

CMPUT 566, Introduction to Machine Learning

Final Project

Mobina Mosannafat

Table of Contents

Problem Definition..... 2

Dataset..... 2

Classification Models 4

 Logistic Regression 4

 Support Vector Machines (SVM)..... 4

 2-Layer Perceptron..... 4

Hyper Parameter Tuning..... 6

Results..... 7

Problem Definition

In the context of this project, our foremost goal is to use information on various wine samples, systematically categorized as either good or bad quality.

Dataset

Input Variables (Physicochemical Tests or x) in our problem are:

1. Fixed Acidity
2. Volatile Acidity
3. Citric Acid
4. Residual Sugar
5. Chlorides
6. Free Sulfur Dioxide
7. Total Sulphur Dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol

The output variable (Sensory Data or y) is Quality ('good' and 'bad'). These attributes represent features obtained from physicochemical tests, with the target variable being the wine quality derived from sensory data. The primary objective is to leverage this dataset for training and evaluating classification models to effectively categorize wines as either 'good' or 'bad' based on these attributes. Table 1 shows several samples of this dataset and Figure 1 shows the Correlation between different features of wines.

To facilitate the classification task, the categorical labels of the 'Quality' are encoded. Specifically, the label 'bad' is mapped to 0, and 'good' is mapped to 1. This numerical representation allows for seamless integration with classification algorithms.

Table 1 Examples of wine samples

index	fixed acidity	Volatile acidity	Citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
1	7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8	bad
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8	bad
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8	good
4	7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad

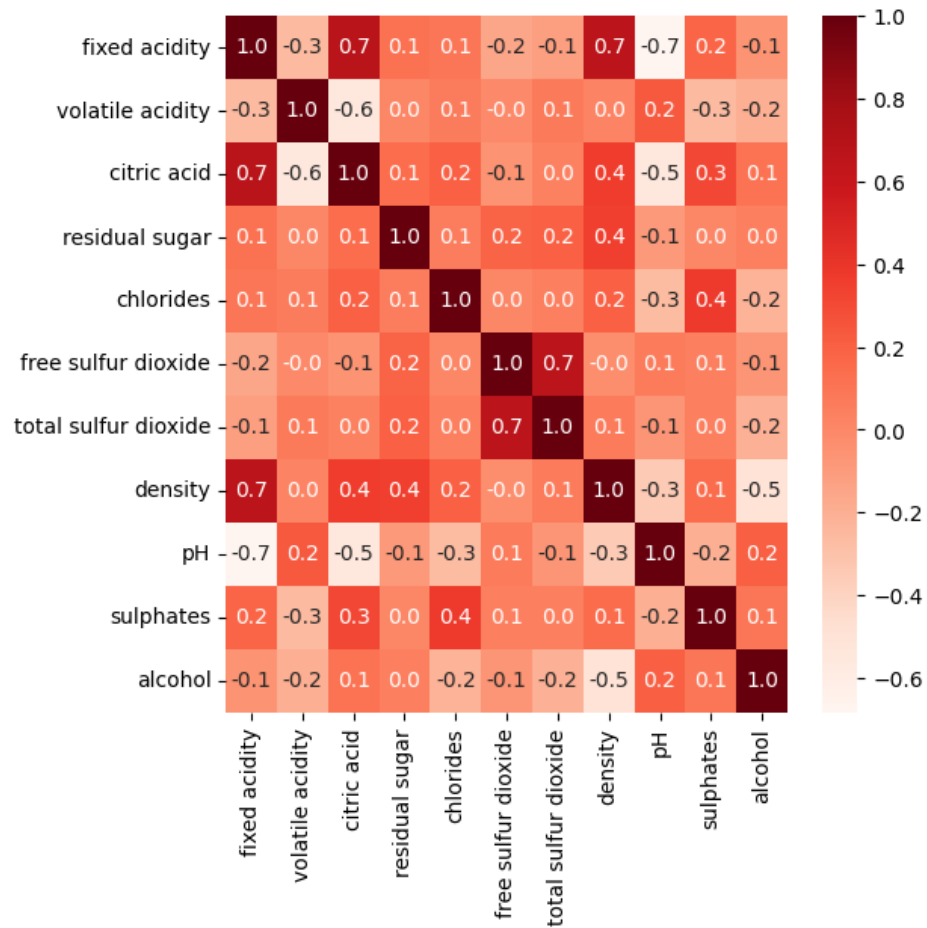


Figure 1 Correlation between different wine's features

The dataset is initially split into three categories: train, test, and validate, with proportions set at 0.6, 0.2, and 0.2, respectively. Subsequently, normalization is applied to all three subsets.

Classification Models

In our project, we implement three distinct classification models to address the wine quality categorization task. Each model offers unique characteristics suited for binary classification:

Logistic Regression

- **Type:** Linear model designed for binary classification.
- **Idea:** Logistic Regression models the probability that an instance belongs to a particular class. It employs the logistic function to map a linear combination of input features to a value between 0 and 1.
- **Hypotheses Class:** logistic function (sigmoid function) which can be represented as $h_{\theta}(x) = \frac{1}{1 + e^{-(\theta^T x)}}$.
- **Training/Learning Objective:** Minimizing the logistic loss (binary cross-entropy) during training, which can be represented as
$$J(\theta) = -\frac{1}{M} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

Support Vector Machines (SVM)

- **Type:** Linear and non-linear classifier.
- **Idea:** SVM aims to find a hyperplane that optimally separates data points of different classes in a high-dimensional space. It prioritizes maximizing the margin between classes, and for non-linear problems, it can utilize kernel functions to map features into a higher-dimensional space.
- **Hypotheses Class:** Decision function which can be represented as $f(x) = \operatorname{argmax}_k (\langle w_k, x \rangle + b_k)$.
- **Training/Learning Objective:** Maximizing the margin between different classes while ensuring correct classification, which can be represented as
$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i(\langle w, x_i \rangle + b)).$$

2-Layer Perceptron

- **Type:** Fully connected feedforward neural network tailored for binary classification.
- **Idea:** A neural network with one hidden layer processes input features through weights and biases, applying activation functions to introduce non-linearity. This architecture enables the model to capture more intricate relationships in the data compared to linear models.
- **Hypotheses Class:** Composition of multiple layers with non-linear activation functions which can be represented as $h(x) = \sigma(W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2)$.
- **Training/Learning Objective:** minimizing the cross-entropy loss during training, which can be represented as

$$J(W, b) = -\frac{1}{M} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

The measure of success for all classification methods in this project is validation and test accuracy.

Hyper Parameter Tuning

In the hyperparameter tuning phase, we systematically explored the impact of different values for the learning rate and batch size on each classification model based on the validation accuracy. Specifically, we considered three learning rates—0.001, 0.01, and 0.1—and three batch sizes—10, 100, and 1000.

The purpose of this exploration is to identify the combination of learning rate and batch size that optimizes the performance of each model. By varying these hyperparameters, we aim to fine-tune the training process and enhance the overall effectiveness of the models in classifying wine quality.

In the subsequent analysis, we will report and examine the model performance under each combination of learning rate and batch size, providing insights into the hyperparameter settings that yield the best results for our specific classification task.

Results

Due to constraints in hardware resources and time limitations, we were restricted to training the models for up to 100,000 epochs. Notably, the deep neural network (DNN) model did not converge within this specified epoch limit. It is crucial to acknowledge that the final outcome may differ if we had the capacity to train the model for an extended number of epochs. Consequently, the subsequent reports and conclusions are based on the outcomes obtained within the given epoch constraint.

Training Details:

All classification models underwent training for a substantial 10,000 epochs, providing an extensive learning process for each model.

Validation Performance Analysis

Table 2 presents the validation performance (accuracy) of each model across different combinations of learning rates and batch sizes. Notably, the results indicate optimal hyperparameter settings for maximizing performance.

- Optimal Learning Rates:

1. Logistic Regression: 0.01
2. Support Vector Machines (SVM): 0.1
3. Neural Networks: 0.1

- Optimal Batch Sizes:

1. Logistic Regression: 10
2. Support Vector Machines (SVM): 100
3. Neural Networks: 1000

Performance Comparison:

Figure 2 visually represents the comparative performance of Logistic Regression, SVM, and Neural Networks for wine quality classification. The observed trend aligns with the ordered mention of models, indicating that Deep Neural Networks, Logistic Regression and SVM perform better in the same order they were mentioned.

Table 2 Best validation accuracy of classification models for different values for Learning Rate and Batch Size.

Model	Learning Rate	Batch Size	Validation Accuracy
Logistic Regression	0.001	10	0.71
SVM			0.48
Neural Network			0.77
Logistic Regression	0.001	100	0.64
SVM			0.57
Neural Network			0.78
Logistic Regression	0.001	1000	0.64
SVM			0.50
Neural Network			0.79
Logistic Regression	0.01	10	0.73
SVM			0.54
Neural Network			0.78
Logistic Regression	0.01	100	0.72
SVM			0.54
Neural Network			0.79
Logistic Regression	0.01	1000	0.67
SVM			0.57
Neural Network			0.77
Logistic Regression	0.1	10	0.73
SVM			0.50
Neural Network			0.78
Logistic Regression	0.1	100	0.73
SVM			0.71
Neural Network			0.76
Logistic Regression	0.1	1000	0.71
SVM			0.54
Neural Network			0.80

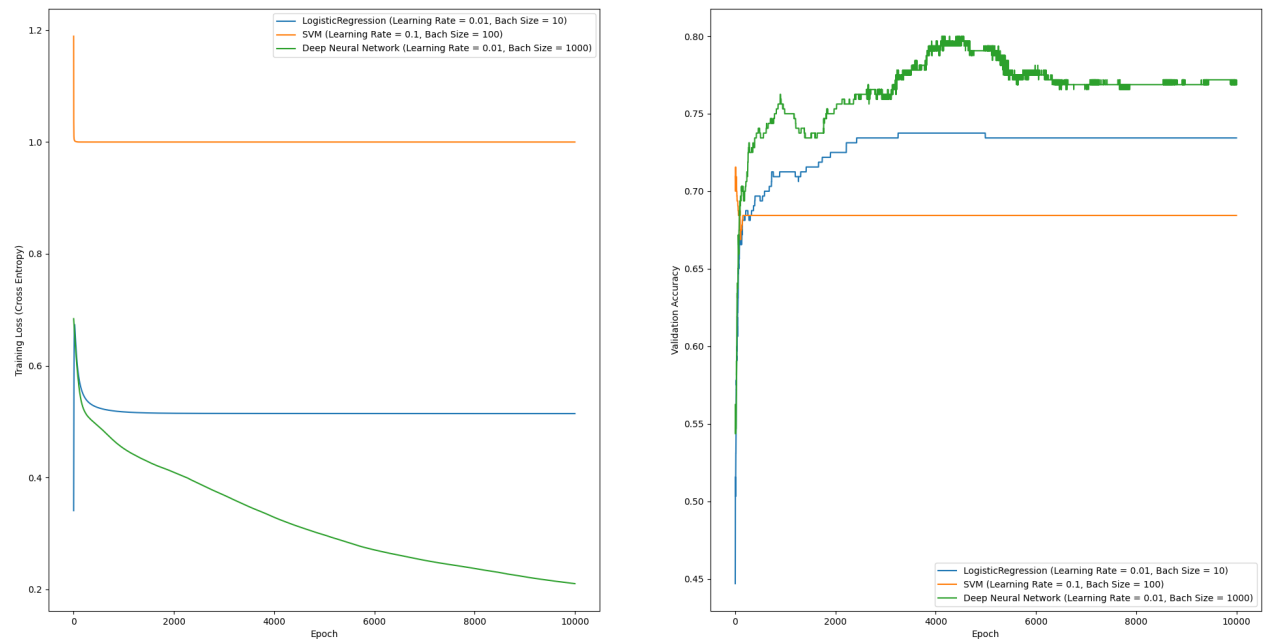


Figure 2 Training loss and validation accuracy of deafferent classification models with different learning rate and batch sizes