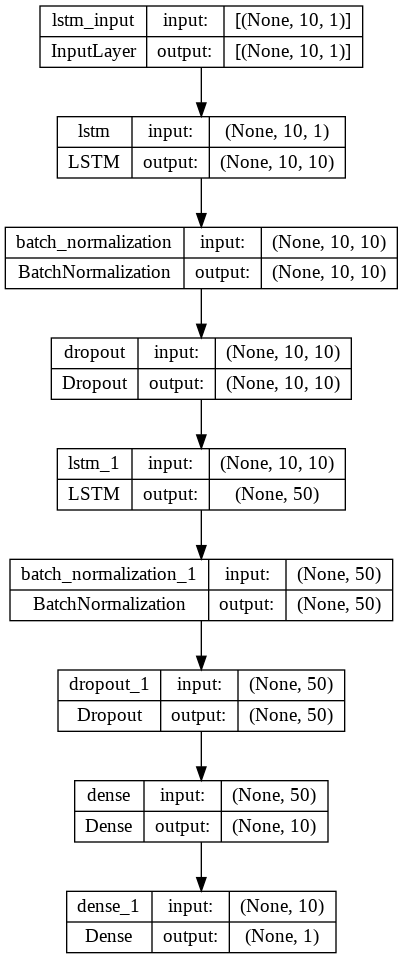
**Homework 4**

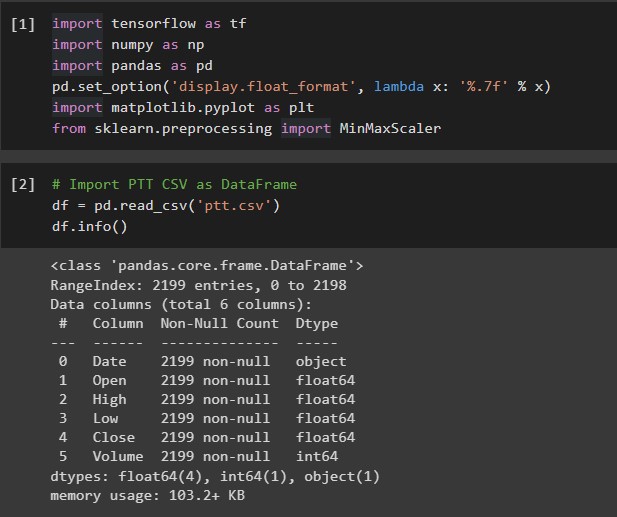
**Diagram of network and parameter used**



A picture containing text, device

Description automatically generated

**Source code**

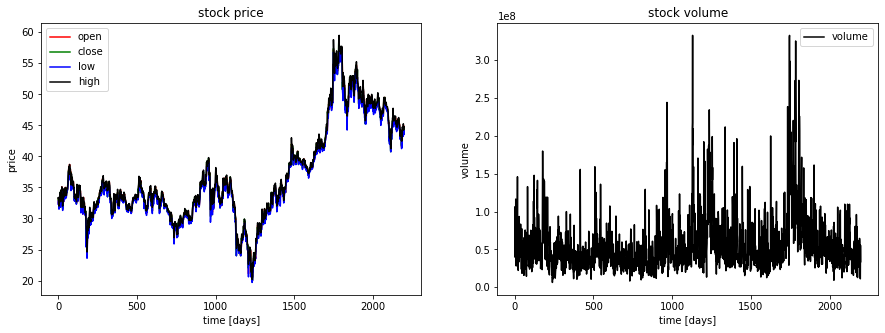


[1] Import libraries, I also set pandas float format to 7 digits.

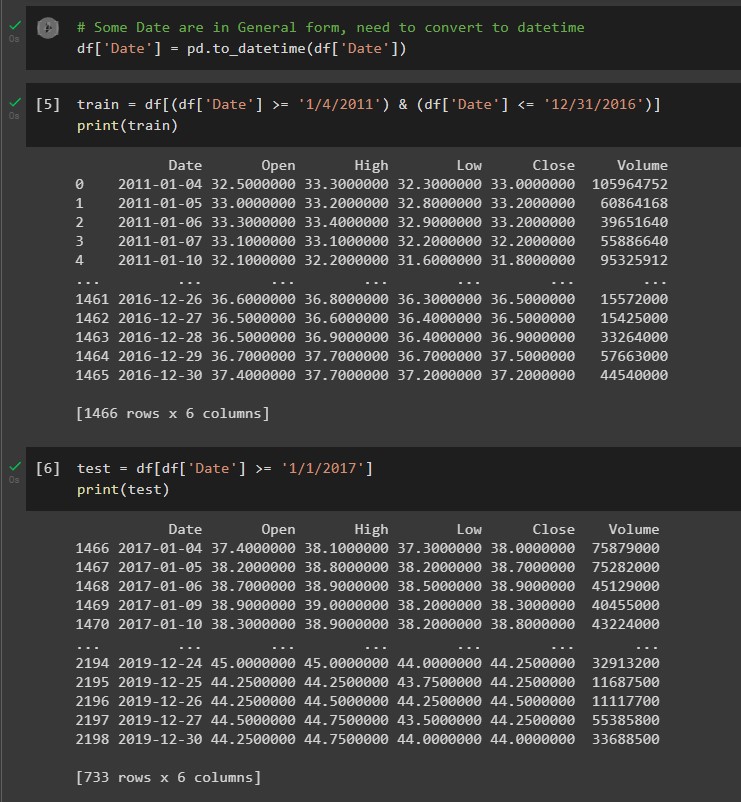
[2] Import ptt.csv into DataFrame and check the information of it.



Next, I visualized the stock price as a graph, I also visualized stock volume.



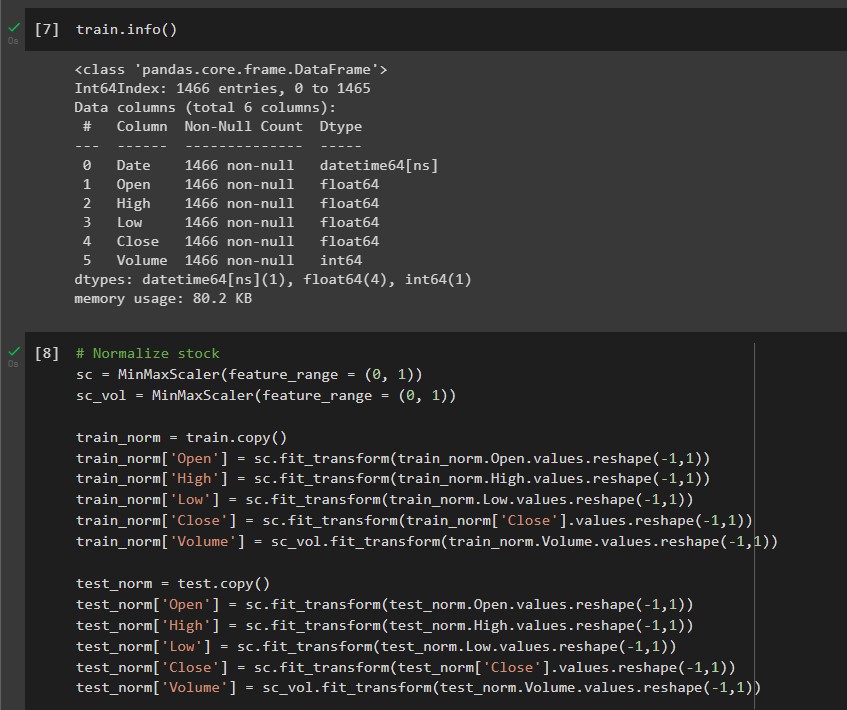
Here is the result of the aforementioned Pyplot code.



[4] Some data in the Date column are in general form (in Excel), I converted the whole column into datetime datatype.

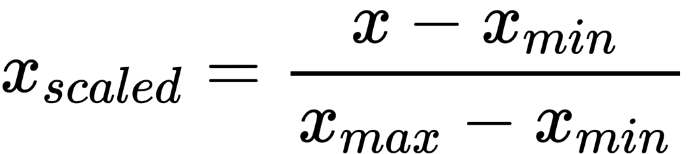
[5] I separated the data into train (2011-2016) dataset.

[6] I separated the data into test (2017-2019) dataset.

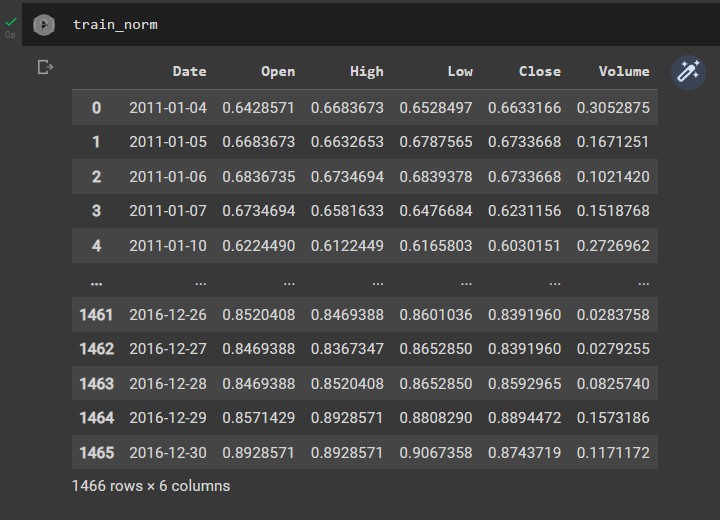


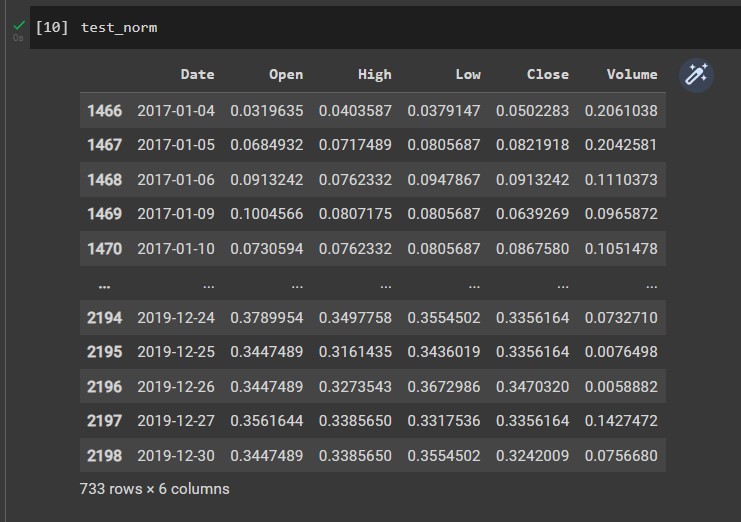
[7] Check if new Date column is in datetime format already or not, it is.

[8] Data normalization using Scikit-learn’s MinMaxScaler, I separated the scaler into sc and sc\_vol. Because the volume data is huge number compared to the prices. So, if we performed inverse transform later, the inversed outcome will be wrong.

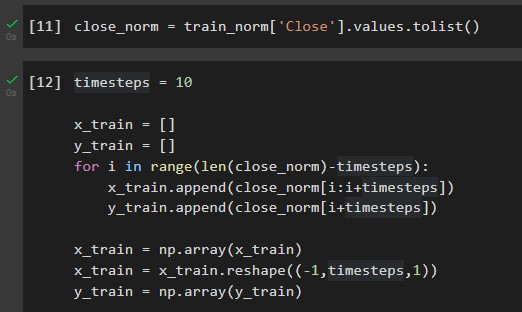


Here is the formula of MinMaxScaler.

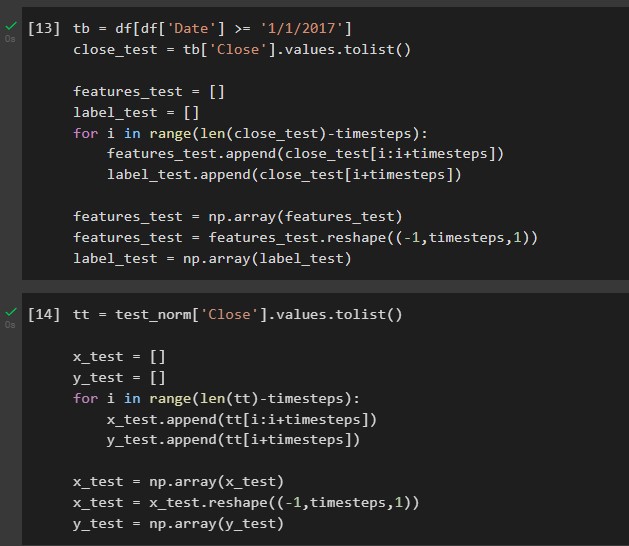




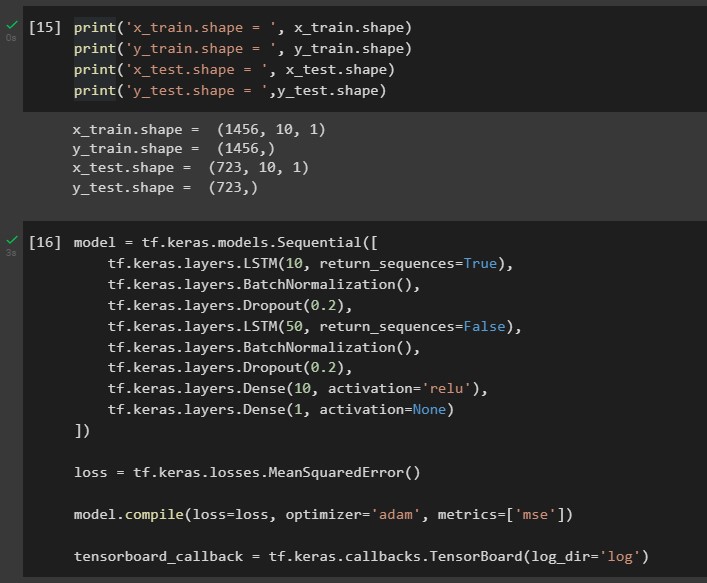
Next, I check the new train and test data that are already normalized.



Next, use the dividing windows timesteps as shown in the class. The time steps is 10, and separated into x\_train (features) and y\_train (label).

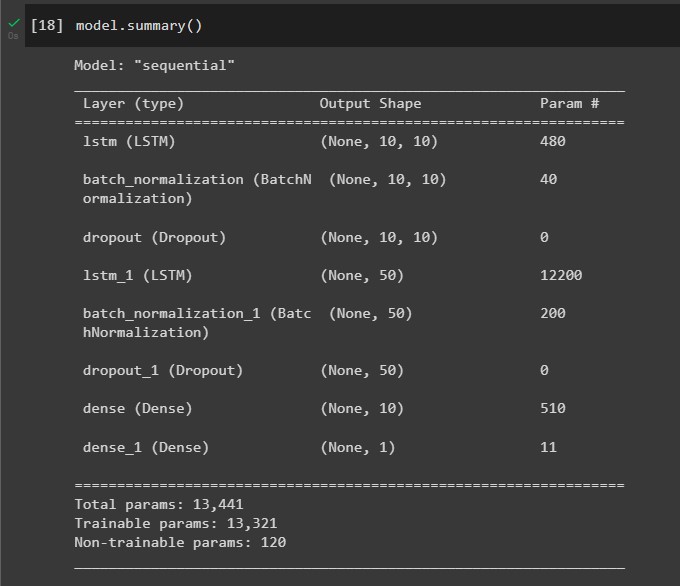


Do the same for testing set. But I did it two times, first is for the raw test data, it is for calculations later. Second is for the normalized test data, it is for the validation in LSTM training iterations.

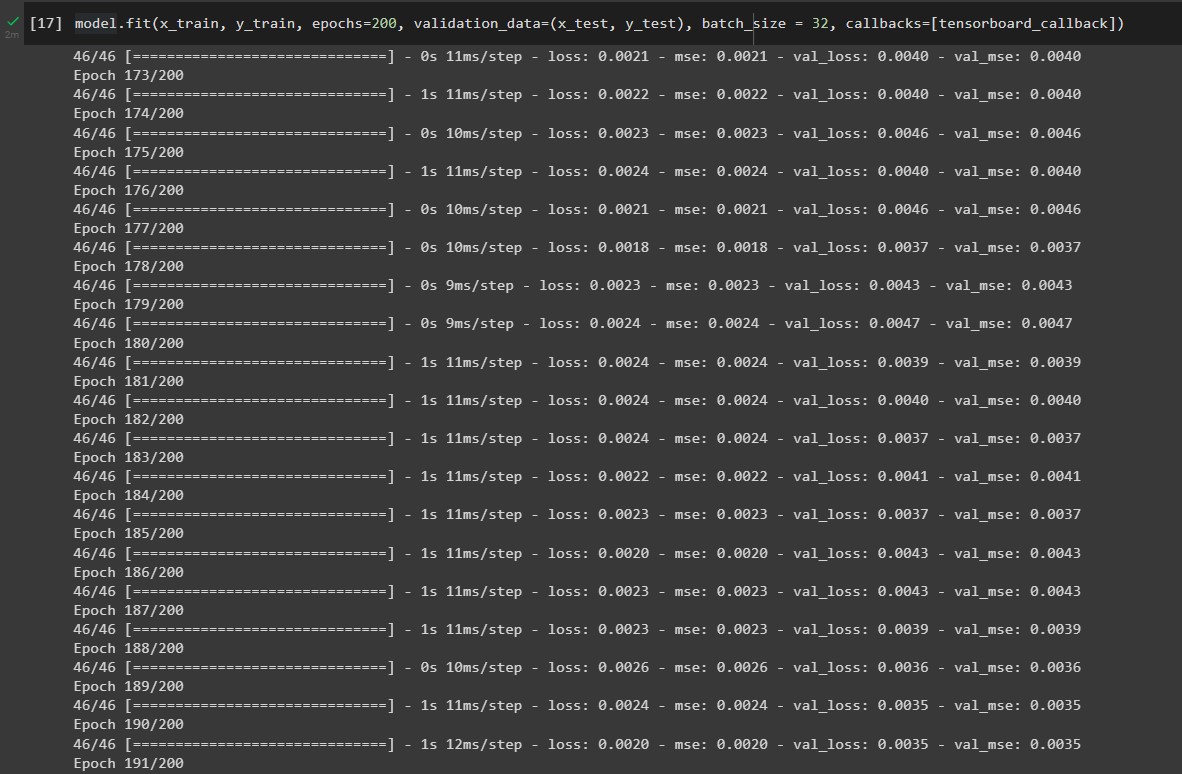


[15] Check the shape of x\_train, y\_train, x\_test, y\_test

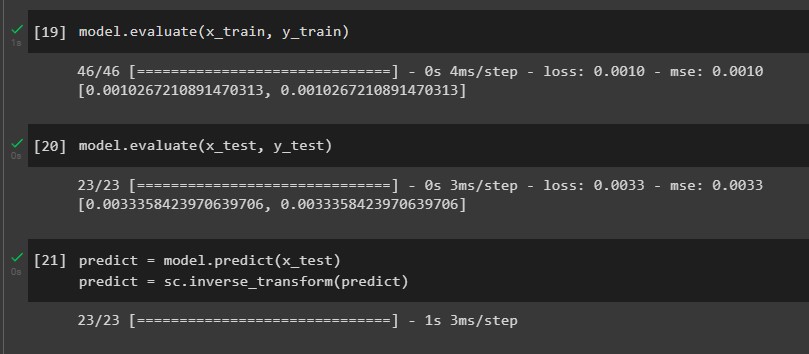
[16] Create the model structures, I use the pattern of LSTM 🡪 BatchNormalization 🡪 Dropout. The reason I use BatchNormalization is because the batch parameter is set at 32 in model fitting. Adam optimizer is used, and mean squared error is for loss measurement.



Here is the model summary result.

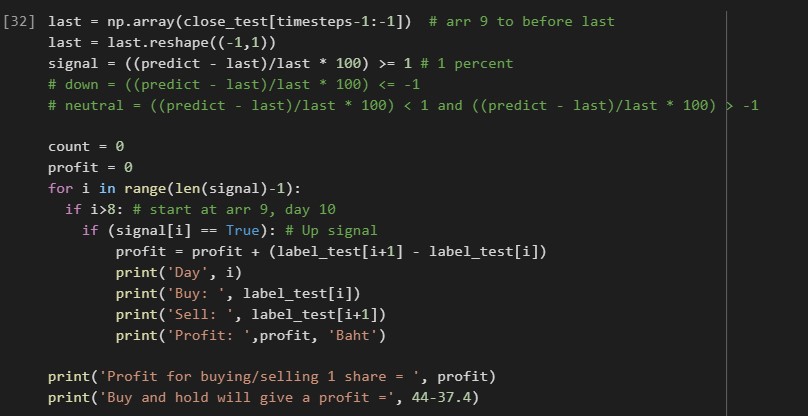


Next, fit the model. Use x\_train as X, y\_train as y, 200 epochs of training, validation data is x\_test and y\_test that we created earlier, batch size is 32, and callback to Tensorboard to observe the loss and accuracy of each epoch trained.

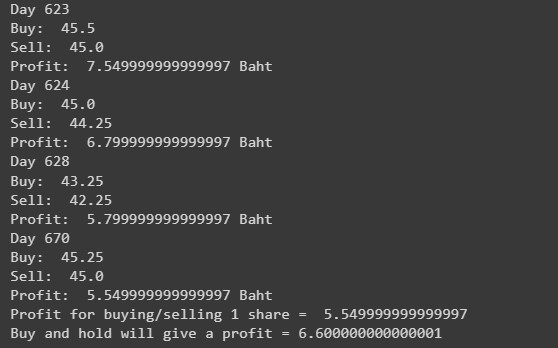


Now I evaluate the model on train and test set.

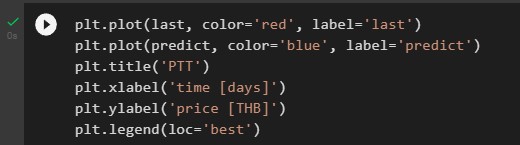
Then I predict using x\_test from the test set. Since x\_test is normalized, we have to use the same scaler to convert it back to normal form (inverse transformation).



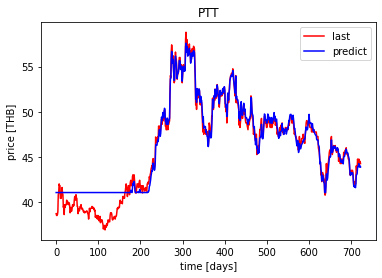
For the buy signal, see if the close price of the 11th day is +1% more than the close price of the 10th day. If so, we buy, if not, we do nothing. Then compared with the buy and hold method.

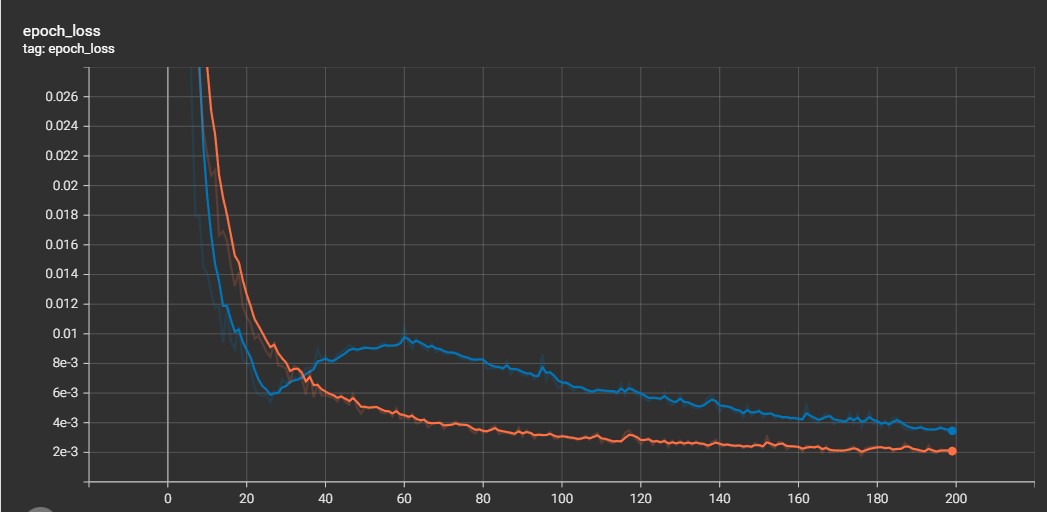


I got 5.55 Baht for this method and 6.6 for buy and hold, so buy and hold is better in this case.

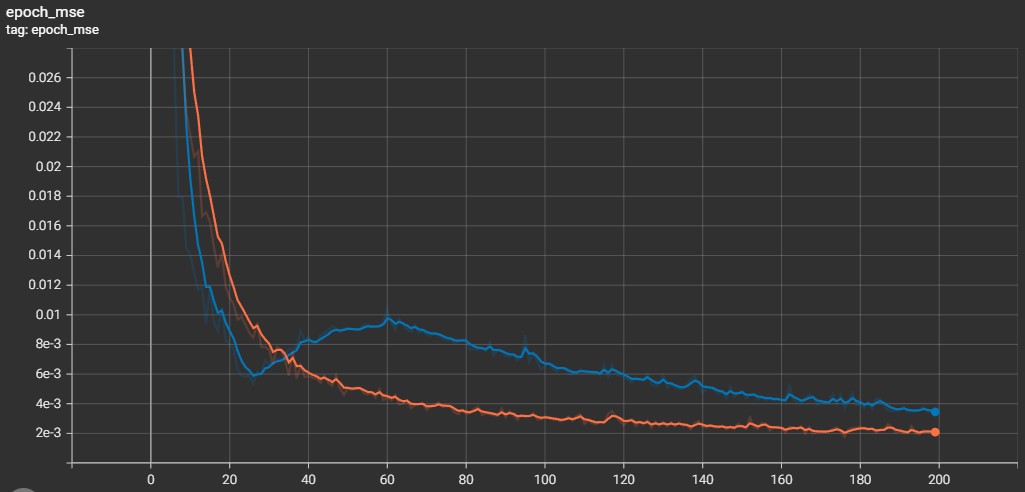


Plot the prediction compared to price





This is graph of loss per epoch, orange is train, blue is validation (test).



This is graph of accuracy (MSE) per epoch, orange is train, blue is validation (test).