# 0016 Project --- Research Project: Predicting Stock Prices

## Abtrast

In this project, some data mining models including Orange Data Mining Tools will be trained to predict stock prices and compared to find out the features of models.

## Data & Training Preparation

With Orange Data Mining Tools and packages in python, I will predict stock prices of 0016 Sun Hung Kai Properties and 0017 New World Development with their previous price data. CSV format file is easy to process. So during the data preparation step, I download the CSV datasets from Yahoo Financa. The data used for training ranges from January 3rd 2019 to June 29th 2022. For test data, it ranges from July 4th to October 28th 2022.

Fig.1 and Fig.2 show the 0016 price trends of training date and test data:

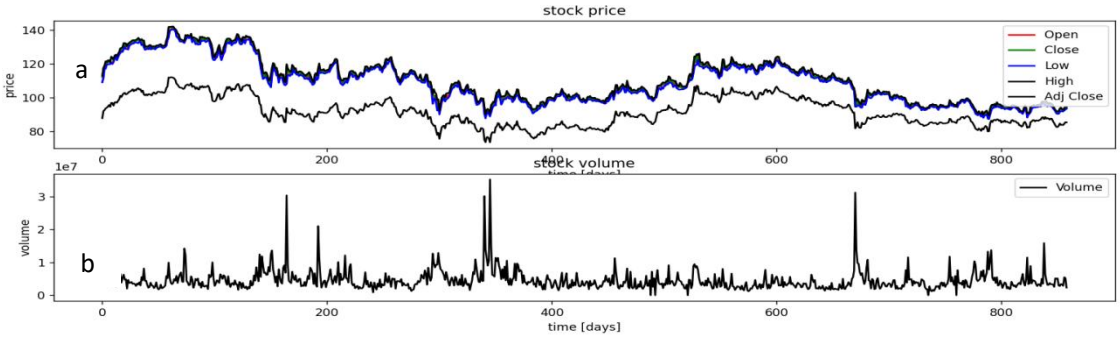



Fig.1.(a) The changes of 0016 HK Training Data. Fig.1.(b)The trading volume of 0016 HK Training Data.

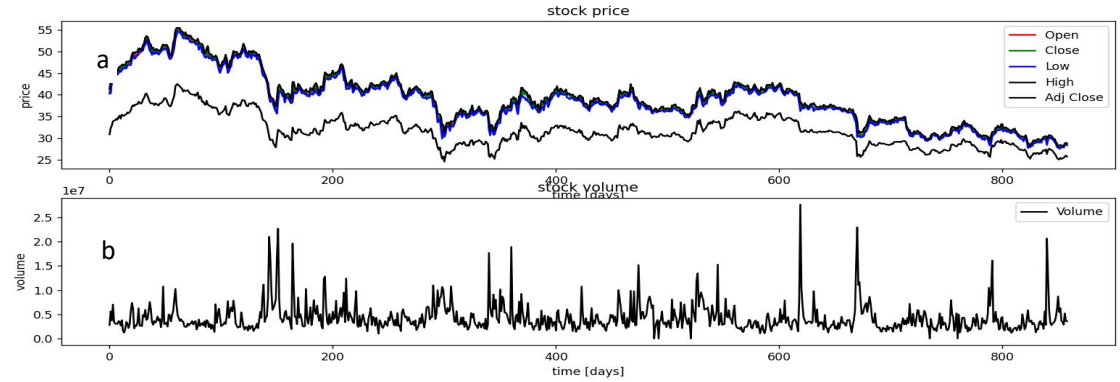



Fig.2(a).The changes of 0017 HK Training Data. Fig.2.(b)The trading volume of 0017 HK Training Data.

## Data preparation

| Date | Open | High | Low | Close | AdjClose | Volume | NextPrice | NextPrice% | 10-day mov | 5-day movi | 10-day for | 5-day fore | Day_n-1 | n-1% | Day_n-2 | n-2% | Day_n-3 | n-3% | Day_n-4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2022/7/18 | 91.849998 | 92.949997 | 91.449997 | 92.650002 | 84.256012 | 2228297 | 83.619431 | -0.755532 | 84.578851 | 84.110507 | 83.514288 | 82.496323 | 83.30114 | 1.1462892 | 83.619431 | 0.7612836 | 84.665245 | -0.483354 | 84.48336 |
| 2022/7/19 | 92 | 92.650002 | 91.099998 | 91.949997 | 83.619431 | 2455332 | 83.755844 | 0.1631355 | 84.565209 | 84.065038 | 83.512772 | 83.753268 | 84.256012 | -0.755532 | 83.30114 | 0.3820968 | 83.619431 | 0 | 84.665245 |
| 2022/7/20 | 92.5 | 92.599998 | 91.800003 | 92.099998 | 83.755844 | 2015675 | 82.891907 | -1.031495 | 84.415157 | 83.892252 | 83.378318 | 83.692725 | 83.619431 | 0.1631355 | 84.256012 | -0.593629 | 83.30114 | 0.5458557 | 83.619431 |
| 2022/7/21 | 91.900002 | 92 | 90.800003 | 91.150002 | 82.891907 | 3763904 | 82.755501 | -0.164559 | 84.246919 | 83.710372 | 83.418315 | 83.936027 | 83.755844 | -1.031495 | 83.619431 | -0.870042 | 84.256012 | -1.619 | 83.30114 |
| 2022/7/22 | 91.650002 | 91.650002 | 90.550003 | 91 | 82.755501 | 2155612 | 83.210205 | 0.5494547 | 84.04685 | 83.564867 | 83.094773 | 83.437979 | 82.891907 | -0.164559 | 83.755844 | -1.194356 | 83.619431 | -1.033169 | 84.256012 |
| 2022/7/25 | 90.300003 | 91.5 | 90.300003 | 91.5 | 83.210205 | 2295578 | 84.256012 | 1.2568254 | 83.783123 | 83.455739 | 82.684663 | 81.591466 | 82.755501 | 0.5494547 | 82.891907 | 0.3839916 | 83.755844 | -0.651464 | 83.619431 |
| 2022/7/26 | 91.5 | 92.650002 | 91.400002 | 92.650002 | 84.256012 | 2377205 | 84.437897 | 0.2158718 | 83.655808 | 83.246578 | 82.788757 | 82.836149 | 83.210205 | 1.2568254 | 82.755501 | 1.8131858 | 82.891907 | 1.6456432 | 83.755844 |
| 2022/7/27 | 92.449997 | 92.949997 | 91.400002 | 92.849998 | 84.437897 | 2045527 | 84.619774 | 0.2153974 | 83.633073 | 83.373894 | 83.301245 | 83.805725 | 84.256012 | 0.2158718 | 83.210205 | 1.4754104 | 82.755501 | 2.0329718 | 82.891907 |
| 2022/7/28 | 92.800003 | 93.050003 | 92 | 93.050003 | 84.619774 | 2668249 | 85.256355 | 0.752284 | 83.610338 | 83.510304 | 83.835398 | 84.550894 | 84.437897 | 0.2153974 | 84.256012 | 0.4317342 | 83.210205 | 1.6939857 | 82.755501 |
| 2022/7/29 | 92.849998 | 93.849998 | 92.699997 | 93.75 | 85.256355 | 4711859 | 84.256012 | -1.173335 | 83.710372 | 83.855878 | 84.189837 | 85.003697 | 84.619774 | 0.752284 | 84.437897 | 0.9693017 | 84.256012 | 1.187266 | 83.210205 |
| 2022/8/1 | 93.849998 | 93.849998 | 91.599998 | 92.650002 | 84.256012 | 2311099 | 82.846443 | -1.67296 | 83.905894 | 84.356049 | 84.889086 | 86.58408 | 85.256355 | -1.173335 | 84.619774 | -0.429878 | 84.437897 | -0.215407 | 84.256012 |
| 2022/8/2 | 92.349998 | 92.349998 | 90.5 | 91.099998 | 82.846443 | 3256325 | 82.71003 | -0.164658 | 83.905894 | 84.56521 | 84.956319 | 84.60862 | 84.256012 | -1.67296 | 85.256355 | -2.826666 | 84.619774 | -2.095646 | 84.437897 |
| 2022/8/3 | 91.099998 | 91.5 | 90.349998 | 90.949997 | 82.71003 | 2180924 | 83.528503 | 0.9895692 | 83.828595 | 84.283296 | 84.361827 | 83.268566 | 82.846443 | -0.164658 | 84.256012 | -1.834863 | 85.256355 | -2.986669 | 84.619774 |
| 2022/8/4 | 90.949997 | 91.949997 | 90.949997 | 91.849998 | 83.528503 | 2048297 | 84.892593 | 1.6330833 | 83.724014 | 83.937723 | 83.914201 | 82.487092 | 82.71003 | 0.9895692 | 82.846443 | 0.8232822 | 84.256012 | -0.863451 | 85.256355 |
| 2022/8/5 | 92.099998 | 93.5 | 91.550003 | 93.349998 | 84.892593 | 2689661 | 85.210884 | 0.3749338 | 83.787673 | 83.719469 | 83.649354 | 82.424348 | 83.528503 | 1.6330833 | 82.71003 | 2.638813 | 82.846443 | 2.4698103 | 84.256012 |
| 2022/8/8 | 92.400002 | 93.800003 | 92.300003 | 93.699997 | 85.210884 | 1712305 | 87.029686 | 2.1344715 | 84.001382 | 83.646716 | 83.643366 | 84.624327 | 84.892593 | 0.3749338 | 83.528503 | 2.01414 | 82.71003 | 3.0236405 | 82.846443 |
| 2022/8/9 | 93.699997 | 97.949997 | 92.699997 | 95.699997 | 87.029686 | 13050021 | 85.529182 | -1.724129 | 84.20145 | 83.837691 | 84.060292 | 85.933258 | 85.210884 | 2.1344715 | 84.892593 | 2.5174081 | 83.528503 | 4.1916027 | 82.71003 |
| 2022/8/10 | 95.699997 | 95.699997 | 93.599998 | 94.050003 | 85.529182 | 4204563 | 87.166092 | 1.9138614 | 84.478818 | 84.674339 | 85.199136 | 87.198332 | 87.029686 | -1.724129 | 85.210884 | 0.3735415 | 84.892593 | 0.7498758 | 83.528503 |
| 2022/8/11 | 95.75 | 96 | 94.699997 | 95.849998 | 87.166092 | 2767283 | 87.075157 | -0.104324 | 84.587946 | 85.23817 | 85.532096 | 86.691377 | 85.529182 | 1.9138614 | 87.029686 | 0.156735 | 85.210884 | 2.2945519 | 84.892593 |
| 2022/8/12 | 96 | 96.199997 | 95.25 | 95.75 | 87.075157 | 1665391 | 86.711395 | -0.417756 | 84.842578 | 85.965687 | 86.442878 | 87.090171 | 87.166092 | -0.104324 | 85.529182 | 1.807541 | 87.029686 | 0.0522477 | 85.210884 |
| 2022/8/15 | 95.75 | 96.050003 | 95.050003 | 95.349998 | 86.711395 | 1266155 | 87.075157 | 0.4195089 | 85.024458 | 86.4022 | 88.247965 | 88.334676 | 87.075157 | -0.417756 | 87.166092 | -0.521644 | 85.529182 | 1.3822335 | 87.029686 |
| 2022/8/16 | 95.599998 | 96.75 | 95.449997 | 95.75 | 87.075157 | 1743692 | 87.75721 | 0.7832923 | 85.269997 | 86.702302 | 88.252413 | 86.903569 | 86.711395 | 0.4195089 | 87.075157 | 0 | 87.166092 | -0.104324 | 85.529182 |
| 2022/8/17 | 96.949997 | 96.949997 | 96 | 96.5 | 87.75721 | 2202884 | 87.166092 | -0.673583 | 85.692868 | 86.711397 | 88.102944 | 87.238711 | 87.075157 | 0.7832923 | 86.711395 | 1.2060872 | 87.075157 | 0.7832923 | 87.166092 |
| 2022/8/18 | 96.650002 | 96.699997 | 95.599998 | 95.849998 | 87.166092 | 2174564 | 87.257042 | 0.104341 | 86.197586 | 87.157002 | 88.069481 | 87.345265 | 87.75721 | -0.673583 | 87.075157 | 0.1044328 | 86.711395 | 0.5243798 | 87.075157 |
| 2022/8/19 | 95.849998 | 96.199997 | 95.099998 | 95.949997 | 87.257042 | 1850181 | 87.075157 | -0.208447 | 86.561345 | 87.157002 | 87.785305 | 87.380147 | 87.166092 | 0.104341 | 87.75721 | -0.569945 | 87.075157 | 0.2088828 | 86.711395 |
| 2022/8/22 | 95.949997 | 96.5 | 95.449997 | 95.75 | 87.075157 | 1543597 | 86.165756 | -1.044386 | 86.79779 | 87.193379 | 87.924729 | 87.784494 | 87.257042 | -0.208447 | 87.166092 | -0.104324 | 87.75721 | -0.777205 | 87.075157 |
| 2022/8/23 | 95.75 | 95.75 | 94.150002 | 94.75 | 86.165756 | 2362650 | 84.847122 | -1.530346 | 86.984217 | 87.266132 | 87.471124 | 87.074729 | 87.075157 | -1.044386 | 87.257042 | -1.250657 | 87.166092 | -1.147621 | 87.75721 |
| 2022/8/24 | 94.75 | 94.75 | 92.650002 | 93.300003 | 84.847122 | 2319853 | 84.938072 | 0.1071928 | 86.897824 | 87.084251 | 87.105987 | 86.280449 | 86.165756 | -1.530346 | 87.075157 | -2.558749 | 87.257042 | -2.761863 | 87.166092 |

Fig.3 shows the data I use during later process, n-day moving average is calculated by the embedded function AVERAGE of Excel and n-day forecast is calculated by the FORECAST function.

## Orange proceedings

For the Orange data mining part, I choose five data mining models which include Random Forest, Neural Network, Linear Regression, kNN and SVM to analyze the characters of the stock prices.

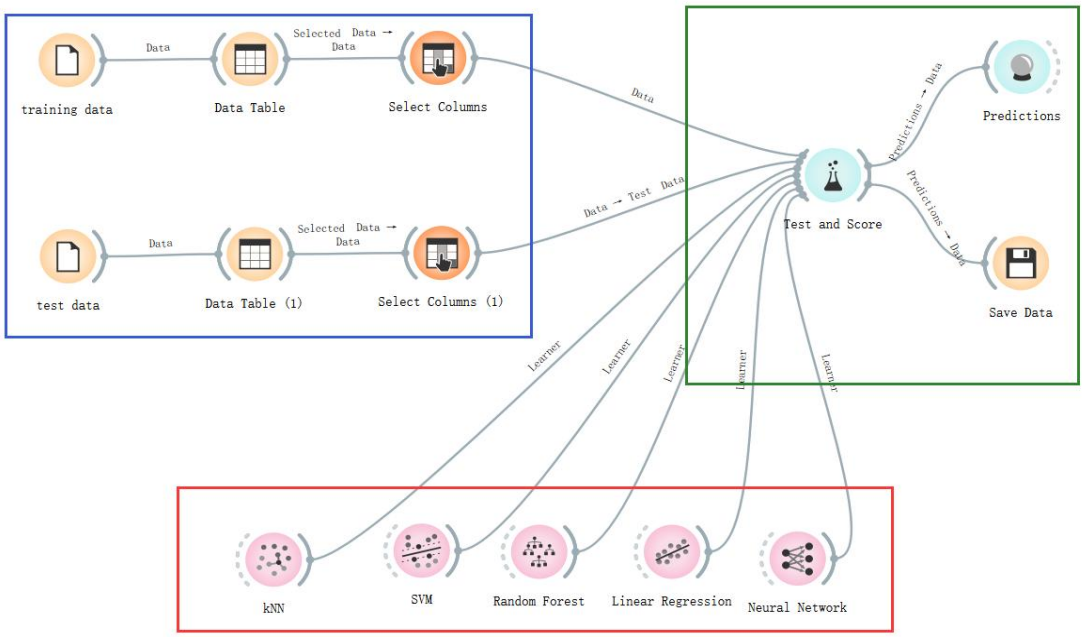



Fig.4 shows the whole training process(Blue: Data preparation and Data choosing, Red: Training models & Testing, Green: Predict & Saving Data)

## Choosing colums to train

## Features 1

I use three ways to divide the data to train the models provided by Orange Data Mining platform. For the first group, I use High, Open, Low, Volume, Close and AdjClose to train the models. The results are as Fig.4.

0016

| Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.009 | 0.000 | 197.713 | 14.061 | 9.669 | -1.202 |
| SVM | 0.020 | 0.002 | 88.051 | 9.384 | 2.680 | 0.019 |
| Random Forest | 0.027 | 0.001 | 74.653 | 8.640 | 1.745 | 0.169 |
| Linear Regression | 0.003 | 0.001 | 73.577 | 8.578 | 1.673 | 0.181 |
| Neural Network | 0.495 | 0.001 | 665.939 | 25.806 | 19.355 | -6.416 |

0017

| Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.007 | 0.003 | 96.960 | 9.847 | 9.482 | -12.568 |
| SVM | 0.019 | 0.003 | 61.942 | 7.870 | 6.810 | -7.668 |
| Random Forest | 0.030 | 0.002 | 18.039 | 4.247 | 3.346 | -1.524 |
| Linear Regression | 0.004 | 0.000 | 0.168 | 0.410 | 0.287 | 0.976 |
| Neural Network | 0.501 | 0.002 | 69.872 | 8.359 | 7.513 | -8.778 |

Fig.5 The Train Time, Test Time, MSE, RMSE, MAE and R2 of Models with High, Open, Low, Volume, Close and AdjClose.

## Features 2

For the second group, I ues the data Day_n-1, Day_n-2, Day_n-3, Day_n-4, Day_n-5 and their change rate as well to train the models and the data NextPrice to be the Taget, with Date as the meta. All the data I use above which is not provided by Yahoo Finance is calculated by myself with the assistance of Excel. The results are as Fig.5.

0016

| Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.007 | 0.001 | 75.228 | 8.673 | 1.828 | 0.162 |
| SVM | 0.014 | 0.002 | 102.665 | 10.132 | 2.209 | -0.143 |
| Random Forest | 0.034 | 0.002 | 71.791 | 8.473 | 1.434 | 0.200 |
| Linear Regression | 0.007 | 0.001 | 2.284 | 1.511 | 0.328 | 0.975 |
| Neural Network | 0.575 | 0.002 | 12118.508 | 110.084 | 27.132 | -133.958 |

| 0017 Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.008 | 0.002 | 20.781 | 4.559 | 3.924 | -1.908 |
| SVM | 0.014 | 0.003 | 51.265 | 7.160 | 5.667 | -6.174 |
| Random Forest | 0.033 | 0.002 | 8.355 | 2.891 | 2.136 | -0.169 |
| Linear Regression | 0.006 | 0.001 | 1.692 | 1.301 | 0.609 | 0.763 |
| Neural Network | 0.558 | 0.002 | 32.374 | 5.690 | 4.939 | -3.530 |

Fig.6 The Train Time, Test Time, MSE, RMSE, MAE and R2 of Models with Day_n-1, n-1%, Day_n-2, n-2%, Day_n-3, n-3%, Day_n-4, n-4%, Day_n-5, n-5%.

## Features 3

The third group of data I choose to use as feature is the 5-day moving average, 10-day moving average, 5-day forecast and 10-day forecast. All these datas are calculated with Excel as well. Nextprice is set to be the Target with Date as the meta. With these data, the training results are as Fig.6 shows.

| 0016 Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.003 | 0.002 | 77.225 | 8.788 | 2.723 | 0.140 |
| SVM | 0.011 | 0.002 | 84.794 | 9.208 | 2.605 | 0.056 |
| Random Forest | 0.021 | 0.001 | 80.329 | 8.963 | 2.826 | 0.105 |
| Linear Regression | 0.004 | 0.001 | 79.909 | 8.939 | 2.622 | 0.110 |
| Neural Network | 0.504 | 0.001 | 505.602 | 22.486 | 16.913 | -4.631 |

| 0017 Model | Train | Test | MSE | RMSE | MAE | R2 |
|---|---|---|---|---|---|---|
| kNN | 0.004 | 0.001 | 24.801 | 4.980 | 4.075 | -2.471 |
| SVM | 0.013 | 0.002 | 85.615 | 9.253 | 8.390 | -10.981 |
| Random Forest | 0.024 | 0.001 | 181.846 | 13.485 | 12.919 | -24.447 |
| Linear Regression | 0.006 | 0.000 | 3.444 | 1.856 | 0.911 | 0.518 |
| Neural Network | 0.505 | 0.001 | 145.227 | 12.051 | 10.218 | -19.323 |

Fig.7 The Train Time, Test Time, MSE, RMSE, MAE and R2 of Models with 5-day moving average, 10-day moving average, 5-day forecast and 10-day forecast.

# Models and datasets analyze

| Models / Datasets | kNN | SVM | Random Forest | Linear Regression | Neural Network | Average |
|---|---|---|---|---|---|---|
| 0016Feature1 | -1.202 | 0.019 | 0.169 | 0.181 | -6.416 | -1.4498 |
| 0017Feature1 | -12.568 | -7.668 | -1.524 | 0.976 | -8.778 | -5.9124 |
| 0016Feature2 | 0.162 | -0.143 | 0.200 | 0.975 | -133.958 | -26.5528 |
| 0017Feature2 | -1.908 | -6.174 | -0.169 | 0.763 | -3.530 | -2.2036 |
| 0016Feature3 | 0.140 | 0.056 | 0.105 | 0.110 | -4.631 | -0.844 |
| 0017Feature3 | -2.471 | -10.981 | -24.447 | 0.518 | -19.323 | -11.3408 |
| Average | -2.9745 | -4.1485 | -4.277666 | 0.5871666 | -29.4393 | |

Fig.8 shows the R2 scores of different models and features

From Fig.8 we can know that Linear Regression has the highest R2 scores and Neural Network has the poorest R2 scores. From this we learn that models has no priority. Which model will I choose depends on that whether the model is suitable.

For the time cost, kNN and Linear Regression have the least time cost, SVM and Random Forest rank second, Neural Network has the most expensive time cost.

In summary, Linear Regression is the most suitable model for these datasets. As for kNN, SVM and Random Forest, they are better than Neural Network, but they are still not suitable. Neural Network in Orange is completely unsuitable for thses datasets for the reason that it has the longest training time and the poorest R2 scores.

| Stocks | 10-day moving average | 5-day moving average | 10-day forecast | 5-day forecast | Machine learning based on previous prices | Machine learning based on previous price change % |
|---|---|---|---|---|---|---|
| 0016 | 1.946031108 | 1.474533752 | 1.46184226 | 1.271818095 | 1.673 | 0.609 |
| 0017 | 0.829025001 | 0.61777859 | 0.610074765 | 0.512898511 | 0.281 | 0.911 |

Fig.9 shows the MAE of the stocks

# New Methods(TCN)

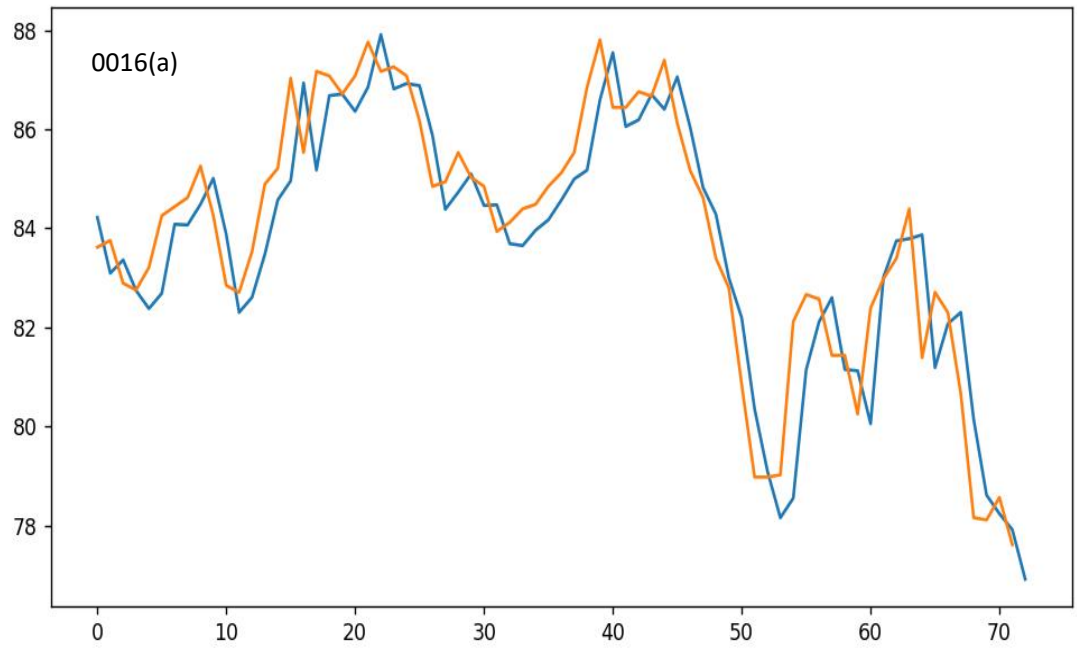
I choose TCN model (Temporal Convolutional Network) as the new method I use to predict stock prices. I choose this model for the simple reason that time plays an important role in the process of predicting prices. This model combines two different models CNN and RNN which enable TCN has brilliant performance when predicting stock prices.

The theory of TCN is from "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". And thanks to code on CSDN platform(https://blog.csdn.net/FrankieHello/article/details/113409856) and python package from philipperemy on Github (https://github.com/philipperemy/keras-tcn).

The dilated convolution operation F on element s of the sequence is defined as below, x is the input of a 1-D sequence and the filter f.

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i}$$

The Figures below show the results.



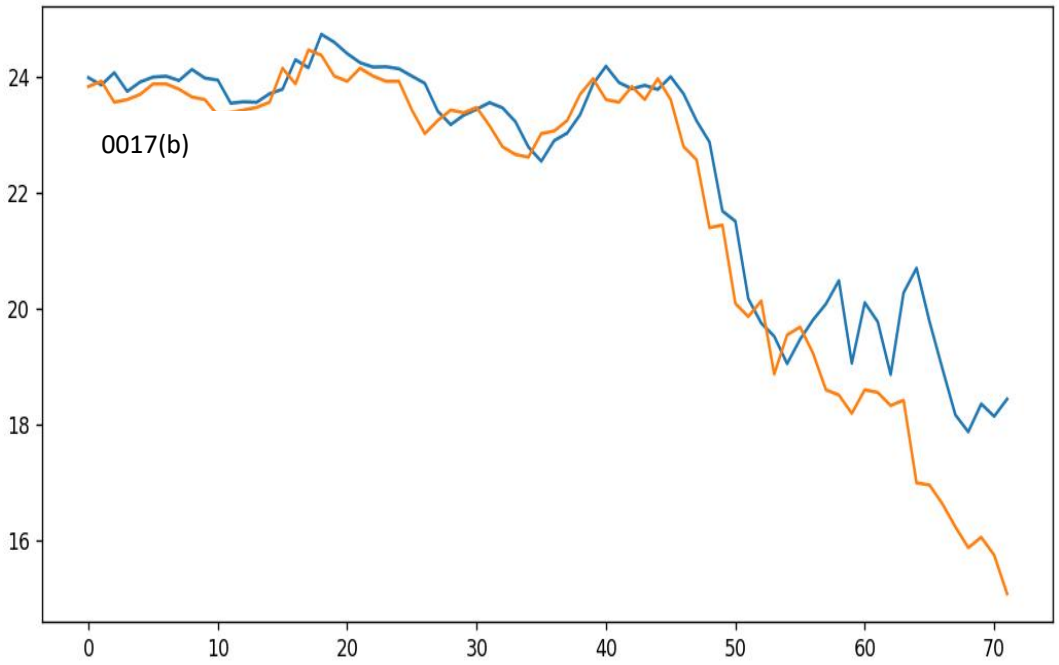



Fig.10(a). shows the prediction curve(orange) and the real curve(blue) of 0016 stock

Fig.10(b). shows the prediction curve(orange) and the real curve(blue) of 0017 stock

The training process is as following:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 tcn (TCN)                   (None, 10)                2940

 dense (Dense)               (None, 1)                 11

=================================================================
Total params: 2,951
Trainable params: 2,951
Non-trainable params: 0
_________________________________________________________________
Epoch 1/200
22/22 [==============================] - 1s 13ms/step - loss: 0.3600 - mae: 0.3600 - val_loss: 0.0414 - val_mae: 0.0414
Epoch 2/200
22/22 [==============================] - 0s 4ms/step - loss: 0.0728 - mae: 0.0728 - val_loss: 0.0290 - val_mae: 0.0290
Epoch 3/200
22/22 [==============================] - 0s 4ms/step - loss: 0.0483 - mae: 0.0483 - val_loss: 0.0292 - val_mae: 0.0292
Epoch 4/200
22/22 [==============================] - 0s 5ms/step - loss: 0.0417 - mae: 0.0417 - val_loss: 0.0255 - val_mae: 0.0255
Epoch 5/200
22/22 [==============================] - 0s 4ms/step - loss: 0.0423 - mae: 0.0423 - val_loss: 0.0297 - val_mae: 0.0297
```

```
Epoch 200/200
22/22 [==============================] - 0s 4ms/step - loss: 0.0242 - mae: 0.0242 - val_loss: 0.0182 - val_mae: 0.0182
```

Fig.11 shows the training process of 0016.

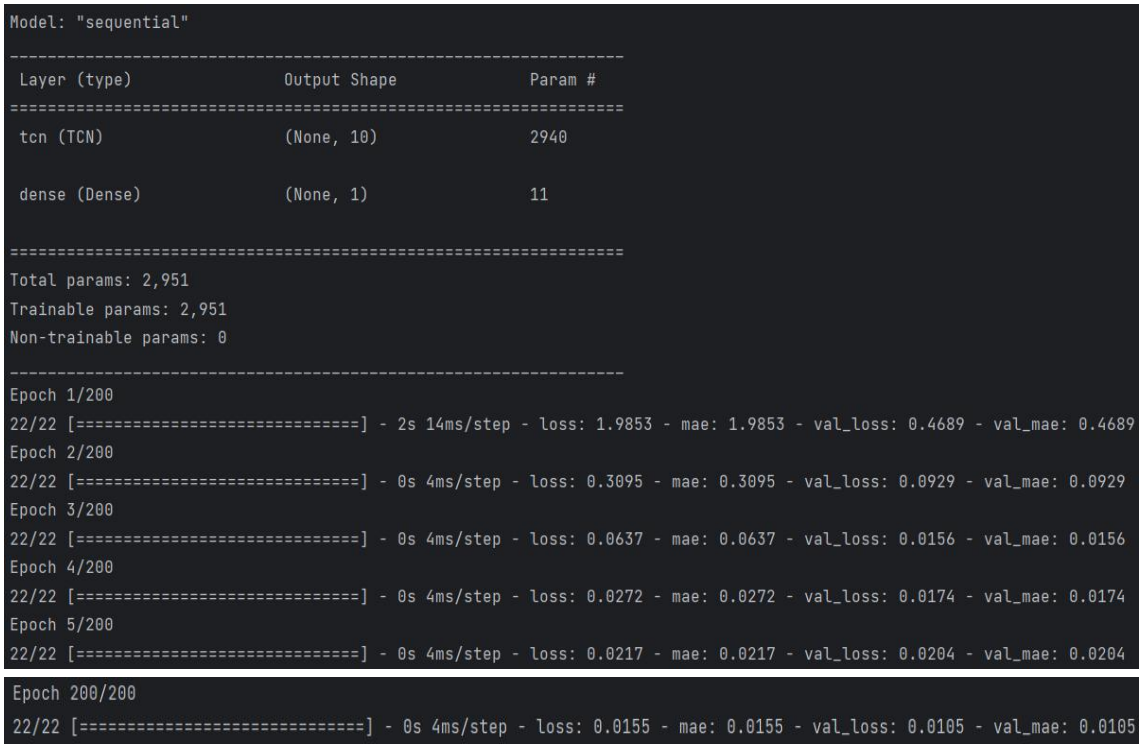



Fig.11 shows the training process of 0017.

From the training process above we can find that the MAE is much lower than any kind of predictions in Orange and Excel. After 200 epoches the MAE finally reach 0.0242 and 0.0155 which is 30 times smaller than the models I use in Orange and Excel.

To summarize, TCN model is much more suitable for this application because it is connected with time and stock price vary with time. In other models time is considered as a feature more. Other models won't memorize the data in previous time. However, TCN could remember that for a long time to make proper predictions.

# Discussion

Through this project, the most important finding I have is not the TCN model but the experience of choosing the most suitable model for different applications. I used to consider neural network as an advanced model and linear regression as a inferior method.

But the training results are much different from what I thought. Neural network has the longest training time and the poorest performance. However, linear regression is the quickest model and has the highest R2 scores.

I can conclude that there is no priority between models. In different situations, we should choose different models to adapt to the situations. A best model for everything is impossible. Different models has different characters and are suitable for different features. Models vary with features.

The second finding I have is that choosing features is important for a good training process. After data preparation, I have lots of features to choose to train the model. But some features are beneficial some are harmful. If you add some useless feature into the model, it will increase the training time without higher R2 scores or lower MAE. This is why we need correlation coefficient to analyze the relationship between datas. I used to have a vague imperssion about this but now the reason is clear.

By and large, to dig out the information hidden in the raw data or to analyze the relationships between data, we must use various features and different models to get suitable results. To achieve this, the characters of models should be taken into account and the theory of it should be clear.

# Reference

[1] Shaojie Bai, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling” 19-Apr-2018.[Online].Available:https://arxiv.org/pdf/1803.01271.pdf

[2] Dastan Hussen Maulud,“A Review on Linear Regression Comprehensive in Machine Learning”31-December-2020[Online]Available:https://jastt.org/index.php/jasttpath/article/view/57

[3] Vladimir Cherkassky, “Practical selection of SVM parameters and noise estimation for SVM regression”,January-2004,[Online].Available:https://www.sciencedirect.com/science/article/pii/S0893608003001692

[4] Yahoo Finance, “0016 Sun Hung Kai Properties stock historical prices & data,” *Yahoo! Finance*, 28-Nov-2022. [Online]. Available: https://finance.yahoo.com/quote/0016.HK/history. [Accessed: 28-Nov-2022]

[5] Yahoo Finance, “0017 New World Development stock historical prices & data,” *Yahoo! Finance*, 28-Nov-2022. [Online]. Available: https://finance.yahoo.com/quote/0017.HK/history. [Accessed: 28-Nov-2022]

[6] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.