UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF ENGINEERING AND SCIENCE

UECS2053 UECS2153 UEMH3073 UEMH3163

Artificial Intelligence May 2022 Trimester

Lab 3: Supervised Learning

Practical Group: P3 – Group 8

Group Members:

Name	Student ID	Year/Trimester	Programme
Lau Yong Yang	2003578	Y3/T1	MH
Ong Kar Ming	1803674	Y4/T1	AM
Siow Wen Hao	2003860	Y3/T1	MH
Teh Shao Yi	1904381	Y3/T1	BI

Lecturer: Dr Ng Oon-Ee

Date of Submission: 06/09/2022, 11:59pm

INTRODUCTION

Supervised machine learning algorithms are designed to learn by example. The name "supervised" learning originates from the idea that training this type of algorithm is like having a teacher supervise the whole process. In this lab, we are focusing on training supervised learning algorithms to predict the number of deaths per day due to Covid-19 pandemic.

As most the supervised models are designed just for classification problems while the outcome (number of deaths) for this lab is continuous. Regression problem supervised models such as Linear Regression and Artificial Neural Network (ANN) are trained to predict the number of deaths per day due to Covid-19 pandemic. Linear Regression predicts a dependent variable (target) based on the given independent variables (features), this regression technique finds out a linear relationship between a dependent variable and the other independent variable given. Artificial Neural Network (ANN) uses the processing of the brain as a basis to develop algorithms that can be used to model complex patterns and prediction problems. The network architecture has an input layer, hidden layer, and the output layer. The hidden layer distils some of the important patterns from the inputs and passes it onto the next layer to see. It makes the network efficient by identifying only the important information from the inputs leaving out the redundant information. Then, activation function between the layers helps convert the input into a more useful output.

There are a total of 17 features (variables) that were chosen by us (**cases_new, cases_active, cases_child, cases_adolescent, cases_adult, cases_elderly, cases_pvax, cases_fvax** under cases_malaysia.csv, **beds_covid, admitted_covid** under hospital.csv, **beds_icu_covid, vent, icu_port, icu_covid, vent_covid** under icu.csv and **beds, admitted_covid** under pkrc. csv) as these features are highly correlated with the target variable which is the **number of deaths** per day due to Covid-19 pandemic.

To begin, we need to know the new daily cases and active cases. This is because if both of these are high, the mortality rate will be higher. Moreover, we also need to know the cases among different levels of people (children, adolescents, adults, and elderly) as there are 2 categories of people, children and the elderly that have low antibodies and a high possibility of getting the infection which may eventually causing deaths. Covid patients with partially vaccinated and Covid patients with fully vaccinated are less likely to get infected because the vaccine will boost the human antibody. We would also like to know the total beds in the hospital, the number of patients who stayed in hospital, ICU beds in hospital, non-portable

ventilators in ICU, portable ventilators in ICU, Covid patients admitted to ICU, Covid patients who acquire the ventilator, covid beds in the quarantine centre, and Covid patients admitted to the quarantine centre. This is because the healthcare facilities are of utmost importance for severe Covid patients to receive the necessary treatment to avoid their condition from getting worse, which will cause death.

These 17 features will be taken to determine the number of deaths 7 days after the particular date starting from **July 1, 2021, till December 31, 2021**. The target output (number of deaths per day due to Covid-19 pandemic) was not taken on the exact date but 7 days after the particular date as this would be the average time for a higher risk Covid-19 patients facing a chance of dying.

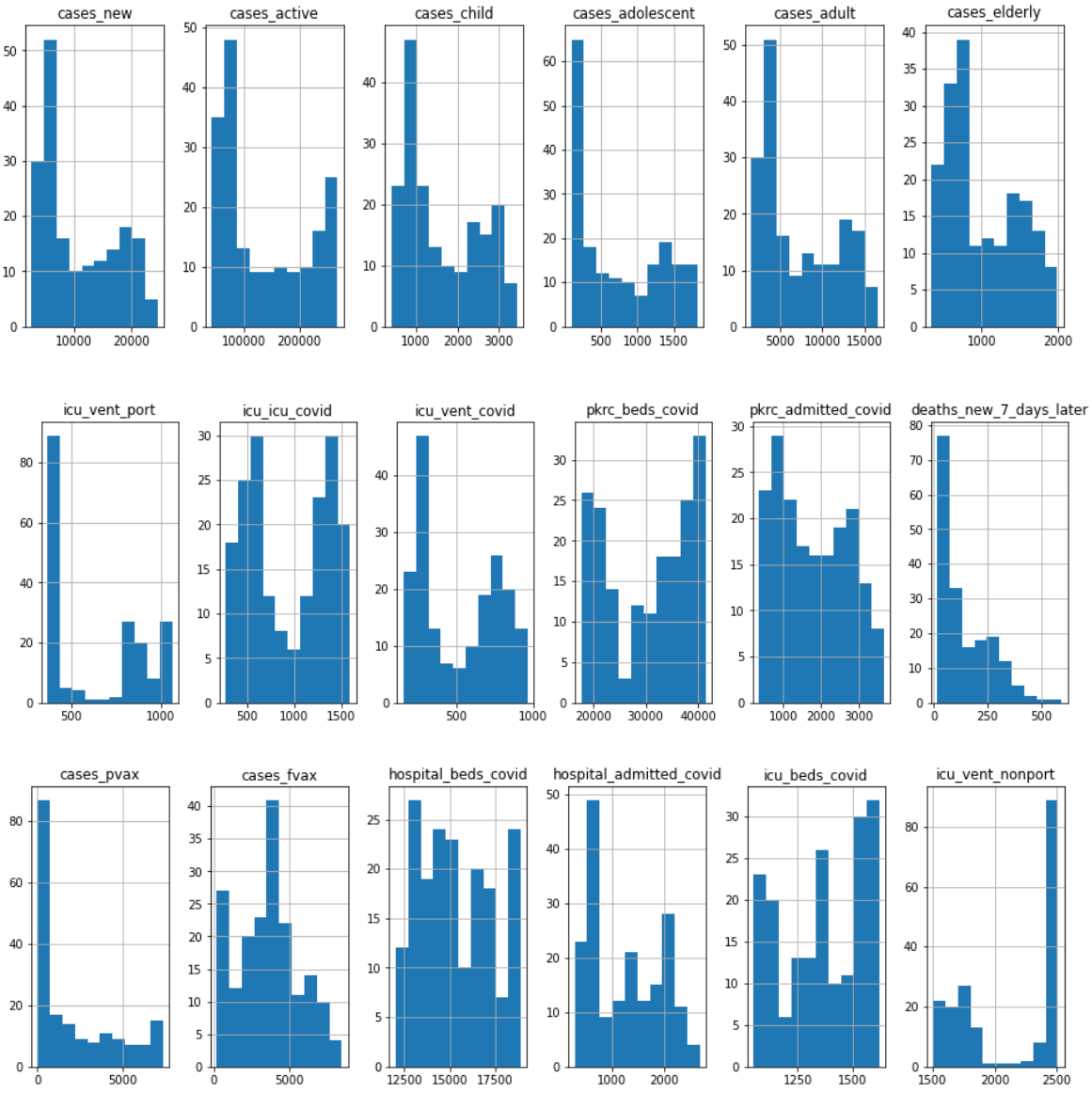

Figure 1: Bar Graph for Features and Target

RESULTS

Linear Regression Method

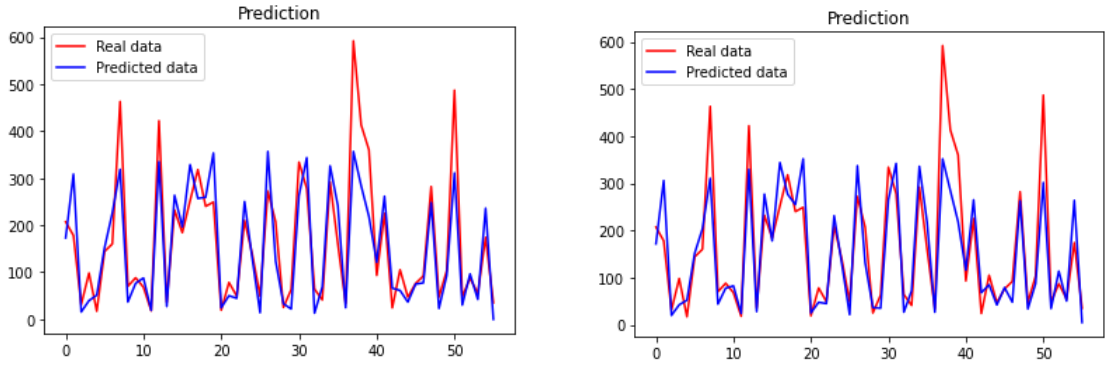

Figure 2: Prediction Graphs before and after hyperparameter tuning.

The best hyperparameters was found to be 11 with the coefficient of determination of 0.754417.

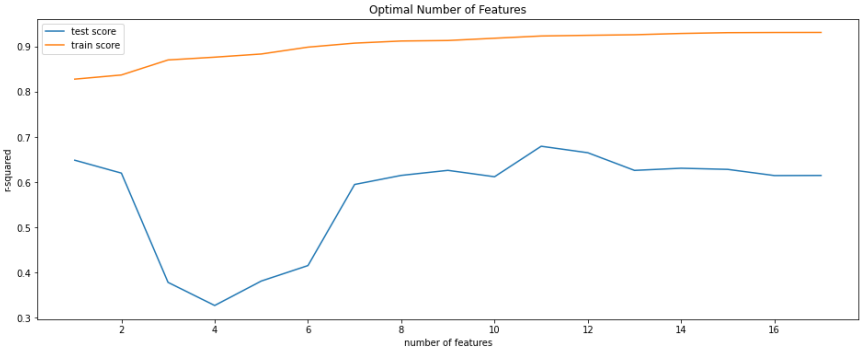

Figure 3: Cross Validation Results for Linear Regression

Artificial Neural Networks (ANN) Method

Neural network algorithms tend to have overfitting and underfitting problems. A graph is plotted to test whether this ANN model exist any overfitting or underfitting phenomenon.

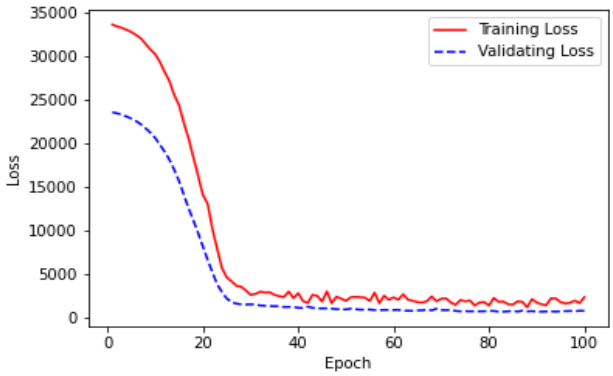

Figure 4: Graph for test overfit and underfit problems

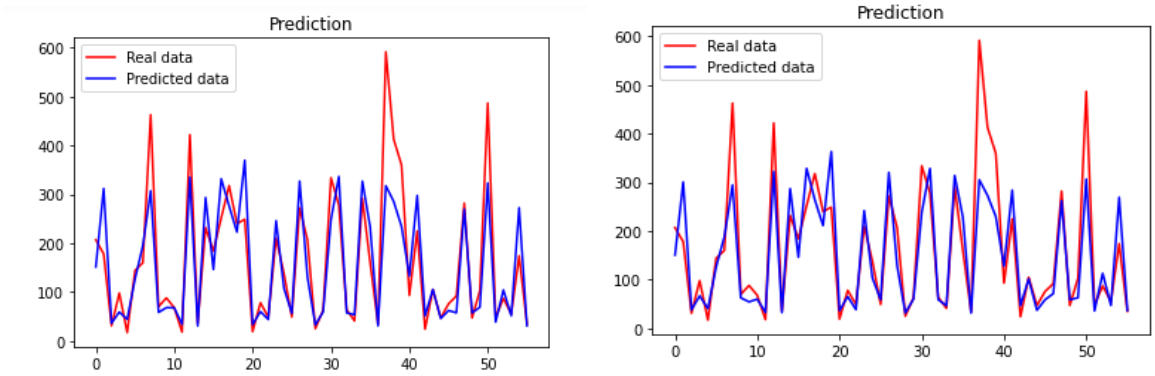

Figure 5: Prediction Graphs before and after hyperparameter tuning.

ANALYSIS

Performance Evaluation

Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MSE) are applied to evaluate the performance of regression models used. Mean Square Error (MSE) is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset. The squaring also has the effect of inflating or magnifying large errors. That is, the larger the difference between the predicted and expected values, the larger the resulting squared positive error. This has the effect of "punishing" models more for larger errors when MSE is used as a loss function. It also has the effect of "punishing" models by inflating the average error score when used as a metric. The Root Mean Squared Error, or RMSE, is an extension of the mean squared error. Importantly, the square root of the error is calculated, which means that the units of the RMSE are the same as the original units of the target value that is being predicted. Mean Absolute Error, or MAE, is a popular metric because, like RMSE, the units of the error score match the units of the target value that is being predicted. Unlike the RMSE, the changes in MAE are linear and therefore intuitive. That is, MSE and RMSE punish larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give more or less weight to different types of errors and instead the scores increase linearly with increases in error.

Mean Square Error for Linear Regression Method

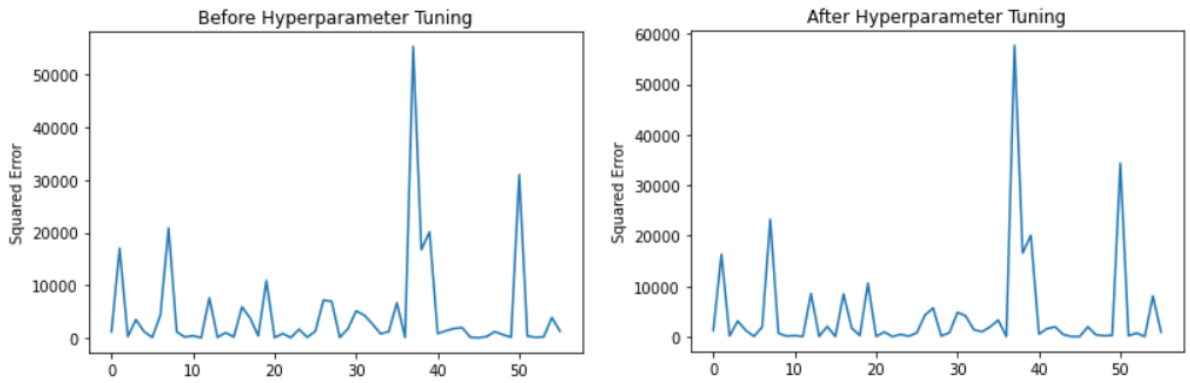

Figure 6: Graphs before and after hyperparameter tuning.

Mean square error before hyperparameters tuning: 4587.821433073013

Mean square error after hyperparameters tuning: 4559.9508071355

Root Mean Square Error for Linear Regression Method

Root mean square error before hyperparameters tuning: 67.3345874140057

Root mean square error after hyperparameters tuning: 67.52740782182816

Mean Absolute Error for Linear Regression Method

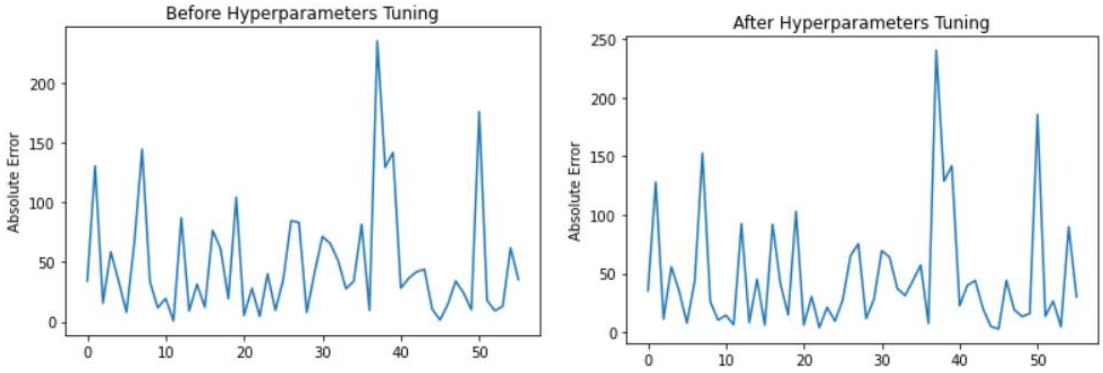

Figure 7: Graphs before and after hyperparameter tuning.

Mean absolute error before hyperparameters tuning: 48.314073334931436

Mean absolute error after hyperparameters tuning: 48.314073334931436

Mean square error for Artificial Neural Networks (ANN)

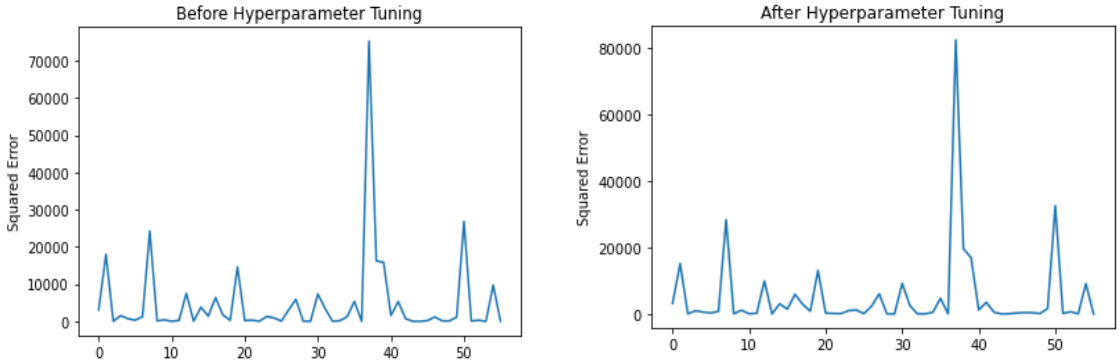

Figure 8: Graphs before and after hyperparameter tuning.

Mean square error before hyperparameters tuning: 4818.612737047222

Mean square error after hyperparameters tuning: 5098.7185329488875

Root Mean Square Error for Artificial Neural Networks (ANN)

Root mean square error before hyperparameters tuning: 69.4162281966344

Root mean square error after hyperparameters tuning: 71.40531165780938

Mean Absolute Error for Artificial Neural Networks (ANN)

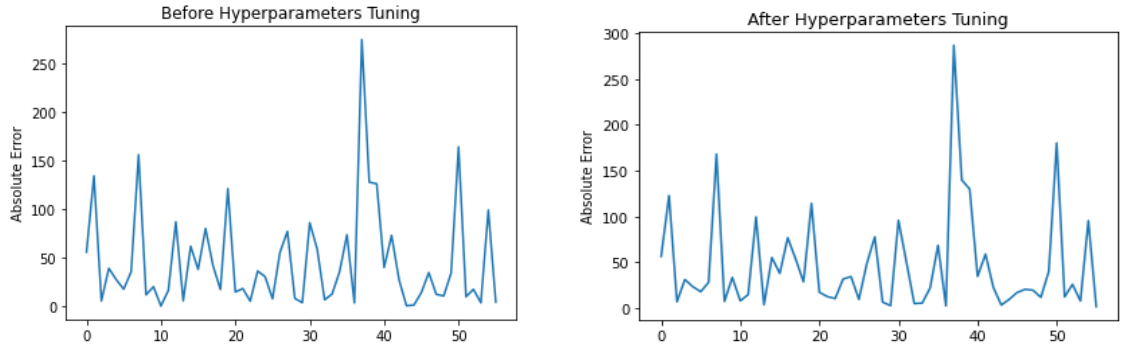

Figure 9: Graphs before and after hyperparameter tuning.

Mean absolute error before hyperparameters tuning: 45.85137483051845

Mean absolute error after hyperparameters tuning: 46.62446222986494

CONCLUSION

To compare the performance of two supervised learning techniques, we must look at the results of various metrics. We must also monitor the training speed of both supervised learning techniques.

	Linear Regression	Neural Networks
Mean Squared Error	4559.950807	5098.718533
Root Mean Squared Error	67.527408	71.405312
Mean Absolute Error	46.527755	46.624462
Model Training Speed (seconds)	0.003986	3.015996
GridSearchCV Training Speed (seconds)	1.956459	111.978318

Figure 10: Results of various evaluation metrics.

As seen from the table above, the Linear Regression model has better performance than the Neural Network model. From the view of MSE, RMSE, and MAE, the results obtained from the metrics of Linear Regression Model are all lesser than Neural Network Model. From the graph (the difference between the predicted Y values and the real Y values) obtained, the predicted value is quite close to what the real data should be. However, both models fail to predict the sudden peak in the of amount of death during a particular period. Hence, as a conclusion, both models are performing considerably fine (errors obtained are not very big) in predicting the amount of death, given 17 features that we have chosen above, but both models are not sensitive enough to sense the sudden peak of amount of death.

Moreover, we can see that Linear Regression model are very much faster in training speed than Neural Network model in term of model training speed and GridSearchCV training speed. This is expected as Neural Networks are much complex than Linear Regression model and will only get more complicated when the number of hidden layers and the number of nodes in each layer increase. Here, we are using 3 hidden layers, and 32 nodes for each layer. The GridSearchCV has very slow training speed as there are many hyperparameters chosen to be tuned to get the optimal model.

All in all, Linear Regression model can be said as a better model in predicting the number of deaths, given 17 features as listed above, than Neural Network model.

REFERENCES

Aidan Wilson (2019). *A Brief Introduction to Supervised Learning*. [online] Medium. Available at: https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590

Brownlee, J. (2019). *Keras Tutorial: Develop Your First Neural Network in Python Step-By-Step*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/.

InData Labs. (2018). *Predictive Models Performance Evaluation and Why It Is Important*. [online] Available at: https://indatalabs.com/blog/predictive-models-performance-evaluation-important.

Jahnavi Mahanta (2017). *Introduction to Neural Networks, Advantages and Applications*. [online] Medium. Available at: https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207

Nikhil Sai. (2019). *Cross-Validation with Linear Regression.* [online] Available at: https://www.kaggle.com/code/jnikhilsai/cross-validation-with-linear-regression [Accessed 6 Sep. 2022].

Researcher, M.S., PhD (2020). *Simple Guide to Hyperparameter Tuning in Neural Networks*. [online] Medium. Available at: https://towardsdatascience.com/simple-guide-to-hyperparameter-tuning-in-neural-networks-3fe03dad8594.