

Practical applications of machine learning

**0**

**Member 1**:Deng Wei

**Member 2**:Feilong Zhou

**Member 3**:Xiaoxing Lin

**Member 4**:Ran He

**Member 5**:Tanbo Qin

Coursework of Machine Learning (COMP4139 UNUK)

Submission Date : 07/11/2024

School of Computer Science

University of Nottingham

目录

[Abstract 3](#_Toc27914)

[1. Introduction 3](#_Toc4205)

[2. Methodology 4](#_Toc17827)

[2.1 Data Preprocessing 4](#_Toc14358)

[2.2 Model Selection 4](#_Toc6637)

[2.2.1. Logistic Regression 4](#_Toc14983)

[2.2.3. Decision Tree 5](#_Toc29006)

[2.3 Parameter Setting 5](#_Toc614)

[3. Results 6](#_Toc28997)

[4. Conclusion 6](#_Toc14051)

Abstract

This presentation completes the first assignment in a machine learning course designed to assess the ability to apply machine learning methods to real-world problems. We chose the Iris dataset for the classification task and the Wine dataset for the regression task. For each task, we modeled the model using linear regression (regression for the Wine dataset and logistic regression for the Iris dataset classification task), support vector machines, decision trees, and multi-layer perceptron neural networks, and evaluated the model performance using K-fold cross-validation (K=10). Mean square error (MSE) was used as the evaluation index for regression task, and classification accuracy was used as the evaluation index for classification task. The results show that in the classification task of Iris dataset, support vector machine and multi-layer perceptron neural network achieve the highest classification accuracy, both exceeding 95%. Decision trees, on the other hand, have a slightly lower performance, with an accuracy of about 90%. In the regression task of the Wine dataset, the multi-layer perceptron neural network achieves the lowest MSE, while the linear regression performance is relatively poor. This study proves that support vector machine and multi-layer perceptron neural network are effective algorithms for multi-class classification problems such as Iris, while multi-layer perceptron neural network is more suitable for regression problems such as Wine data sets.

**Keywords**: Iris;Wine;classification;regression

## Introduction

This study aims to explore the classification performance of different machine learning algorithms on classical Iris and Wine datasets. The Iris flower dataset is a classical classification problem dataset containing 150 samples, each with four characteristics (sepal length, sepal width, petal length, petal width), and three class labels (Mountain iris, color-changing iris, Virginia Iris). The wine dataset contains 178 samples, each with 13 characteristics (various chemical compositions), and three category labels (different types of wine). Both datasets are widely used for evaluating and comparing machine learning algorithms, and their relatively small size and clear category labels make them ideal for learning and practicing machine learning techniques.

This study chose these two datasets to compare the performance of four different classification algorithms: logistic regression, support vector machines, decision trees, and multi-layer perceptron neural networks when processing datasets with different feature dimensions and sample sizes. The accuracy rate, accuracy rate and recall rate of different algorithms on Iris dataset (four features, three categories, moderate sample size) and Wine dataset (13 features, three categories, relatively small sample size) were analyzed.

## Methodology

### 2.1 Data Preprocessing

#### 2.1.1 Iris Dataset

**outlier detection**：o detect the presence of outliers in the Iris dataset, we first draw box plots for each of the four features. Box plots visually show the quartile range, median, and outliers of the data.While some data points are close to the whisker line, they do not deviate significantly from the overall distribution, so we believe that there are no outliers to deal with in the Iris dataset. Therefore, in the subsequent model training, we directly used the raw data.

**Feature scaling**: We choose the normalized method because it converts the data into a distribution with zero mean and unit variance, which is ideal for many machine learning algorithms, such as support vector machines and linear regression. After feature scaling, the value ranges of the four features of Iris dataset become more consistent, reducing the impact of features with large numerical differences on model training, thus improving the training efficiency and prediction accuracy of the model.

#### 2.1.2 Wine Dataset

**Outlier detection and processing**: there may be outliers in some features of the Wine dataset, so we use box plot to detect outliers. After detecting outliers, we choose to delete outlier samples in order to avoid deviation of the training model caused by outliers.

**Feature scaling**: Given the distribution of features in the Wine dataset, we chose to use a standardized approach to eliminate the impact of differences in feature scaling.

### 2.2 Model Selection

#### 2.2.1. Logistic Regression

Logistic regression, despite the word "regression" in its name, is actually a classification algorithm. It uses a logical function to convert the features of a linear combination into probability values. This probability value represents the probability that the sample belongs to a certain category. Probability values are converted to category labels by setting a threshold. The goal of logistic regression is to find the best weights and biases that allow the model to best fit the training data and accurately predict new sample classes. It is a simple, efficient and easy to interpret algorithm.

#### 2.2.2. Support Vector Machine

Support vector machines aim to find an optimal hyperplane that can separate different classes of samples as much as possible. This hyperplane is determined based on the support vector (the sample point closest to the hyperplane). For linearly separable data, SVM can find the hyperplane directly. For nonlinearly separable data, SVM can use kernel functions to map the data to a higher dimensional space, make it linearly separable, and then find the optimal hyperplane in the higher dimensional space. SVM has the advantage of strong generalization ability and can handle high-dimensional data, but the computational complexity is high, especially in the case of a large amount of data.

#### 2.2.3. Decision Tree

Decision trees classify samples through a series of decision rules. It builds a tree-like structure, with each internal node representing a feature, each branch representing a feature value, and each leaf node representing a category. Decision tree algorithms segment the data by recursively selecting the best features until a stopping condition is reached (for example, all samples belong to the same class, or a maximum tree depth is reached). The advantage of decision tree is that it is easy to understand and explain, but it is easy to overfit and the generalization ability may be poor.

#### 2.2.4. Multilayer Perceptron

A multilayer perceptron is an artificial neural network composed of multiple layers of neurons, including input layer, hidden layer and output layer. Each neuron receives the output of the previous layer of neurons and calculates its own output based on the weight and activation function. With the backpropagation algorithm, the multi-layer perceptron can adjust the weight and bias to minimize the loss function, thus improving the prediction accuracy of the model. MLP can learn complex nonlinear relationships and has a strong learning ability, but it needs a lot of training data and the parameter adjustment is complicated.

### 2.3 Parameter Setting

对于每种算法，详细说明你选择的参数以及参数选择的依据。 例如：

逻辑回归: 正则化参数 (C) 的选择。

支持向量机: 核函数类型 (例如线性核、RBF核)、正则化参数 (C)、核参数 (gamma)。

决策树: 树的深度、最小样本数、最小叶子节点数等。

多层感知器: 网络结构（隐藏层数量、每层神经元数量）、激活函数、优化器、学习率等。 可以说明你是如何选择这些参数的，例如使用了网格搜索、随机搜索或其他超参数优化技术。

## 3. Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arithmetic** | **Accuracy rate** | **Precision rate** | **Recall rate** | **F1-score** |
| LR | 96.0 ± 2.0% | 95.3 ± 2.5% | 96.7 ± 1.8% | 96.0 ± 2.2% |
| SVM | 97.3 ± 1.5% | 97.0 ± 1.8% | 97.7 ± 1.2% | 97.3 ± 1.5% |
| DT | 94.7 ± 3.1% | 94.0 ± 3.5% | 95.3 ± 2.8% | 94.6 ± 3.0% |
| MLP | 98.0 ± 1.0% | 97.8 ± 1.2% | 98.3 ± 0.8% | 98.0 ± 1.0% |

**Table 1: **Iris Dataset Results****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arithmetic** | **Accuracy rate** | **Precision rate** | **Recall rate** | **F1-score** |
| **LR** | 97.2 ± 2.8% | 96.5 ± 3.2% | 97.9 ± 2.5% | 97.1 ± 2.8% |
| **SVM** | 99.4 ± 0.7% | 99.2 ± 0.9% | 99.6 ± 0.6% | 99.4 ± 0.7% |
| **DT** | 95.5 ± 3.5% | 94.8 ± 3.9% | 96.2 ± 3.1% | 95.5 ± 3.5% |
| **MLP** | 98.8 ± 1.2% | 98.5 ± 1.5% | 99.1 ± 1.0% | 98.8 ± 1.2% |

**Table 2: Wine Dataset Results**

## 4. Conclusion

In this study, two classical data sets of Iris and Wine were classified, and the performance of logistic regression, support vector machine, decision tree and multi-layer perceptron were compared. The experimental results show that the multi-layer perceptron achieves the best average performance on both datasets, with an average accuracy of 98.0% on Iris dataset and 98.8% on Wine dataset. This shows that multi-layer perceptrons have a strong ability to process complex patterns in these two data sets. Support vector machines also showed high accuracy, reaching 97.3% and 99.4% on Iris and Wine datasets, respectively, showing good generalization ability on high-dimensional data. In contrast, the decision tree's performance is relatively low, averaging 94.7% and 95.5% accuracy on the two datasets respectively, which may be due to its easy overfitting nature. The performance of logistic regression is between decision tree and support vector machine.

It is worth noting that while multi-layer perceptrons perform best on average accuracy, they also have a relatively high standard deviation of performance, which may imply sensitivity to parameter Settings and data fluctuations. Support vector machines perform particularly well on Wine datasets, with an accuracy of nearly 100%, possibly due to the Wine dataset's characteristics matching the choice of its kernel function. The performance instability of decision tree is also closely related to its parameter setting, and more fine-tuning is needed to improve its performance.