**Final Project Report**

**Deep Learning for Artificial Intelligence**

**Prof: Huynh The Dang**

Student’s name: Bui Ngoc Linh

ID: 210223

# Nasdaq stock price prediction

Google Colab:

<https://colab.research.google.com/drive/1ExtOOP2_kNGI3cmd1ONJCPHTWCjcmdzf?usp=sharing>

## Experiments & Observation:

The main task here is to predict the price of NASDAQ Stock Price. From my research, I acknowledge that NASDAQ Stock Market has been the most active stock trading venue in the US by volume, and has been established for a very long time compared to other markets. Thus, NASDAQ stock market, in the long run, can be expected to be quite stable. This helps the training process of the model become more simple than that of new and “immature” market like Vietnam.

Regarding the data extraction and data preparation process, I first used the method of uploading csv files directly to runtime and reading csv directly. However, I later realized that this would be troublesome for others to run Colab themselves (as they would have to upload files again every time, which would take 10-15 minutes). Therefore, I later used the os, sys package and drive.mount('/content/drive') to get files directly from Google Drive. For others’ reusing, I can enclose the Colab link with the Drive folders for them to upload.

Also, to reduce the time filtering files by files, I concatenate all files first and filter them all together later. First, data which has missing columns, or values, must be filtered out. Then, I filtered out the companies which have too few transaction dates available or small trading volume (signaling immaturity or too small scale), as these factors would add more complexities that cannot be dealt with yet.

Window: I have tried different window sizes (30, 45, 64,..) for training model, which return little differences in terms of mean squared error or loss. I think that stocks tend to exhibit patterns or cycles that occur within specific timeframes, such as weeks (5 trading days), months (about 20 trading days), quarters (90 days), or years (360 days); therefore, since these cycles are typically shorter than the window sizes being tested (64 and 30 days), the information captured within these different window sizes may not differ significantly. Eventually, I sticked with the 30-day-before window size.

Regarding model building phase, I choose LSTM model similar to example in the sample code for two reasons: it is compatible with time-series data; and it can model time dependencies (capturing temporal and long-term dependencies). Also, I experimented with some modifications to improve the model accuracy so that a common model can be applicable for individual companies.

* I modified the input and output to take into account all six features instead of just one separate.
* Convert Panda dataFrame type to time-series numpy dataset for more flexible manipulation
* Use standardization instead of Max-min normalization (Z-score normalization)

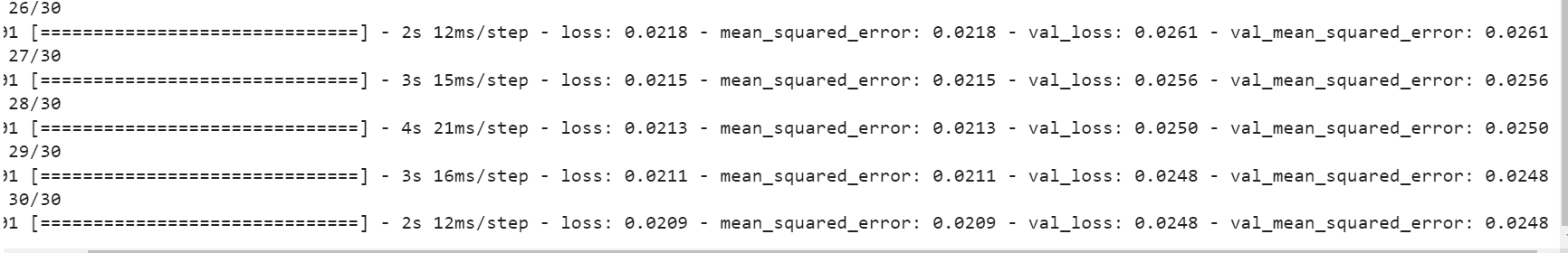
Regarding validation and test size, I stuck to the familiar ratio of 1/5. So, with 5000 recent days dataset, day 1 to day 3500 will be training data, day 3500 to 4000 will be validation data, and testing data will be from day 4000 to 5000. I also stuck to the mean squared error as main metrics.

Regarding learning rate schedule, I added decay rate to avoid overfitting and with starting rate of 0.1 (optimal for the batch size). I experimented with different batch sizes (512, 1024, 2048) – the higher, the faster – and with different numbers of epochs. It is observed that the changes in batch size did not make much difference, but the figure for epochs standing at around 20-30 is optimal (since from epoch 10-15, the MSE is quite volatile, but after that, it increased quite sustainably and converged.

## Findings

Overall, this is the main result:

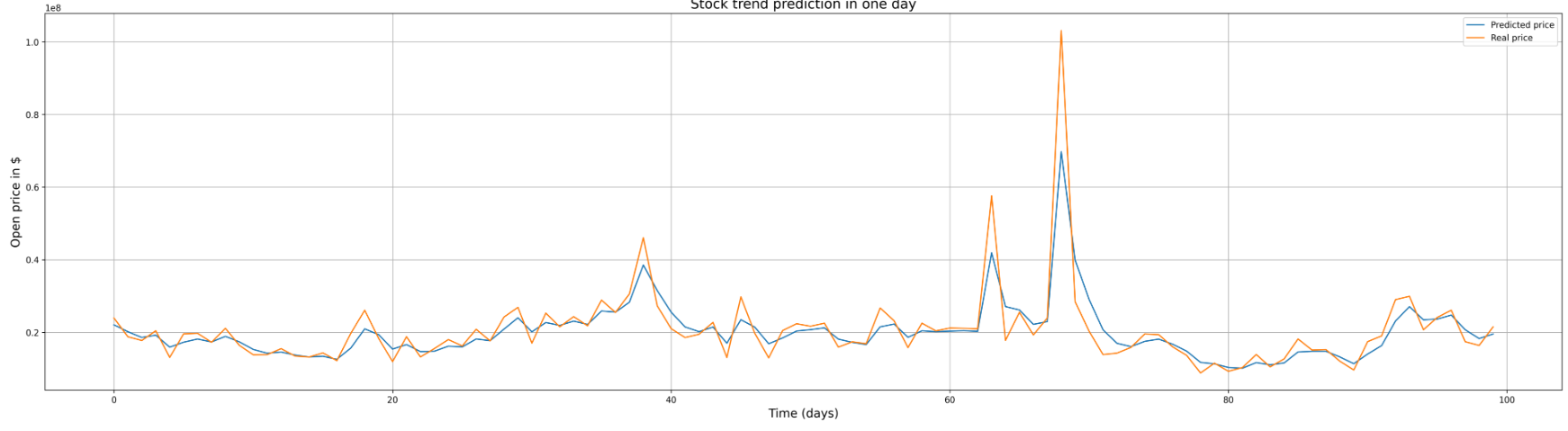
* At epoch 30, val MSE is reduced to 0.0248, which is good.



* The MSE for test data is relatively comparable.



* The predicted stock prices closely align with real stock price:



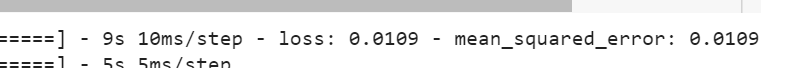
The sample chosen for this visualization may be from a volatile period, with days between 60 and 70 showing many abrupt strikes. I suppose that this period is near or after the 2008 financial crisis.

## Conclusion:

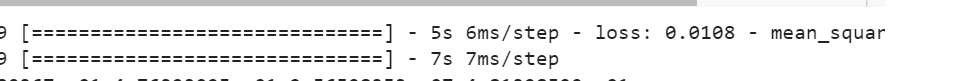
It can be observed that this LSTM model works well with one-day-next prediction, evident through the close alignment. It can capture the rule and predict most of the abrupt spikes.

From the general model, I also applied it to the testing data set of APPL, AMZN, and MSFT, the giant tech corporations in the US. I expect a very low MSE here, and that is true.

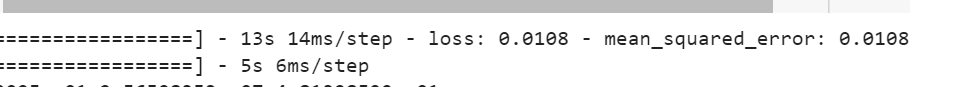
MSE of APPL



MSE of AMZN



MSE of MSFT



# Vietnam stock price prediction

<https://colab.research.google.com/drive/1kFw8FBKe9ITyun3GQs_9IBmtQvCI0GWk?usp=sharing>

The Vietnam Stock market is much more volatile and unpredictable compared to NASDAQ Stock Market, which I assume is due to the immaturity and susceptibility of external factors. Also, Vietnam’s economy has experienced rapid growth in recent years, but it also faces inflation. These can have a significant impact on the stock market, leading to increased volatility. Vietnam also has quite a small number of investors compared to other developed markets, which also leads to volatility. Besides, inconsistencies or gaps in regulations may also be contributing factors.

With that in mind, I first chose the oil & petrol industry for input data. This industry has both pros and cons in terms of compatibility for training model. First, it is heavily influenced and dominated by the Vietnamese government: all key players are state-owned. Second, there has been an expansion of petrol production activities in Vietnam in recent years, since Doi Moi 1986. Third and most importantly, the stocks of these state-owned companies are mostly reserved for internal officials instead of open to the public. All these three factors would make the petrol industry a challenging case, with huge differences from the patterns observed in the giant corporation in the US.

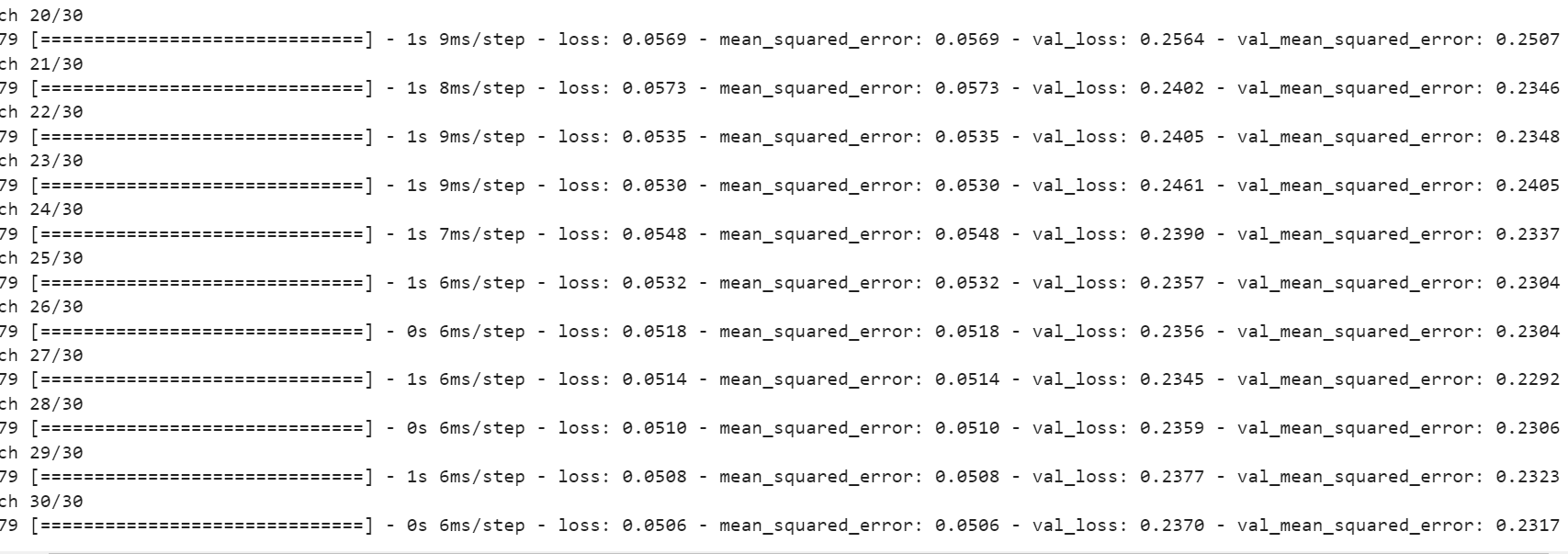
Regarding additional Vietnam data such as dividend history, industry analysis, financial ratio, I did extract and process them, but not yet found an optimal way to include them in the input; historical data is still the key dataset here. I also made use of ‘try and except’ to consider companies in both HOSE and HNX, as I assumed that HNX market is still worth considering. Companies in Upcom should not be included because of immaturity and excessive volatility.

Regarding dataset preparation, I stuck to the ratio of 1:5, and window size 30, as it makes no difference. In this task, the objective is still to predict the next day only.

Regarding data normalization, I chose Z-score normalization to take into account the possible significant difference between volume and price.

Regarding model construction, I have experimented with many batch size and learning rate schedules to find an optimal pair of figures. First, I tried a batch size of 1024, but the MSE for both train and val is bad. I went to research more and realized that with many noisy or outliers in VN dataset, using smaller batch sizes can help mitigate the impact of individual noisy samples. By averaging gradients computed from a smaller batch, the influence of individual outliers or noisy samples is reduced, leading to more stable updates to the model's parameters.

Thus, I tried with gradually smaller batch size. The optimal one so far, in terms of speed, memory, and MSE & val\_MSE, is 128 batch size, and num\_epochs = 30. With the same model used for NASDAQ, here is the result:

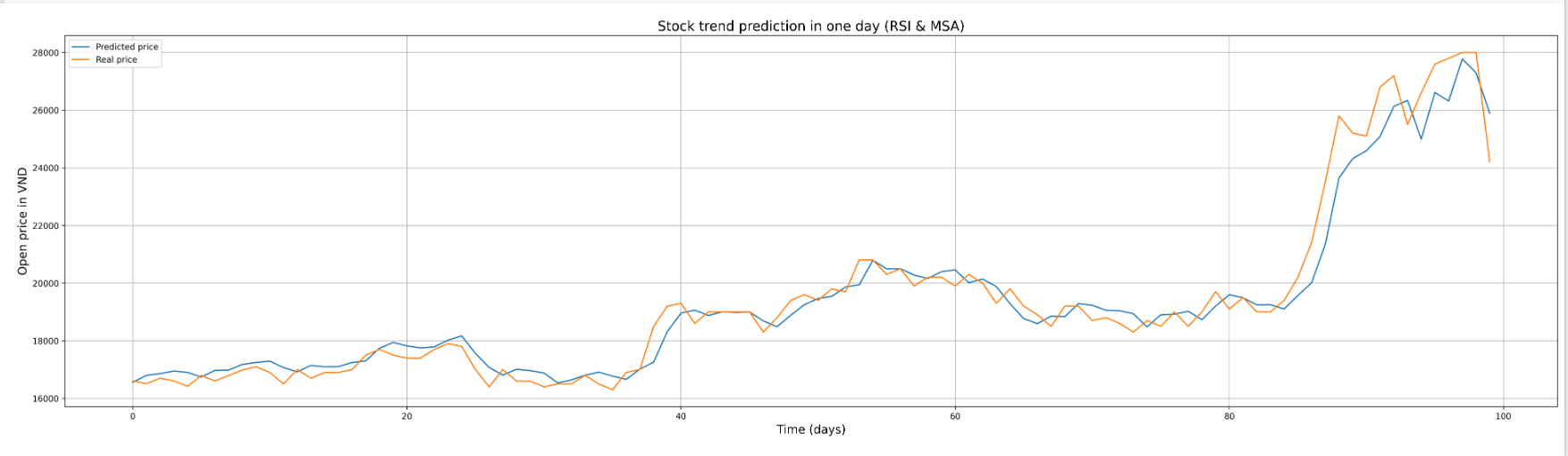


At epoch 30, MSE reaches 0.0506, which is still much higher than that of NASDAQ, but is quite promising in the context of Vietnam market. I suppose that this performance is enough for real-life application. The val\_MSE, though still very high (10X compared to that of NASDAQ), is the best performance I can get. This is still at an acceptable level for usage.

Also, on the test data, the performance is quite close to train MSE. This suggests a threat of overfitting, but not too much, or just randomness. I would do more experiments to find out its nature.

Results

Here, the prediction line is not as closely aligned to the real data as seen in NASDAQ.



# Nasdaq/VN trading point identification

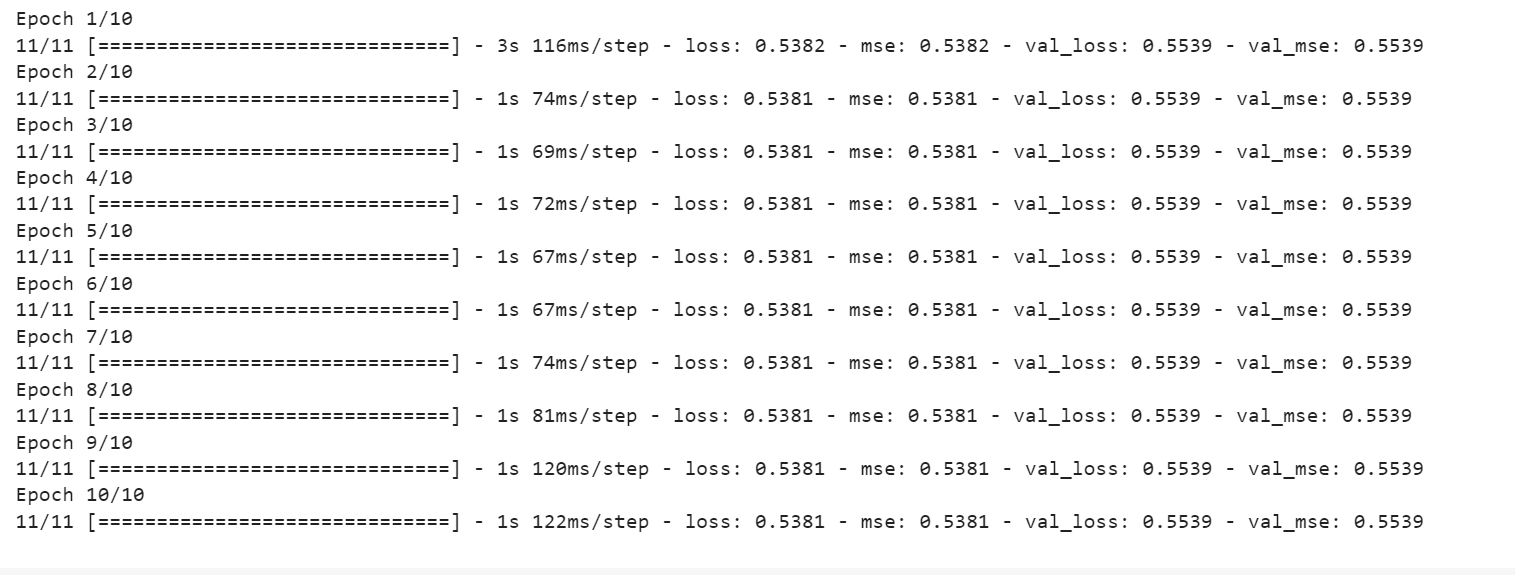
Colab link: <https://colab.research.google.com/drive/15WrqRpQEEspppZmHy5sW1863CQvrps3N?usp=sharing>

I decided to do trading point identification for VN Index 30 for these reasons:

* The VN Index 30 (VN30) is a stock market index that represents the performance of the 30 largest and most liquid stocks listed on the Ho Chi Minh City Stock Exchange (HOSE) in Vietnam.
* It serves as a benchmark for assessing the overall performance of the Vietnamese stock market and is widely followed by investors.
* These stocks are also considered to be representative of various sectors of the Vietnamese economy.

Regarding normalization, I stuck to MinMax normalization.

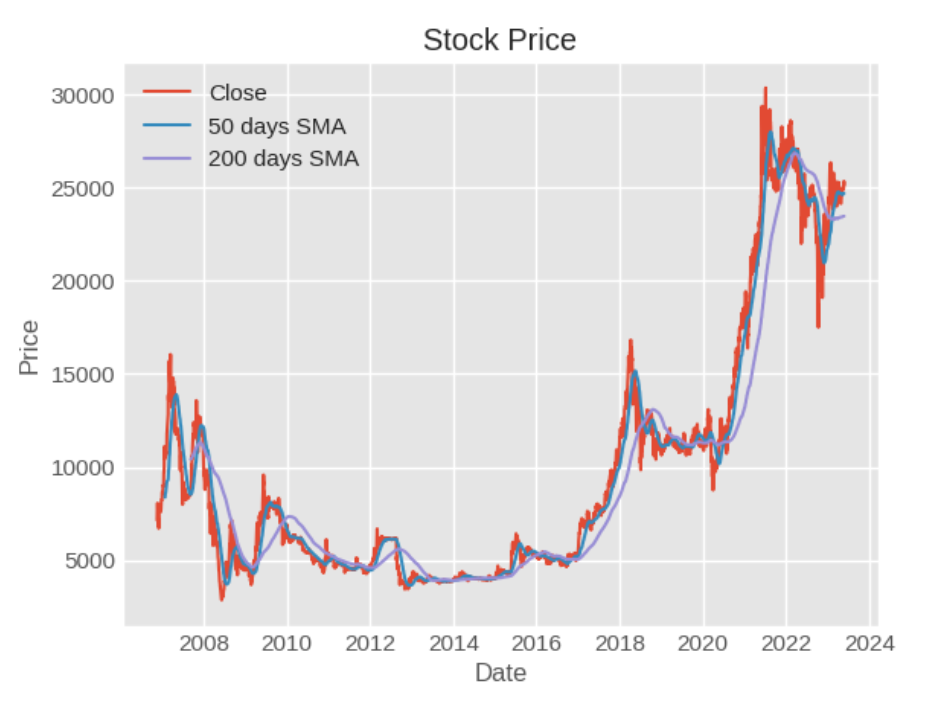
Regarding the training process, the performance of model training is not good: theval\_mse remains mostly unchanged throughout epochs. This means that the model here is not learning. Due to my quick review, I suppose that this problem may be due to improper hyperparameter tuning, insufficient training data, or wrong model architecture. I have experimented with different hyperparameters and window sizes to diagnose the problems, but it cannot be fixed yet.

****

Regarding model architecture, I have also tried add a flattening layer according to instructor’s suggestion, change optimizers,… but these changes eventually did not make much difference.

Still, I went on researching about how to do the manual feature engineering such as Simple Moving Average (SMA) to determine the training point. I found a sample code on Github <https://github.com/huseinzol05/Stock-Prediction-Models> that is related to identifying buying and selling signals based on thresholds. I decided to employ this method. In summary, the most common crossover is the 50-day SMA and the 200-day SMA. When the shorter-term SMA (e.g., the 50-day SMA) crosses above the longer-term SMA (e.g., the 200-day SMA), is considered a bullish signal, which means that the shorter-term price trend is becoming stronger than the longer-term trend, indicating potential upward momentum in the price. The reverse is true for bad (selling) signals.

Here is the result so far after some small modifications:



Besides, I have also experimented with multiple-day prediction, reusing the previous model in task 1 and 2. The method here is to predict 7 days – a week – and identify the lowest price for buying and highest price for selling. This is based on assumption that there are available funds, and the trading fee & tax is not significant. However, it did not work well and not returned meaningful results.

So, in conclusion, I suppose that building a trading point prediction, for both NASDAQ and Vietnam market, is highly challenging. Still, the manual feature engineered from SMA can be a good input for further training for higher accuracy and usability.

# 

# **Nasdaq stock risk/portfolio management:**

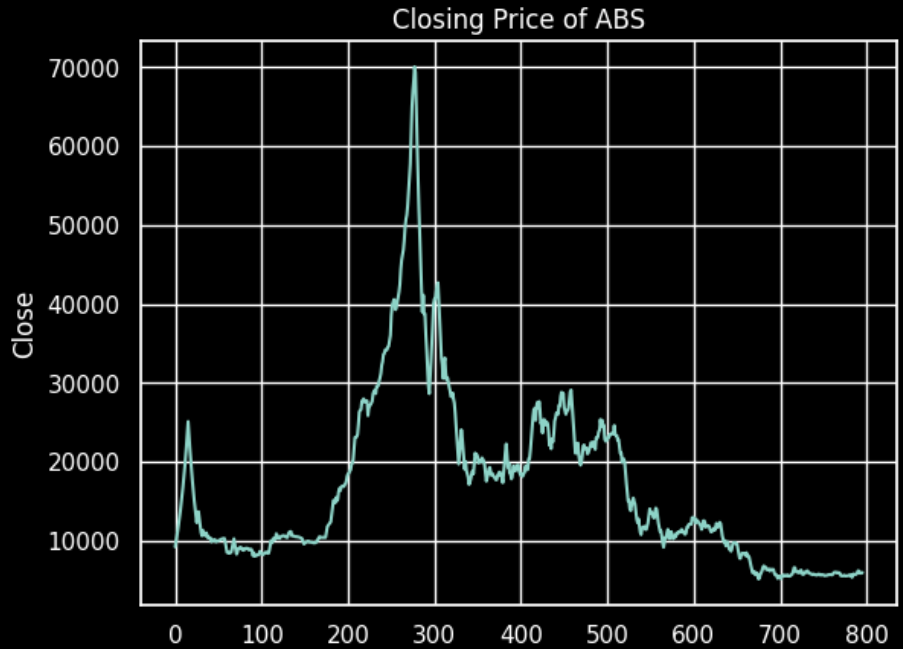
Link to Google Colab:

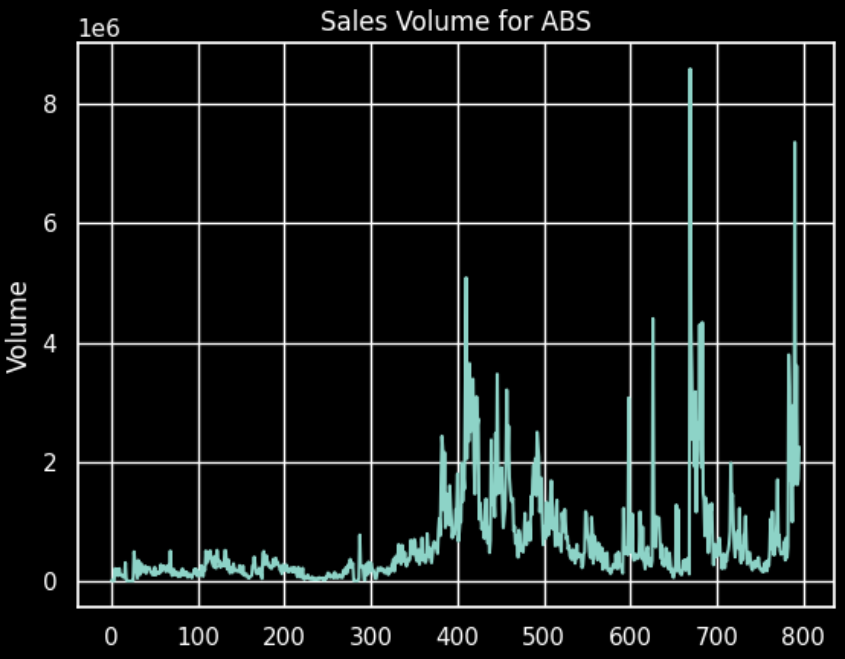
<https://colab.research.google.com/drive/1FqI4y9PFWGqgb7KnIvpdC_J6l4VMgYme?usp=sharing>

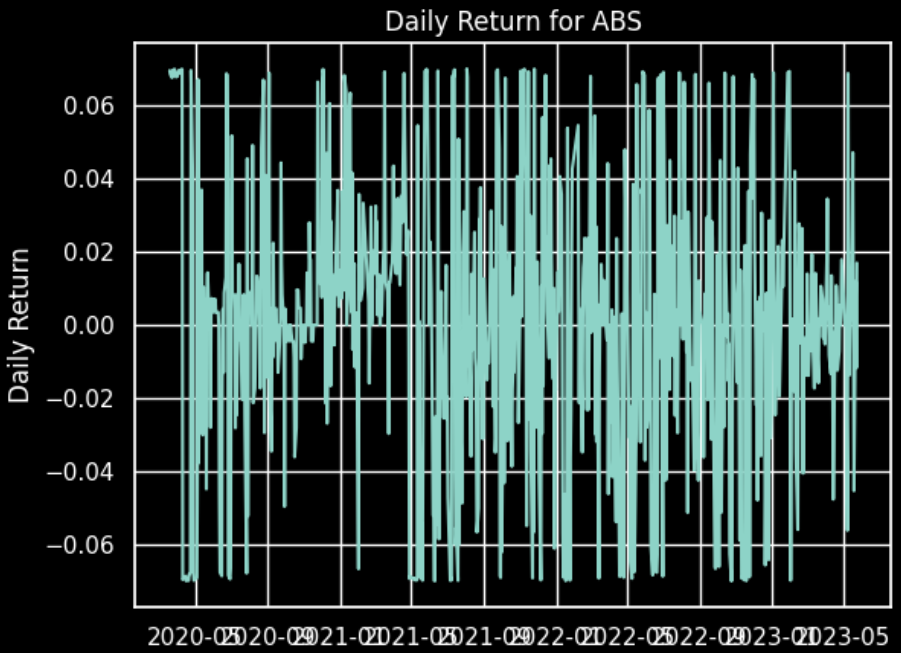
In task 4, I chose Retail, Real Estate, and Banking industry for visualization and analysis. Reasonings are as follows:

* Retail Industry:
  + has been growing at a rapid pace, driven by increasing consumer purchasing power, urbanization, and a growing middle class. The country's economic growth and rising disposable incomes have bolstered consumer spending, leading to the expansion of the retail sector.
  + Modern retail formats, including supermarkets, hypermarkets, shopping malls, and convenience stores, have gained popularity in urban areas. In the list, they are FPT, The gioi di dong, and Digiworld.
  + There are huge discrepancies between industry leaders like The gioi di dong, FPT and the others.
* Real Estate:
  + Vietnam's stock market has a significant number of real estate companies listed, showing the importance of the real estate sector in the country's economy growth.
  + Real estate companies on the stock market in Vietnam often have ongoing development projects, including residential complexes, commercial buildings, or mixed-use developments. These projects contribute to revenue generation and asset growth.
  + Vietnam's real estate market offers significant growth potential due to urbanization, increasing middle-class population, and rising demand for residential and commercial properties.
* Banking:
  + Many of the dominant players in Vietnam economy & stock markets are banks, such as Vietcombank, ACB, BIDV, MB,… They are mostly quite stable, and play an important role to other industries (real estate, production,… )

Next, I visualize the closing price, volume, and the percent changed for each day for every companies in these 3 lists.

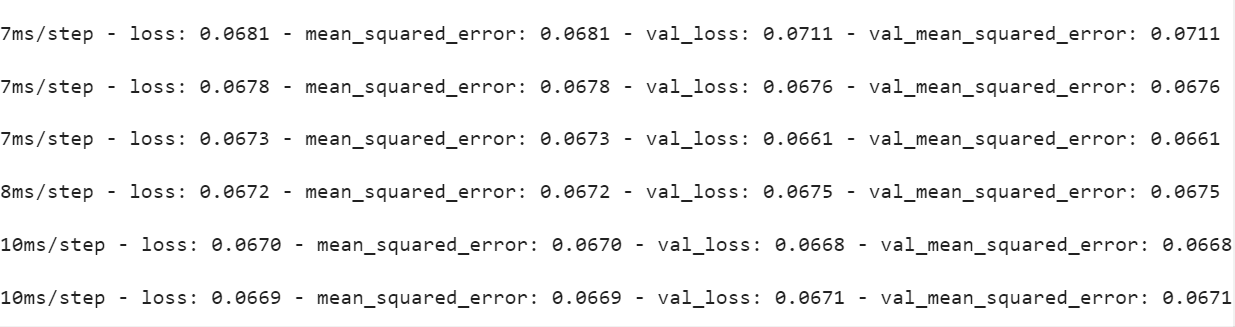


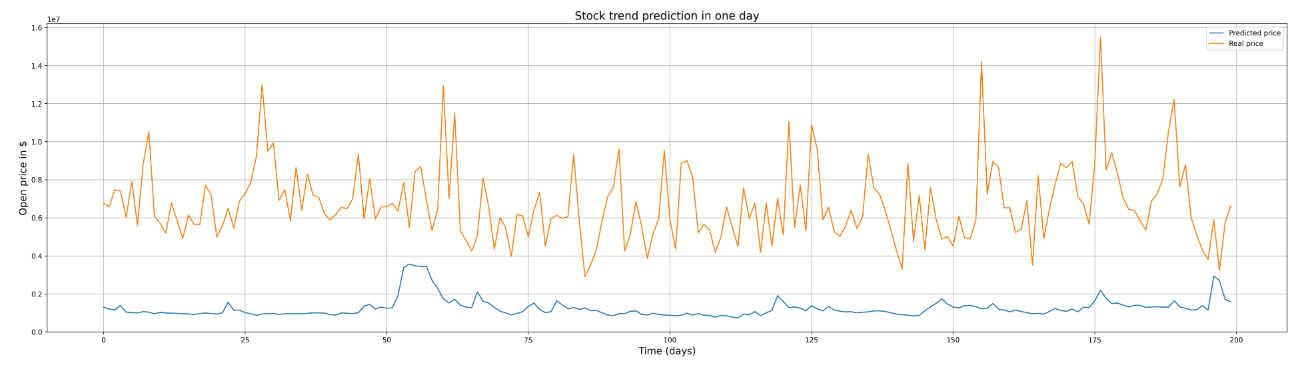




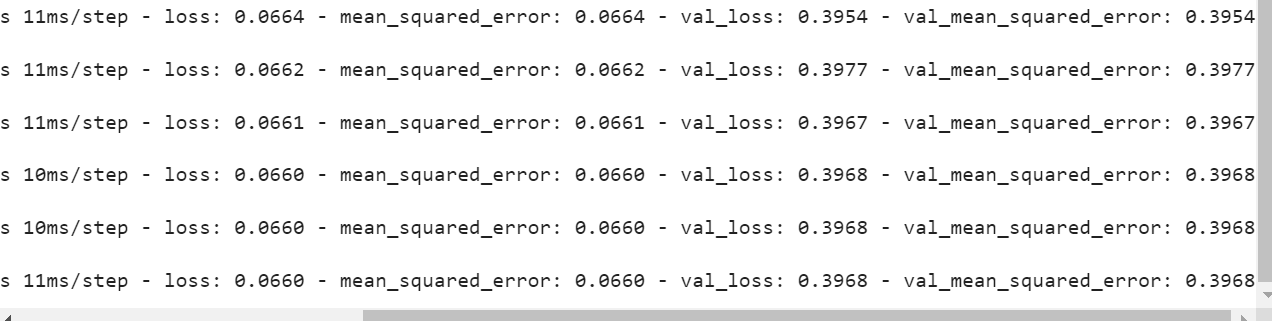
I continue to employ the simple moving average method used in Task 3 for industry analysis here, but I realize that it is not meaningful because it does not provide predictive value. Therefore, I tried to reuse the model for NASDAQ in task 1, 2 here.

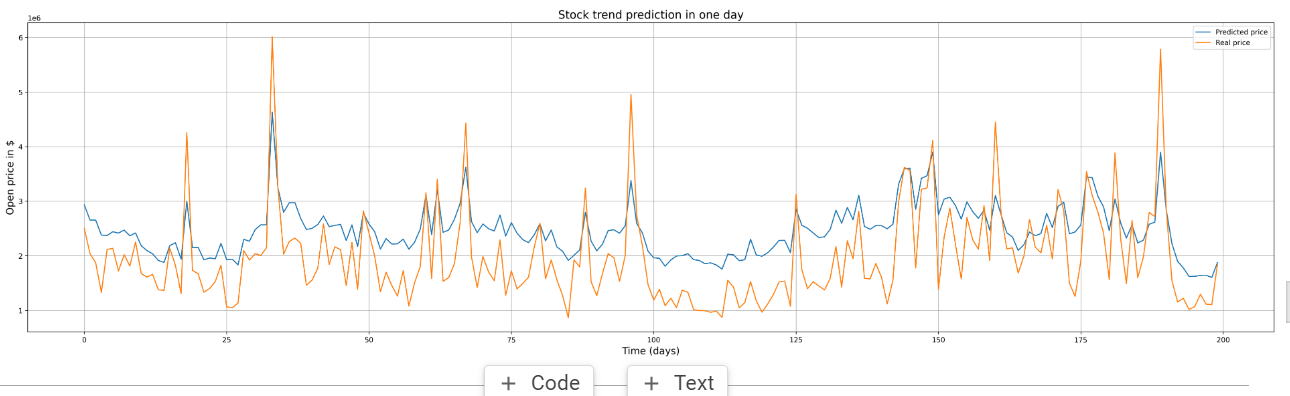
First, I only chose companies in the list NASDAQ 100 for higher credibility and fewer outliers, then filter out companies by industry. I experimented with two industries, biotechnology (6 companies) and soft drink (2 companies) first. Results are as follows: **Biotechnology:**





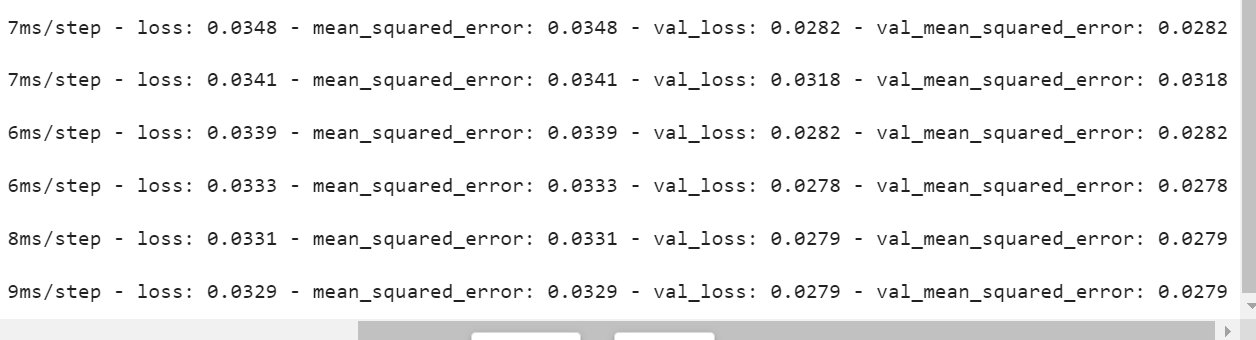
Soft drink:



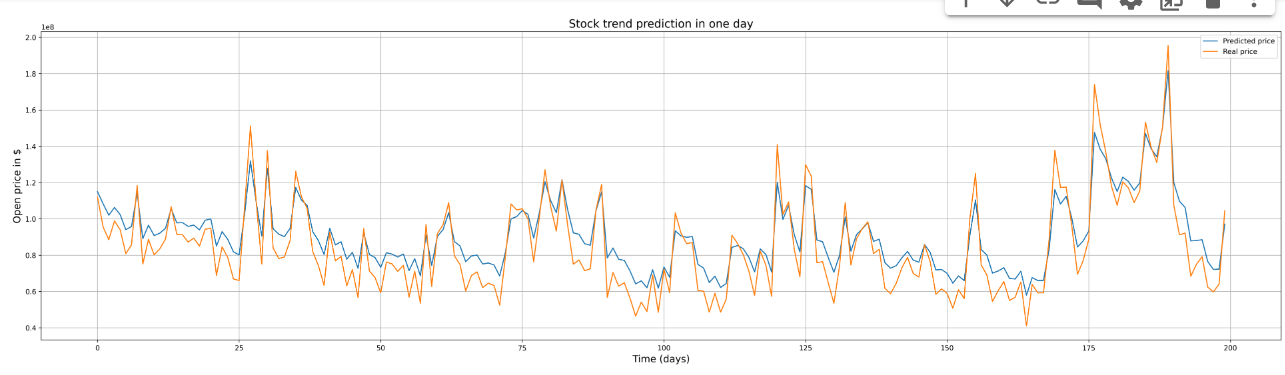


For individual industries, the MSE is still quite high. I assume that the more and higher-quality input data, the better the performance is.

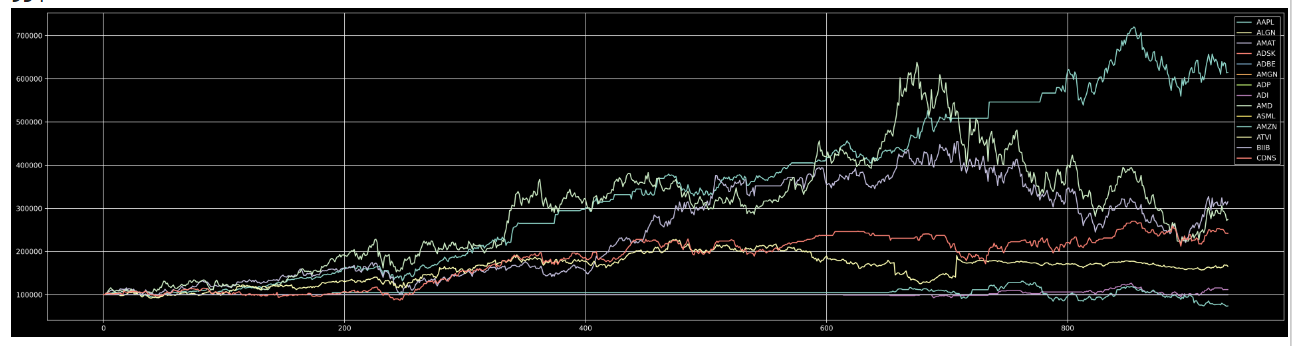
Therefore, I decided to include all NASDAQ100 companies. However, as this exceeded the limit of Colab and my RAM, I randomly chose 15 companies from NASDAQ. The performance i**s significantly improved.**



MSE is reduced to 0.03 and 0.02 respectively.



Closely aligned prediction



Portfolio management simulation for these random companies.

• What is the list of companies to hold? What is the profit within a certain  
period?

* AAPL and AMAT have the highest return rates.
* With initial fund of 100000 dollars, AAPL increased to 700000 USD, and AMAT around 300000 USD
* Therefore, it is advisable that AAPL and AMAT are to hold.

• What is the list of companies to get rid of?

With lowest (negative) performance, AMD and ADI are supposed to be gotten rid of.

What should be the list of companies to hold if investors are risk-taking or prudent?

Based on my experiment with NASDAQ and literature review, I hypothesize that risk-taking investors would hold technologies companies (ex: Apple (APPL), Tesla (TSLA),…), biotechnology (Moderna, Pfizer,…) or startup like Airbnb (ABNB)

Prudent investors should hold companies with a history of stable earnings and dividends (although I cannot take this into account in the training process yet) such as Coca Cola (KO), PepsiCo (PEP), AT&T.

The simulation also backs up this hypothesis, as tech companies will have much higher volatility, while utility companies and consumer staples will be much more stable in the long run.