TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**MACHINE LEARNING**

**FINAL PROJECT REPORT**

**…**

Lecturer: **Assoc. Prof. Lê Anh Cường**

Full Name: **Nguyễn An Khánh**

**Trương Trung Hiếu**

Student ID: **519H0107**

**519H0164**

Class: **19H50204**

Course: **23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**MACHINE LEARNING**

**FINAL PROJECT REPORT**

**…**

Lecturer: **Assoc. Prof. Lê Anh Cường**

Full Name: **Nguyễn An Khánh**

**Trương Trung Hiếu**

Student ID: **519H0107**

**519H0164**

Class: **19H50204**

Course: **23**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

**FIRST WORDS**

*Firstly, I sincerely thanks to Mr. Lê Anh Cường who is my Machine Learning instructor. During my learning process, I received a lot of useful advices and enthusiastic support from my instructor. Mr. Cường is the person who helps us take a few very little first steps in this course. From what I have learned and from my personal knowledge, I was able enough to finish my final report.*

*Because of my lack of experience as well as my limited knowledge, this report definitely cannot avoid the incorrectness. I willingly and hopefully to be received some comments from my instructor to make this report become more and more complete.*

*Lastly, I would like to greet you a good health, success and happiness.*

*Best regard.*

PROFESSOR’S EVALUATION

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Nguyễn An Khánh

Trương Trung Hiếu

**CONTENTS**

[**PROBLEM 1** 6](#_Toc138668866)

[**PROBLEM 2** 6](#_Toc138668867)

[**PROBLEM 3** 9](#_Toc138668868)

[**PROBLEM 4** 24](#_Toc138668869)

[**PROBLEM 5** 26](#_Toc138668870)

[**PROBLEM 6** 27](#_Toc138668871)

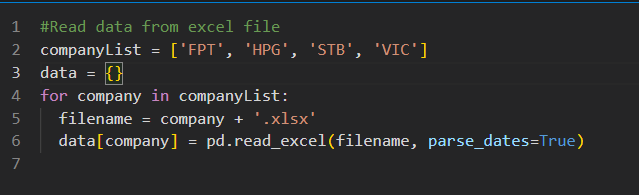
[**REFERENCES** 34](#_Toc138668872)

# **PROBLEM 1**

**Data Collection**

- We collect data from investing.com. We collect companies including: HPG, FPT, STB and VIC.

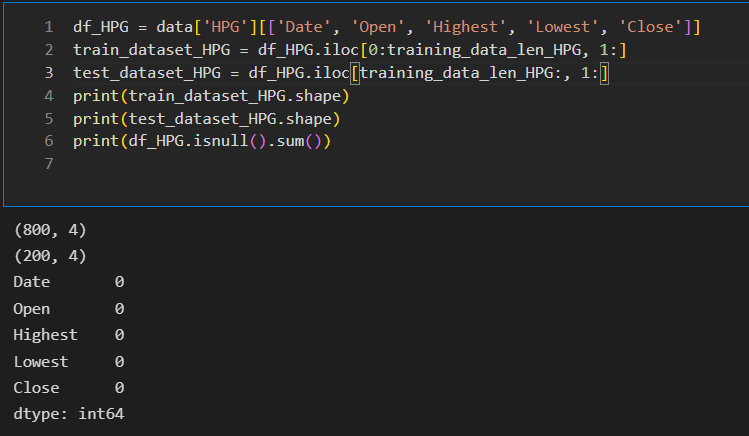
- Data is in excel format (.xlsx), it has 1001 lines of stock data from 1/3/2017 to 31/12/2020



# **PROBLEM 2**

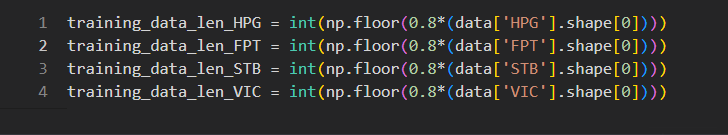
**1. Feature Selection**

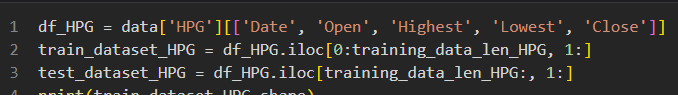
* In this problem, we will use Open, Highest, Lowest and Close Price.

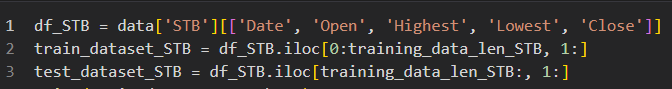
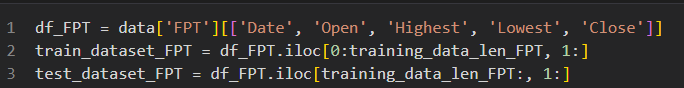


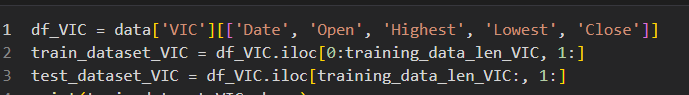
**2. Preprocessing data**

- Split data into train dataset and test dataset

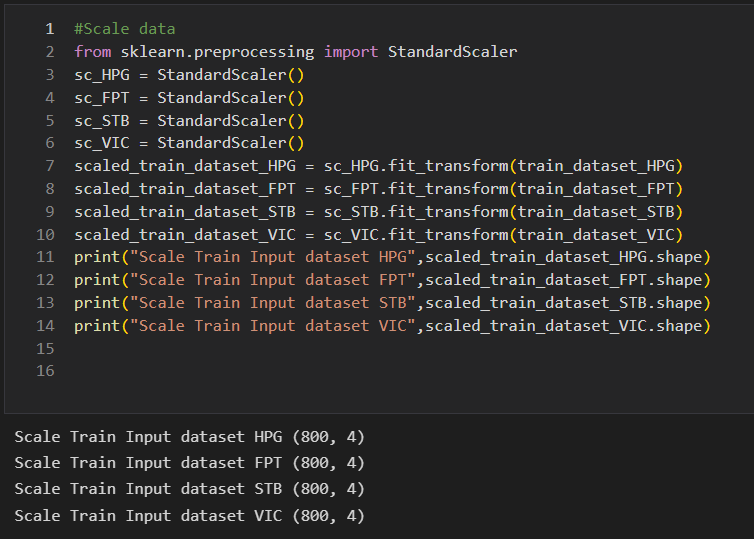




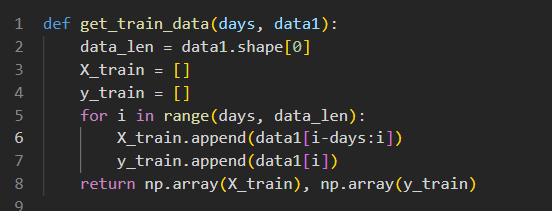




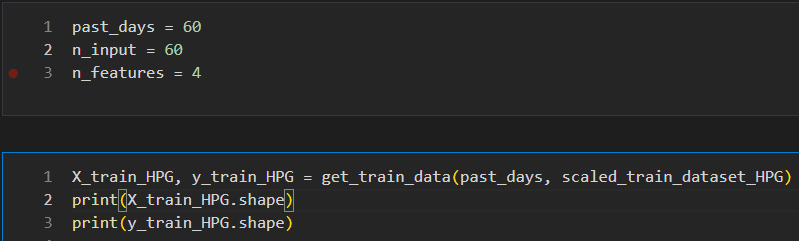
- Scaling dataset using StandardScaler.

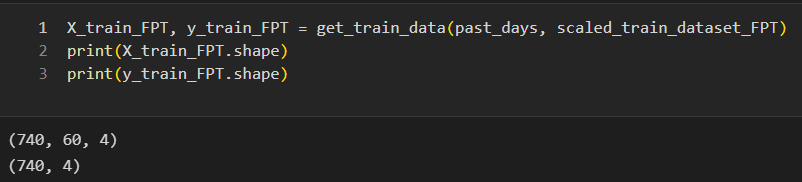


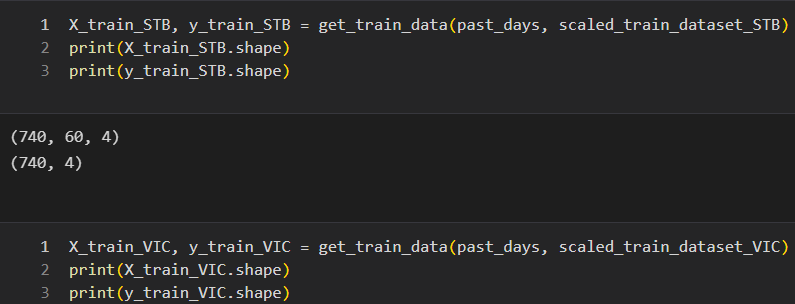
- Because this is time-series data, we must not use sklearn's train\_test\_split but use the below function. Here we use 60 days to predict 1 day.



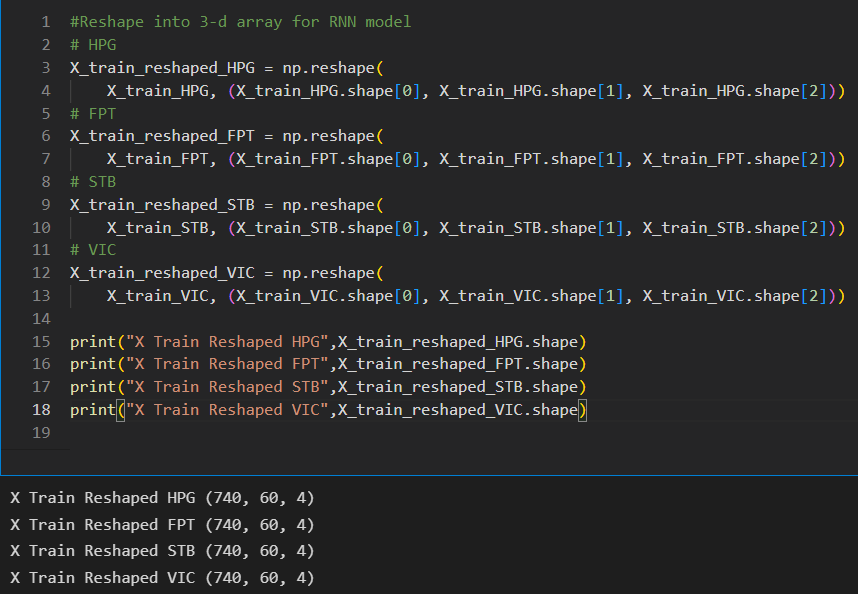
- Split X\_train and y\_train:







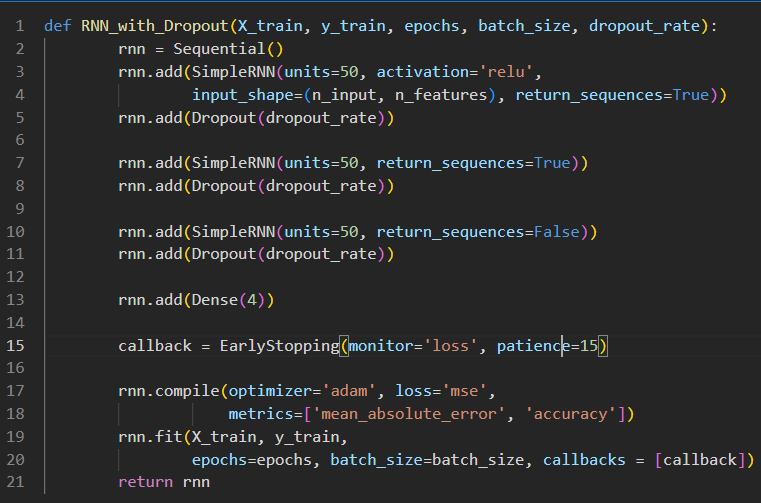
- Reshaped X\_train into 3-d for training:



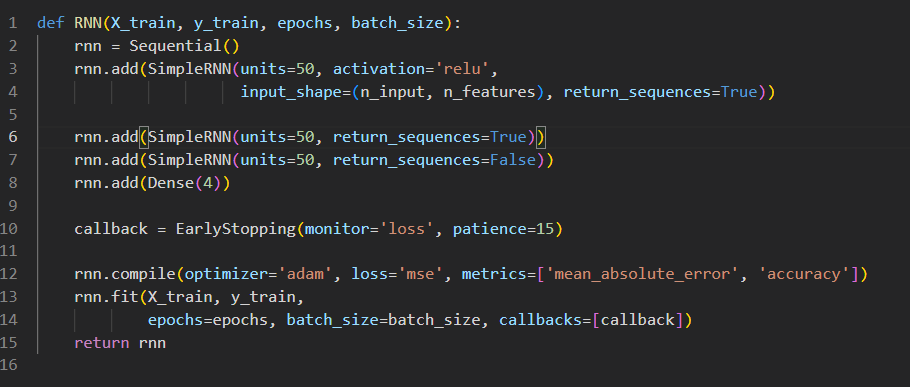
# **PROBLEM 3**

1. **Recurrent Neural Network (RNN)**

- We will compare using Dropout to avoid overfitting and not using Dropout and see how it turns out:

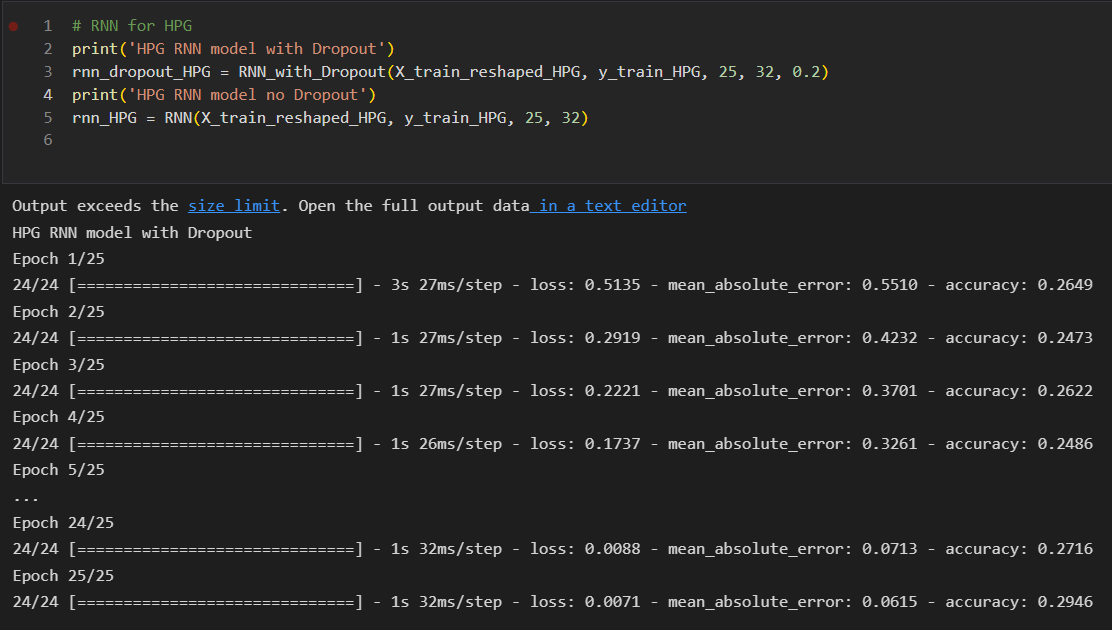


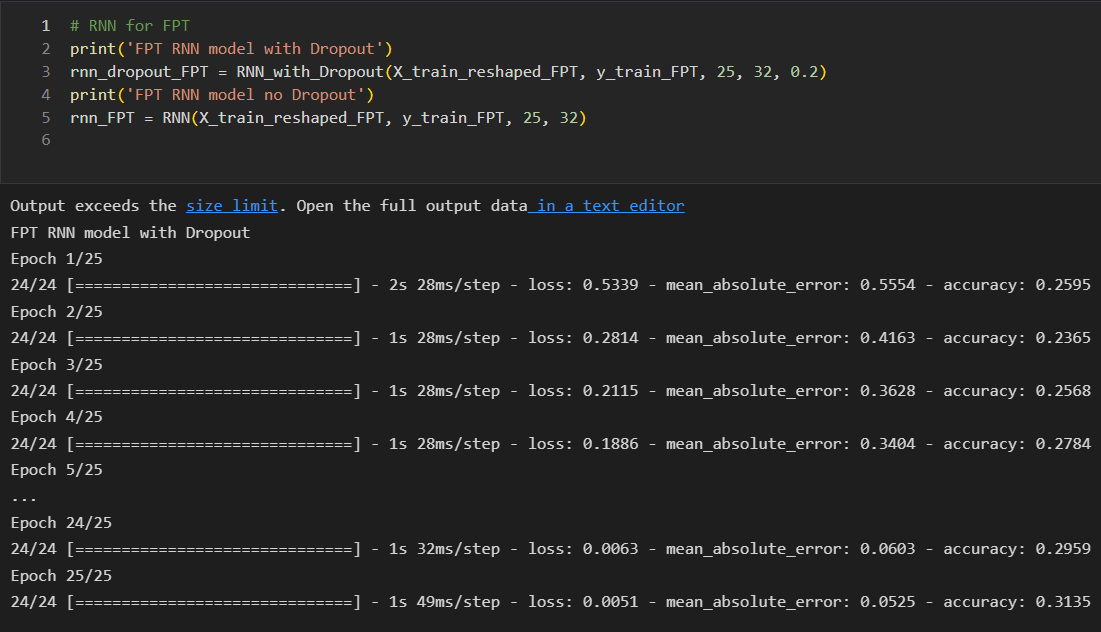
7 layers RNN (with Dropout)

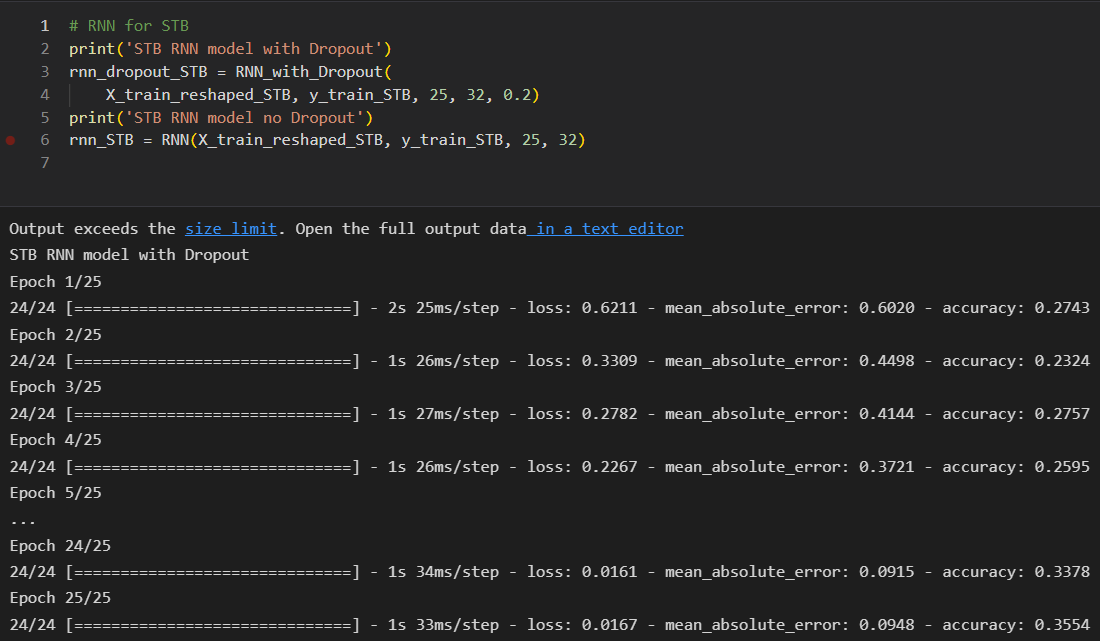


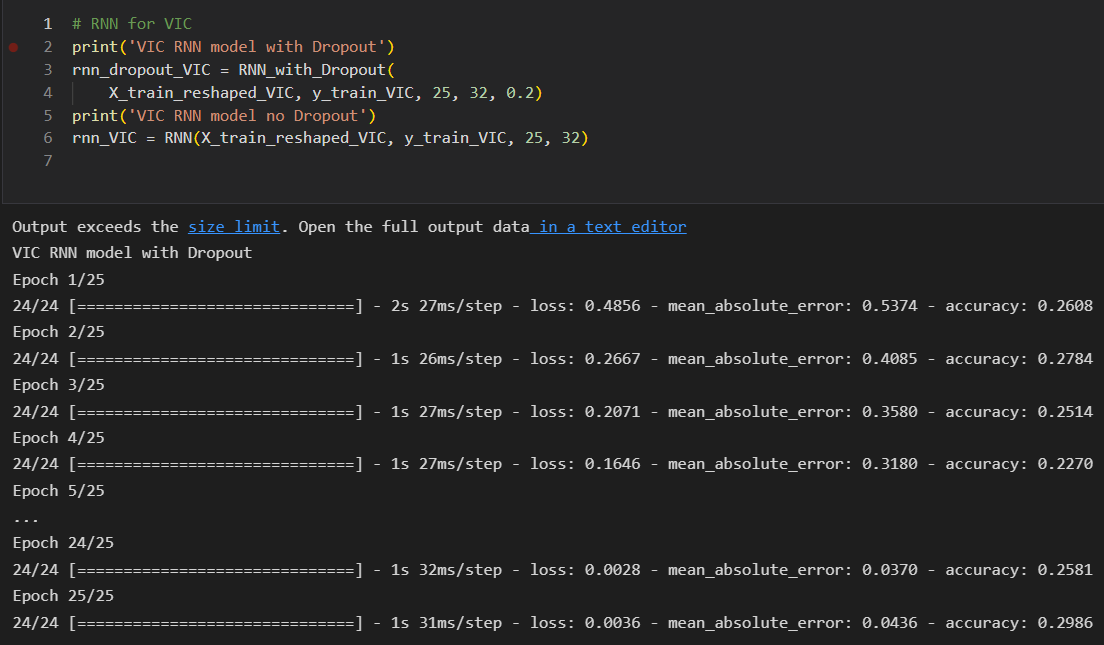
RNN Model with 3 layers

- Next, we just need to call the function to train the model:

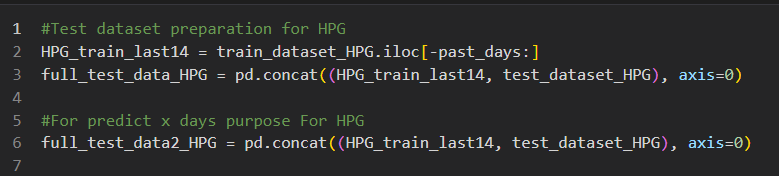


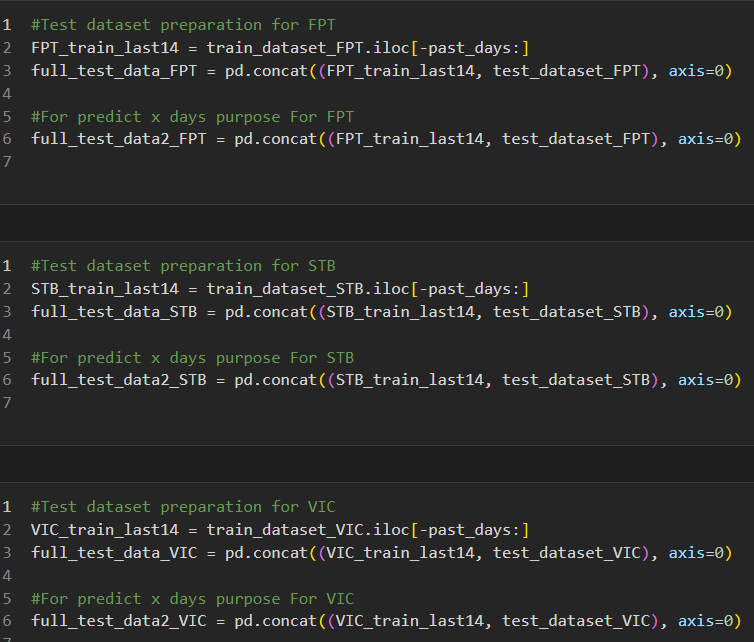




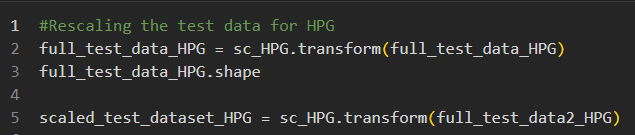


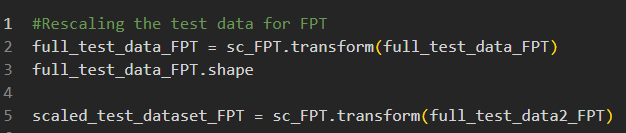
- Next, we need to handle test data like we as done above with train data, we also use last 14 days from traindata to add into test data.

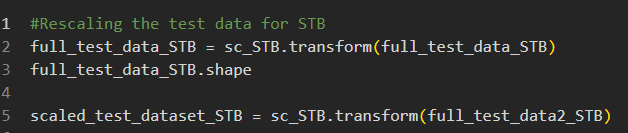


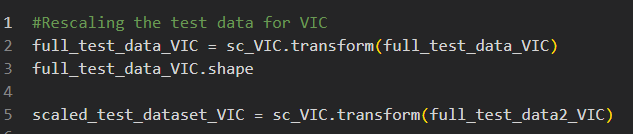


- Scaling test data:

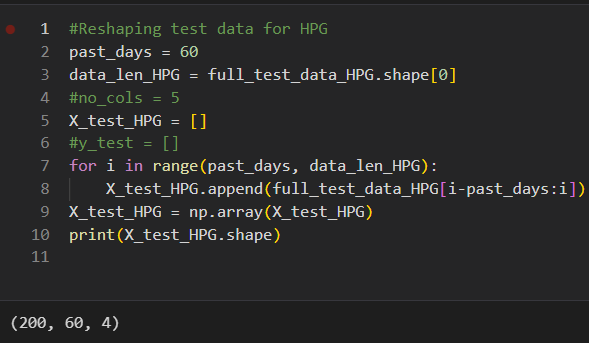


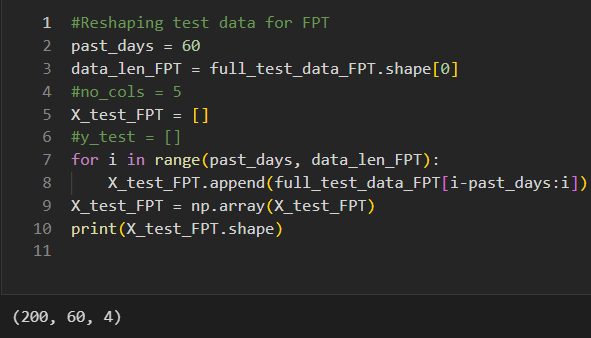


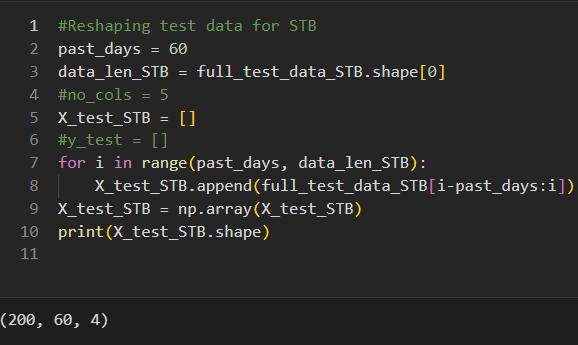


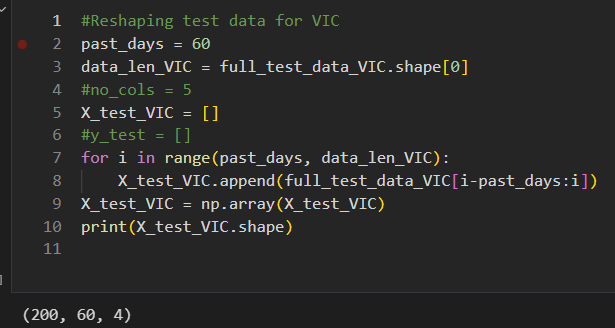


- Reshape test data for prediction:

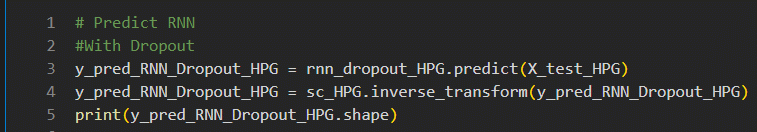


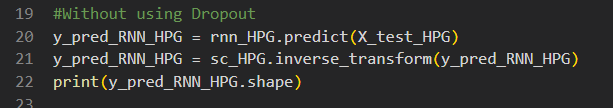






- Next, we will predict to see how the model work (Because this part is quite long, our group only sreenshot HPG, it all the same):

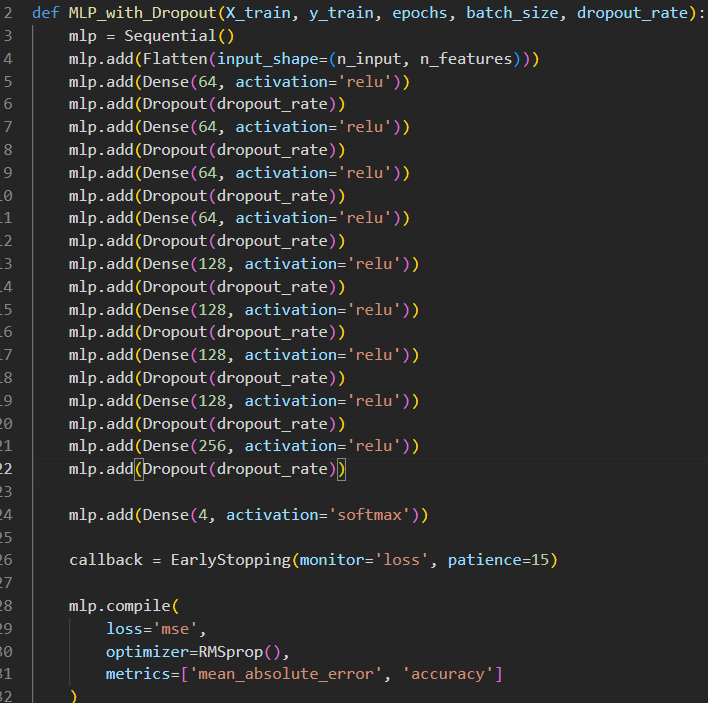




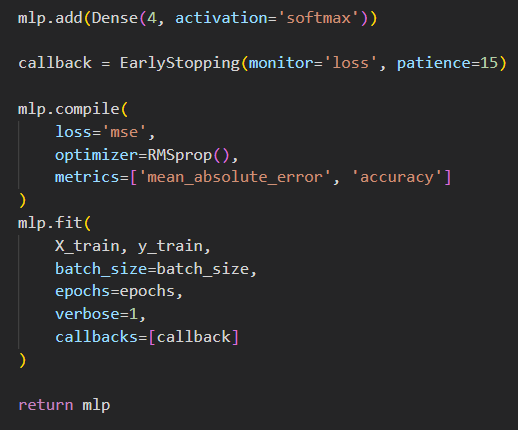
- The plot is below (in the MLP part)

1. **Multilayer Perceptron (MLP)**

- Like the RNN, we also create 2 function, 1 using Dropout and 1 is not. We use many layers in this part to increase the accuracy



19 layers MLP (including Flatten and Dropout)

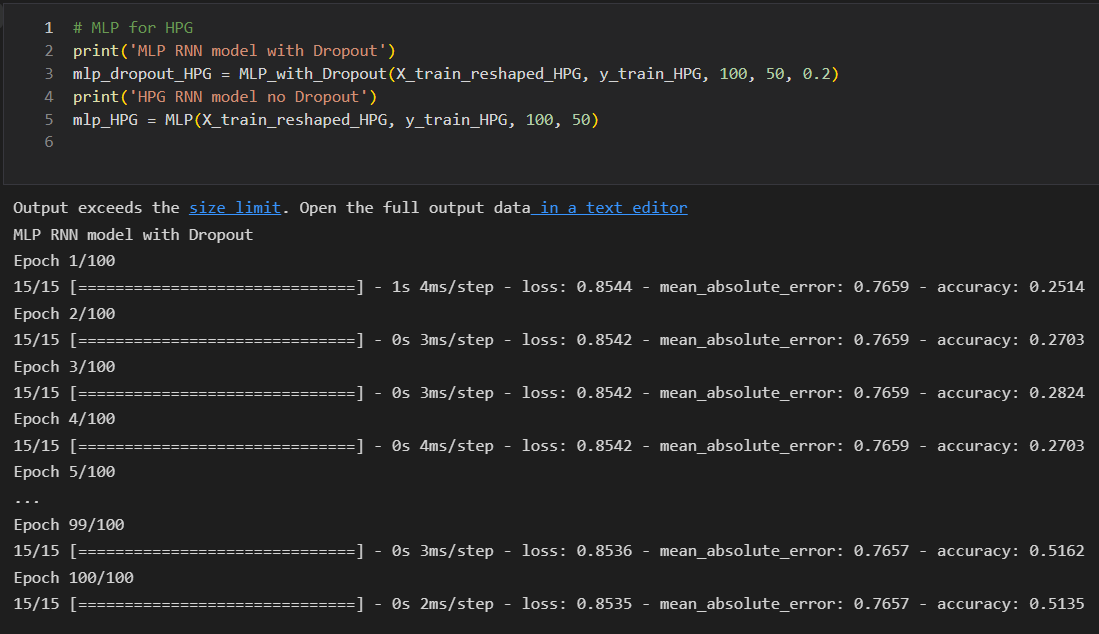


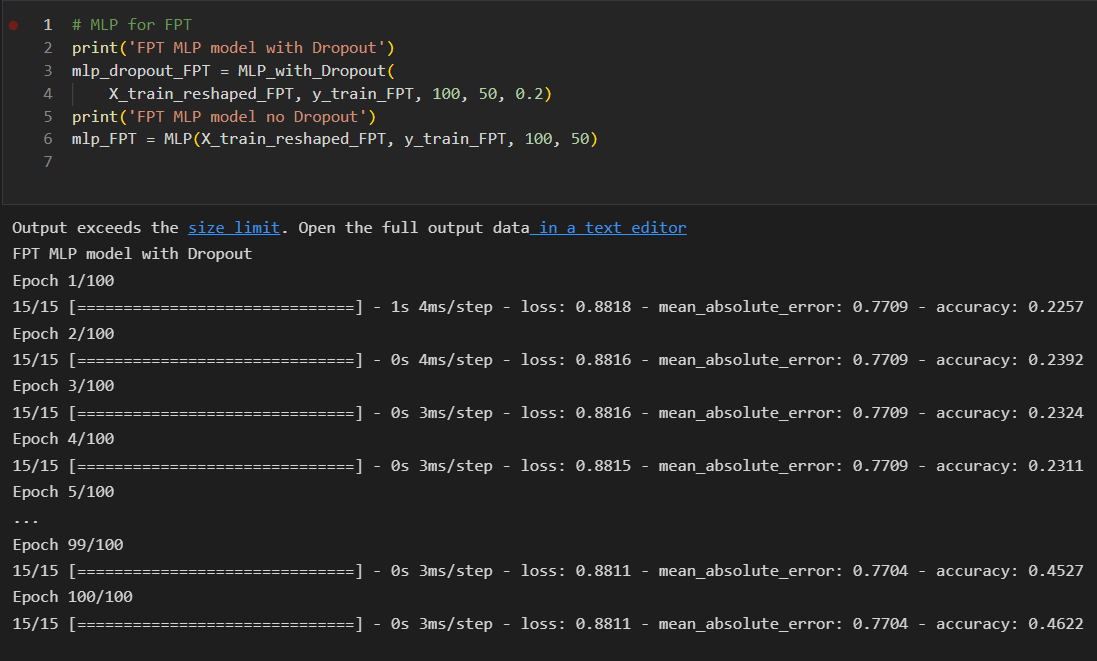
- The MLP without using Dropout:

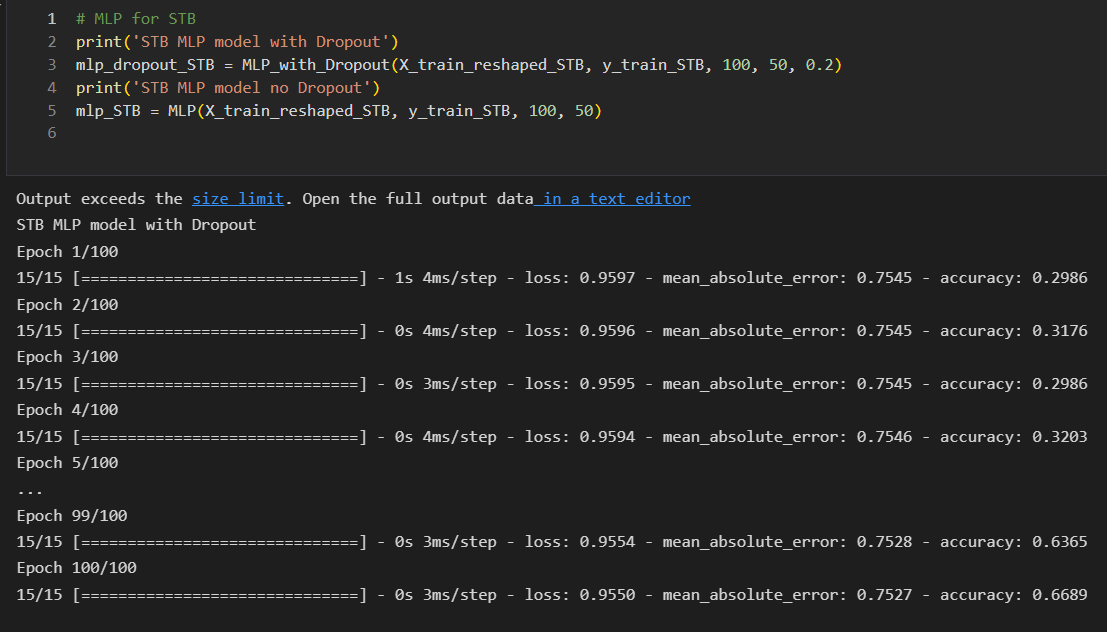


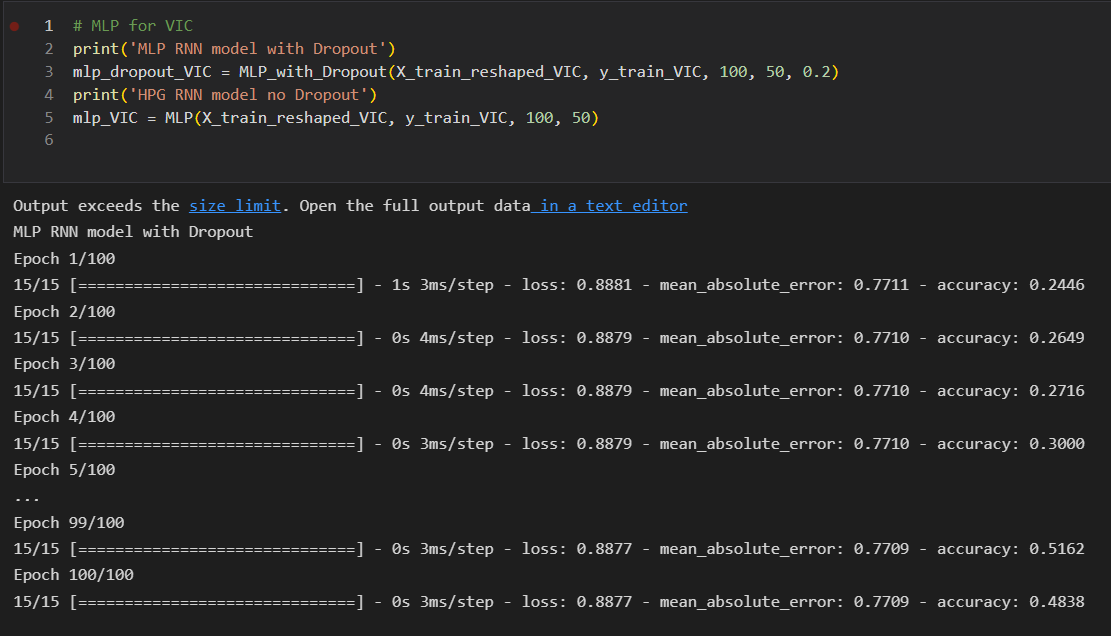
9 layers MLP

-Training model:

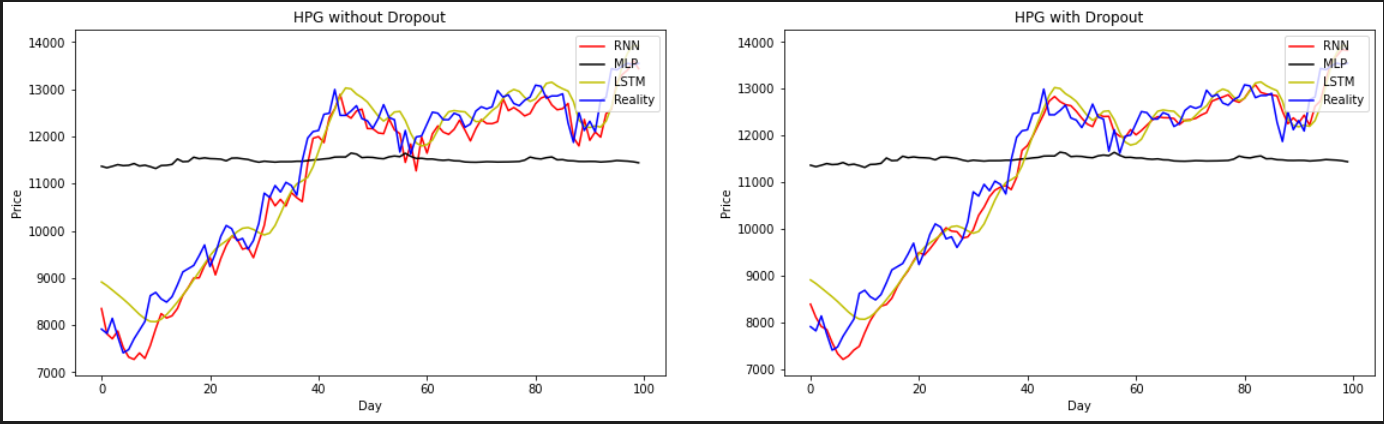


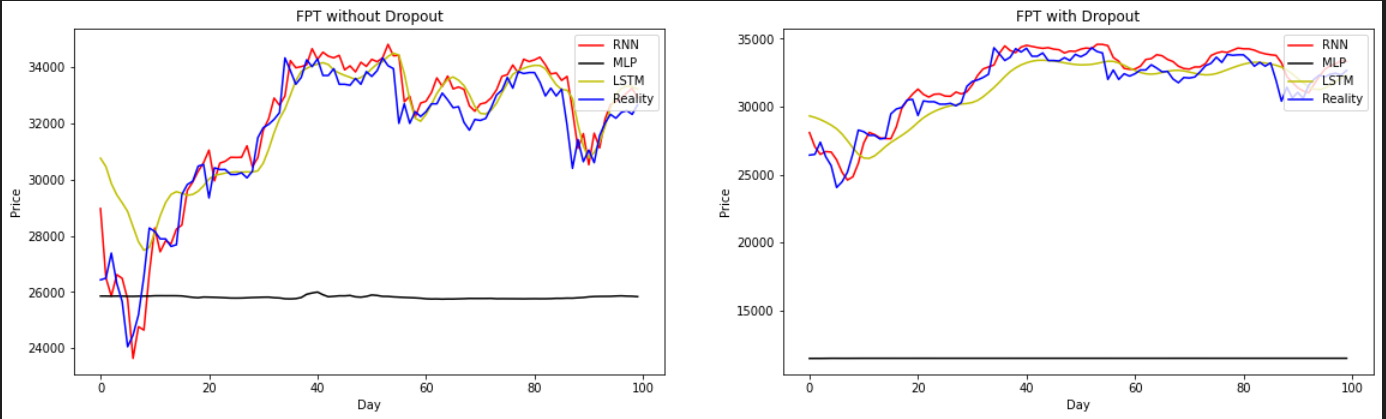


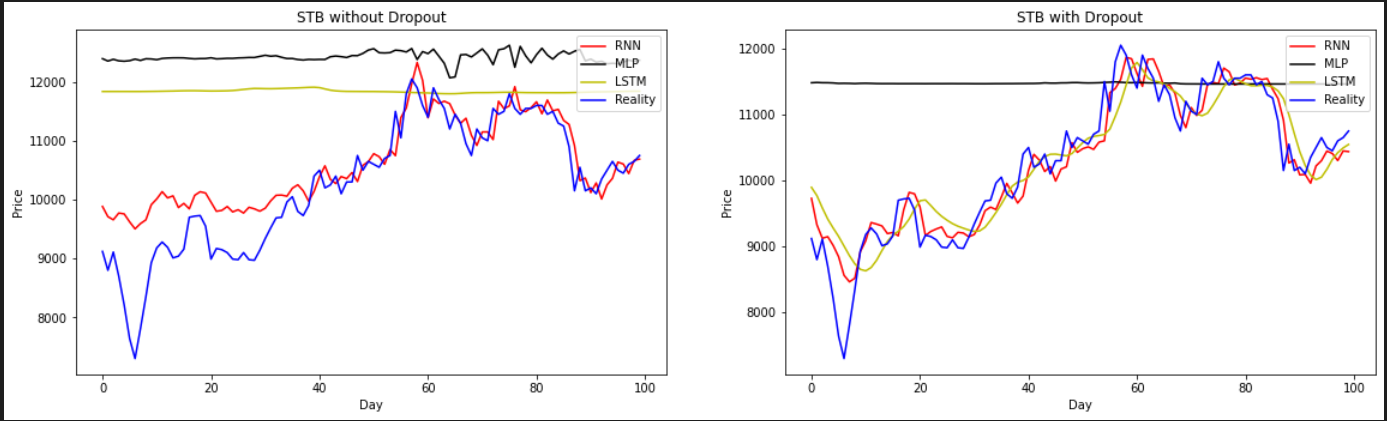


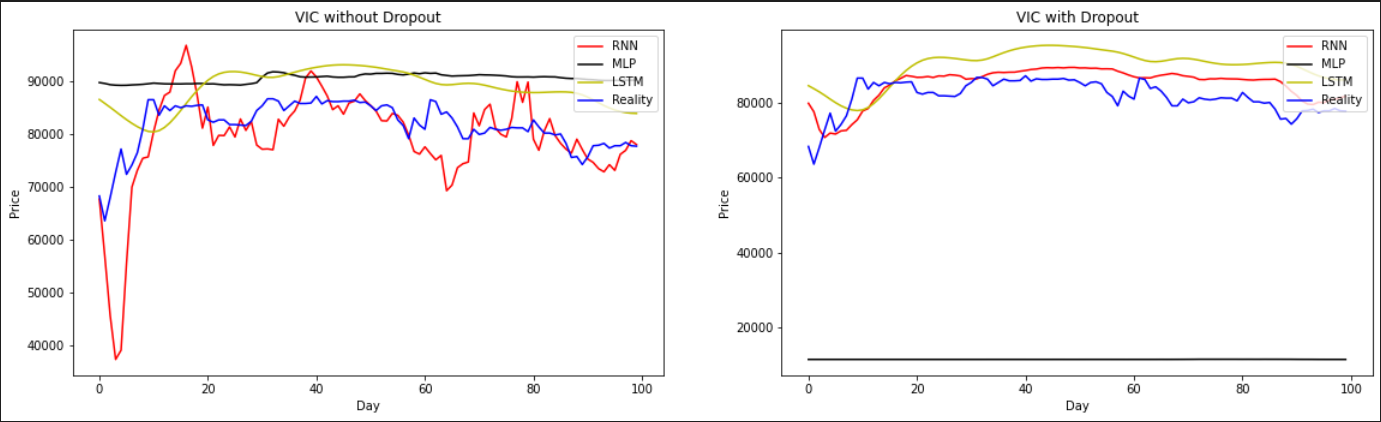


- We can see that the accuracy, MAE and loss varies depend on the data. Now we will see the result of RNN and MLP:









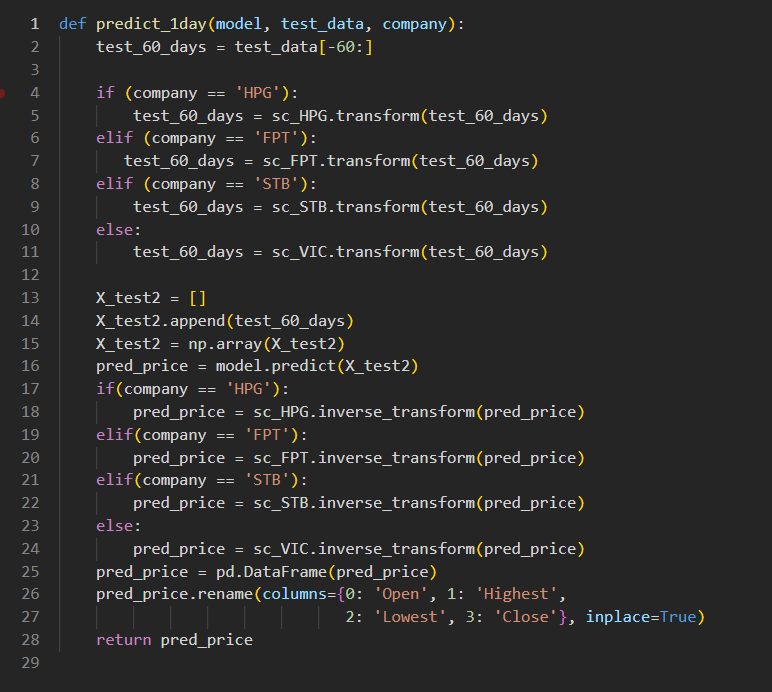
- We can see that HPG and FPT has the best performance. STB with Dropout is better without using it and VIC is the same.

- So we can see that Dropout play an important role in overfitting problem. It increased the accuracy a lot!

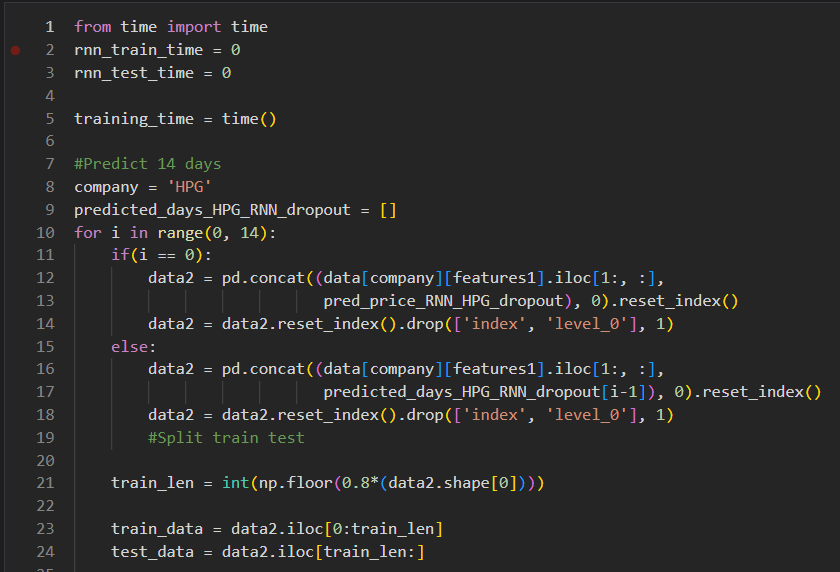
**3. Predict the next 14 days**

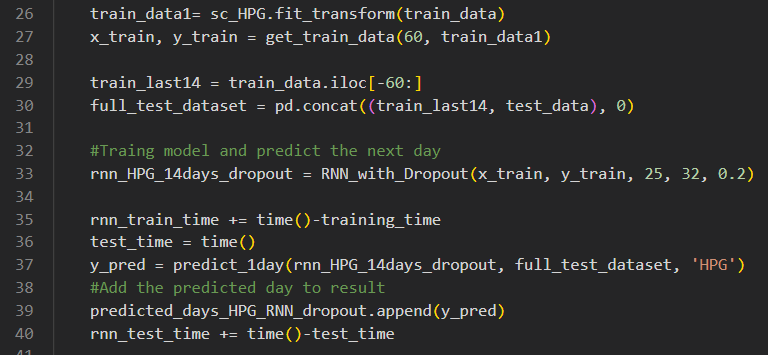
- To predict the next 14 days, we need to predcit 1 days first then we add this day to the data and drop the first row in the data (shift it to the right). Then we train the model again and predict the next next day.

- First we will create a function to predict 1 day:

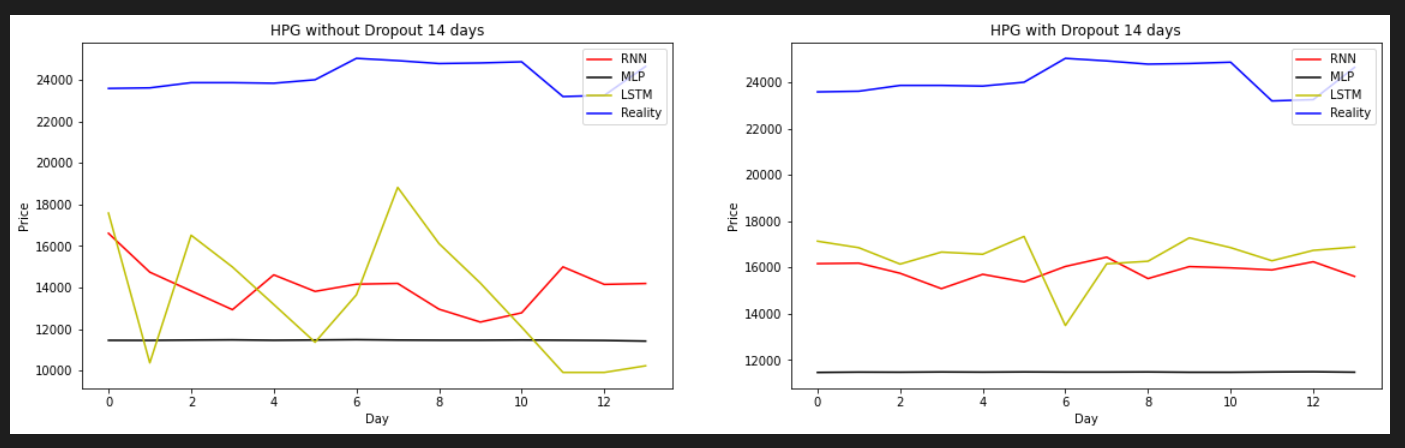


- Then we will predict the next 14 days using for loop:



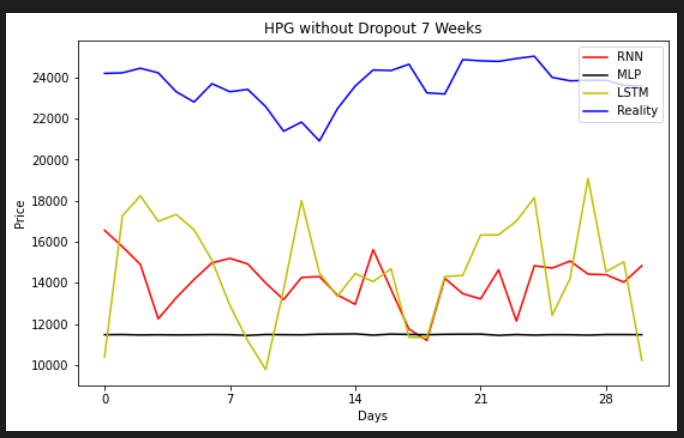


- This is the result:



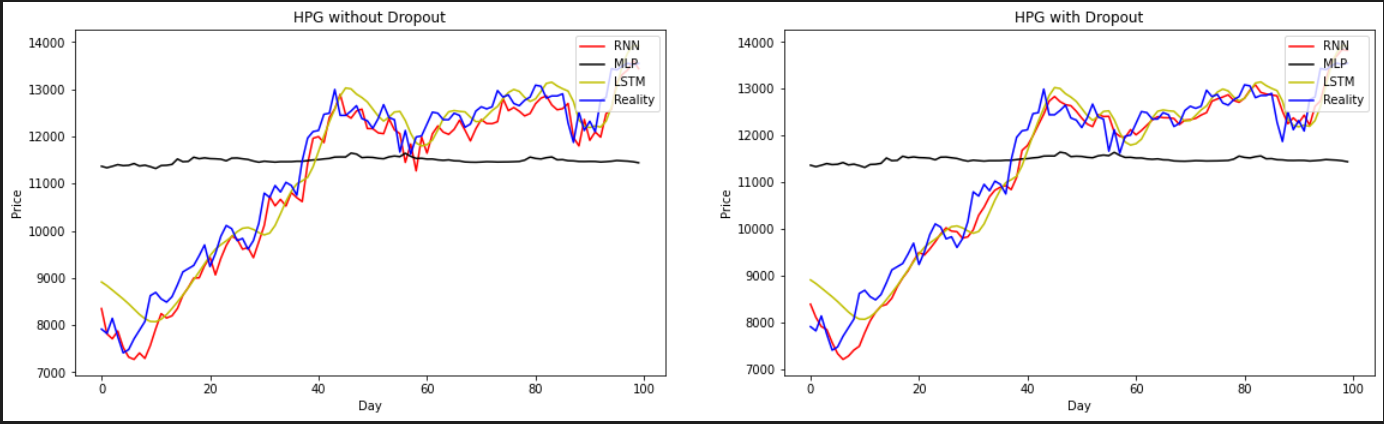
As we can see, the accuracy is very low because we are using predicted value by machine to predict the future. Although this is not accurate, we can predict the trend of a stock in the future.

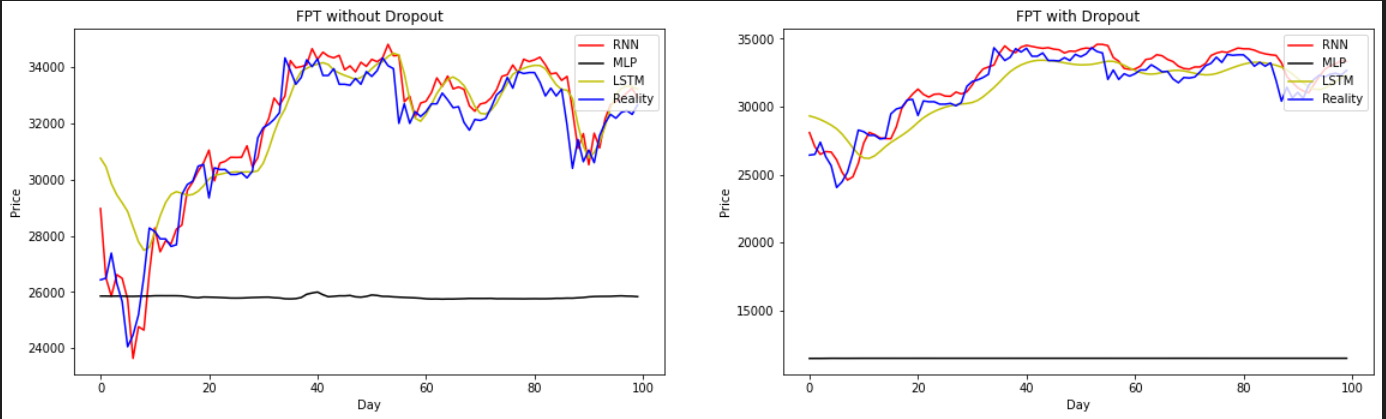
- This is the result of predicting 7 weeks:

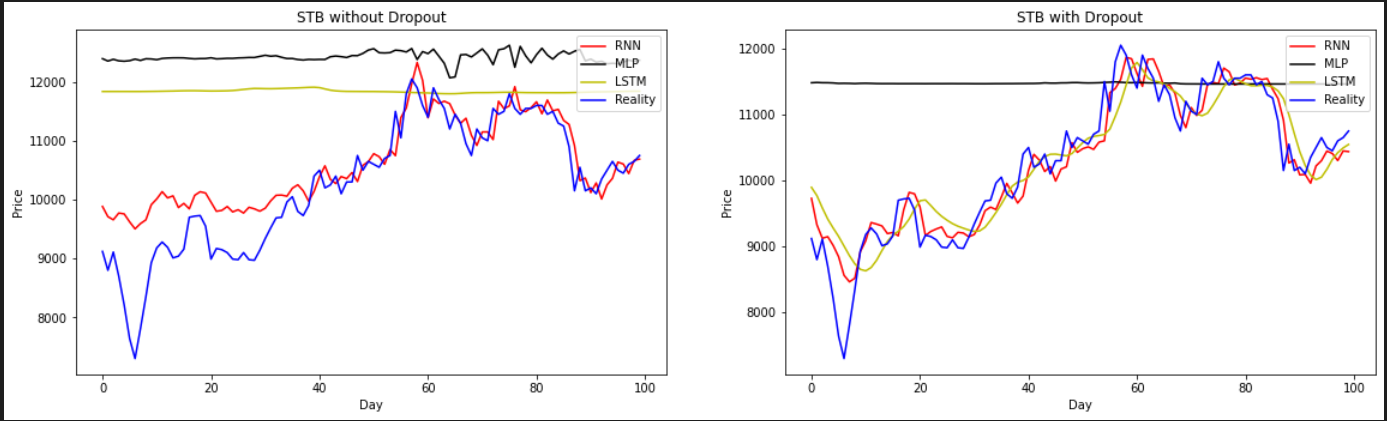


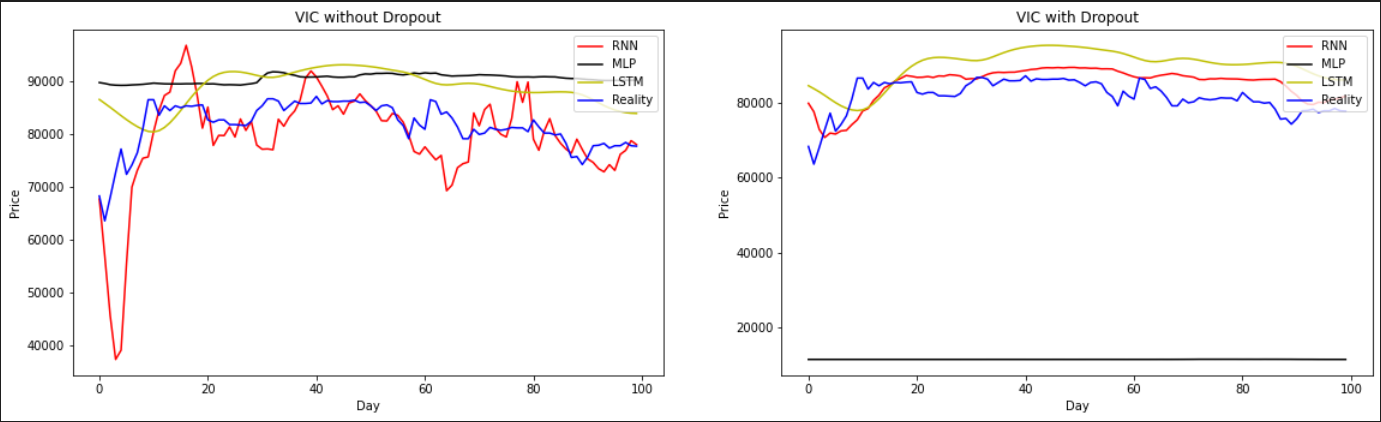
# **PROBLEM 4**

- As mentioned in problem 3, our group prevent overfitting using Dropout (=0.2) and Early Stopping technique with patience = 15







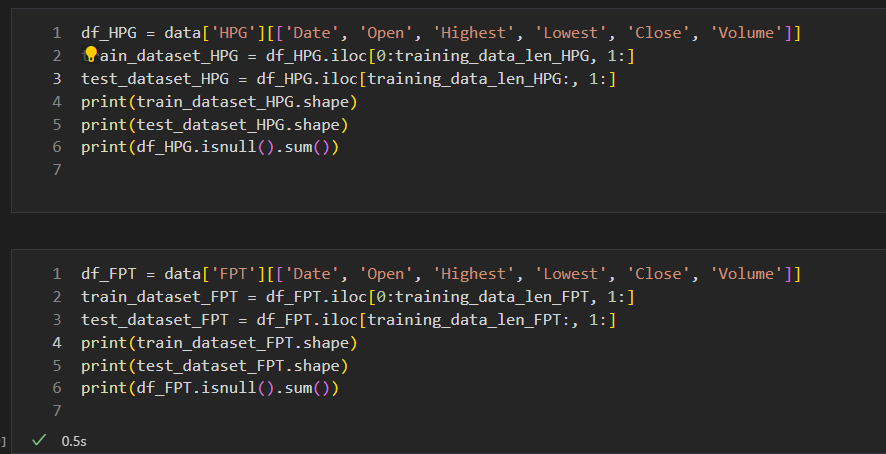


- We can see the result here, without dropout, the result is very bad, especcially with STB and VIC.

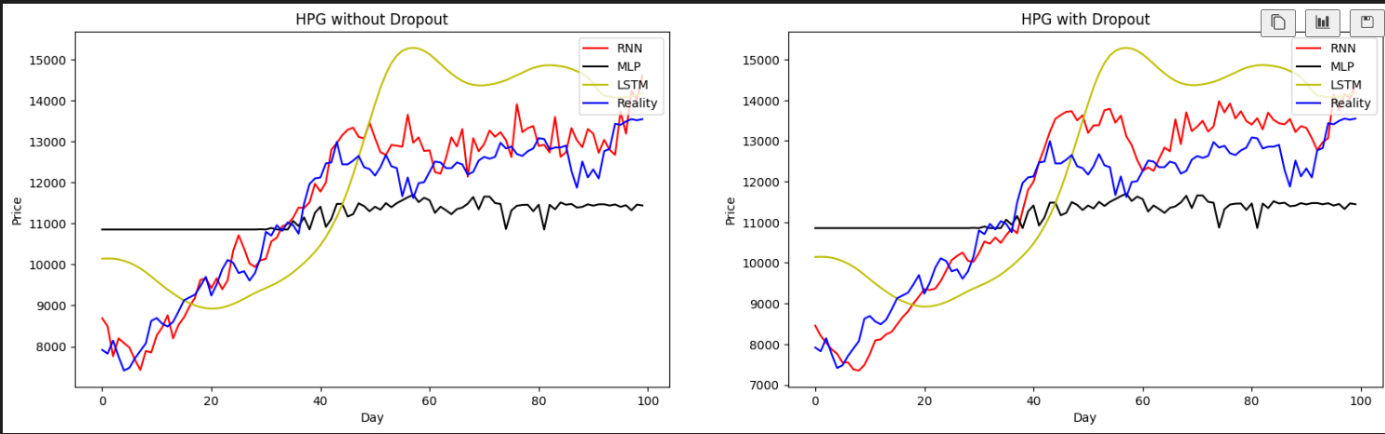
- With LSTM, we can see that in STB company, if we not using Dropout, it look like a straight line.

# **PROBLEM 5**

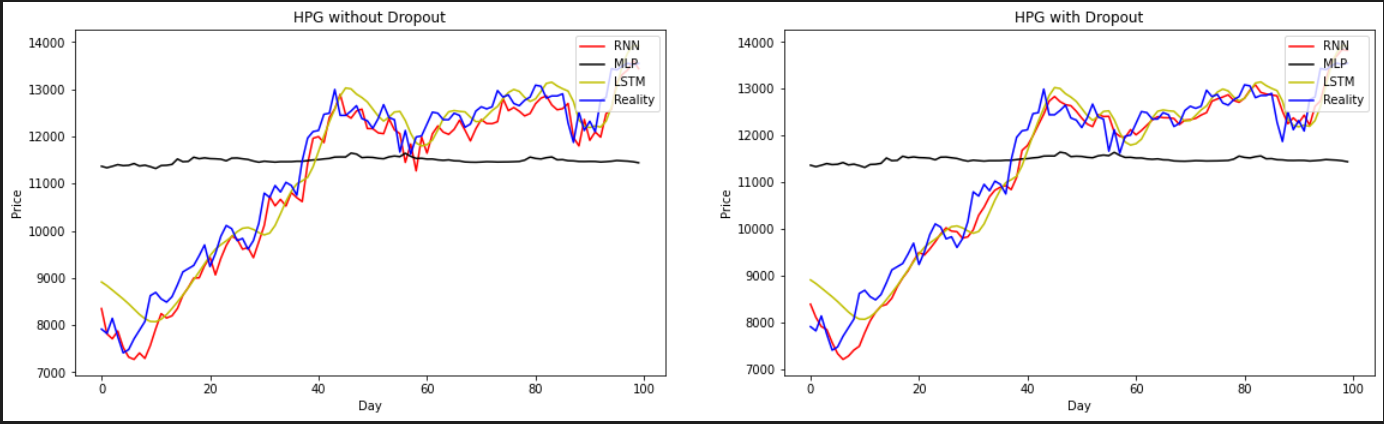
- In this problem, we will add ‘Volume’ feature into our dataset to see how it affect the result:



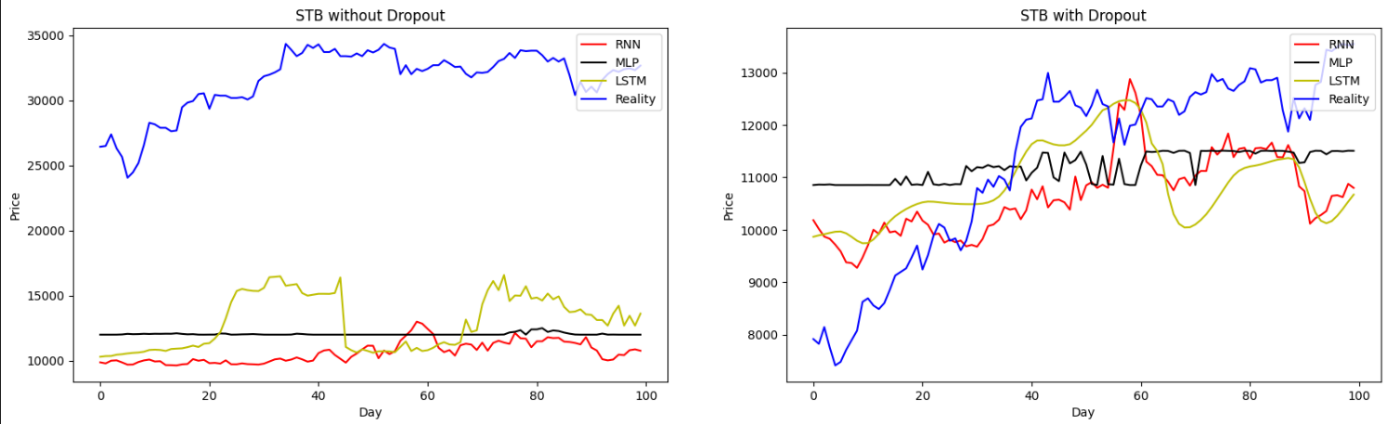
This is the result with ‘Volume’ feature added:



This is the result without ‘Volume’ feature:



- We can see that with Volume feature, the result is very bad, even with adding Dropout when training model. We can see this more clearly with STB company:



As we can see without Dropout the result is very bad.

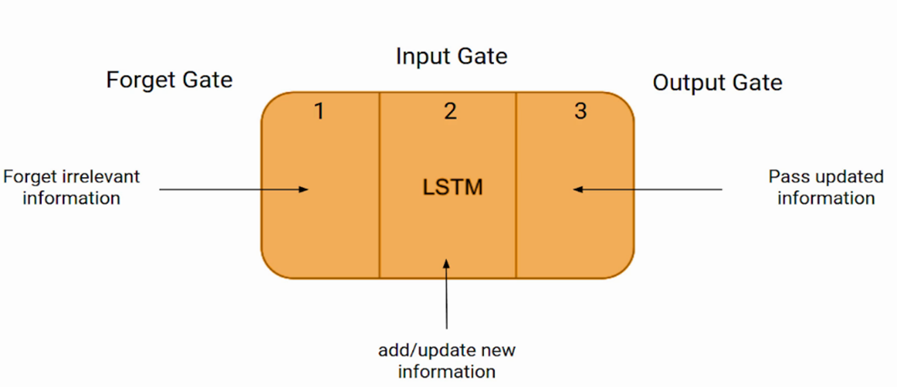
# **PROBLEM 6**

1. **Long short term memory**
   1. **Definition**

To solve the problem of Vanishing and Exploding Gradients in a Deep Recurrent Neural Network, many variations were developed. One of the most famous of them is the Long Short Term Memory Network (LSTM). LSTM recurrent unit tries to “remember” all the past knowledge that the network is seen so far and to “forget” irrelevant data. This is done by introducing different activation function layers called “gates” for different purposes.

* 1. **LSTM model**

The LSTM consists of three parts (Gates), as shown in the image below and each part performs an individual function.

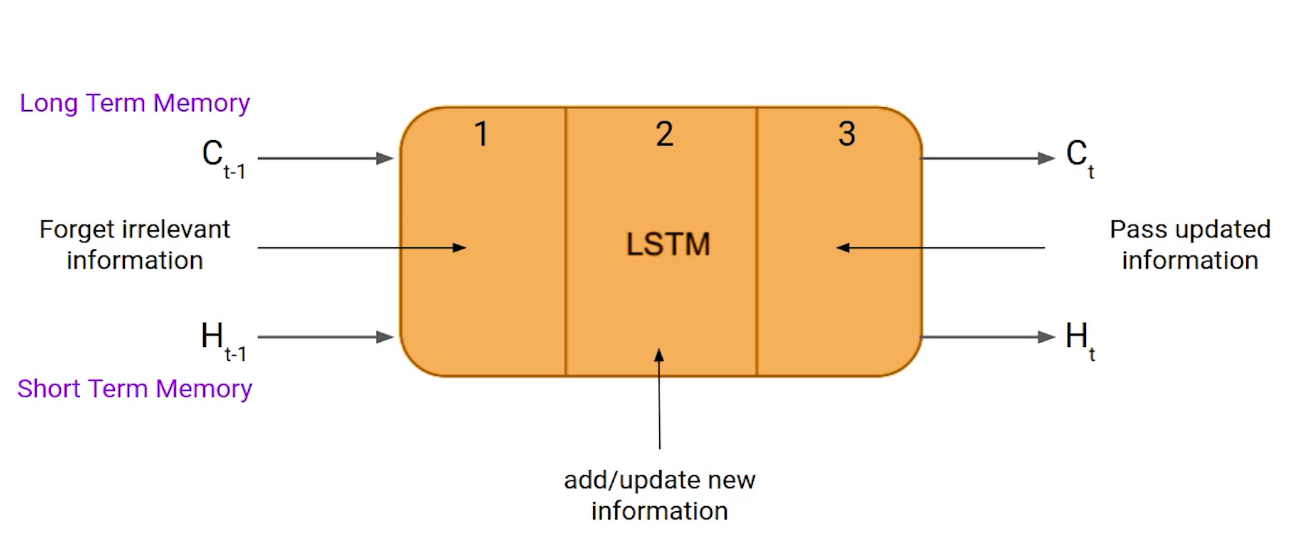


Picture 1. LSTM model

* **The first part** chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. (**Forget gate**)
* **In the second part**, the cell tries to learn new information from the input to this cell. (**The Input gate**)
* **In the third part**, the cell passes the updated information from the current timestamp to the next timestamp. (**The Output gate**)

Similar with a simple RNN, an LSTM also has a hidden state where H\_{t-1} represents the hidden state of the previous timestamp and H\_t\ is the hidden state of the current timestamp. In addition to that LSTM also have a cell state represented by C\_{t-1} and C\_t for previous and current timestamp respectively.

Here the hidden state is known as “Short term memory” and the cell state is known as “Long term memory”. Refer to the following image.

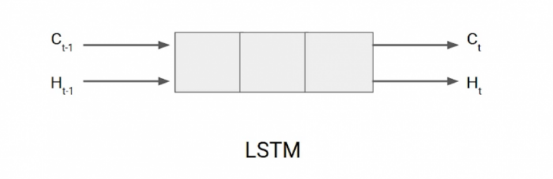


Picture 2. LSTM model and states

It is interesting to note that the cell state carries the information along with all the timestamps.

* 1. **Example**

Example sentences: **Today is a good day. Tomorrow will be a rainy day.**



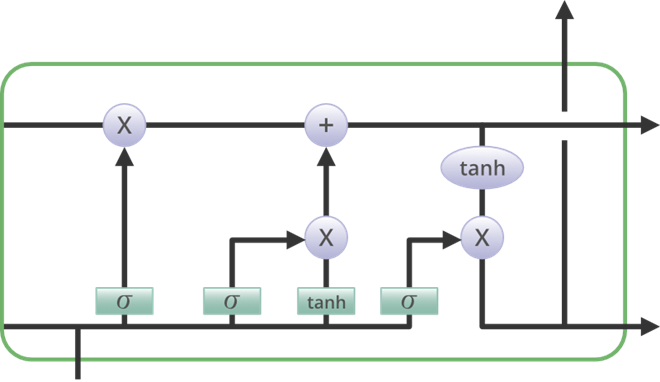
Picture 3. LSTM example

Here we have two sentences separated by a full stop. The first sentence is “Today is a good day” and the second sentence is “Tomorrow will be a rainy day”. It is very clear that the first sentence we are talking about “**Today**” and as soon as we encounter the full stop(.) we started talking about “**Tomorrow**”.

As we move from the first sentence to the second sentence, our network should realize that we are no more talking about **Today**. Now our subject is **Tomorrow**. Here, the Forget gate of the network allows it to forget about it.

* 1. **Structure**

LSTM has a chain structure that contains four neural networks and different memory blocks called cells.

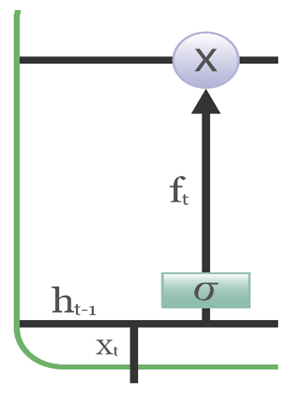


Picture 4. LSTM model structure

Information is retained by the cells and the memory manipulations are done by the **gates**. There are three gates:

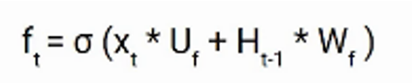
* + 1. **Forget gate:**

The information that is no longer useful in the cell state is removed with the forget gate. Two inputs (input at the particular time) and (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.



Picture 5. LSTM’s forget gate structure

**Formula:**



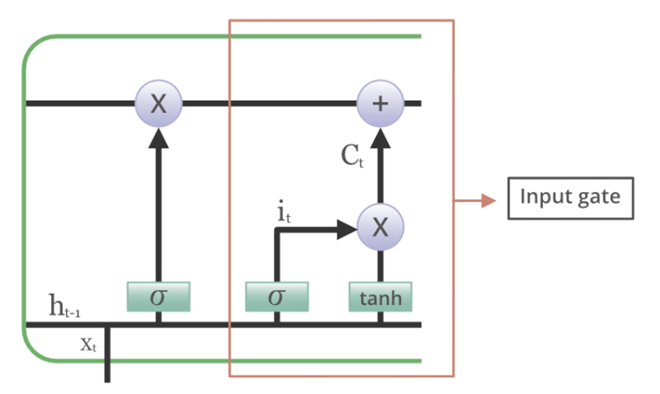
Note that:

* : input to the current timestamp.
* : weight associated with the input
* : The hidden state of the previous timestamp
* : It is the weight matrix associated with hidden state

**For example**: Today is a good day. Tomorrow will be a rainy day. The first sentence was talking about **“today”** and after a full stop, the network will encounter **“tomorrow”**, in an ideal case the network should forget about **“today”.**

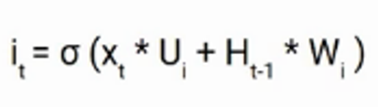
* + 1. **Input gate:**

The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs and . Then, a vector is created using *tanh* function that gives an output from -1 to +1, which contains all the possible values from and . At last, the values of the vector and the regulated values are multiplied to obtain the useful information

****

Picture 6. LSTM’s input gate

**Formula:**

****

Note that:

* : Input at the current timestamp t
* : weight matrix of input
* A hidden state at the previous timestamp
* : Weight matrix of input associated with hidden state

Again, we have applied sigmoid function over it. As a result, the value of I at timestamp t will be between 0 and 1.

**For example:** Lan knows how about machine learning. She chatted with me on the phone that she had learnt IT for 3 years.

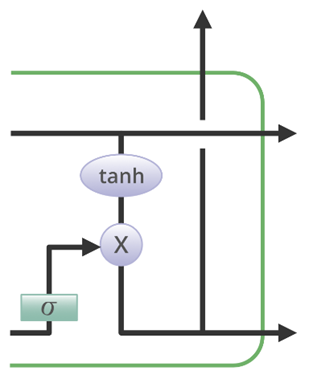
In these two sentences, we are talking about Lan. Both sentences give different kind of information about her.

* First sentence, we talk about Lan know ML
* Second sentence, Lan chat with me and learn IT for 3 years.

Based on the context given in the first sentence, which information of the second sentence is critical. In the second sentence, we all see that first clause is about how Lan transfer her information to me so which mean the next clause would be some vital information. The fact that learnt IT for 3 years is important and this is the task of the Input gate.

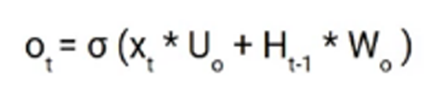
* + 1. **Output gate:**

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs and. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

****

Picture 7. LSTM’s output gate structure

**Formula:**

****

Its value will also lie between 0 and 1 because of this sigmoid function. Now to calculate the current hidden state we will use and tanh of the updated cell state. As shown below.

Ot and tanh

It turns out that the hidden state is a function of **“Long term memory” ()** and the current output. If you need to take the output of the current timestamp just apply the softmax activation on hidden state .

SoftMax

Here the token with the maximum score in the output is the prediction.

* 1. **What is the main difference between RNN and LSTM**

The main difference between RNN and LSTM is in terms of which one maintain information in the memory for the long period of time.

# **REFERENCES**

[1] <https://towardsdatascience.com/stock-market-predictions-with-rnn-using-daily-market-variables-6f928c867fd2>

[2] <https://towardsdatascience.com/temporal-loops-intro-to-recurrent-neural-networks-for-time-series-forecasting-in-python-b0398963dc1f>

[3]<https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/>

[4]<https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>