Comparative Study of Classification Algorithms on Breast Cancer Wisconsin Data Set

MK Jayanth Shanmugam
Computer Science & Engineering
Manipal Institute of Technology

Ashwin Gupta
Computer Science & Engineering
Manipal Institute of Technology

ABSTRACT
This project aims to compare the performance of four classification algorithms in machine learning: Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors. The performance metrics we have considered are the algorithm's accuracy after fitting the data, and the time each algorithm takes to classify a test data set.

KEYWORDS
Comparative, Classification, Logistic Regression, SVM, Decision Tree, KNN, Accuracy, and Inference Time.

1. INTRODUCTION
Classification algorithms in machine learning are a group of supervised learning algorithms wherein the algorithm is trained with labeled data and is tasked with predicting the class or label of new data. The accuracy and inference time of the algorithm depends on many factors, including the quality of the data set, the number of epochs, and the choice of hyperparameter. Some of the main applications of classification algorithms include Email spam detection, Speech recognition, Classification of Tumor cells, and Biometric identification.

Supervised algorithms have a wide range of applications and are one of the most widely used techniques in machine learning. With machine learning being a sought-after skill in the industry and the increase in demand for machine learning practitioners, it is becoming increasingly important to have at least a cursory understanding of basic machine learning algorithms.

Our motivation behind this project is to study the workings of some of the most widely used classification algorithms, understand the methodology used for classification, and compare the performance of these algorithms. By doing so, we will get a basic understanding of machine learning which can be used as a platform to build upon our knowledge of machine learning and explore more avenues in it.

2. LITERATURE REVIEW
2.1 Logistic Regression
Logistic regression is a popular binary classification algorithm used in machine learning. Given an input, the algorithm gives an output of zero or one, indicating the class or label the input belongs. This algorithm assumes a linear relationship between the input features and tries to find the parameters that best fit the training data set. The linear sum of the inputs is then passed to a Sigmoid function to give the output.

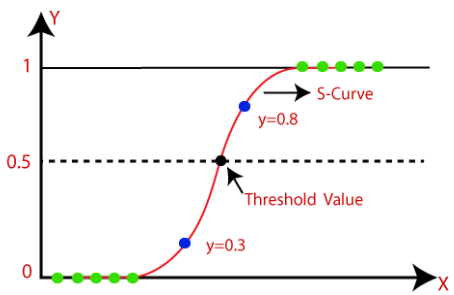


Fig. 1. Logistic Regression Visual

2.2 Support Vector Machines
Support Vector Machines (SVM) work on the notion of a margin. Essentially, the further away a data point is from a margin, the more confident we are about its class. The advantage of SVMs is that they can be used for very high dimensional data, and they also work for data that is linearly non-separable.

The goal of the SVM is to find the best decision boundary that separates the data into distinct classes. SVM chooses extreme vectors called support vectors to create the separating hyperplane.

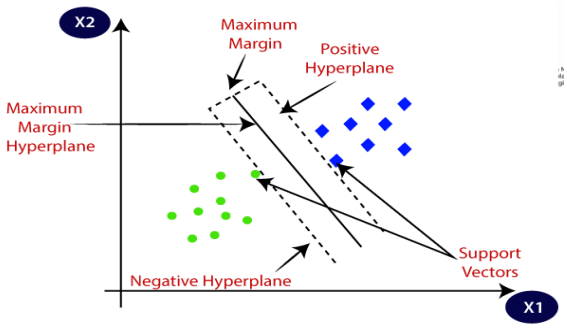


Fig. 2. SVM Visual

2.3 Decision Tree
The decision tree algorithm is a versatile algorithm that can be used for regression and classification tasks. A decision tree consists of two types of nodes: a Decision Node used to make decisions with multiple branches and Leaf Nodes representing the output of those decisions. It is a graphical representation of all possible solutions to a problem/decision based on given conditions.

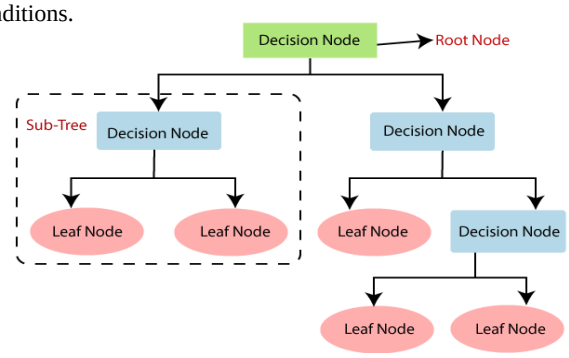


Fig. 3. Decision Tree Visual

2.4 K-Nearest Neighbors (KNN)
KNN is one of the simplest classification algorithms in machine learning. It is also a non-parametric algorithm, i.e., it does not make any assumptions about the underlying data. It stores all the

available data based on the similarity of the data, and during inference, it assigns that label which for which the data point is the closest or similar.

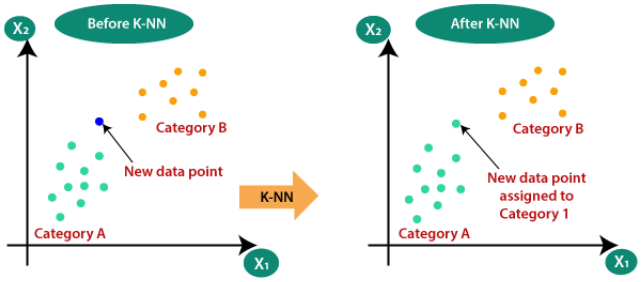
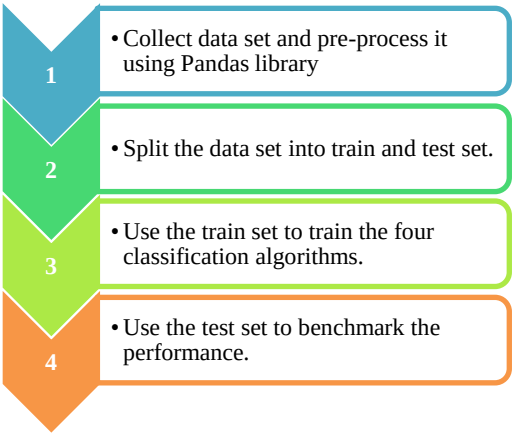


Fig. 4. KNN Visual

3. METHODOLOGY

The code for this project was entirely written in Python 3.10 using popular machine-learning libraries such as Sci-Kit Learn, NumPy, and Pandas. Sci-Kit Learn is a machine learning library bundled with functions to implement widely used machine learning algorithms such as Linear Regression, Logistic Regression, etc. NumPy is another helpful library used to efficiently represent data using NumPy arrays, which are highly optimized C arrays. The benefit of using NumPy over standard Python lists is NumPy arrays are much easier to handle, and computations on them are done much quicker by utilizing vectorization.

Before the data set can be used to train the algorithms, it must be pre-processed. Pre-processing of data involves removing any null or unwanted values that may affect the performance of the algorithm. This step is done using the Pandas library which is a popular library for data analysts to clean and pre-process data. Once the data has been pre-processed, the data set is split into a train and test set with approximately 70% and 30% of the original data set in the train and test sets respectively. The train set is used to train the different algorithms, and the test set is used to benchmark the performance of the algorithms.



The code for the project is available on this GitHub repository.

4. RESULT AND DISCUSSION

After fitting the training data to the algorithms, the testing data was used to benchmark the performance. The benchmarking parameters that we used were training accuracy, testing accuracy, prediction time, and confusion matrix analysis.

4.1 BENCHMARKING ANALYSIS

Algorithm	Training Accuracy	Testing Accuracy	Prediction Time(µs)
Logistic Regression	98.83%	98.6%	164
SVM	99.06%	97.20%	698
Decision Tree	100%	92.3%	186
KNN	96.00%	96.50%	25100

Table 1.0 Benchmarking Results

4.2 CONFUSION MATRIX ANALYSIS

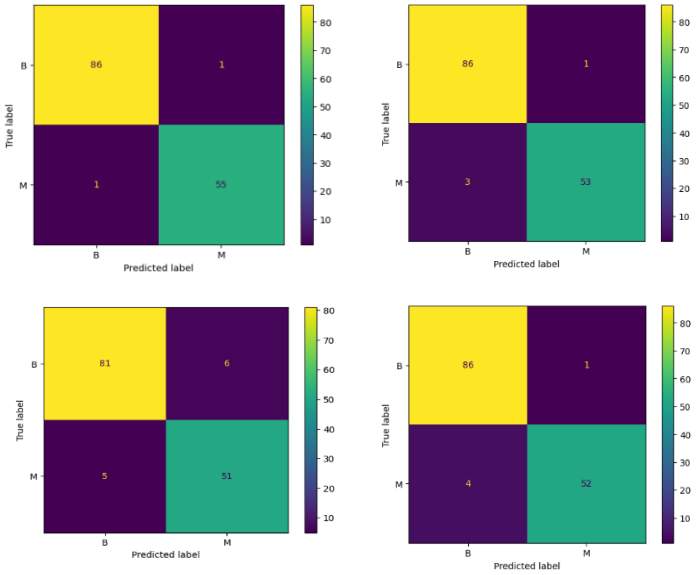


Fig. 6 (clockwise from top left) Confusion Matrices for Logistic Regression, SVM, Decision Tree, and KNN

With the confusion matrix we can calculate the recall, precision and precision. Recall gives an idea of how many examples we predicted correctly out of the positive classes. Recall is calculated as

$$R = \frac{True\ positives}{True\ positives + False\ negatives}$$

We want recall to be as high as possible. Precision gives us an idea that out of all the examples we predicted, how many are actually positive. Precision is calculated as follows

$$P = \frac{True\ positives}{True\ positives + False\ positives}$$

Algorithm	Recall	Precision
Logistic Regression	0.98	0.98
SVM	0.96	0.98
Decision Tree	0.94	0.93

K-Nearest Neighbors	0.95	0.98

Table 2. Precision and Recall scores

The results were obtained with ease as the Sci-Kit Learn library has many built in functions to measure the benchmarking parameters mentioned such as Accuracy, and Confusion Matrix.

After analyzing the results, all the algorithms have similar results except for KNN on the prediction time which is much higher when compared to the other algorithms. But on the contrary, the KNN algorithm seems to have generalized well and not overfit the data like in the case of the other algorithms. In terms of overall results, Logistic Regression is the most promising on this dataset, as the training and testing accuracy are good and the prediction time is the lowest among the algorithms. Furthermore, Logistic Regression also has very good precision and recall scores. Decision Tree and SVM seem to be overfitting the data, which can cause a problem as it does not generalize well. With the following dataset, Logistic Regression seems to be giving the best results when it comes to training with marginally lesser overfitting, and with respect to testing with a very good testing accuracy and prediction time.

5. CONCLUSION

Classification algorithms are some of the most widely used algorithms used in machine learning that are used to put a label on a given input. Choosing the right algorithm with respect to our data is very important as it will determine the accuracy and the time it takes for a prediction. We want our model to have a high testing accuracy and predict the output in a reasonable time frame. Furthermore, the algorithm should not overfit the data. An overfit of the data can be identified by a very high training accuracy. If our algorithm overfits the data, then it is not generalized properly, and will lead to anomalous results when given new data for prediction. Our method is a quick and dirty way to test different algorithms but the results cannot be definitive as the data set used I too small.

The results that we have obtained are only for one small dataset. Further testing on large datasets will make sure that our algorithm does not overfit the data and we can more definitively choose the right classification algorithm.

6. REFERENCES

[1] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[2] Machine Learning (2022, 17th October). In javatpoint. https://www.javatpoint.com/machine-learning.

[3] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.