-day cricket tournament is collected. Here, over-wise run and fall of wickets data is collected. During the process, 80% of the data were used for training, and 20% of the data were used for testing.

Preprocessing

- *Data cleaning:* Duplicate, irrelevant, and noisy data were removed during data processing.
- *Train model using data*: To train the data model, approximately 65 international one-day match data were provided.
- *Feature normalization and encoding*: During this process, data rescaling, cleaning, and recycling were performed.

Tools, Algorithm, and Environment

- *Tools*: For practical implementation, various tools such as Python, Jupyter notebook, and Anaconda are used.
- *Algorithm*: Various algorithms, such as logistic regression, DecisionTreeClassifier, DecisionTreeRegressor, SVC, SVR RandomForestRegressor, etc., are implemented for the comparison of prediction results.
- *Library used*: Pandas, Seaborn, Matplotlib, train_test_split library was used.

EVALUATION METRICS

The evaluation metrics listed in Table 1 have been widely adopted to assess the classification effectiveness, reliability, and computational efficiency of AI.

RESULTS AND DISCUSSION

This section analyzes comparative algorithm performance, highlighting the accuracy, efficiency, and scalability differences across the machine learning models (Table 2).

Observations

- Random forest provides the highest accuracy and balanced performance.
- Logistic regression is not applicable to this type of dataset.

Table 1. Evaluation metrics.

Metric	Description
Accuracy	Ratio of correctly predicted instances
Precision	Proportion of true positives among predicted positives
Recall	Proportion of true positives among actual positives
F1-score	Harmonic mean of precision and recall
Training time	Time taken to train the model

Table 2. Comparative analysis of algorithms.

Algorithm	Accuracy	Precision	Recall	F1-score	Training time
Logistic regression	N.A.	N.A.	N.A.	N.A.	N.A.
Decision trees (regression)	99.13%	99.09%	99.00%	99.00%	Very low
SVM (regression)	98.52%	98.12%	98.10%	98.10%	Medium
KNN (regression)	99.19%	99.11%	99.10%	99.09	High
Random forest	99.58%	99.49%	99.42%	99.41%	High

- SVM performs well but requires tuning of parameters and is slower.
- KNN is simple and gives higher accuracy, but less scalable with large datasets.
- Decision trees are fast but may overfit without pruning.

CONCLUSION

This comparative analysis shows that no single algorithm is the best for all tasks. Random Forest generally provides the highest accuracy and robustness. Nonetheless, in scenarios where model transparency and computational efficiency are important, logistic regression or decision trees are more suitable choices. Algorithm selection should be guided by the nature of the dataset, required performance metrics, and computational constraints.

REFERENCES

1. Acharya BB. Comparative analysis of machine learning algorithms: KNN, SVM, decision tree and logistic regression for efficiency and performance. Int J Res Appl Sci Eng Technol. 2024;12(11):614–619. doi:10.22214/ijraset.2024.65138.
2. Sutanto T, Aditya MR, Budiman H, Noor Ridha MR, Syapotro U, Azijah N. Comparison of logistic regression, random forest, SVM, KNN algorithm for water quality classification based on contaminant parameters. J Data Sci. 2024. doi:10.61453/jods.v2023no48.
3. Silva H, Bernardino J. Machine learning algorithms: An experimental evaluation for decision support systems. Algorithms. 2022;15(4):130. doi:10.3390/a15040130.
4. Waqas M, Zaman Q, Mahsood F, Shahnaz A. A hybrid approach to T-20 cricket team selection: Combining probabilistic and machine learning techniques. Dialogue Soc Sci Rev. 2025;3(1):978–996.
5. Ul Mustafa R, Nawaz MS, Ullah Lali MI, Zia T, Mehmood W. Predicting the cricket match outcome using crowd opinions on social networks: A comparative study of machine learning methods. Malays J Comput Sci. 2017;30(1):63–76. doi:10.22452/mjcs.vol30no1.5.
6. Karimi R, Mousavi-Sadr M, Haghighi MH, Tabatabaei FS. Machine learning for exoplanet detection: A comparative analysis using Kepler data. [Preprint]. 2025. arXiv:2508.09689. doi:10.48550/arXiv.2508.09689
7. Wickramasinghe I. Applications of machine learning in cricket: A systematic review. Mach Learn Appl. 2022;10:100435. doi:10.1016/j.mlwa.2022.100435.
8. Elstak I, Salmon P, McLean S. Artificial intelligence applications in the football codes: A systematic review. J Sports Sci. 2024;42(13):1184–1199. doi:10.1080/02640414.2024.2383065.
9. Shah SAA, Zaman Q, Wasim D, Allohibi J, Alharbi AA, Shabbir M. Optimal model for predicting highest runs chase outcomes in T-20 international cricket using modern classification algorithms. Alex Eng J. 2025;114:588–598. doi:10.1016/j.aej.2024.11.113.
10. Priya S, Gupta AK, Dwivedi A, Prabhakar A. Analysis and winning prediction in T20 cricket using machine learning. 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India. 2022. p. 1–4. doi:10.1109/ICAECT54875.2022.9807929.
11. Jaeger S. The golden ratio in machine learning. 2021 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA. 2021. p. 1–7. doi:10.1109/AIPR52630.2021.9762080.
12. Rashid A, Biswas P, Nasim MD, Gupta KD. Power plays: Unleashing machine learning magic in smart grids. [Preprint]. 2024 Oct 20. arXiv:2410.15423. doi:10.48550/arXiv.2410.15423.

