ECE 9603/9063B – DATA ANALYTICS FOUNDATIONS

Assignment 2

SAHANA CHAKRAVARTY

Table of Contents

1. Introduction

- **Description of the selected problem**

The purpose of this report is to analyze and prediction of the resale price of different types of apartments in Singapore using different models.

The selected prediction problem is to find the resale price of apartments in Singapore with respect to town, flat_type, street name, storey_range, floor_area_sqm, flat_model and lease_commence_date and block. This means that the column ‘resale price’ is the target variable and all the other columns except 'month' are predictor variables.

New features such as ‘block’ has been considered in this problem.

The factors that contribute positively to the predictability of the resale price are that the above columns are relevant, consists of 45000 number of rows which is quite large and the forecasts do not affect the resale price of the apartments. Also, the forecasting problem is “normally assumed” and is expected to continue in the similar fashion in the future.

- **Description of available data (attributes, context, quantity…) indicating attributes used.**

The dataset used for the problem is the same as the one used in Assignment 1.
The source of the data is Singapore Housing and Development Board, December 8, 2016 (Singapore Open Data Licence).
The 10 attributes are month, town, flat_type, block, street name, storey_range, floor_area_sqm, flat_model, lease_commence_date and resale_price.
The context of this data holds that the resale apartment prices depend on the neighborhood, street and size of the flat. The dataset consists of a decade-long listing.
In the selected forecasting problem, the column ‘resale price’ is the target variable and all the other columns except 'month' are predictor variables.
In the selected model, Outliers are removed with IQR method. Also, encoding of categorical values are done along with standard scaling. Data normalization was not required in the data set.

2. Description of the selected Neural Network Model

The selected models for the problem are **Multi-variable Linear Regression** and **Feed Forward Neural Network.**

1. **Multi-variable Linear Regression**

Multivariate linear regression is a supervised machine learning algorithm that involves multiple data variables for analysis. It consists of one dependent variable and multiple independent variables. The prediction of output is done on the basis of multiple independent variables. It tries to find out how factors in variables respond simultaneously to changes.

The generalized equation for the multivariate regression model is as follows:

$y = \beta 0 + \beta 1.x1 + \beta 2.x2 + ... + \beta n.xn$

where n represents the number of independent variables, β0~ βn represents the coefficients and x1~xn, is the independent variable.

2. **Feed Forward Neural Network**

For the selected problem, I have used **feedforward neural network** and tuned it with different hyperparameters.
A **feedforward neural network** is an artificial neural network which moves in only forward direction from the input nodes to the output nodes through the hidden nodes. Outputs of the model are not fed back to the model.
The basic kind of neural network is a single-layer perceptron network. It consists of a single layer of output nodes and the inputs are directly fed into the outputs through a series of weights.
Multi-layer perceptron consists of multiple layers of computational units which are interconnected in a feed-forward way. Every neuron in one layer has directed connections to the neurons of the subsequent layer.
The most popular form of multi-layer network is back-propagation where the output values are compared with correct answer to compute the value of some predefined error-function. Using different techniques, the error is fed back, and the algorithm adjusts the weights of each connection to reduce the value of error function.
However, one of the significant disadvantages of backpropagation is the **Vanishing Gradient Problem.**

In this problem, when more layers using certain activation functions are added to neural networks, the gradients of the loss function approach zero and makes the network hard to train.

3. Description of the Neural Network Model and properties using in tuning

In the selected FFNN model, resale price is the dependent variable while town, flat_type, street name, storey_range, floor_area_sqm, flat_model, lease_commence_date are the independent variables. An extra feature 'block name' is added to the model.
The model was trained with samples of around 45000 with 80% of the total dataset was used for training, the rest was used for testing.
At first, outliers are removed from the data using IQR method and then Label Encoding was applied to categorical columns such as town, flat_type, street_name, storey_range, flat_model and block. Data was then split into 80:20 ratio between train and test. These train and test data are then standardized using the scikit-learn object StandardScaler and then they are fit and transformed according to the applied scalar function.
After this, the Keras tuner package is applied to get the best model. The sequential model API is used to create deep learning models in which an instance of sequential class is created, and layers are added to it. Since FFNN is a fully connected layer, Dense function is used to compute with both ReLu and Leaky_Relu. Dropout is added in a range of .02 to .04 and 'Adam' is used as an optimizer to compile the model with a loss of mean_squared_error. The starting value of number of layers used are 5 and are

trained till 10 layers are reached. The number of neurons or units has a minimum value of 8 and a max value of 512 with a step value of 32. To avoid overfitting, early stopping is applied with a total epoch of around 100. As a result, in my model only run 14 epochs were run until the algorithm found the best fit. For predictions, the train predictions are inverse transformed and plotted to do comparison.

For evaluation of the FFNN model and Linear Regression model, the metrics considered are Root Mean Squared Error, mean_squared_error and mean_absolute_error.

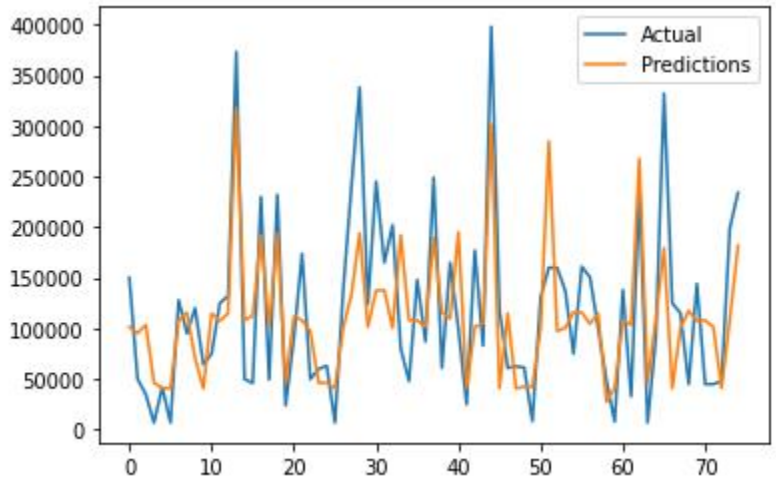
4. The Comparison of results obtained with different algorithms

The Comparison of results obtained from the selected algorithms are

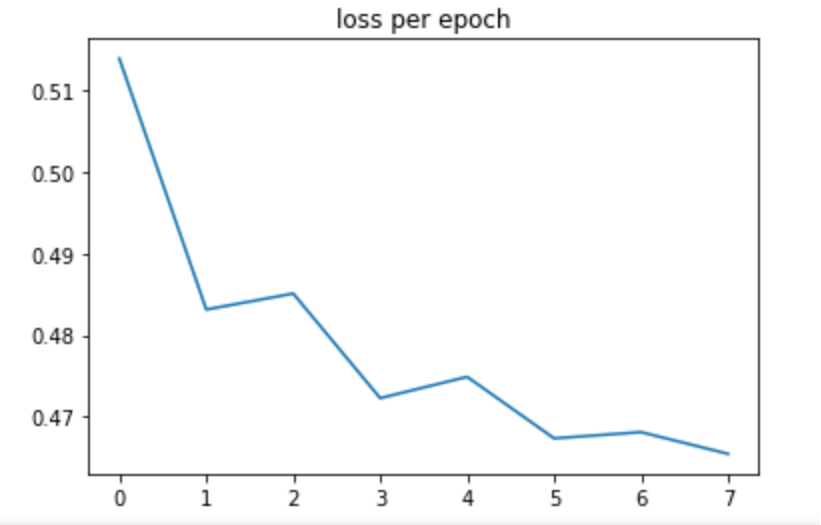
Algorithm	RMSE	MAE
Linear Regression	107372.84	80258.44
FFNN	76724.64	58527.38

From the above results we can see that the FFNN has outperformed Linear Regression Model in both MAE and RMSE error validation.

Further, below graph shows how FFNN model is able predict with respect to the actual values:



Also, it can be seen in the below graph that after training the model, for 7 epochs the loss decreased to below .25.



5. Reference

https://en.wikipedia.org/wiki/Feedforward_neural_network
https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/
https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7
https://en.wikipedia.org/wiki/Multilayer_perceptron