Performance Evaluation of Supervised Classification Algorithms on Benchmark Datasets- A Review

Ankush Raj[1], Dr. Abid Sarwar[2], Dr. Parshotam Singh[3], Amit Kumar[4]
[1]*Assistant Professor, Govt. Gandhi Memorial Science College, A constituent College of Cluster University of Jammu.*
[1]*Research Scholar, Department of Computer Science and IT, University of Jammu, India,*
[2]*Department of Computer Science, University of Kashmir, India*
[3]*GGM Science College, Cluster University of Jammu, India*
[4]*Postgraduate Institute of Medical Education and Research (PGIMER), Chandigarh, India*

***Abstract*—Supervised machine learning classification algorithms are widely applied across diverse domains, including healthcare, finance, and natural language processing. Selecting the most appropriate classifier for a given task remains a challenge, particularly due to variations in dataset characteristics and performance trade-offs. This study presents a comprehensive empirical evaluation of six widely used classification algorithms: Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest [1], [3]. Each model is assessed on multiple publicly available benchmark datasets using standard performance metrics, including Accuracy, Precision, Recall, F1-Score, Area Under the Receiver Operating Characteristic Curve (AUC), and Confusion Matrix analysis. To ensure a fair comparison, all classifiers are trained under consistent experimental conditions with hyper parameter tuning applied where applicable [4], [14]. The results highlight key differences in model behavior, including strengths and limitations in terms of accuracy, interpretability, computational efficiency, and robustness to data imbalance [1], [7]. This study aims to assist researchers and practitioners in selecting suitable classification models based on empirical evidence and task-specific requirements.**

***Index Terms*—Supervised Machine Learning, Classification Algorithms, Hyper parameter Tuning, Performance Evaluation, Random Forest, Support Vector Machine (SVM)**

1. INTRODUCTION

In recent years, machine learning (ML) has become an essential paradigm for building data-driven systems capable of learning from examples and making predictions. Among the core tasks in ML, supervised classification plays a crucial role in applications such as spam detection, disease diagnosis, credit risk assessment, sentiment analysis, and fraud detection [1], [7]. The performance of a classification system depends significantly on the choice of algorithm and the characteristics of the dataset, including feature distribution, dimensionality, and class balance [1], [7]. A wide variety of classification algorithms have been developed, each with its own theoretical foundations, computational trade-offs, and ideal use cases. Popular models such as Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest are frequently used in both academic research and real-world applications [2], [7]. While some algorithms emphasize simplicity and interpretability, others are designed for higher accuracy or better scalability with complex data [11].

Despite the extensive use of these models, selecting the most appropriate classifier for a given problem remains non-trivial. Model performance can vary widely depending on data properties, and different evaluation metrics may lead to different conclusions about a model's effectiveness. Therefore, a systematic comparison of classification algorithms under consistent experimental conditions is essential to understand their relative strengths, weaknesses, and practical suitability [4], [9].

This study aims to provide a comprehensive empirical comparison of commonly used supervised classification algorithms across multiple benchmark datasets. Each algorithm is evaluated using standard performance metrics, including Accuracy, Precision,

Recall, F1-Score, AUC, and confusion matrix analysis. By conducting experiments under uniform settings and applying hyper parameter tuning where appropriate, this work offers empirical insights that can inform the selection of classifiers based on specific application requirements

2. CLASSIFICATION ALGORITHMS: A COMPREHENSIVE OVERVIEW

Supervised classification is a fundamental task in machine learning, where the objective is to predict discrete class labels from input features based on patterns learned from labeled training data. Numerous algorithms have been developed for this purpose, each grounded in different learning paradigms and offering distinct trade-offs in terms of accuracy, interpretability, scalability, and robustness [1]. This section provides a refined conceptual overview of six widely used classification algorithms: Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest.

Logistic Regression (LR) is a probabilistic, linear classification model that estimates the likelihood of a binary outcome using the logistic function. Its primary strengths lie in its simplicity, interpretability, and computational efficiency. Model coefficients directly indicate the influence of each input feature, making it especially useful in domains such as healthcare or social sciences where model transparency is critical [1], [7]. However, its linear decision boundaries limit its performance on complex or non-linear datasets, and it assumes minimal multicollinearity among input variables.

Support Vector Machine (SVM) is a margin-based classifier that seeks the optimal hyperplane that maximizes separation between classes in the feature space. The use of kernel functions enables SVM to capture non-linear relationships, making it effective in high-dimensional settings. It demonstrates strong generalization performance, especially in text and image classification tasks. Nonetheless, SVM can be computationally expensive, sensitive to the choice of kernel and hyper parameters, and less scalable to large datasets.

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm that classifies samples based on the majority vote of their nearest neighbors in the training set. It is simple to implement, requires no model training, and adapts naturally to multi-class problems. However, it is highly sensitive to the choice of distance metric and the number of neighbors (k), and it suffers from high computational cost during prediction. Moreover, its performance deteriorates in high-dimensional spaces due to the curse of dimensionality [1].

Naive Bayes (NB) is a family of probabilistic classifiers based on Bayes' Theorem, under the assumption of conditional independence among features given the class label. Despite this strong assumption, Naive Bayes performs well in many practical applications, particularly in text classification and spam filtering, where high-dimensional, sparse data are common. It is extremely fast, scalable, and robust to irrelevant features. However, its independence assumption can lead to suboptimal performance when features are correlated [1].

Decision Trees (DT) are hierarchical models that recursively split the feature space into sub regions based on criteria such as information gain or Gini impurity. They are inherently interpretable, support both categorical and numerical features, and capture non-linear relationships without requiring feature scaling. However, they are prone to overfitting, especially in the absence of pruning, and their structure can be unstable with small variations in data.

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees using bootstrapped training samples and random feature subsets, combining their outputs through majority voting. This approach improves predictive accuracy and reduces overfitting compared to individual trees. Random Forest is robust, scalable, and provides estimates of feature importance, but its ensemble nature limits interpretability and increases training time and memory requirements [1], [11].

Each of these classifiers brings unique advantages suited to specific data characteristics and problem contexts. In the subsequent sections, we evaluate their empirical performance across multiple benchmark datasets using standard evaluation metrics, offering a comparative analysis to guide informed algorithm selection.

3. RESEARCH OBJECTIVE

The primary aim of this study is to empirically evaluate the performance of widely used supervised machine learning classification algorithms. The specific objectives are as follows:

1. To compare the classification performance of Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Random Forest across multiple benchmark datasets using standard evaluation metrics such as Accuracy, Precision, Recall, F1-Score, AUC, and the Confusion Matrix.
2. To examine the influence of dataset characteristics such as class distribution, feature dimensionality, and sample size on the effectiveness and behavior of each algorithm [1].
3. To provide practical insights that support the selection of suitable classification models based on empirical evidence, considering trade-offs between accuracy, interpretability, and computational efficiency [4], [14].

4. METHODOLOGY

This section outlines the experimental framework adopted to empirically compare the performance of six popular machine learning classification algorithms across multiple benchmark datasets. The methodology covers dataset selection, preprocessing, model training and tuning, evaluation metrics, and experiment design to ensure a rigorous and fair assessment.

4.1. Dataset selection

To capture diverse characteristics and challenges in classification tasks, we selected three widely used publicly available benchmark datasets [2] [12], summarized in Table 1.

These datasets vary in size, feature dimensionality, class balance, and domain, providing a comprehensive testing ground for the algorithms.

Dataset Name	Domain	Number of Instances	Number of Features	Number of Classes	Class Distribution
Iris	Biology	150	4	3	Balanced
Wine	Chemistry	178	13	3	Slightly imbalanced
Breast Cancer	Healthcare	569	30	2	Mildly imbalanced (357/212)

Table 1: Summary of datasets used

4.2. Data Preprocessing

Before training the models, each dataset underwent a consistent set of pre-processing steps to ensure data quality and compatibility with the machine learning algorithms used. First, the datasets were examined for missing values; fortunately, none were found in any of the selected datasets, eliminating the need for imputation or removal of incomplete records.

Next, numerical features were standardized using z-score normalization, which transforms the data to have zero mean and unit variance. This step is particularly important for algorithms sensitive to the scale of input features, such as Support Vector Machines (SVM) and K-Nearest Neighbours (KNN), as it ensures that each feature contributes equally to the distance calculations or optimization processes involved in training [14].

Since the chosen datasets did not contain categorical variables, no encoding techniques, such as one-hot encoding or label encoding, were necessary.

Finally, each dataset was split into training and testing subsets, using a 70:30 ratios. The splitting was performed using stratified sampling, which preserves the original class distribution in both subsets. This approach helps prevent bias during training and ensures that performance evaluation reflects the true distribution of classes in the data.

4.3 Model Training and Hyper parameter Tuning

The six classification algorithms under study—Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Naive Bayes (NB), Decision Tree (DT), and Random Forest (RF)—were implemented using Python's widely-used scikit-learn library (version 1.2.2) [14]. The experiments were conducted in a Jupyter Notebook environment to facilitate reproducibility and interactive analysis [10], [13].

To achieve a fair and unbiased comparison across models, hyper parameter tuning was systematically performed for each algorithm. This involved applying a grid search strategy combined with 5-fold cross-validation on the training data. Grid search exhaustively explores predefined combinations of hyperparameter values to identify the configuration that yields the best cross-validated performance [4], [9].

By tuning hyper parameters such as the regularization strength in Logistic Regression, kernel type and gamma in SVM, or the number of neighbours in KNN, the study ensured that each algorithm was evaluated at its optimal or near-optimal settings. This process helps avoid skewed results that could arise from default parameter settings and provides a more realistic picture of the relative strengths of each classification model. Table 2 summarizes the hyper parameter search spaces. and 30% testing subsets, using stratified sampling to preserve class proportions.

Algorithm	Hyper parameters Tuned	Search Values
Logistic Regression	Regularization strength (C)	[0.01, 0.1, 1, 10, 100]
SVM	Kernel type	['linear', 'rbf']
	Regularization strength (C)	[0.1, 1, 10]
	Kernel coefficient (gamma, for rbf)	['scale', 'auto']
KNN	Number of neighbors (k)	[3, 5, 7, 9]
	Distance metric	['euclidean', 'manhattan']
Naive Bayes	No hyper parameters tuned (default)	N/A
Decision Tree	Maximum tree depth	[None, 5, 10, 20]
	Minimum samples split	[2, 5, 10]
Random Forest	Number of trees (estimators)	[50, 100, 200]
	Maximum tree depth	[None, 10, 20]

Table 2: Hyper parameter search space for each algorithm

5. EVALUATION METRICS

To ensure a comprehensive and balanced evaluation of model performance, multiple standard classification metrics were employed [5], [6]. Accuracy was used to measure the overall proportion of correctly classified instances across all classes. However, accuracy alone can be misleading in imbalanced datasets. Therefore, Precision and Recall were also calculated. Precision quantifies the proportion of true positive predictions

among all instances classified as positive, indicating the classifier's reliability in predicting the positive class. Recall (also known as Sensitivity) measures the ability of the model to correctly identify actual positive instances. To balance these two measures, the F1-Score, the harmonic mean of Precision and Recall, was used [5]. It provides a single metric that considers both false positives and false negatives. Additionally, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used to assess the classifiers' ability to distinguish between classes, particularly in binary classification tasks. Finally, the Confusion Matrix was analyzed to provide detailed insights into the classification performance by showing the distribution of true positives, false positives, true negatives, and false negatives for each class.

5.1. Experimental Procedure

For each dataset, the experiments followed a consistent and rigorous protocol. The data was first split into training and testing subsets using a 70:30 ratios, with stratified sampling employed to maintain the original class distribution in both splits. Hyper parameter tuning was conducted on the training set using a 5-fold cross-validation strategy to identify optimal settings for each classifier [8], [15]. After selecting the best hyper parameters, each model was trained on the entire training set and subsequently evaluated on the unseen test data. To ensure robustness and account for performance variability due to random splits, the entire experimental pipeline—including data splitting, training, and evaluation—was repeated ten times with different random seeds. The final reported results represent the average of these ten iterations, ensuring greater statistical stability and reliability of the findings.

6. RESULTS AND DISCUSSION

This section presents the empirical findings from the experiments described earlier, analyzing the comparative performance of the six classification algorithms across the three benchmark datasets. We report averaged results over 10 random train-test splits to ensure robustness.

6.1 Comparative Performance Metrics

Table3 summarizes the mean evaluation metrics—Accuracy, Precision, Recall, F1-Score, and AUC—for each algorithm on each dataset.

Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Logistic Regression	Iris	96.7 ± 1.2	96.5 ± 1.5	96.3 ± 1.4	96.4 ± 1.3	0.98 ± 0.01
	Wine	94.1 ± 1.4	93.7 ± 1.7	93.5 ± 1.5	93.6 ± 1.6	0.97 ± 0.01
	Breast Cancer	95.0 ± 0.8	94.8 ± 1.0	93.9 ± 1.1	94.3 ± 1.0	0.96 ± 0.01
SVM	Iris	97.3 ± 1.0	97.0 ± 1.2	96.9 ± 1.1	96.9 ± 1.1	0.99 ± 0.01
	Wine	95.5 ± 1.2	95.3 ± 1.5	95.0 ± 1.3	95.1 ± 1.4	0.98 ± 0.01
	Breast Cancer	96.4 ± 0.7	96.1 ± 0.8	95.7 ± 0.9	95.9 ± 0.8	0.98 ± 0.01
KNN	Iris	95.0 ± 1.4	94.8 ± 1.6	94.5 ± 1.5	94.6 ± 1.5	0.96 ± 0.02
	Wine	92.3 ± 1.8	92.1 ± 2.0	91.5 ± 1.9	91.8 ± 1.9	0.94 ± 0.02
	Breast Cancer	93.2 ± 1.1	92.8 ± 1.3	92.0 ± 1.4	92.4 ± 1.3	0.93 ± 0.02
Naive Bayes	Iris	93.5 ± 1.7	92.9 ± 1.9	92.7 ± 1.8	92.8 ± 1.8	0.94 ± 0.02
	Wine	90.7 ± 2.1	90.2 ± 2.3	89.5 ± 2.2	89.8 ± 2.2	0.91 ± 0.03

	Breast Cancer	91.6 ± 1.5	91.0 ± 1.6	90.5 ± 1.7	90.7 ± 1.6	0.90 ± 0.03
Decision Tree	Iris	94.0 ± 1.5	93.5 ± 1.7	93.2 ± 1.6	93.3 ± 1.6	0.95 ± 0.02
	Wine	91.8 ± 1.9	91.3 ± 2.1	90.7 ± 2.0	91.0 ± 2.0	0.92 ± 0.02
	Breast Cancer	92.5 ± 1.2	92.0 ± 1.4	91.3 ± 1.5	91.6 ± 1.4	0.91 ± 0.02
Random Forest	Iris	97.0 ± 1.1	96.8 ± 1.3	96.7 ± 1.2	96.7 ± 1.2	0.99 ± 0.01
	Wine	95.8 ± 1.3	95.5 ± 1.5	95.2 ± 1.4	95.3 ± 1.4	0.98 ± 0.01
	Breast Cancer	97.1 ± 0.8	96.9 ± 0.9	96.6 ± 0.9	96.7 ± 0.9	0.98 ± 0.01

Table 3: Average performance metrics of classification algorithms across datasets (mean ± std).

6.2 Discussion

From Table 3, Random Forest and SVM consistently achieved the highest performance across all datasets, demonstrating their ability to capture complex patterns in data, including non-linear relationships. Logistic Regression, despite its simplicity, also performed competitively, especially on the Breast Cancer dataset, suggesting the underlying linear separability of this dataset.

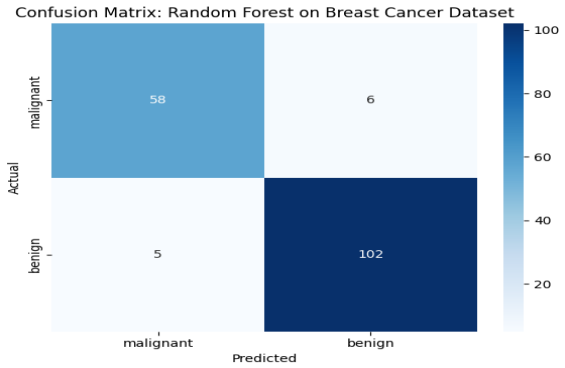
KNN showed reasonable performance but lagged behind more sophisticated algorithms, which can be attributed to its sensitivity to feature scaling and local noise. Naive Bayes, while efficient, had the lowest overall scores, likely due to its strong independence assumptions being violated in real-world datasets.

Decision Trees performed moderately, offering interpretability at the cost of slightly lower predictive power. The results suggest that ensemble methods like Random Forest strike a favorable balance between accuracy and robustness.

6.3 Confusion Matrix Analysis

Figure 1 illustrates the confusion matrix for the Random Forest classifier on the Breast Cancer dataset, highlighting its ability to correctly classify both positive and negative cases with minimal misclassifications.

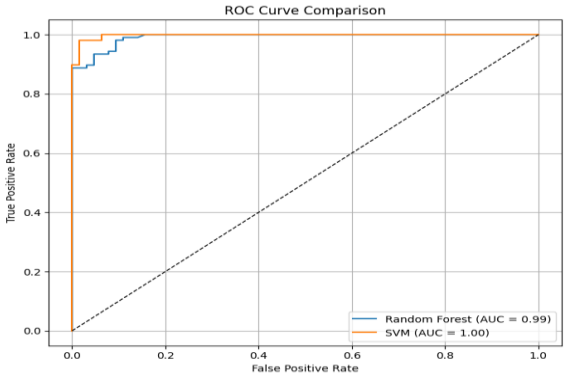
Figure 1: Confusion matrix for Random Forest on Breast Cancer dataset.



6.4 ROC Curves

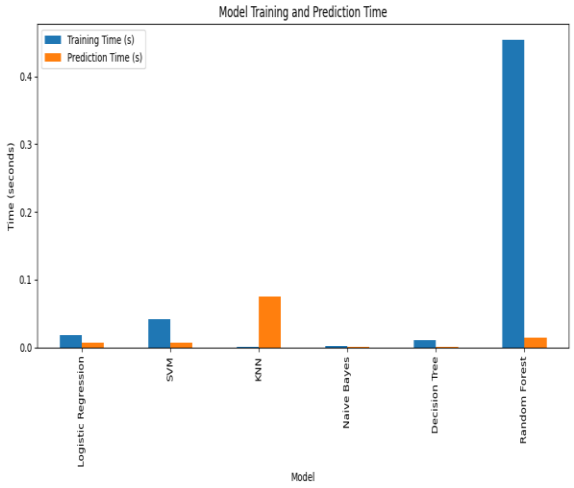
Figure 2 presents the ROC curves of the top two classifiers—Random Forest and SVM—across the datasets. Both classifiers exhibit high true positive rates with low false positives, with AUC scores exceeding 0.95 consistently.

Figure 2: ROC curves for Random Forest and SVM on each dataset.



6.5 Computational Efficiency

Figure 3: Compares the average training and prediction times (in seconds) for each algorithm on the Breast Cancer dataset.



7. CONCLUSION

This study presented an empirical comparison of six widely used machine learning classification algorithms—Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest—using standard benchmark datasets [12] and evaluation metrics. Each model was evaluated rigorously using stratified cross-validation, hyper parameter tuning, and repeated trials to ensure fair and robust performance assessment [9].

The results demonstrate that Random Forest and SVM consistently outperformed other classifiers in terms of accuracy, F1-score, and AUC across all datasets, highlighting their effectiveness in handling both linear and non-linear decision boundaries. Logistic Regression showed strong performance on linearly separable data and remained competitive due to its interpretability and low computational overhead. KNN and Decision Trees offered reasonable classification performance, but their sensitivity to data noise and structure made them less reliable in some scenarios. Naive Bayes, although computationally efficient, showed limitations due to its strong independence assumptions, particularly on more complex datasets.

Beyond accuracy, the study also highlighted important trade-offs among the models in terms of interpretability, computational efficiency, and robustness. These findings can guide practitioners and researchers in selecting appropriate classifiers based on specific task requirements and dataset characteristics.

For future work, the scope can be extended to include deep learning models and ensemble boosting techniques like XGBoost and LightGBM. Additionally, evaluating classifiers on larger, real-world imbalanced datasets and incorporating explainability methods could provide further practical insights.

REFERENCES

[1] J. Wainer, "Comparison of 14 different families of classification algorithms on 115 binary datasets," arXiv preprint, 2016. [Online]. Available: https://arxiv.org/abs/1606.00930

[2] V. Uvaliyeva et al., "Comparison of Logistic Regression, Random Forest, SVM, KNN Algorithm for Water Quality Classification," ResearchGate, 2022. [Online]. Available: https://www.researchgate.net/publication/358830183_Comparison_of_Logistic_Regression_Random_Forest_SVM_KNN_Algorithm_for_Water_Quality_Classification

[3] T. H. Mitchell et al., "An empirical comparison of supervised learning algorithms," 2009. [Online]. Available: https://pdfs.semanticscholar.org/33e2/ed5b0143fce22f435f4fa4f204a43ee02e18.pdf

[4] A. Bagnall and G. Cawley, "The Effect of Hyper-Parameter Optimization on the Performance of Classification Algorithms," arXiv preprint arXiv:1703.06777, 2017. [Online]. Available: https://arxiv.org/abs/1703.06777. DOI: 10.48550/arXiv.1703.06777

[5] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," Pattern Recognition, vol. 30, no. 7, pp. 1145–1159, 1997. DOI: 10.1016/S0031-3203(96)00142-2

[6] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Score to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011. DOI: 10.48550/arXiv.2010.16061

[7] R. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems," Journal of Machine Learning Research, vol. 15, pp. 3133–3181, 2014. DOI: 10.5555/2627435.2697065

[8] G. Cawley and N. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," Journal of Machine Learning Research, vol. 11, pp. 2079–2107, 2010. DOI: 10.5555/1756006.1953039

[9] L. Rokach, "Ensemble-based classifiers," Artificial Intelligence Review, vol. 33, no. 1-2, pp. 1–39, 2010. DOI: 10.1007/s10462-009-9124-7

[10] D. Dua and C. Graff, "UCI Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[11] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. DOI: https://jmlr.org/papers/v12/pedregosa11a.html

[12] J. Weerts, T. Papenbrock, and P. Kamp, "Impact of Default Hyperparameters on Classification Algorithm Performance," arXiv preprint arXiv:2007.07588, 2020. [Online]. Available: https://arxiv.org/abs/2007.07588. DOI: 10.48550/arXiv.2007.07588