Study for Iris Classification Based on Multiple Machine Learning Models

Yihan Zhou*

Department of Computer Science, Southern Methodist University, Dallas, United States

*Corresponding author: yihanz@smu.edu

Abstract. Classification algorithms in machine learning aim to classify data into different kinds, which is important in data mining. It is important and challenging to select the classification model with high accuracy and efficiency. To address this, this paper compares and analyzes Multilayer Perceptron (MLP)with different machine learning methods, including K-NearestNeighbor (KNN), Support Vector Machine (SVM), Logistic regression, decision tree, and random Forest. All models are trained to classify different kinds of Iris to show further comprehension of different models. All the related models are evaluated with metrics called accuracy and confusion matrices. The experimental results show that the confusion matrices based on various models are similar but MLP overperforms other models in terms of accuracy. With sufficient nodes per hidden layer for backpropagation, MLP has a remarkable ability to analyze and provide projections. With this superiority, it is expected that the neural network can have better performance in classification soon. After understanding the strength and limitations of various classification algorithms, the optimization models are expected to be developed to better solve real-world problems.

Keywords: MLP; KNN; SVM; Random Forest model; Neural Network; Iris dataset.

1. Introduction

Many real-world problems can be casted as classification. Machine learning classification methods aim to assign a class label in terms of features after training input data [1]. Choosing the most suitable classification model to deal with specific classification problems is of great significance. Recent research has analyzed and compared different models and summarized their characteristics.

To achieve better performance in terms of accuracy and efficiency, various methods are proposed to develop a more powerful algorithm. Recent classical classification models in machine learning mainly include K-nearest neighbors (KNN) [2], Support Vector Machine (SVM) [3], Logistic Regression [4], Decision Tree [5], and Random Forest [6]. KNN calculates the distance between two points and makes predictions based on k number of nearest observed points. However, KNN is sensitive to outliers and slow for the large dataset due to a large number of computations. SVM use kernels to find a decision boundary called hyperplane to maximize the distance between data points from different classes. The limitation of SVM is selecting the appropriate kernel is tricky and it is slow for a large dataset. Logistic regression uses the sigmoid function to map the probability of an event occurring between 0 and 1. The limitation is it has poor performance when the decision boundary is nonlinear. Decision tree splits the dataset into branches based on features and makes a prediction of a target variable. The disadvantages include trees are prone to overfitting. Random forest is an ensemble of multiple decision trees. The limitation is the low interpretability. Thus, this study considers Multilayer Perceptron (MLP)which is efficient in unstructured data with better self-learning capabilities [7].

Inspired by the structure of the human brain, neural network shows great potential in learning. Deep learning methods have gained lots of attention for their groundbreaking applications. They utilize multiple layers that transform data to learn the characteristics of the input efficiently [8]. In the age of big data, deep learning performs very well in computer vision, Natural Language Processing (NLP), and predictive modeling.

In this paper, different classification models including MLP, KNN, SVM, Logistic regression, Decision tree, and Random Forest are compared. After data visualization and analysis, the Iris dataset

is applied to various algorithms [9]. The experimental results show that the MLP model can achieve the best performance compared to other algorithms.

2. Method

This section describes the process of the comparison of all the related methods. First, this study visualizes and analyzes the Iris dataset (Sec. 2.1). Then, the MLP model is introduced (Sec. 2.2).

2.1. Data Visualization and Analysis

First, this study analyzes the relationship and correlation between these four features for the same class of flowers. Fig. 1 describes the correlation between sepal length and sepal width on the iris dataset. From Fig. 1, it can be found that for Iris Setosa flowers, the scatterplot shows a high, positive association between the sepal length and width. However, considering Iris Versicolor and Iris Virginica, the correlation is not obvious. At the same time, the points are more spread out on the graph and do not form a cluster like Setosa.

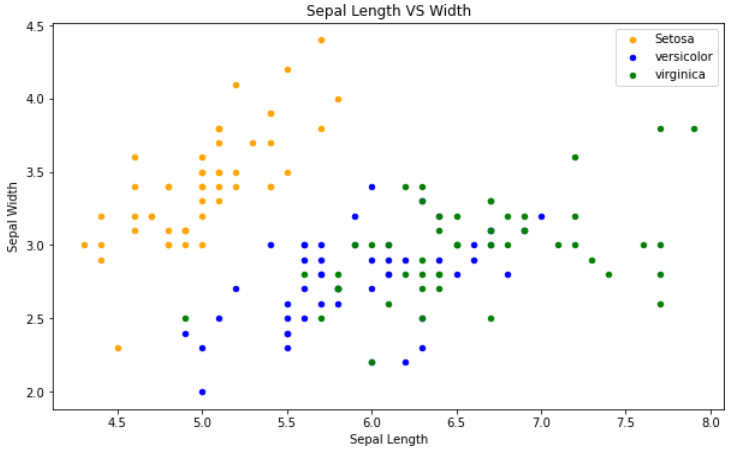

Fig. 1 The correlation between sepal length and sepal width on the iris dataset.

Fig. 2 shows the correlation between petal length and petal width on the iris dataset. It can be observed that there is a positive correlation in all classes of flowers. Comparing Fig. 1 and Fig. 2, it can be found that the petal figures are showing better clusters than sepal figures. This indicates that petal figures performs better and more accurate in species prediction over sepal.

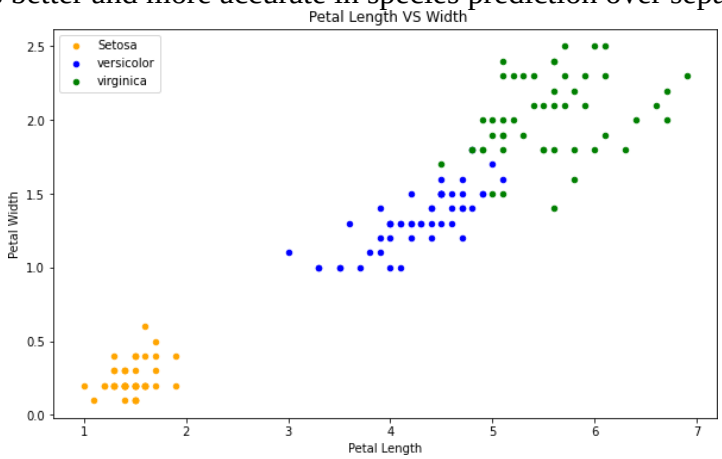

Fig. 2 The correlation between petal length and petal width on the iris dataset.

Fig. 3 shows how length and width are distributed in 3 different species and the comparison between them. It is indicated the statistics of each attribute in Iris classes and the probability density of each feature. It can be also found that a wider area represents that the iris flowers have a higher probability in that numerical value of features, whereas the skinner sections represent the lower probability. From Fig. 3, the data distribution of the three species can be found.

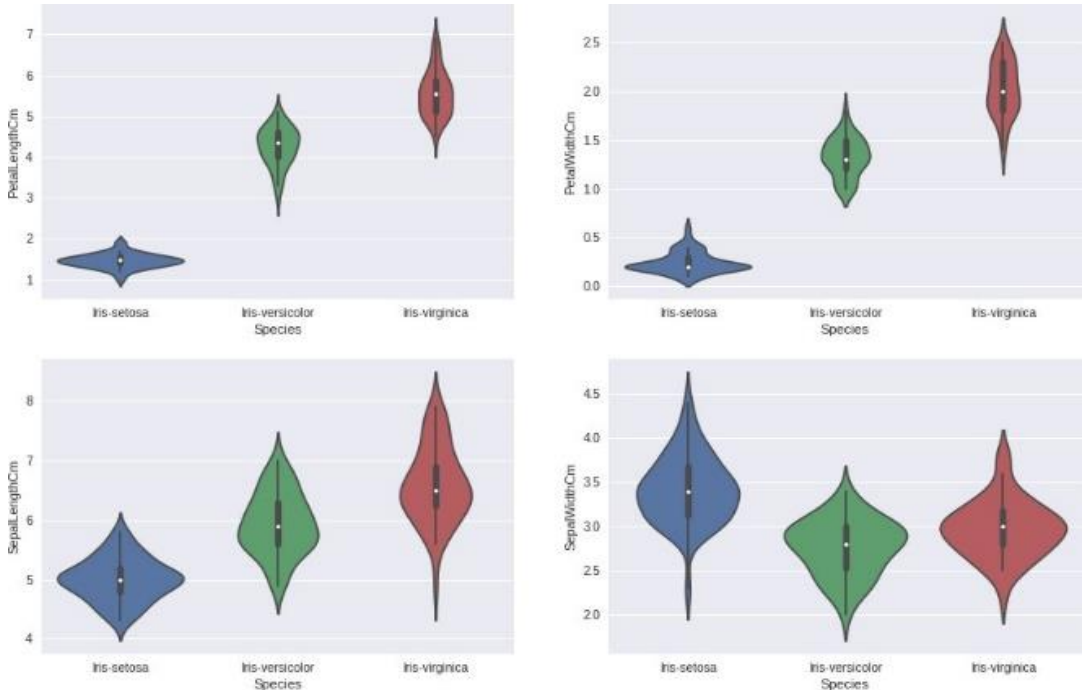

Fig. 3 Violin plots of 4 features (i.e. PetalLengthCm, PetalWidthCm, SepalLengthCm and SepalWidthCm) in 3 species

Before training the model, this study further visualizes the correlation matrix of four features of the collected data in Fig. 4 to show the correlation between four features based on the pearson coefficient. The value in a cell shows the correlation, where a positive number represents the positive relation. Since many features will reduce the accuracy, the heatmap is employed to select features. From Fig. 4, it can be observed that the width and length of the sepal are not correlated. The width and length of the petal are highly correlated. To train, this study will use all 4 features. In addition, 2 not correlated features will be used to check the accuracy of each algorithm.

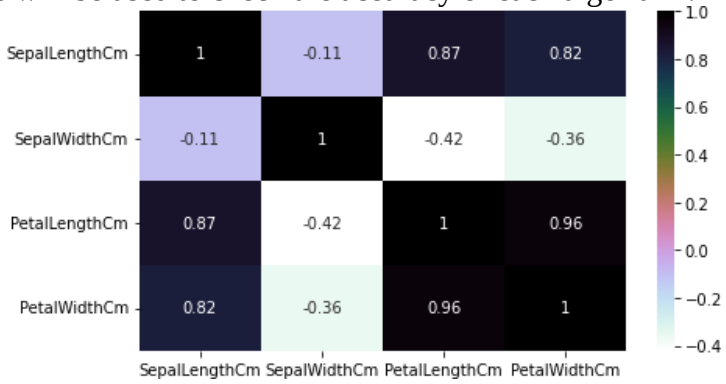

Fig. 4 The correlation between four features based on the pearson coefficient.

2.2. Model Construction

An MLP consists of at least three layers, namely an input layer, an output layer, and at least one hidden layer [10]. The input layer consists of number of neurons representing the features with corresponding weights. The output layer transforms the value using nonlinear functions known as activation functions. Each of the nodes in a multilayer perceptron is a neuron that uses an activation function. MLP is complicated since it required to modify hyperparameters such as neurons and iterations. Starting with a randomized value of weights and bias, MLP transforms all input values to the output layer [11]. Then, it performs backpropagation for training that updates the weight and bias in the inner layers. MLP repeats the propagation backpropagation. In the first step, MLP develops propagation algorithms to transform values. The algorithm of net activation is defined as:

$$L_j^l = \sum_i w_{ji}^l x_i^l = w_{j,0}^l x_0^l + w_{j,1}^l x_1^l + w_{j,2}^l x_2^l + \cdots + w_{j,n}^l x_n^l,$$
$$Y_j^l = g^l\left(\mathrm{L}_j^l\right)$$
$$\{y_i, x_{i1}, \ldots, x_{ip}\}_{i=1}^n \quad (1)$$

Where:

- L_j^l is the net activation
- x_i^l is the ith input of layer l
- y_j^l is the jth output of layer j
- w_{ji}^l is the weight of the jth neuron of input i
- $g^l(\cdot)$ is the activation function of layer l

Next, this study calculates the error between currently stored neuron-produced results in the output layer and the expected value, which is defined as (2):

$$E(k) = \frac{1}{2}\sum_{k=1}^{K}\left(d_j(k) - y_j(k)\right)^2. \quad (2)$$

Then, the Sigmoid Function defined as (3) is employed, Hyperbolic Tangent Function defined in (4) and ReLU Function defined in (5). In the next step, MLP does back-propagate algorithms.

$$f(x) = \frac{1}{1+e^{-x}} \quad (3)$$

$$f(x) = tanh(x) \quad (4)$$

$$f(x) = max(0, x) \quad (5)$$

For the output layer, the MLP firstly calculates the error in the output layer and then updates all the weights. Finally, it updates the bias. For the input layer, the MLP firstly calculates the error in the hidden layer. Then, it modifies all the weights. Finally, it will again update all the biases in the output layer. This process is called the training process.

For a clear comparison, this study further tunes the number of the hidden layer, neurons in a layer, and different activation functions of the MLP.

3. Experimental Settings

In this section, the experiment settings are described. First, this study introduces the iris dataset and the data attributes (Sec. 3.1). Next, the baseline classification algorithms are introduced, including KNN, SVM, Logistic Regression, Decision Tree, and Random Forest (Sec. 3.2). Finally, this study explained the evaluation indexes of the classification model (Sec. 3.3).

3.1. Dataset

The Iris flower dataset, describing three species of irises is used as the dataset in this study to evaluate the performances of all the related methods. For Iris Setosa, Versicolor, and Virginica, the dataset has 50 instances each. The dataset includes four features are measured: Sepal Length, Sepal Width, Petal Length, and Petal Width in centimeters. The 150 records are stored in a 150×5 array. Based on these 4 attributes, the species of the iris plant can be predicted. Table 1 shows attributes and corresponding types on the iris dataset

Table 1. Attributes and corresponding types on the iris dataset

Column	Attribute	Type
0	SepalLength	float64
1	SepalWidth	float64
2	PetalLength	float64
3	PetalWidth	float64
4	Species	object

3.2. Baselines

KNN: The K-nearest neighbors algorithm uses proximity to make predictions based on a grouping of one data point [12, 13]. K is defined by the user and the choice of k is depending on the input data. The algorithm finds the k nearest neighbors of a given point in the training set so that to assign a class label to that point. In order to find the k-nearest points, this study uses Euclidean distance $(p = 2)$ in the function to calculate the distance of a test point from other observations. The distance is defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}. \quad (6)$$

Since it takes up more memory and data storage, KNN may not perform well for a larger dataset [14]. In KNN algorithm, finding the value of K to achieve the best accuracy is challenging. The value of K indicates the number of nearest neighbors. KNN model computes the distance between test points and label points. Since the K value is highly dependent on the dataset, a plot between accuracy and K that helps is drawn to find the optimal K. The best K value found by scientists is usually the square root of N (the total number of samples). Fig. 5 represents the relationship between the accuracy and the value of the chose K.

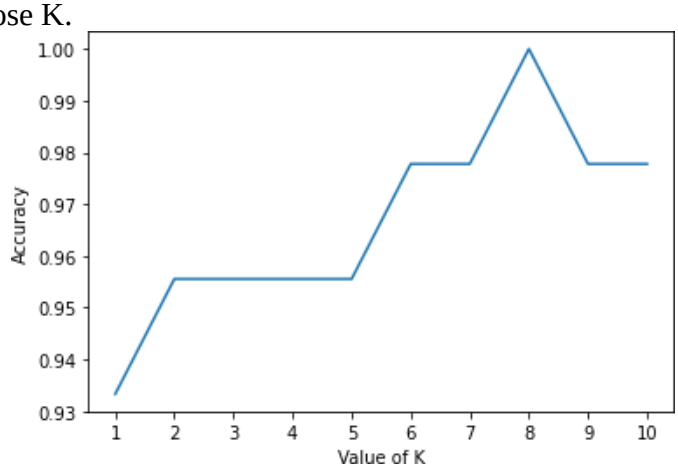

Fig. 5 The relationship between the accuracy and the value of the chose K

SVM: Support Vector Machines (SVM) map the data to a N-dimensional space and categorized the data points. SVM find the hyperplane with maximum margin using support vectors. The kernel function is used for transformation for non-linear data. First, this study uses the svc function with default parameters. To search for the best parameter, hyperparameters is used to train data using 5-fold cross-validation through a function called GridSearchCV().

Logistic Regression: Based on the concept of probability, logistic regression modeling transforms data to the value of probability using logistic functions. The input variables (i.e. features) are used to predict a categorical outcome variable (i.e. label). When data has too many predictor variables, it may be overfitting.

Decision Tree: Decision trees are methods that continuously split data according to a decision parameter learned from features [15]. In a decision tree, leaves indicate the decision of class labels,

and branches indicate the splitting rules. The tree can be visualized to show all possible solutions. However, a decision tree cannot generate the over-complex data well which is called overfitting.

Random Forest: Random Forest utilizes ensembled learning to combine the output of multiple decision trees. It is an extension of the bagging method which utilizes the feature randomness to create uncorrelated decision trees. Random forest reduced the risk of overfitting of decision trees since the uncorrelated decision tree lower the variance.

3.3. Evaluation Metrics

Accuracy： Accuracy is the most important index when evaluating classification models. The accuracy is a fraction of the right predictions we made. In classification, accuracy can be defined in terms of actual or predicted positives and negatives. Accuracy is defined as (7):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

Confusion Matrix: Confusion matrix is a table with 4 grids that counts the number of predicted values and actual values, which is shown in Table 2. Confusion matrix is useful when analyzing the Accuracy, Recall, Precision, and AUC-ROC curves.

Table 2. The definition for confusion matrices

Actual Value / Predicted Value	Positive	Negative
Positive	TP	FP
Negative	FN	TN

4. Results and Discussion

The performance of all six classification algorithms in terms of accuracy is shown in Table 3. From Table 3, it can be found that all the related methods have excellent performance since the size of the dataset is not large. To analyze, MLP whose accuracy comes to 1.0 outperforms other models. As a multiple-layer neural network, it could efficiently solve the nonlinear problem. With their hidden layers, MLP could make the approach to any differentiable functions. Its complexity makes a great capability to do classification efficiently. The accuracy of KNN is high since this study has made an approach to choose the best value of K. SVM could utilize kernel tricks to solve non-linear problems and have a good performance. However, due to the correlation between variables, the logistic regression only has the accuracy of 0.955556. As an extension of bagging, the random forest utilizes the randomness to create the uncorrelated decision. Thus, it outperforms decision tree methods due to feature randomness.

Table 3. The performance of all six classification algorithms in terms of accuracy

Model	Accuracy
KNN	0.977778
SVM	0.955556
Logistic Regression	0.955556
Decision Tree	0.933333
Random Forest	0.977778
MLP	1.0

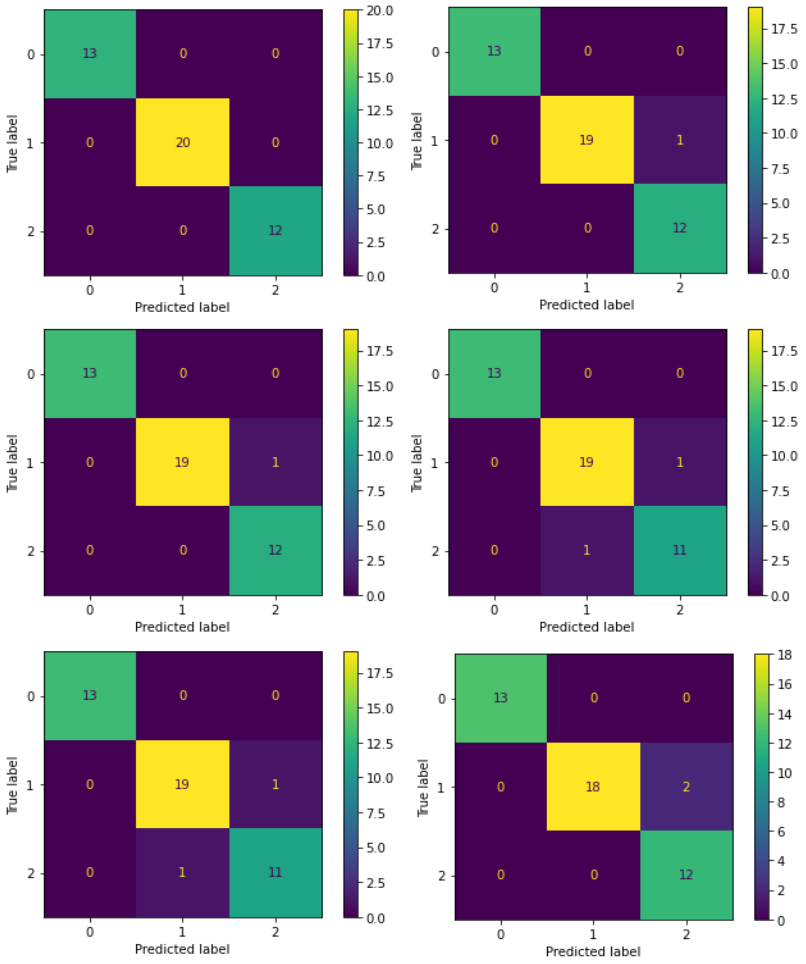

Fig. 6 The performance of various methods based on confusion matrices

Fig. 6 shows the output confusion matrix of six algorithms. It can be found that because of the size and neatness of the dataset, there is little difference in confusion matrices. In the future, comparing the confusion matrices of various algorithms with more complex datasets is worthy of attention.

5. Conclusion

Various typical machine learning algorithms called MLP, KNN, SVM, Logistic Regression, Decision Tree, and Random Forest are compared in detail on the Iris dataset. The accuracy and confusion matrices are taken as the evaluation indexes to compare the performance of the classification algorithms. The experiment results demonstrates that the MLP algorithm overperforms basic models in terms of accuracy. In the future, further study plans to continue studying the classification models for data mining. Ensemble models will be also considered to test the dataset. It is expected that the neural network can be the most powerful model in classification soon.

References

[1] S. Ali, K. A. Smith, On learning algorithm selection for classification. Applied Soft Computing, 2006, 6(2): 119-138.

[2] J. M. Keller, M. R. Gray and J. A. Givens, A fuzzy K-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics, 1985, SMC-15(4): 580-585.

[3] N. Cristianini and J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods. Cambridge: Cambridge University Press, 2000.

[4] S. Dreiseitl, L. Ohno-Machado, Logistic regression and artificial neural network classification models: a methodology review. Journal of Biomedical Informatics, 2002, 35: 352-359.

[5] Y. Y. Song, Y. Lu, Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry, 2015,27(2): 130-135.

[6] L. Breiman, Random Forests. Machine Learning, 2001, 45: 5-32.

[7] M. W Gardner, S. R Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment, 1998, 32(14-15): 2627-2636.

[8] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 2015, 521: 436-444.

[9] Information from: https://archive.ics.uci.edu/ml/datasets/iris

[10] S. K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, and classification, IEEE Transactions on Neural Networks.1992, 3(5): 683-697.

[11] D. W. Ruck, S.K. Rogers, M. Kabrisky, Feature selection using a multilayer perceptron. Neural Network Comput, 1990, 2: 40-48.

[12] Z. Y. Deng, X. S. Zhu, D. B. Chenget. Efficient kNN classification algorithm for big data, Neurocomputing, 2016, 195: 143-148.

[13] A. Shokrzade, M. Ramezani, F. Akhlaghian, A novel extreme learning machine based KNN classification method for dealing with big data. Expert Systems with Applications, 2021.183: 115293.

[14] J. X. Deng. B. Xie. D. D. You, Process parameters design of squeeze casting through an improved KNN algorithm and existing data. Journal of Manufacturing Processes, 2022, 84 (1320-1330)

[15] T. G. Dietterich, an experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Machine Learning, 2020 40: 139-157.