***Naivigyan Journal of Multidisciplinary Research* (NJMR)**
Volume 2, Issue 2, April 2026, Pages: 69-76
Original Article | ISSN (O): 3108-1487

Comparative Analysis of Various Machine Learning Models – A Case Study for the Classification of Flowers

Soni P M[1]*, Seema M.A[2] and **Sini Francis[3]**

[1]Associate Professor, Naipunnya Institute of Management and Information Technology, Pongam, Thrissur- 680308, Kerala, India
[2]Assistant Professor, Presentation College of Applied Sciences, Puthenvelikara, Kerala, India
[3]Assistant Professor, Presentation College of Applied Sciences, Puthenvelikara, Kerala, India
*Corresponding Author's Email: sonipm@naipunnya.ac.in

Abstract

In many areas, machine learning plays a key role in how we handle data, helping sort things into groups, spot trends, or forecast outcomes. The Iris flower collection is introduced by Ronald Fisher in 1936. This one pack 150 records covering three types of Irises, such as setosa, versicolor and virginica, and each tagged with four clear measurements as length and width of sepal and petal dimensions in the same way. Those numbers are what models learn from when doing grouping jobs. Because it's clean, evenly split, and the groups stand apart clearly, folks lean on this set to test how well different learning methods actually work. To sort data, six common methods were tested- Decision Trees, k-Nearest Neighbours, Logistic Regression, Support Vector Machines, Random Forests, and Neural Networks. In the Iris dataset, every method was trained and then checked for how well it worked. Performance came down to four things: accuracy, precision, recall, and how fast each one ran. Even though all did well on getting answers perfect, some fit certain tasks better than others. Where SVM and Random Forests scored high in correct predictions, they weren't always the clearest to follow, and on the flip side, Decision Trees and k-NN made decisions easier to trace, simpler to grasp, but in the case of slower learners, Logistic Regression kept pace quickly, stayed straightforward despite trailing a bit in exactness. Model selection means looking past just how accurate it is. After evaluating all the methods, we can say no one method wins everywhere. Depending on the situation, strengths show up in different places.

Keywords: KNN, ANN , SVM , Logistic Regression , Random Forest

Introduction

Nowadays, machines can pick up patterns just by looking at information, no step-by-step coding needed. So, sorting items into groups has become a basic task in smart software. Back in 1936, Ronald Fisher shared some flower measurements that are still used a lot today. That holds exactly 150 examples- no more, no less- split evenly across three types of irises. Each plant comes with four members attached – two for sepals and two for petals. Even now, researchers reach for this set when trying out new ways to classify things. One reason researchers keep coming back to this data? It plays nicely with lots of model types- from Decision Trees up through Neural Nets. When using trees, clarity meets function- they lay out decisions where you can see them. On the flip side, multiple trees together in a Random Forest tighten results and avoid getting too cosy with noise. When boundaries are set to classify the data without any hesitation. Then there's k-NN – simple guts, works decently when data stays compact (Singh & Singhal,2024). Funny thing, though – nailing accuracy isn't the whole story here. What floats one user's boat might sink another's, and whether someone can actually follow what it did depends on how fast it runs.

Even if Logistic Regression works fast, it struggles if groups overlap too much. Instead of staying simple, Neural Nets learn tricky patterns- though they might be overkill on tidy data such as Iris, seeing as basic methods do just fine (Academia.edu, 2023). From this, we can understand that tool choice should fit what the data looks like and what you actually need. This isn't just sorting flowers- it shows core ideas behind teaching machines to learn. Because one model may guess right more often, another might explain its reasoning better. When these methods move into hospitals or farms, grasping such differences becomes critical. Even though the Iris dataset was made long ago, this collection still helps people practice and probe new techniques in a safe and clear way. After looking at something closely, something stands clear: one-size-fits-all doesn't work when picking classifiers. Depending on what needs doing, different models rise or fall in usefulness. A careful look shows that fit matters more than fame. Where a method lands depends heavily on the setting, purpose, and even data shape. Choices gain strength when tied tightly to real conditions instead of trends. The structure of the paper is as follows. The literature review, materials and methods, results and discussion, conclusion, and future scope are described in the next section.

One reason for coming back to the Iris data is that it offers a clean way to test how well machine learning methods handle sorting tasks. Back in 1986, Quinlan demonstrated that decision trees could make sense of patterns plainly, thanks to their straightforward structure. Still, others noticed these trees sometimes latch onto noise unless trimmed carefully. Even now, particularly when working with limited data, one thing about k-Nearest Neighbors (k-NN) is how straightforward it feels. Back in 1967, Cover and Hart introduced the idea, planting a seed that still grows in basic classification tasks. On the Iris dataset, results often line up well with more complex models- accuracy holds steady under the right conditions. Yet here's the catch: change the scale of features or pick a different k value, then outcomes shift, sometimes sharply (Zhang, 2016). Starting off strong, Cortes and Vapnik introduced Support Vector Machines (SVM) back in 1995- a real game changer for sorting data into categories. Using SVM on the classic Iris flower set often leads to better results compared to older methods, especially when lines between groups get twisty rather than straight (Hus, Chang, & Lin, 2010).

One forest of many trees took shape back in 2001, thanks to Breiman's design that moved beyond lone decision paths. Rather than relying on just one tree, this method builds many- each slightly different. Even though each stands alone at first, they grow together into a stronger collective. Results since then show it handles data more consistently than basic models. Accuracy rises when predictions come from crowds of trees rather than solo splits. Liaw and Wiener pointed this out early, noting how well the group performs across varied tests.(Liaw, A., & Wiener, M. 2002). Inspite of being unable to capture complex curves, Logistic Regression remains popular because it runs quickly and results are straightforward to grasp (Hosmer, Lemeshow & Sturdivant, 2013). In recent times, researchers have turned attention toward Neural Networks- these deliver high accuracy although experts debate whether such complexity adds value when working with compact data such as Iris, seeing that basic methods perform just as well (Goodfellow, Bengio, & Courville, 2016). It is prominent when looking through past studies – SVM, along with forests, often hit higher marks in getting predictions right. However, Decision Trees still hold ground because people can follow how they reach decisions. Accuracy matters a lot, yet so does how fast a model runs, plus whether someone can make sense of its logic. What many seem to agree on comes down to weighing these three things together.

Materials and Methods

The data consist of one hundred fifty Iris flower examples, which are spread equally across three types, each described by four number- based traits. Every trait was scaled to match ranges for fair testing, avoiding size differences that might skew results. The group balance was maintained during the subsequent split, with seventy percentage going towards training and thirty percent being used for performance evaluation. Six different learners were involved instead of just one method: Decision Tree made choices using rules; neighbours voted in k- NN. Logistic Regression drew lines through data points while SVM pushed boundaries wider. Random Forest developed many trees to reduce errors, whereas an ANN imitates brain-like connections across layers. When tuning was important, the best settings came from trying combinations in structured sweeps. Each model developed its own methods for sorting flowers without copying the logic of others. Figures like accuracy, precision, recall, F1- score, along with confusion matrices, helped check how well the model worked. To make results more reliable and reduce bias, a ten-fold cross- validation approach was applied. Python is used as the main tool for running every test. Scikit-learn handles standard models, whereas neural networks are handled by TensorFlow and Keras instead. Consistency was maintained throughout each step taken. A visual outline of this process appears in Figure 1.

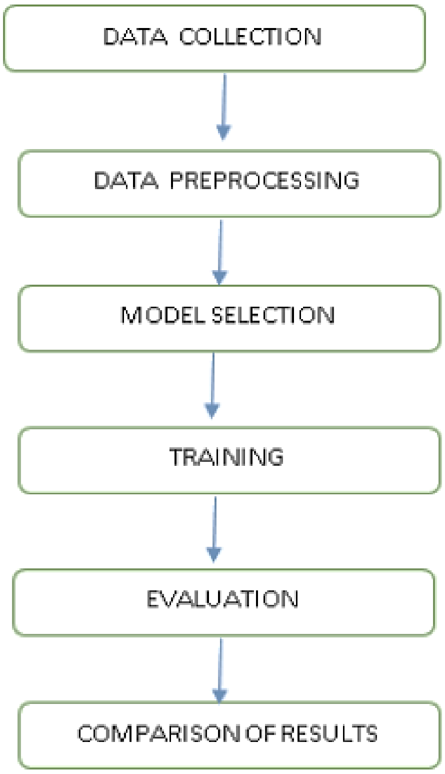

Fig.1 Visual Outline of the Process

Dataset

Ronald Fisher published this research back in 1936, using the Iris data collection. He examined 150 flowers, fifty from each of three types: setosa, versicolor, and virginica. For every flower, he measured and recorded the sepal length and width, followed by the petal dimensions length and width. Regardless of its small size, its structure allows many model trails without heavy computing needs. Because the groups are evenly spread and distinct, t has become a reliable and favourite example in algorithm training. Because of its simplicity, few datasets helped beginners to match its clean layout for early- stage practice.

Pre- processing

With the data already well-ordered, little pre-processing work was required. However, some adjustments were needed to ensure the fairness between models like cleaning gaps here, adjusting scales there, and lining up labels too.

- Feature Scaling: With standardisation, feature scaling adjusts every value to fit one range – important because methods such as k- Nearest Neighbors (k- NN) or Support Vector Machines (SVM) rely on distances between points.
- Train -Test Split**:** Splitting the data came first – seventy percent went to training, thirty to testing. A different method kept species ratios intact across both parts. Stratified sampling handled that balance without extra steps. Each group ended up reflecting the original mix on its own.
- Cross-validation**:** Averaging results across ten different splits helped check how steady the model performs. Splitting data this way cuts down on skewed outcomes by testing more ways than just once.

Models

A mix of six machine learning setups came into play, each showing a different way to sort things out.

- Decision Tree Classifier: It helped to check how clear results appear when mapped out. Fine – tuned settings like maximum depth, along with minimum samples per leaf, to reduce overfitting risks. During trials, model transparency, along with Visualization strength, was evaluated. Overfitting stayed low because adjustments balanced complexity carefully behind the scenes.
- k-Nearest Neighbors (k-NN)**:** When k-NN was tested with neighbors set at three, five, or seven, performance changed. Distance between points got measured using straight- line geometry. The number of nearby examples that were included determines how close predictions remained.
- Logistic Regression: Logistic regression was used first as a basic linear approach. Regularisation of type L2 helped it perform better on unseen data. The aim stayed clear by keeping things stable and avoiding overfitting. Each step followed this quiet rule. The settings- like how strict the boundaries should be got fine-tuned by testing many combos one after another. Each adjustment aimed at better separation without overdoing it.
- Random Forest Classifier**:** A group of decision trees works together in this model to help it stay stable across different data. Instead of one tree, many split the information, with each adding a vote. We changed the depth and number of each tree to get the ideal speed and accuracy. Too few trees or minor divisions lose detail, and many others slow things down. Performance goes up first, then levels off as more trees join. Each run modifies these settings slightly, aiming for steady results without wasting resources.

- Artificial Neural Network (ANN): One hidden layer sits inside this artificial neural network. When proceeding, data moves through ReLU switches that turn on or off based on input. Instead of working with full data sets, training uses small, random batches – each tweak adjusts weights slightly. This setup learns patterns without looping back, just straight ahead. The method keeps adjusting its predictions step by step until the results settle close enough.

Evaluation Metrics

The following metrics are used to evaluate the performance:

- Accuracy: Proportion of correctly classified instances.
- Precision, Recall, and the F1 -score are calculated individually for each class, giving a better understanding of how mistakes tend to gather across various categories.
- Confusion Matrix: Examining the mistakes that were made when classifying species, a chart showed exactly where errors happened. Instead of just numbers, colours highlighted which ones got confused most often. Errors appeared more often between the same types of groups than others. This view made it easier to spot patterns nobody had noticed before. A bigger story about overlap in traits was revealed by each misclassification.
- Cross–Validation Scores: By checking on multiple splits of data, it shows the consistency of the results. Each piece gets its turn, revealing patterns that hold or shift. Splitting things differently each time keeps assumptions in check. Results wander? That shows up here first. Stability becomes clear through repetition. One follows another, building a fuller picture slowly.

Tools and Environment

Every test was run on Python 3.10. Older methods took shape there in place of scikit-learn, while Tensor Flow paired with Keras brought the brain- like network to life. With just 16 GB of memory and an Intel i7 chip under the hood, all computations were completed by a standard desktop; no heavy lifting was required because the data remained small.

Results and Discussions

Not one model struggled when tested on the Iris data. Each algorithm's success was probably aided by neat grouping and even distribution. Performance stayed strong throughout, regardless of how the flowers were sorted.

Comparative Results of Machine Learning Models

The comparison of accuracy obtained by various machine learning models is listed in Table 1 below. The table also explained the strengths and weaknesses of each model.

Table 1: Comparative Analysis of Various Models

Model	**Accuracy (%)**	**Strengths**	**Weaknesses**
Decision Tree	95	Easy to interpret, visualizable	Risk of overfitting
k-Nearest Neighbors	96	Simple, effective for small data	Sensitive to scaling, choice of *k*
Logistic Regression	94	Fast, interpretable	Struggles with non-linear boundaries

Support Vector Machine	98	High accuracy, handles non-linear	Requires tuning, less interpretable
Random Forest	98	Robust, reduces overfitting	Less transparent than single tree
Neural Network	97	Models complex patterns	Computationally heavier, overkill

Errors mostly popped up when telling apart Iris versicolor from Iris virginica – those two share too many similar traits. On the flip side, Iris setosa hardly ever got mixed up, standing out clearly through its unique measurements.

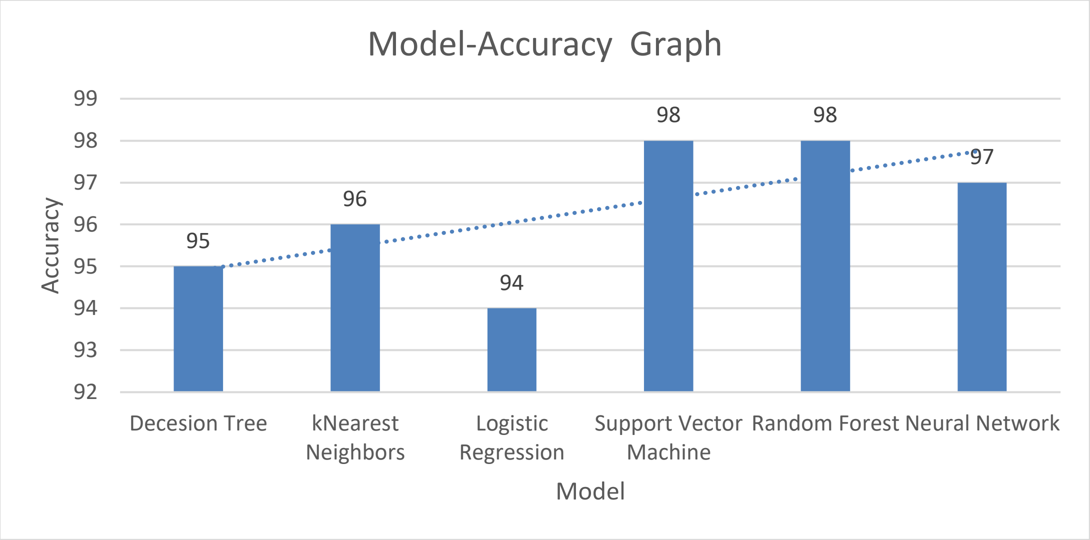

Fig.2 Model Accuracy Graph

SVM and Random Forest achieved peak accuracy by demonstrating that they handle complex patterns well. SVM reshapes data into extra layers, which makes tangled groups easier to split apart – not just because of its raw power. Meanwhile, Random Forests skip overfitting traps by blending results from many branching choices. Their edge comes from structure, not luck.

Decision Trees stand out by clearly showing every step, even though they are not quite as precise. Their straightforward logic fits well in classrooms or when answers must be explained easily. But with k-NN**,** it handled small data smoothly because similar points grouped naturally. However, if settings or scales shift, it behaves quite differently. That vulnerability holds it back outside controlled cases.

Quick to move, Logistic Regression make a mistake when patterns are twisted too much, sometimes guessing wrong. Neural networks finally got the details right after a lot of work, but all that power felt like cracking a nut with a sledgehammer.

The truth is that one size never fits every problem here. Picking a model relates to what the data looks like and what the task really needs. Tiny, tidy sets? In most cases, a simple approach suffices. Only when precision is crucial or when the data becomes disorganised and enormous do fancy methods make sense.

A look at the Iris data still reveals how choices in modelling come with compromises, showing what each method can handle alongside where it falls short. Although helpful for instruction, its simplicity also suggests limitations in the real world when using algorithms outside of textbook examples.

Conclusion

This comparison of machine learning techniques for sorting flowers by type reveals that each approach is effective, but the selection process relies on balancing correct response with speed and clarity. High scores often come from Support Vector Machines or groups of Decision Trees, especially if getting it correct matters most. However, these require careful adjustments; some of the steps taken along the way are also hidden by the tree clusters. Yet Decision Trees and k- NN provide strong results while remaining easy to set up. This clarity is helpful in classrooms or when understanding how choices are made is important. On the flip side, Logistic Regression works quickly and simply but falters once data twists in complex ways. Even so, Neural Networks demand more computing power and often miss the point with tiny sets such as Iris, where basic methods do just fine. One thing is clear after examining the results. No single method succeeds every time. Depending on the data, some models perform better than others. What matters most boils down to what kind of information you have, how much computing power is within reach, and exactly what question needs answering. Even today, researchers use the Iris dataset to test how well different methods sort things into groups.

Though this work looks at how different models handle the Iris data, next steps might involve messier, real- life conditions. High-resolution photos of actual flowers, paired with CNNs, could uncover patterns harder to spot before. Instead of choosing one method, mixing ensembles with neural nets might hold better results. Let the system adjust its own settings, reshape input in clever ways, and pull in features it already knows. When tested out in fields or labs, such tools can show what they truly do under pressure.

References

Academia.edu. . (2023). *Comparison on Iris dataset using classification techniques*. Retrieved from https://www.academia.edu

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. Wiley.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2010). A practical guide to support vector classification. *National Taiwan University*.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18–22.

Mondal, S., Roy, A., Akram, S. W., Zakir, S. M., Biswas, A., Bhattacharjee, K., & Das, A. (2024). Comparative study on classification techniques through IRIS data analysis. *International Journal of Recent Trends in Multidisciplinary Research, 4*(2), 82–87.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Singh, A. G., & Singhal, M. K. (2024). A survey on prediction models for the Iris dataset: Comparative analysis of machine learning and CNN approaches. *International Journal of Research in Computer Science*.

Zhang, H. (2016). The optimality of naive Bayes. *AAAI Conference on Artificial Intelligence*.