EE3-23: Coursework for Introduction to Machine Learning

Zheng Yang Lee
CID: 01042500
Department of Electrical and Electronic Engineering
Imperial College London
zyl115@ic.ac.uk

1. Abstract

The goal of the coursework is to access ability to solve a machine learning project. In this coursework, dataset (b) was used to predict the quality of wine given a number of features. Different machine learning algorithms were tested and their performances were evaluated. Techniques such as regularisation and cross-validation to prevent over-fitting were explored and discussed. A solution is proposed at the end of this report to solve the given problem.

2. Problem Presentation

The learning problem for dataset (b) is to predict the wine quality from the scale of 1 to 9 given a set of features, hence a regression problem. The goal of the machine learning model is to minimize the difference between predicted quality and actual quality of the wine. Mean-squared error was chosen to be the loss function for this learning problem, as larger deviation from the actual quality is penalized more, which fits the learning problem nicely. This loss function also conveniently returns a positive-valued error in all cases. Other loss functions can be defined based on the use case of the model, such as preference of overestimation over underestimation of quality, and vise versa. In this coursework, no preferences were assumed and hence overestimation and underestimation of the quality were penalised equally.

3. Data Preprocessing

The data given were split into two different sets, one for red wine and one for white wine. Since the type of wine is not a concern, and the model is designed to predict quality for both red and white wine, the two datasets were simply merged into one for training and testing.

The combined dataset is then split into training set and test set, with the ratio of 8 to 2. This is to enable evaluation of model performance by using it on the test set. The result will be somewhat representative of the actual performance of the model, as the test set were not involved in the training of model.

The features of training set is then used to build a scaler that would transform the features of training set to have a mean of 0, and variance of 1. This scaler is then applied to features of both training and test set. This process is called features normalisation, and is required for many machine learning algorithms to behave optimally.

Note that the dataset mainly consists of wine of average quality, that is between 4-6. This will cause the trained model to be better at predicting average wine, and worse at predicting quality of good and bad wine. A trade-off between accuracy and precision can be made by changing the output from values of 1 to 9, to classes of good, average, and bad wine. This is worth considering given the heavily biased dataset. However, due to the problem definition, the value needs to be a value from 1 to 9, hence this modification is not implemented.

4. Baseline Predictor

As mentioned above, this is a regression problem. A suitable baseline predictor is multiple linear regression[1]. The dataset contains 11 different features, and the learning algorithm will generate a weight vector of 12 coefficients (including the intercept term) that minimizes the loss function, which is the mean-squared error. This is suitable as a baseline predictor as it is a simple linear model that is computationally cheap. Linear regression is applied onto the training set, and the result is shown in Table 1.

Linear regression as the baseline algorithm managed to train the model suprisingly well. The training error is 0.63017, which is acceptable given that it is only a simple linear model. It also seems to generalise well, and performed similarly on the test set, with an error

Training Error	Test Error	Generalisation Error
0.63017	0.64615	0.01598

Table 1. Performance of Linear Regression

of 0.64615. The property of good generalisation is expected as the complexity of the hypothesis set is very low. The intercept term and the coefficients are shown in Table 2.

Feature Name	Coefficient
Intercept term	5.82471
Fixed Acidity	0.08879
Volatile Acidity	-0.21466
Citric Acid	-0.00895
Residual Sugar	0.22642
Chlorides	-0.01717
Free Sulfur Dioxide	0.09634
Total Sulfur Dioxide	-0.14072
Density	-0.18653
pH	0.07524
Sulphates	0.12023
Alcohol	0.31254

Table 2. Coefficients for linear regression

5. Advanced Algorithm

The complexity of linear regression model is low, and there is a high probability of under-fitting. A few models that are more complex than linear regression were tested, and precautionary measures were taken to stay away from the other end of spectrum, which is over-fitting.

5.1. Polynomial Regression

One simple way to increase model complexity is through non-linear transformation. The features of dataset were transformed with a polynomial transformation of degree 4. The number of coefficients after the transformation can be calculated using the following equation.

Let $g \epsilon \mathbb{F}[x_1, ..., x_n]$ be a polynomial of degree d with n variables. Number of its coefficients is $\binom{n+d}{d}$.

With a polynomial transformation of degree 4, the new dataset now has 1365 features. This new dataset is then used to train the model using linear regression. The result is shown in table 3.

Training Error	Test Error	Generalisation Error
0.27844	2.78799	2.50955

Table 3. Performance of Polynomial Regression

As seen from table 3, the best hypothesis in this class yields much better training error than that of linear regression. This is expected as a more complex model will be able to fit the data points better. However, this model performs badly out of sample, which is shown by the high test error in table 3. This is also expected, as the complexity penalty term will increase with model complexity, causing the model to generalise badly. This phenomena is called over-fitting, and is caused by the noisy targets in the dataset. This calls for regularisation: a technique commonly used to reduce risk of over-fitting.

5.2. Ridge Regression

Ridge regression is a form of linear regression with regularisation of L2 penalty. Conceptually speaking, it constrains the sum of the coefficients squared to a constant, hence reducing the complexity of the model. Detailed explanation of the algorithm can be found in the lecture notes [2]. The parameter $alpha$ determines the regularisation strength used to train the model, and the choice of alpha is extremely important. The dataset with polynomial transformation of degree 4 was used to train the model using ridge regression with different alpha values. The model is then used to predict output for test data to evaluate the out-of-sample performance. The results can be seen from the graph in Figure 1.

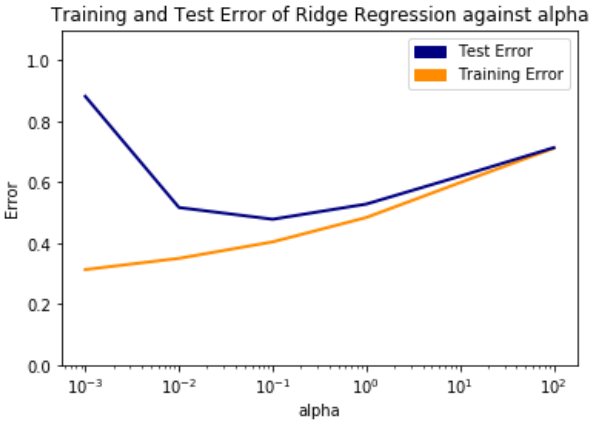

Figure 1. Error of Ridge Regression against alpha

The graph in Figure 1 clearly shows the behavior of the model trained with different alpha values. When alpha

is too small, the regularisation strength is weak, causing the model to generalise badly. When alpha is too big, generalisation error becomes very small. However, both training and test error increases as alpha increases, resulting in under-fitting. We can deduce from the graph that the optimal value for alpha is between 10^{-2} and 10^{0}, where there is a good compromise between training error and generalisation error.

One way to obtain a good alpha value from the training data alone is through cross-validation[3]. Cross validation can be performed on each alpha value, and the alpha value that gives the minimum cross-validation score is selected. The cross-validation curve is plotted along with the test error curve in Figure 2.

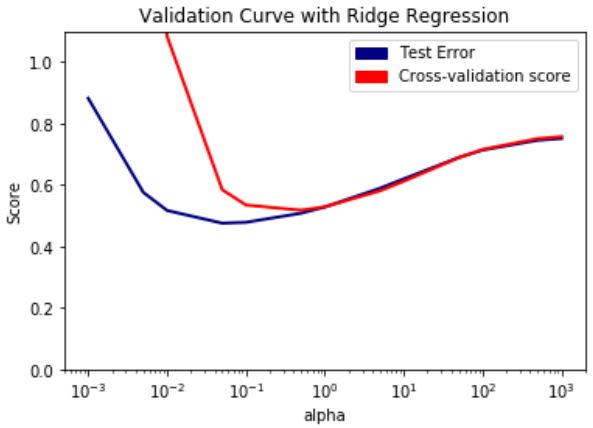

Figure 2. Cross-validation curve of Ridge Regression

As shown in Figure 2, the cross validation curve is highly correlated to the test error. The alpha that gives us the lowest cross-correlation score in this case is alpha = 0.5, which is within the range of 10^{-2} to 10^{0}. The model is then trained with alpha = 0.5, and the result is shown in Table 4.

Training Error	Test Error	Generalisation Error
0.45646	0.50823	0.05177

Table 4. Performance of Ridge Regression with alpha = 0.5

From the result, we can see that even though the model performs worse in-sample, it actually performs better out-of-sample, giving a test error of 0.50823. The property of good generalisation is highly valued as it means that the model will be more reliable when predicting out-of-sample data.

5.3. Neural Network: Multilayer Perceptrons

Multilayer Perceptrons is a neural network that is non-linear. This implies that features transformation is not needed as non-linearity can be dealt with by the algorithm. The detailed explanation of the algorithm can be found in lecture notes 5 [4]. The trade-off of using neural network as compared to the other linear algorithm is that it is significantly slower in training. This is due to it not having a closed-form solution, and the solution needs to be obtained iteratively. Besides, it contains many hyperparameters and can be hard to optimise. In this section, the effects of number of hidden layers and number of nodes will be discussed.

The model was first trained with different number of hidden layers, with the number of nodes being 10 in each layer. Cross validation were performed and the results were plotted in a graph in Figure 3.

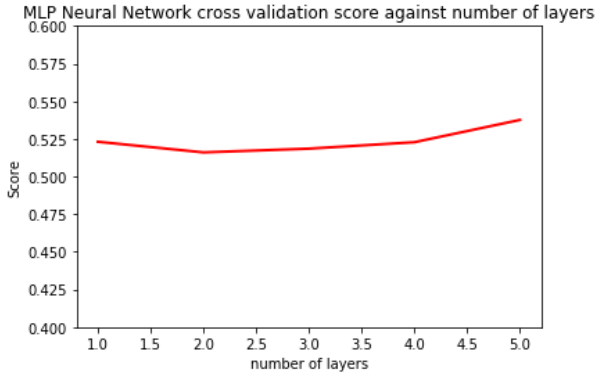

Figure 3. Cross-validation curve of Multilayer Perceptron: number of hidden layers

As can be seen from the graph in Figure 3, the number of layer does not have much of an impact to the performance of the algorithm. Hence, one hidden layer were chosen due to the significantly shorter training time needed. Next, the effect of number of nodes in the hidden layers were explored. The model was trained with one hidden layer of different number of nodes, and similarly, cross validation was performed, and the results were plotted on a graph in Figure 4.

The graph in Figure 4 shows that the cross validation score decreases as the number of nodes increases. It levels off at around 90, and shows no significant improvement after that. Therefore, the final parameters chosen for MLP are one hidden layer with 90 nodes. The model is trained again with the chosen parameter, and its performance is shown in Table 5.

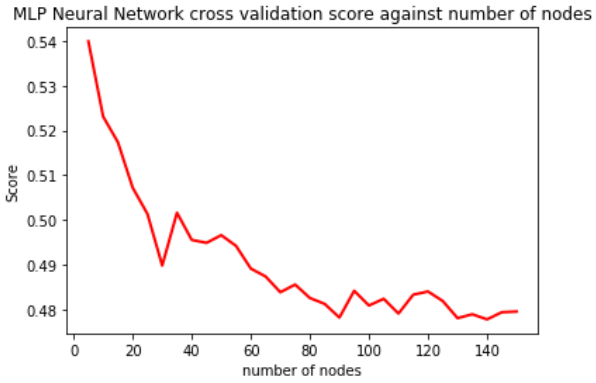

Figure 4. Cross-validation curve of Multilayer Perceptron: number of nodes

Training Error	Test Error	Generalisation Error
0.42779	0.48027	0.05248

Table 5. Performance of MLP with one hidden layer of 90 nodes

The result shows slight improvement from the ridge regression model. It was able to give a smaller training error and similar generalisation performance.

6. Proposed Solution

Table 6 shows the performance of different models that has been tested in this coursework.

Model	Training Error	Test Error	Generalisation Error
MLR	0.63017	0.64615	0.01598
PLR	0.27844	2.78799	2.50955
RR	0.45646	0.50823	0.05177
MLP	0.42779	0.48027	0.05248

Table 6. Performance of MLP with 1 hidden layer of 90 nodes

Out of the 4 different models that were tested, Multilayer Perceptron has the best performance. The proposed solution is therefore Multilayer Perceptrons, with 1 hidden layer of 90 nodes.

7. Conclusion

In conclusion, non-linear algorithms such as multilayer perceptron is much more powerful than linear algorithms. However, one needs to have access to great computational power in order to use the model effectively. Even so, by just tweaking two hyperparameters (number of hidden layers and number of nodes), a better result can be achieved compared to ridge regression, which is the best performer of all the linear models tested. Ridge rigression however is still a valid alternative to multilayer perceptron for this particular problem, as it does offer similar in-sample and out-of-sample performance.

From the coefficients in Table 2, we can see that some of the features does not contribute much to the result. This prompts us to consider feature selection, where features that are not important, or have a low correlation to the output can be discarded. This allows us to implement models that are more complex with less computational time.

As mentioned at the beginning of the report, the dataset is split into training and test data. One possibility to further improve the performance of the model is to obtain the parameters that has achieved the best performance when training set is used, and use these parameters to train model with the entire dataset. This is safe as the generalisation property has been checked when choosing parameter, and will result in a model that is more accurate in predicting real life cases.

8. Pledge

I, Zheng Yang Lee, pledge that this assignment is completely my own work, and that I did not take,borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

References

[1] György, András (2018). EE3-23: Introduction to Machine Learning, Lecture 2 (p. 8-14). Imperial College London.

[2] György, András (2018). EE3-23: Introduction to Machine Learning, Lecture 3 (p. 36-43). Imperial College London.

[3] Mikolajczyk, Krystian (2018). EE3-23: Introduction to Machine Learning, Lecture 4 (p. 6-12). Imperial College London.

[4] Mikolajczyk, Krystian (2018). EE3-23: Introduction to Machine Learning, Lecture 5. Imperial College London.