
Introduction to Machine Learning and Data Mining - Project 2

Authors

Ion Chetraru - s204710

Irene Abigail Sotomayor Munoz - s205720

Zihao Fu - s202533

Distribution of work	Ion	Abigail	Zihao
1. Regression part a	40%	60%	-
2. Regression part b	65%	35%	-
3 Classification	-	-	100%
4.1 Discussion - regression	40%	60%	
4.2 Discussion - classification	45%	-	55%
5. Problem 1 and 6	-	-	100%
6. Problem 2 and 5	-	100%	-
7. Problem 3 and 4	100%	-	-

1 Regression part a

For the regression model part we are interested in predicting the value of *ldl* based on all the other attributes. To accomplish this we first need to do some data transformation in the original data. In *report 1* we transformed the *famhist* attribute into binary data where 1 corresponds to *Present* and 0 corresponds to *Absent*. However in this report and in our model we will not use binary values and therefore the data set to be used for the regression part will not contain the *famhist* and the *chd* attributes. At the same time since *ldl* is our attribute of interest we will remove it from the data set as well. Moreover, the data set will be standardized by subtracting the mean and dividing by the standard deviation.

We are interested in finding a regularization factor so that we can control the complexity of our model and so preventing data over-fitting that leads to a high generalization error. We have decided to use a range of $\lambda = [10^{-4}, 10^7]$ with a logarithmic step and a 10-fold cross validation was performed in order to get an evolution of Test Error and Training Error with regards to the regularization factor which allows us to choose an optimal value of regularization factor for our model. This can be seen in the figure below.

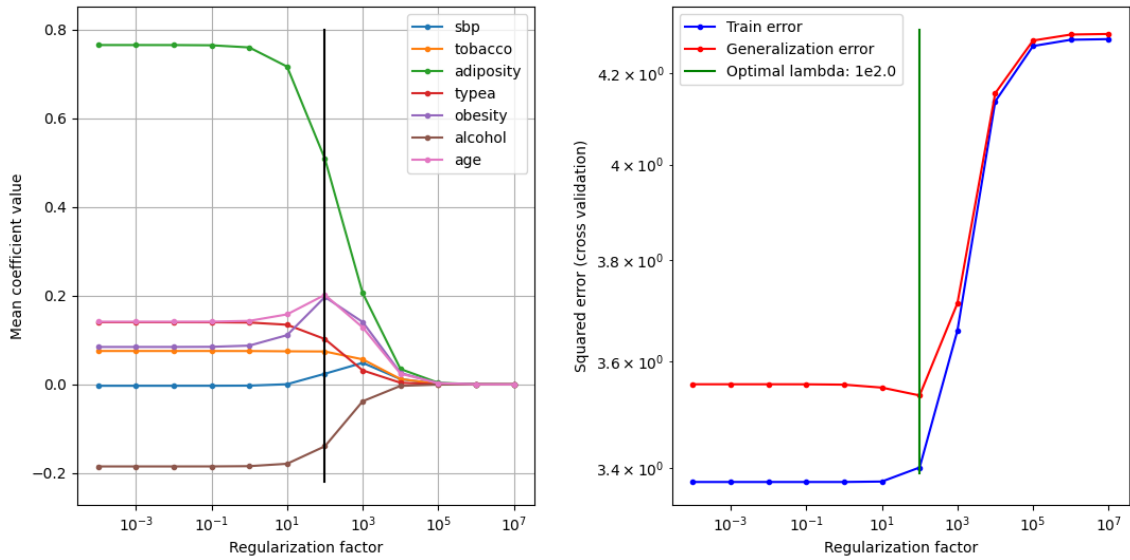


Figure 1: Regularization Factor

In Figure 1 we can see that the optimal lambda is that which correspond to the lowest generalization error. The graph on the right-side thus indicates that the optimal lambda for our model is $\lambda = 100$.

If we have a look to the graph on the left-side in Figure 1 we can see that for the optimal lambda value ($\lambda = 100$), if we were to predict a new observation the attributes which will have the most contribution in our model would be *adiposity*, followed by *age* and *obesity*. This results make sense since in report 1 we saw that there is a correlation between *ldl* and *obesity*, *adiposity*, *age*.

2 Regression part b

In this part we have implemented a two level cross-validation with $K_1 = K_2 = 10$ for model selection on the three different models. The models tested are baseline, linear regression and an artificial neural network. In the two latest we are interested in finding an optimal value of regularization strength λ_i^* and an optimal value of number of hidden layers h_i^* , respectively.

For this reason we will test a range of values of regularization strength and a range of numbers of hidden units in the inner loop of the model selection algorithm. The range chosen for the values of regularization strength is as mentioned above $\lambda = [10^{-4}, 10^7]$ with a logarithmic step and for

number of hidden units the range is $h_i = [1, 15]$ with a linear step = 1. Once done that, we will find the test error based on optimal values found in the inner loop and compute the test error for the outer folds. This data is shown in the following table.

Table 1: Two-level cross-validation to compare the three models

Outer Fold i	Baseline E_i^{test}	Linear Regression		ANN	
		λ^*	E_i^{test}	h_i^*	E_i^{test}
1	2.273	100	1.688	1	1.694
2	5.401	100	4.862	1	4.749
3	3.463	10	2.314	1	2.405
4	4.033	100	3.351	1	3.186
5	3.119	100	2.435	1	2.346
6	1.791	10	2.013	1	2.102
7	6.510	100	5.274	1	5.249
8	4.113	10	3.890	1	3.890
9	7.332	10	6.894	1	6.807
10	3.421	10	2.657	1	2.670
Mean	4.146	55	3.538	1	3.510

Table 1 shows the test error for all the three different models, the optimal λ_i^* and optimal number of h_i^* for the i 'th outer cross-validation fold. In the bottom of the table we have chosen to show the mean of the columns. This mean allows us to comment on the fact that the biggest test error happens in the Baseline model whereas for the Linear Regression model and the Neural Network this test error decreases compared to the Baseline model however it does not vary a lot if compared between them.

As for the number of hidden units we can see that the optimal value is clearly $h = 1$ as it was chosen all the times, and for lambda we can see there is a variation between $\lambda = 10$ and $\lambda = 100$. However, since the lowest test error corresponds to the regularization factor $\lambda = 100$, we choose it as our optimal lambda. For each outer cross validation, we use the same test and train data for the three models in order to allow statistical analysis afterwards.

Now we would like to see if there is a significant difference between models. Thus we will perform a pairwise comparison and obtain their p -values and confidence intervals in order to talk about the models. We would expect from the statistical analysis that the models that are compared to the baseline are significantly different since the baseline holds the highest test error according to Table 1. The results are shown in the table below.

Table 2: Statistical Analysis

	$E_{gen}^{LR} - E_{gen}^{ANN}$	$E_{gen}^{LR} - E_{gen}^{Base}$	$E_{gen}^{ANN} - E_{gen}^{Base}$
p -value	0.31678	0.00213	0.00244
Confidence Interval	$[-0.686, 0.210]$	$[0.304, 1.331]$	$[0.322, 1.454]$

Table 2 shows the different p -values and the confidence intervals for the pairwise statistical analysis between the three different models. In order to be able to analyse this table is important to know what is it that we are looking for. For this reason we would like to mention that our significance level $\alpha = 5\%$. Thus, if the p -value is smaller than 0.05, the null hypothesis can be rejected and conclude that a significant difference does exist between the models. We can see that this happens in two occasions which is the analysis between the baseline vs linear regression model and

between the baseline vs neural network, as expected. In the comparison between Linear Regression and neural network the p -value was greater than 0.05 which means that we can accept the null hypothesis and cannot conclude that there is a significant difference between the models.

Another way to confirm this null hypothesis is to look at the confidence intervals of the test error of the models and see whether 0 is contained within the interval or not. We have chosen to plot the confidence intervals and a line through the value 0 to see which interval crosses it. If zero is not contained within the interval then it means that the null hypothesis is rejected and we can conclude that there is a significant difference between models and vice versa.

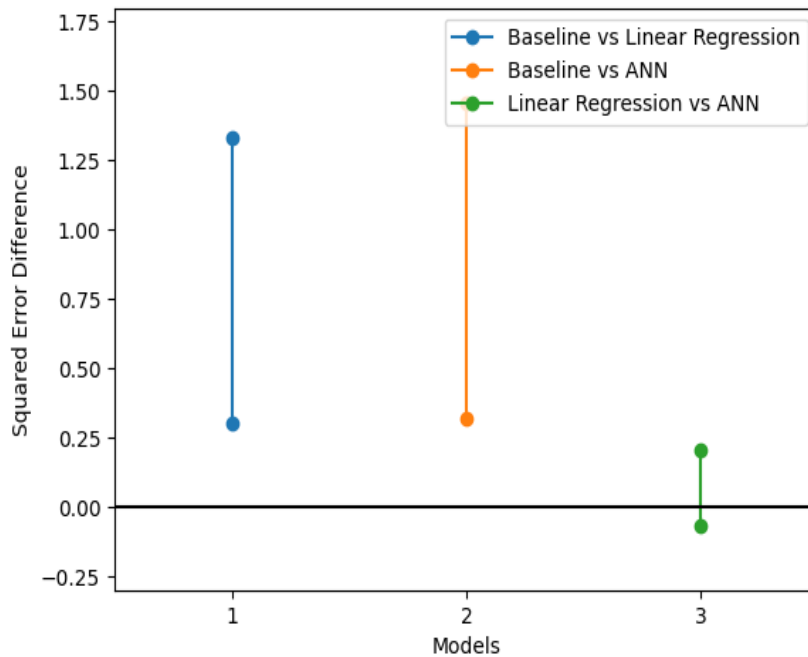


Figure 2: Confidence Intervals

Figure 2 shows and confirms our discussion about the p -values since zero is not contained by models 1 and 2 - See x-axis.

3 Classification Part

For the regression model part we are interested in predicting the value of *chd* based on all the other attribute. *Chd* is the attribute which talks about the response to a coronary heart disease, and it is something worth predicting. In the original data-set, *chd* is a binary data. Thus, the the classification problem we have chose to solve is a binary classification problem.

We have chosen ANN as our method2, in this ANN, we use quasi-Newton to optimize weights because it performs better on small data-sets [1]. We used 0.0001 as regularization parameter, and relu as the activation function.

Table 3: Two-level cross-validation to compare the three models on classification

Outer Fold	Baseline	Logistic Regression		Method2	
i	E_i^{test}	λ^*	E_i^{test}	h_i^*	E_i^{test}
1	0.2340	100	0.2340	1	0.2580
2	0.3404	100	0.3191	3	0.3871
3	0.3261	100	0.2174	5	0.3548
4	0.2609	100	0.4348	10	0.3441
5	0.4130	100	0.4783	5	0.3226
6	0.3261	100	0.3043	3	0.3548
7	0.2601	100	0.2174	1	0.3226
8	0.4130	100	0.2609	10	0.3871
9	0.4348	100	0.2391	5	0.2796
10	0.4565	100	0.2609	3	0.3978
Mean	0.3466	100	0.2966	4.6	0.3409

Table 3 shows the error rate $E = \frac{\{\text{Number of misclassified observations}\}}{N^{test}}$ for all the three different models. It also shows that in the logistic regression model, the regularization factor is chosen as $\lambda = 100$ in all of the outer folds. Four different numbers of units of ANN (method2) is 1, 3, 5 and 10; 1,3,5 units ANN just used one layer, and the 10 units ANN used two layers. Normally, the more units and deeper layers will help the model perform better, however, it seems not work on our ANN model. The baseline model computes the largest class on the training data, and predict everything in the test-data as belonging to that class. Comparing the results of logistic regression and baseline, we can find the logistic regression model get a obviously better result. Comparing logistic regression and ANN method, ANN even got a worse result. It is out of my mind. After digging into the result, I found the ANN with 10 units and 2 layers did not show better performance than the ANN with less units. Thus, I think the reason is that the data-set we selected is kindly small, complicated model can get over-fitted easily. Hence, the ANN model did not show much more advantage than logistic regression model in our data-set.

Table 4: Statistical Analysis

	$E_{gen}^{LR} - E_{gen}^{base}$	$E_{gen}^{LR} - E_{gen}^{method2}$	$E_{gen}^{method2} - E_{gen}^{Base}$
p -value	0.06	0.05	0.01
Confidence Interval	[0.01, 0.11]	[0.02, 0.08]	[-0.04, 0.06]
$\hat{\theta}$	0.02	0.06	-0.04

Table 4 The McNemar's test was used to estimate the difference in performance $\theta = \theta_A - \theta_B$ between model A and model B. If $\theta > 0$ then the model A is preferable over model B. However, looking into the p -value, we found p -value for logistic regression model and baseline is 0.06, which is larger than α which is 0.05. Thus we cannot do not have a strong enough statistical evidence to conclude the logistic regression model is better than baseline. Also, we do not have sufficient evidence to conclude the logistic model is better than ANN model. Thinking about why is p -value is too strong, we came up with two possible reasons to explain: the first one is our data-set is too small to achieve a classification task [4]. The other one is that we cannot predict the value of chd based on the all attributes of the data-sets actually or the model we selected is not suitable.

The logistic regression model we trained for classification using 100 as the value of λ . For ideal logistic regression model for binary classification, we supposed the y (the target of the data set) follows a continuous distribution between 0 and 1, the closer value y is to 1, the more likely it is to represent a positive class; the closer to 0, the more likely it is to represent a negative class. But

in the fact we only got 1 or 0 for the value of y (in our data it is the value of chd). So we treated our y is the ideal value with some noises. Thus, our target function is like:

$$\text{target function } f(\mathbf{x}) = P(+1 | x) \in [0, 1]$$

Table 5: ideal data vs actual data

Ideal data	Actual data
$(\mathbf{x}_1, y'_1 = 0.913 = P(+1 \mathbf{x}_1))$	$(\mathbf{x}_1, y_1 = 1 \sim P(y \mathbf{x}_1))$
$(\mathbf{x}_1, y'_1 = 0.623 = P(+1 \mathbf{x}_1))$	$(\mathbf{x}_1, y_1 = 1 \sim P(y \mathbf{x}_1))$
$(\mathbf{x}_1, y'_1 = 0.276 = P(+1 \mathbf{x}_1))$	$(\mathbf{x}_1, y_1 = 0 \sim P(y \mathbf{x}_1))$

And then we need to choose a hypothesis to close to this target function and limited the result between 0 and 1. We decided to choose sigmoid function, which is $\theta(s) = \frac{1}{1 + e^{-s}}$, so for the logistic regression, the final hypothesis is like:

$$h(\mathbf{x}) = \theta(\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$

Then we need to find out a w to make the hypothesis $h(\mathbf{x})$ close to the target function $f(\mathbf{x})$ to generate the same y . We calculated likelihood of h , because $h(\mathbf{x}) = \theta(\mathbf{w}^T \mathbf{x})$ satisfied a property: $1 - h(x) = h(-x)$. So the likelihood is:

$$\text{likelihood}(h) = P(\mathbf{x}_1) h(+\mathbf{x}_1) \times P(\mathbf{x}_2) h(-\mathbf{x}_2) \times \cdots \times P(\mathbf{x}_N) h(-\mathbf{x}_N)$$

Our task is to maximum $h(x)$, and we add a \ln to change from product to summation, and add a minus to make the task as a minimize task. And finally, the task is to minimize the loss function shown as below to figure out w to get the logistic regression model:

$$\text{err}(w, \mathbf{x}, y) = \ln(1 + \exp(-y\mathbf{w}\mathbf{x}))$$

4 Discussion

4.1 Regression

In the regression model part we have learned how to apply and compare different models. We have also learned the importance of having a baseline model since it acts as a threshold for our data analysis which will help us compare and talk about how other models perform in our data set. We also learned the importance of performing cross validation. This 'technique' help prevent over fitting of the model since it estimates the error over each test set and thus decide which parameters will work best for the models - Regularization Strength and Hidden Units in our case.

We learned the importance of the regularization strength parameter in the linear regression model. If chosen correctly this parameter helps prevent over fitting of data since it reduces the variance of the model without 'increasing' the bias. However it is important to know when to stop increasing the value of λ since after a point this parameter will actually increase the bias and thus the model will be under-fitted for that reason is important to know how to carefully select the lambda parameter as we have done in regression part a and b.

As for the neural network model we know from theory that choosing the right number of hidden layers allow us to train a neural network faster and without over fitting however in our case we

have seen that $h = 1$ was chosen to be the adequate number of hidden layers. This could probably be caused by the size of our data set. We think that the size of our data is not big enough and for that reason it might not be necessary to have a large number of hidden units as our regression problem is not that complex and if the number of hidden units increased then it would probably increase the accuracy of the model however it might over fit the model without mentioning that the time to train the neural network would increase significantly.

As four previously analysis performed on our data set we have found some that concerned classification models, which we will discuss in the next subsection.

4.2 Classification

The book "Element of Statistical Learning"[2] and the paper "Exploring Machine Learning Techniques for Coronary Heart Disease Prediction"[3] explores the same classification problem for the dataset we have worked on, and the same binary variable is predicted. In the book, the authors fitted a logistic-regression model by a maximum-likelihood, where as a result, the attributes *sbp* and *obesity* were found non-significant, by *analysis of deviance*. In another logistic regression fit, the nonlinearity of the attributes was explored, and as a result, using backward step wise deletion process, some attributes were dropped. It's worth mentioning that the attributes *sbp* and *obesity* were included in the second model.

The paper"Exploring Machine Learning Techniques for Coronary Heart Disease Prediction"[3] use the whole data-set to do the classification work by SVM, MLP, KNN and Logistic regression model. It also mentioned that a mixed model combined with Bidirectional Long-Short Term Memory(BiLSTM) and CNN would get better performance.

Due to the fact we only worked on a small sample of the original data-set, we did not use the result of the paper[3] as a benchmark to compare with. However, we also learned some things from the data-set. One interesting thing is, when we compared the result of different scaling methods, we found out the normalization has a better error rate result than standardization for both ANN and logistic regression model. It told us for different data-set and model, choosing a suitable scaling method is also important. In addition, in the statistical evaluation part, we found the p -value is too strong. We tried to fix it into a smaller value, but we failed. It seems due to our data-set is too small[4], the scale of data-set sometimes is the crucial factor of a classification task.

5 Exercises

Exercise 1

The correct answer is **C**. We compute the false positive rate and look into the chart, only Prediction C corresponds to the right answer. and true positive rate to get ROC. When we consider false positive rate and look into the chart, only Prediction C corresponds to the right answer.

Exercise 2

In this question we are asked to find the impurity gain from a classification tree based on Hunt's Algorithm, using the classification error impurity measure. To do this we first compute the number of observations in the root and then we compute the class error based on how the tree was split in this case $x_7 = 2$.

$$n(r) = 135$$

$$n(v_1) = 1$$

$$n(v_2) = 134$$

$$I(r) = 1 - \frac{37}{135}$$

$$I(v_1) = 1 - \max\left(\frac{0}{1}, \frac{1}{1}, \frac{0}{1}, \frac{0}{1}\right) = 1 - 1 = 0$$

$$I(v_2) = 1 - \max\left(\frac{37}{134}, \frac{30}{134}, \frac{33}{134}, \frac{34}{134}\right) = 1 - \frac{37}{134}$$

Plugging this values into the formula of impurity gain

$$\Delta = 1 - \frac{37}{135} - \frac{1}{135} \cdot 0 - \frac{134}{135} \cdot \left(1 - \frac{37}{134}\right)$$

$$\Delta = 0.00741$$

Correct Answer: **C**

Exercise 3

$$n_h = 10$$

$$noAttributes = 7$$

$$n_o = 4 \text{ (4 classes)}$$

The number of parameters to be trained is equal to the number of links between the neurons, this is, $n_h \cdot noAttributes + n_h \cdot n_o = 10 \cdot 7 + 10 \cdot 4 = 110$

Correct Answer: **C**

Exercise 4

If we analyze *Congestion level 4*, we can see that it is given by the conditions $b_1 \geq -0.16$ and $b_2 \geq -3$. Since we need node A and node C to be True in order to get this boundary - A, B and C can be eliminated, because node C for each of them enforces $b_2 > 0$. In contrast, the nodes in case D creates the Classification Boundary shown in the figure. Notice that $b_1 \geq -0.76$ is being *overwritten* by node C with $b_1 \geq -0.16$

Correct answer: **D**

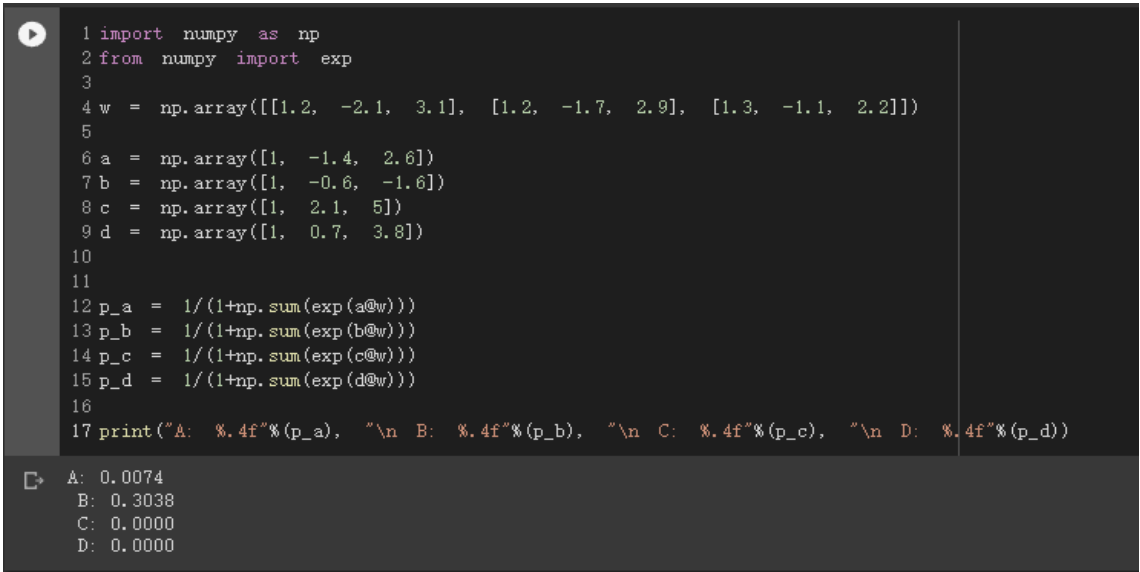
Exercise 5

To calculate the total time for creating the table we need to follow algorithm 6 from the text book. If we focus on the neural network computation. First we start in the inner loop, where we have to train the 5 different values of hidden layers and then test it. Keeping in mind that the training time takes 20ms and the testing time takes 5ms. We do this computation 4 times as this is the value of the k-inner fold. Therefore: $4 \cdot (5 \cdot 5 + 20 \cdot 5)$ To this result we have to multiply by 5 as this is the value of the k-outer fold, also we have train and test the data one last time. Therefore the time for the neural network is $4 \cdot (5 \cdot 5 + 20 \cdot 5) \cdot 5 + 5 \cdot 20 + 5 \cdot 5 = 2625$.

We follow the same procedure for the linear regression model, only this time the training part takes 8ms whereas the testing part takes 1ms. Therefore $4 \cdot (5 \cdot 1 + 8 \cdot 5) \cdot 5 + 5 \cdot 8 + 5 \cdot 1 = 945$. If we then add up both values we get the answer 3570.

Correct Answer: **C**

Exercise 6



```
1 import numpy as np
2 from numpy import exp
3
4 w = np.array([[1.2, -2.1, 3.1], [1.2, -1.7, 2.9], [1.3, -1.1, 2.2]])
5
6 a = np.array([1, -1.4, 2.6])
7 b = np.array([1, -0.6, -1.6])
8 c = np.array([1, 2.1, 5])
9 d = np.array([1, 0.7, 3.8])
10
11
12 p_a = 1/(1+np.sum(exp(a@w)))
13 p_b = 1/(1+np.sum(exp(b@w)))
14 p_c = 1/(1+np.sum(exp(c@w)))
15 p_d = 1/(1+np.sum(exp(d@w)))
16
17 print("A: %.4f"%(p_a), "\n B: %.4f"%(p_b), "\n C: %.4f"%(p_c), "\n D: %.4f"%(p_d))
```

A: 0.0074
B: 0.3038
C: 0.0000
D: 0.0000

Correct Answer: **B**

References

- [1] John E Dennis Jr and Jorge J Moré. ‘Quasi-Newton methods, motivation and theory’. In: *SIAM review* 19.1 (1977), pp. 46–89.
- [2] *Elements of Statistical Learning*. URL: <https://hastie.su.domains/ElemStatLearn/>.
- [3] Hisham Khdair. ‘Exploring machine learning techniques for coronary heart disease prediction’. In: *International Journal of Advanced Computer Science and Applications* 12.5 (2021).
- [4] David S Moore and George P McCabe. *Introduction to the Practice of Statistics*. WH Freeman/Times Books/Henry Holt & Co, 1989.