Dashboard Apple Stock Price Prediction

By - Einav Yaacov 316137918 , Asaf Ben-Menachem 313352049, Gal Sharbat 316072016.

Motivation

Predicting Apple stock price with machine learning. We will compare different models to get the most accurate results and how it can help Apple's stock traders.
We will create an user-friendly UI (as a dashboard) that will show the end user (Apple's stock traders) what is recommended for this day,to buy or sell the stock, and give him visual analysis of the prediction as a comparison to the real results and show trends of the stock using the hiristcul data and the results of the models.
We will compare the results of the model with similar companies' stocks to see how good the model is.

Method

Our method that we will use to find the model that can best predict Apple stock price is comparing few different models such as:

1. Deep Neural Networks.
   We are going to train a neural network that will predict (n+1) - the price using n known values (previous prices). We assume that the time between two subsequent price measurements is constant
2. Introduction to LSTMs for the time-series data
   LSTM, short for Long Short-term Memory, is an extremely powerful algorithm for time series. It can capture historical trend patterns, and predict future values with high accuracy.
3. And consider using more ML models as a comparison.

We will build a Dashboard as a tool to the trader that represents the current day status of the stock and if it is worth buying or selling according to our ML model.

In addition, we will show historical analysis on the stock price and on the models the accuracy of the model in relation to the prediction for that day and the actual result using public data and the same data analysis structure as we do for Apple stock and use APIs to build and update the database of the raw data.

We will create KPI’s to indicate if to buy or sell stocks and give a visual time plot in one dashboard with interactive graphs to help the user understand what is recommended to him on the date and give the user a view to the future results of the model and what it means to the user.

Project structure

API

Using Alpha Vantage API which is a method to obtain historical data for Stock markets, the cost limitation is 500 requests per day and because our project is based on daily data and not hours / Minutes it will be good enguth. The data that we will work on start’s from 1999-11-01 and has all the trading days in the US stock market which are usually Monday through Friday.

The following module shows two function **(attach 1)**:

1. get_apple _stock_api() which reads the private api from a file that is located out of the folder directory for obvious reasons.
2. get_apple_stock_data() uses (and calls) get_apple _stock_api function to get the data and the metadata from alpha_vantage and stores the data in another folder .

Prepare data

This part is for preparing the data in a general way that is relevant for all the use cases of the raw data. This part includes renaming the column names, changing formats and solving the stock split’s values change in the raw data, this was solved with creating a special function that gets the data, change value, and the date of the change and returns the “fixed” values according to today's stock preposition, this was done by searching the stock split days and the change value’s and converting each columns and rescale the old stock values to the current stock preparation value **(attach 2+3).**

Data Analysis

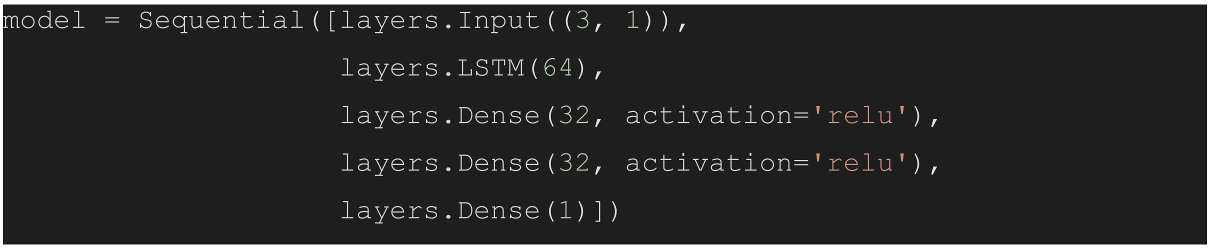
From the first view on the data per column we can conclude that the columns “Open” “Close” “Low” “High " behave very similar one to other specificly when watching all historical data from 1999. we can also observe that the stock behaviour has changed from 2009 **(attach 4)**, we can assume that previous data couldn't the current stock behaviour for today. So our conclusion is to refrence only the data starting from 2009. After dividing the data to 2009 and from 2009, we can observe that the price stock had begun diffrent behaviour from 2009 and this will be the data that we will work on **(attach 5)**.

After we divided the data from 2009 we checked again the columns “Open” “Close” “Low” “High” behaviour. Our conclusion was that they behave similar to columns before we divided. Therefore we are focused on date, close and volume columns.
Because the data came from a reliable source,we do not need to clean the data **(attach 6)**.

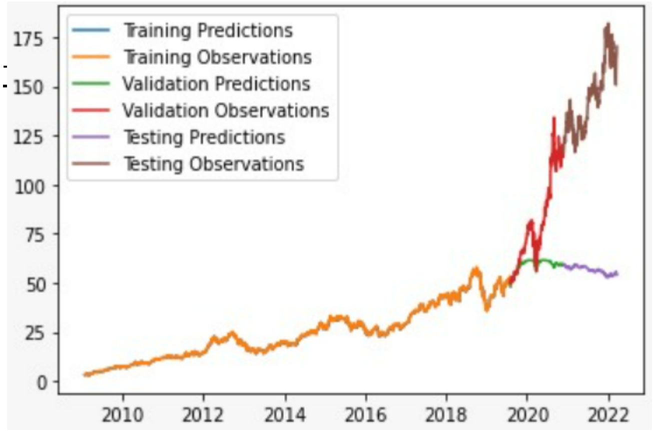
<u>LSTM Model</u>

LSTM, short for Long Short-term Memory, is an extremely powerful algorithm for time series. It can capture historical trend patterns, and predict future values with high accuracy.
Our project focuses on Stock prediction and the historical dada and events can benefit the prediction by memory unic events that may occur in the present sometimes these events occurred in the long past we still remember it to fit it.
The LSTM model contain three gates
1.forget gate- decides which information to throw away from the current cell state by using sigmoid function with output between 0 and 1.
2.input gate- It’s used to choose which new information gets added and stored in the current cell state. In this layer, a sigmoid function is implemented to reduce the values in the input vector (it), and then a tanh function squashes each value between [-1, 1] (Ct)
3.output gate- The output gate is implemented to control the output flowing to the next cell state. Similar to the input gate, an output gate applies a sigmoid and then a tanh function to filter out unwanted information, keeping only what we’ve decided to let through.

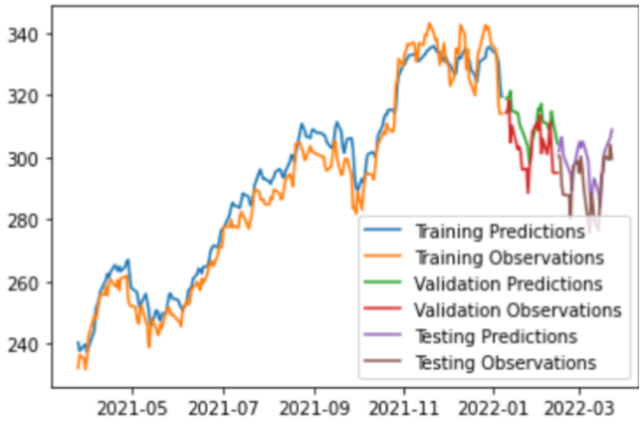
First try for predicting the stock price using LSTM model:
The model variables were:



```
model = Sequential([layers.Input((3, 1)),
                    layers.LSTM(64),
                    layers.Dense(32, activation='relu'),
                    layers.Dense(32, activation='relu'),
                    layers.Dense(1)])
```



LSTM model from 2009.



LSTM model from 2020.

We will get into the model in more depth to understand if the results are overfitting or maybe the LSTM model can bring accouret results for predicting stocks.
We can see that the results are very accurate (maybe too much?).

What each one did:

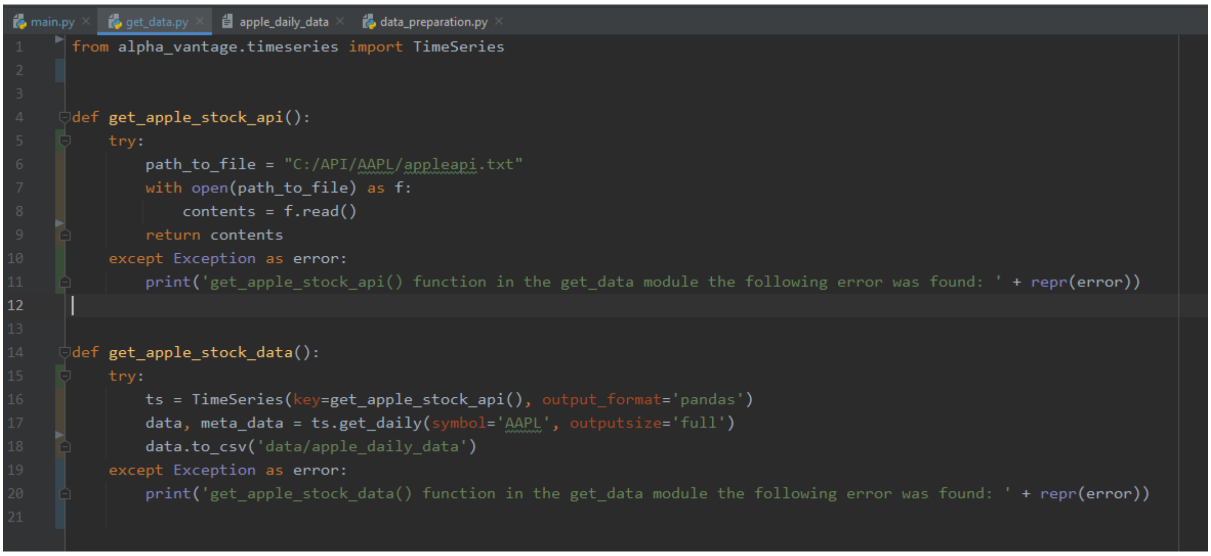
Einav- Algorithm research with implemeting and testing the results of the modles of the project, deciding the most relevant models to try and adjustment to the project data.
Gal - Data Analysis and Data Preparation of the raw data to the data reserved by the models, together with deciding the relevant historical data according to market research.
Asaf - Infrastructure of the project created and maintained the API data flow version control (GIT) of all team members with showing and implementing the System Design.

What next ?

- Accuracy test for LSTM model, adding additional models, comparing the results and conclusion.
- Design and devlopment the final UI for the end user.
- Adding intreactive prodiction selector according to the end user input.
- Testing the model with another similiar stock market companies

Attach

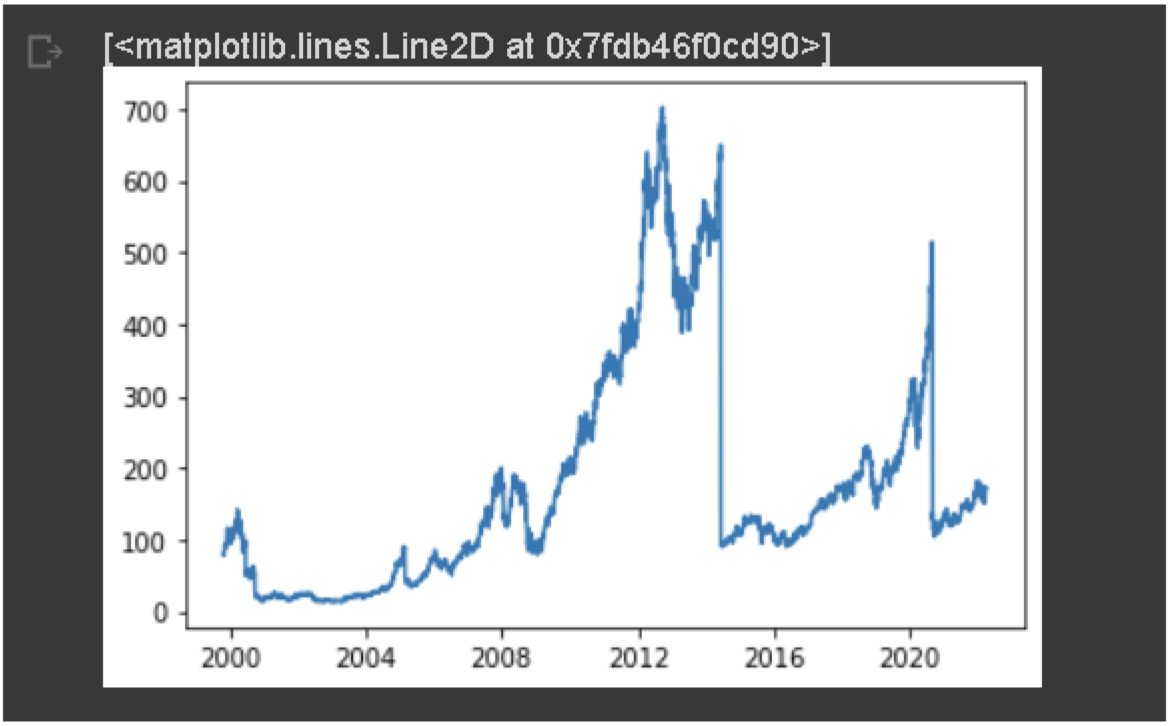
(1)



```
from alpha_vantage.timeseries import TimeSeries


def get_apple_stock_api():
    try:
        path_to_file = "C:/API/AAPL/appleapi.txt"
        with open(path_to_file) as f:
            contents = f.read()
        return contents
    except Exception as error:
        print('get_apple_stock_api() function in the get_data module the following error was found: ' + repr(error))


def get_apple_stock_data():
    try:
        ts = TimeSeries(key=get_apple_stock_api(), output_format='pandas')
        data, meta_data = ts.get_daily(symbol='AAPL', outputsize='full')
        data.to_csv('data/apple_daily_data')
    except Exception as error:
        print('get_apple_stock_data() function in the get_data module the following error was found: ' + repr(error))

```

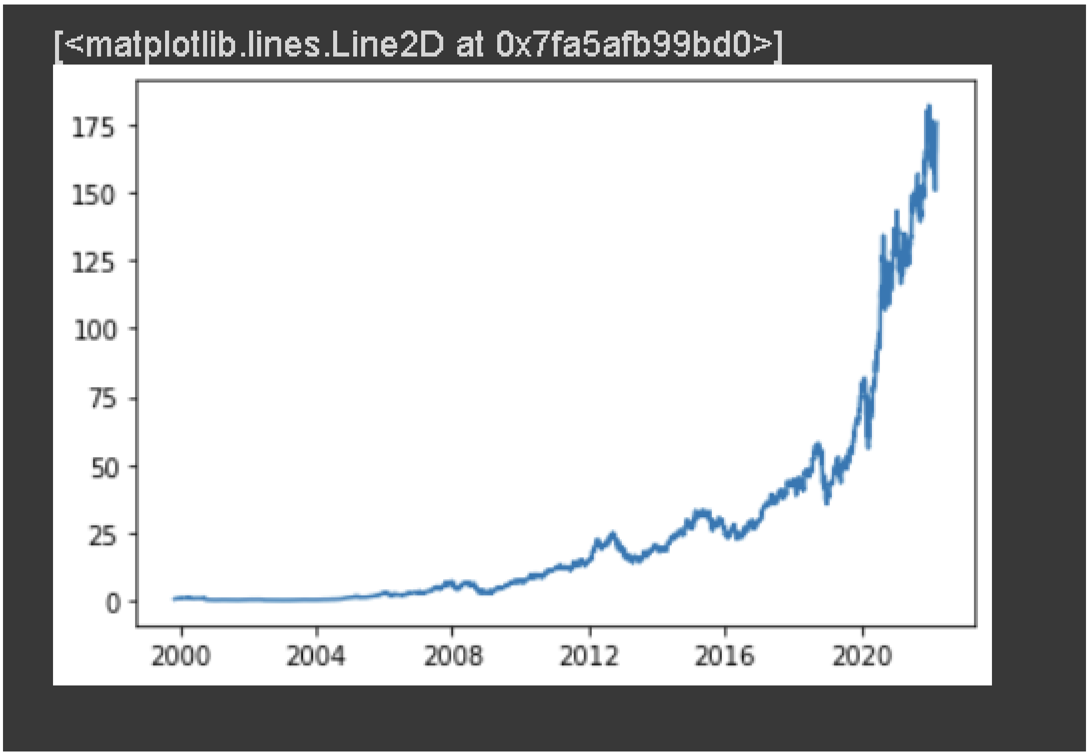
Before

(2)



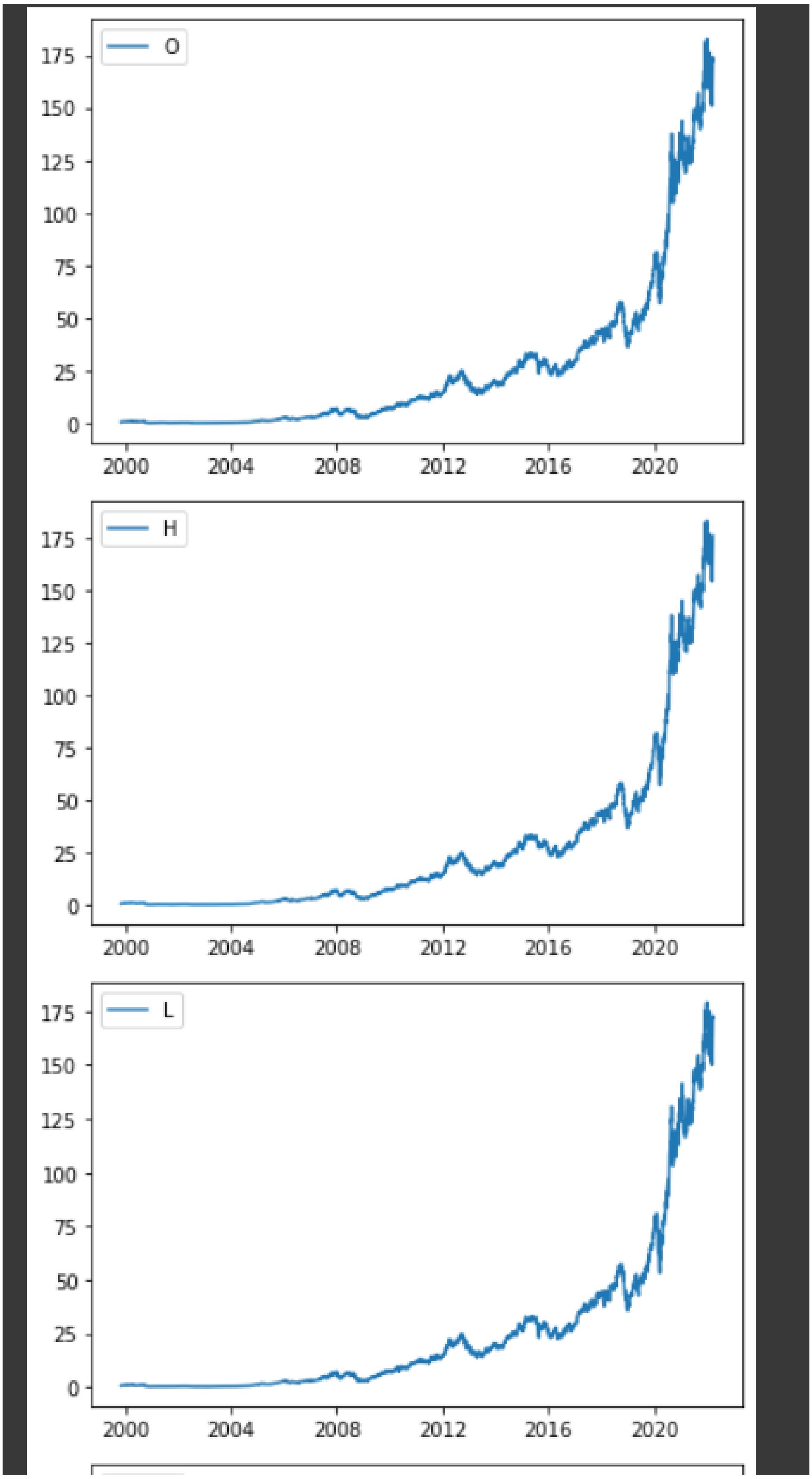
```
[<matplotlib.lines.Line2D at 0x7fdb46f0cd90>]
```

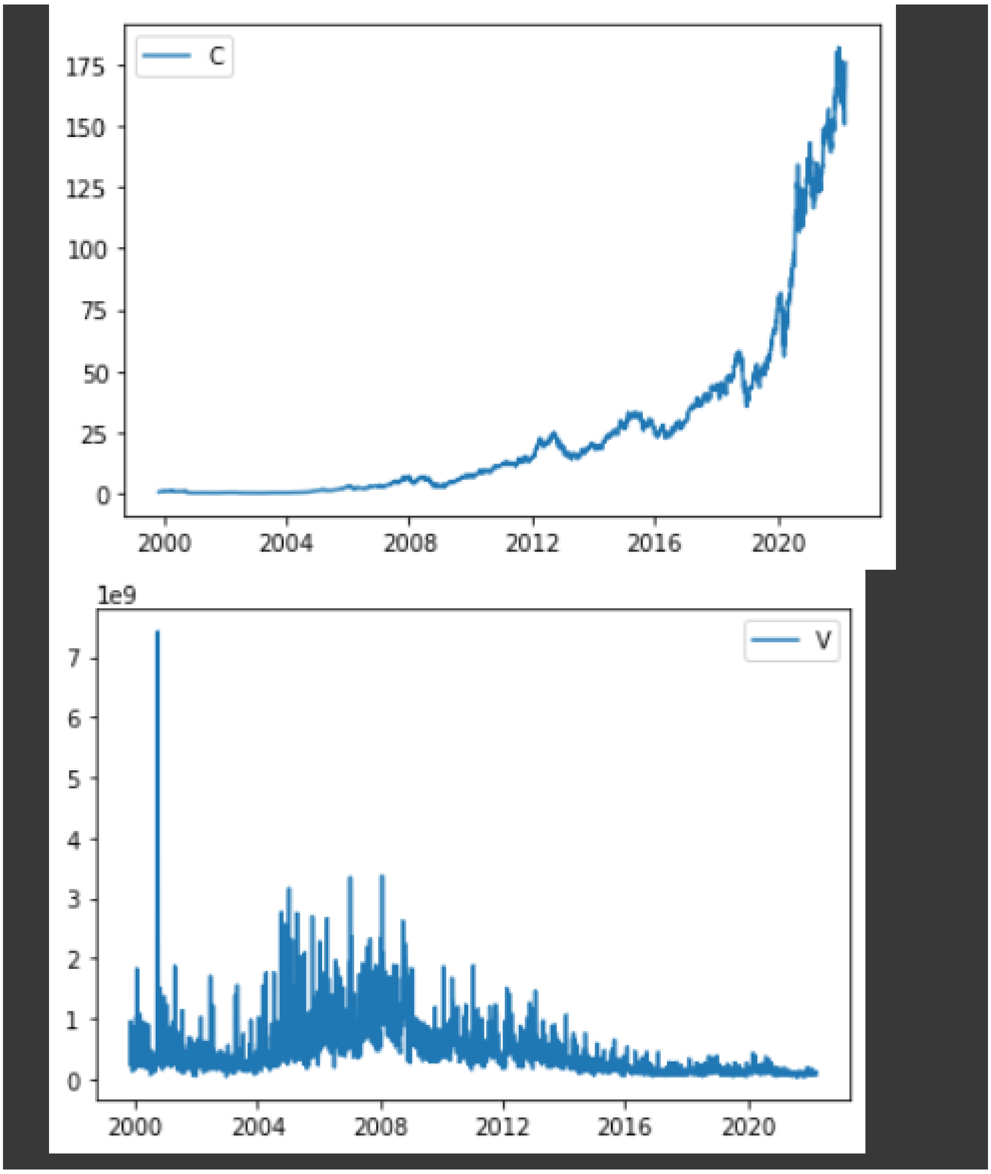
After
(3)



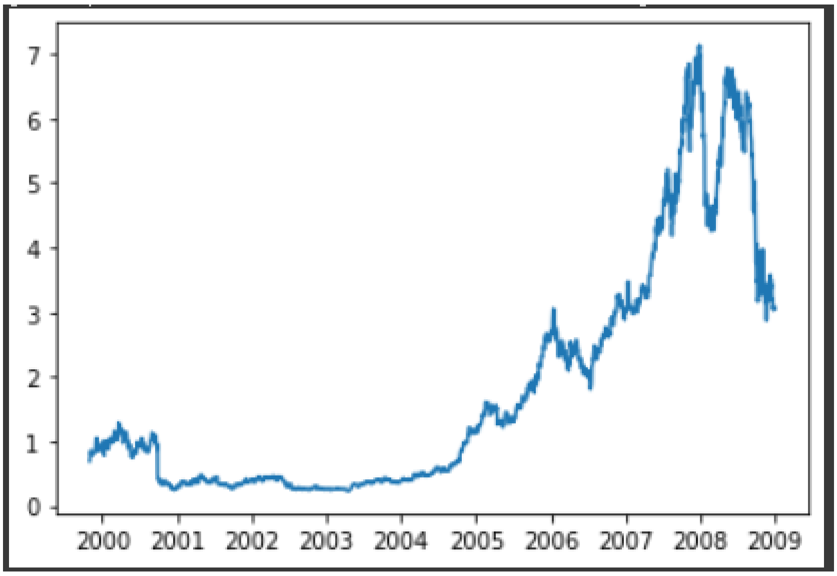
```
[<matplotlib.lines.Line2D at 0x7fa5afb99bd0>]
```

(4)

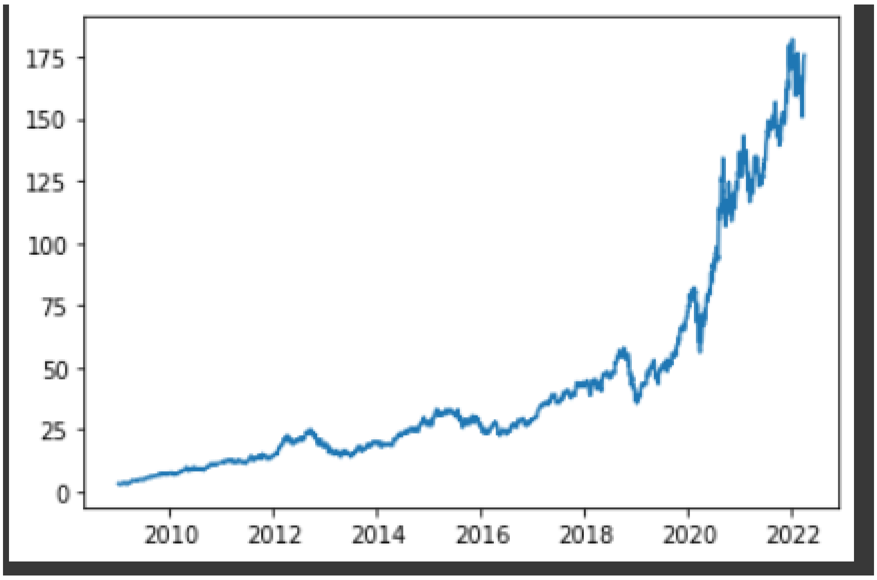

O
175
150
125
100
75
50
25
0
2000
2004
2008
2012
2016
2020
H
175
150
125
100
75
50
25
0
2000
2004
2008
2012
2016
2020
L
175
150
125
100
75
50
25
0
2000
2004
2008
2012
2016
2020


C
175
150
125
100
75
50
25
0
2000
2004
2008
2012
2016
2020
1e9
V
7
6
5
4
3
2
1
0
2000
2004
2008
2012
2016
2020

(5)



(6)



(7)
http://colah.github.io/posts/2015-08-Understanding-LSTMs/