Department of Computer Science & Engineering

Final Year B. Tech. (CSE) – I: 2022-23

5CS462: PE5 - Data Mining Lab

Assignment No. 9 : Mini Project

PRN: 2020BTECS00205

NAME: Monika .V. Chitrakathi

BATCH: B8

Title: 16] Predicting Stock Prices with Neural Networks

Use a neural network to predict stock prices. Download the data from Yahoo! Finance and then train neural network to predict future stock prices.

Objective/Aim:

1. To implement data analysis tool using python programming language.
2. The initial data we will use for this model is taken directly from the Yahoo Finance page
3. To Training and validating data,data normalization,creation of deep learning model LSTM.

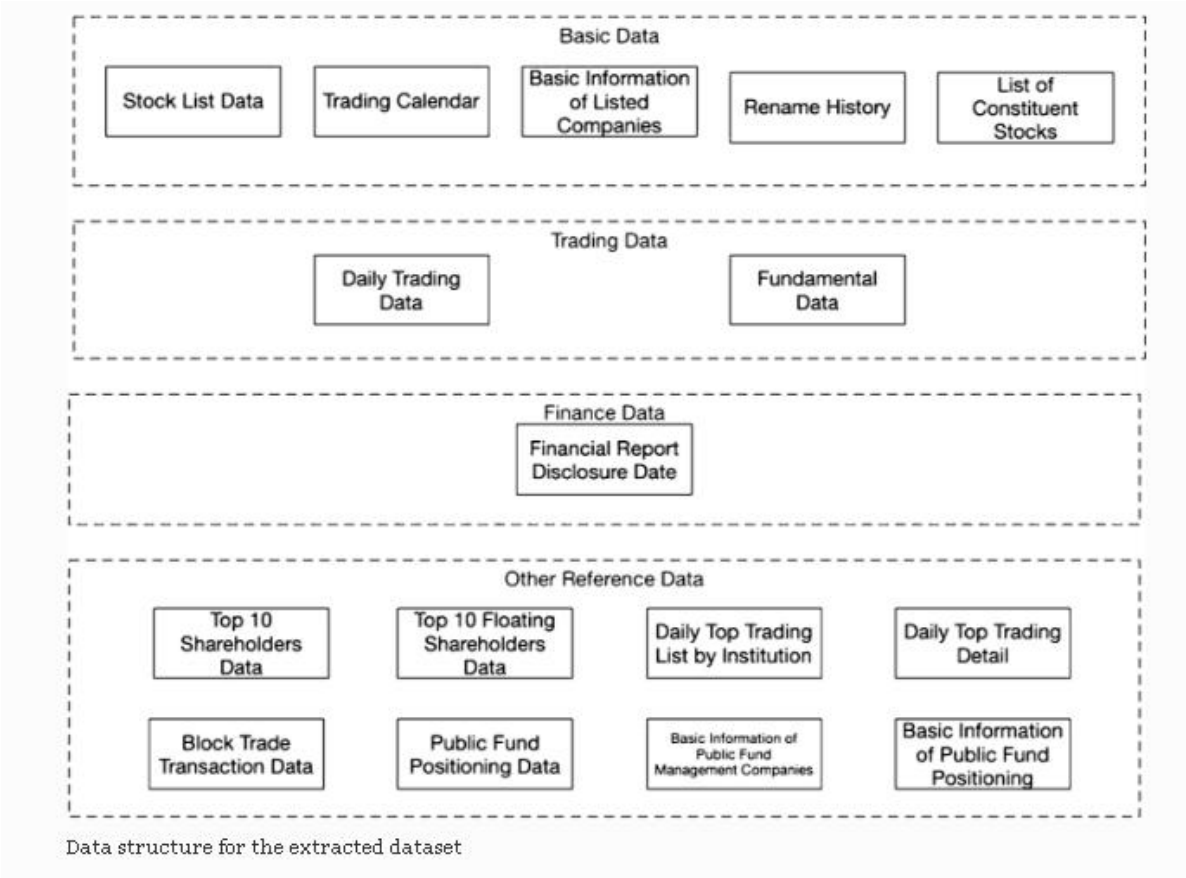
Introduction:

Predicting stock prices is a cumbersome task as it does not follow any specific pattern. Changes in the stock prices are purely based on supply and demand during a period of time. In order to learn the specific characteristics of a stock price, we can use deep learning to identify these patterns through machine learning. One of the most well-known networks for series forecasting is LSTM (long short-term memory) which is a Recurrent Neural Network (RNN) that is able to remember information over a long period of time, thus making them extremely useful for predicting stock prices. RNNs are well-suited to time series data and they are able to process the data step-by-step, maintaining an internal state where they cache the information they have seen so far in a summarised version. The successful prediction of a stock's future price could yyield a significant profit.

Theory:

From last few years there have been many up's and downs in stock market, as there are n factors which can affects a share market. Thus, due to its dynamic nature, it is very highly difficult to predict a stock price. To address this issue there should be some system which can detect the pattern in stock prices when influenced by political, economic and natural environment as well as which can take what are the people's sentiment about the particular company. In this paper, authors have presented a possible mixture of sentiment analysis and historical stock price. Now one of the major concerns before predicting the stock prices is to use the reliable historic stock data, thus pre-processing of data is also must.

Functional Block Diagram:

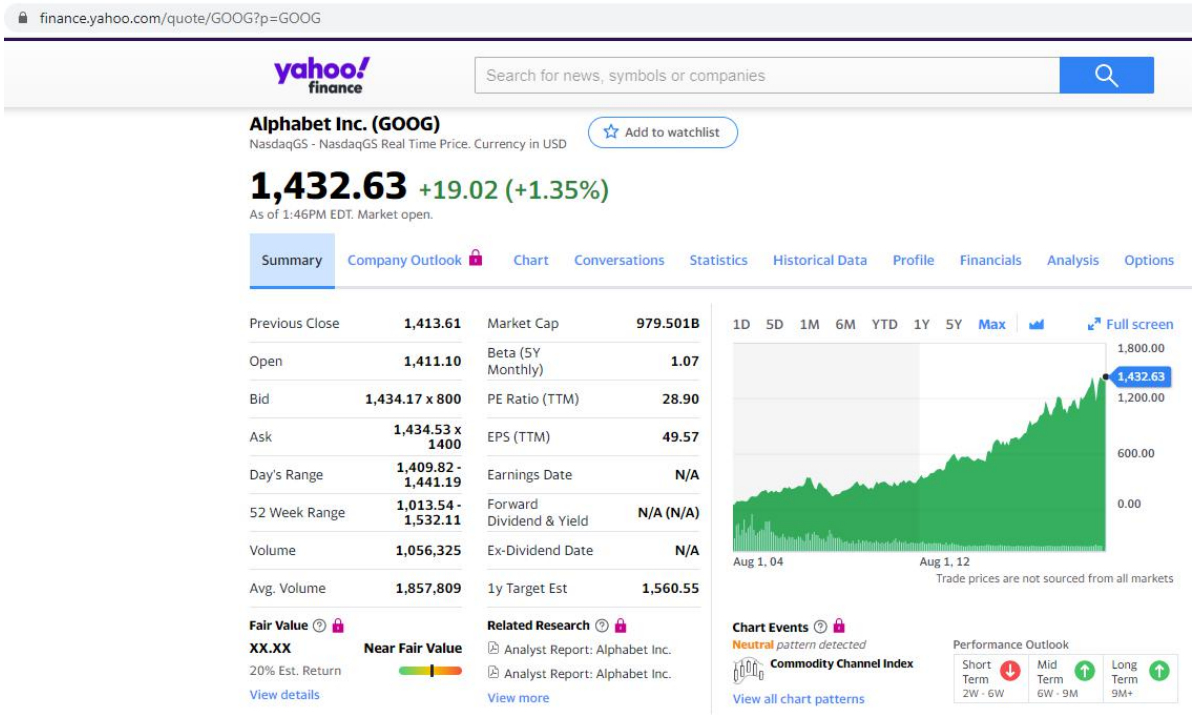


Data structure for the extracted dataset

Procedure:

Given problem statement is solved using python programming language and Deep neural network.Created deep learning model LSTM.

Screenshots:



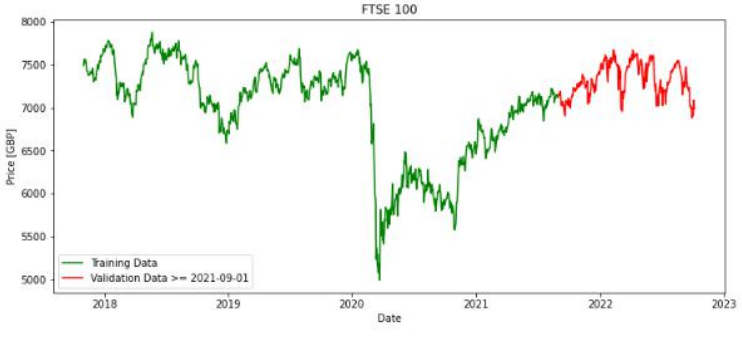
```
import os
import secrets
import pandas as pd
import argparse
from datetime import datetime

from stock_prediction_class import StockPrediction
from stock_prediction_lstm import LongShortTermMemory
from stock_prediction_numpy import StockData
from stock_prediction_plotter import Plotter
from stock_prediction_readme_generator import ReadmeGenerator
from stock_prediction_deep_learning import train_LSTM_network

os.environ["PATH"] += os.pathsep + 'C:/Program Files (x86)/Graphviz2.38/bin/'
```

In [2]:

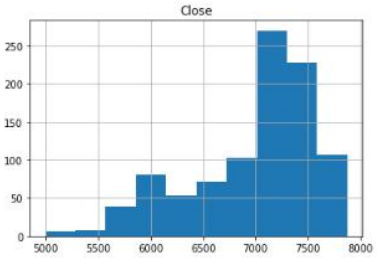
```
STOCK_TICKER = "^FTSE"
STOCK_START_DATE = pd.to_datetime("2017-11-01")
STOCK_VALIDATION_DATE = pd.to_datetime("2021-09-01")
EPOCHS = 100
BATCH_SIZE = 10
TIME_STEPS = 3
TODAY_RUN = datetime.today().strftime("%Y%m%d")
TOKEN = STOCK_TICKER + '_' + TODAY_RUN + '_' + secrets.token_hex(16)
GITHUB_URL = "https://github.com/JordiCorbilla/stock-prediction-deep-neural-learning/raw/master/"
print('Ticker: ' + STOCK_TICKER)
print('Start Date: ' + STOCK_START_DATE.strftime("%Y-%m-%d"))
print('Validation Date: ' + STOCK_START_DATE.strftime("%Y-%m-%d"))
print('Test Run Folder: ' + TOKEN)
# create project run folder
PROJECT_FOLDER = os.path.join(os.getcwd(), TOKEN)
if not os.path.exists(PROJECT_FOLDER):
    os.makedirs(PROJECT_FOLDER)

stock_prediction = StockPrediction(STOCK_TICKER,
                                   STOCK_START_DATE,
                                   STOCK_VALIDATION_DATE,
```

```
stock_prediction = StockPrediction(STOCK_TICKER,
                                   STOCK_START_DATE,
                                   STOCK_VALIDATION_DATE,
                                   PROJECT_FOLDER,
                                   GITHUB_URL,
                                   EPOCHS,
                                   TIME_STEPS,
                                   TOKEN,
                                   BATCH_SIZE)
# Execute Deep Learning model
train_LSTM_network(stock_prediction)
```

```
Ticker: ^FTSE
Start Date: 2017-11-01
Validation Date: 2017-11-01
Test Run Folder: ^FTSE_20221009_48e45baea7d3425f67d1253a209eaf34
End Date: 2022-10-09
[**********100%**********]  1 of 1 completed
mean: [0.68902354]
max 0.9999999999999998
min 0.0
Std dev: [0.20211829]
plotting Data and Histogram
```

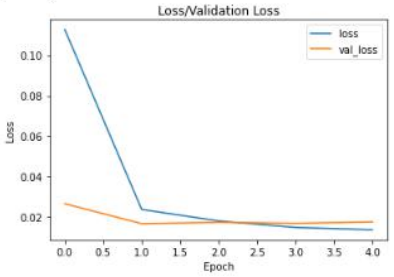


Date




```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 3, 100)            40800
_________________________________________________________________
dropout (Dropout)            (None, 3, 100)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 3, 50)             30200
_________________________________________________________________
dropout_1 (Dropout)          (None, 3, 50)             0
_________________________________________________________________
lstm_2 (LSTM)                (None, 3, 50)             20200
_________________________________________________________________
dropout_2 (Dropout)          (None, 3, 50)             0
_________________________________________________________________
lstm_3 (LSTM)                (None, 50)                20200
_________________________________________________________________
dropout_3 (Dropout)          (None, 50)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 111,451
Trainable params: 111,451
Non-trainable params: 0
_________________________________________________________________
Epoch 1/100
97/97 [==============================] - 22s 148ms/step - loss: 0.1127 - MSE: 0.2420 - val_loss: 0.0266 - val_MSE: 0.1018
Epoch 2/100
```

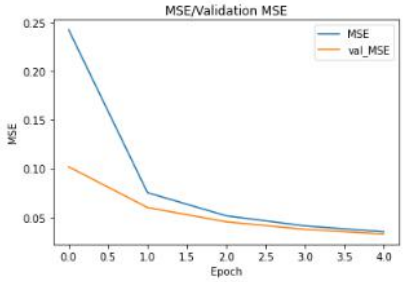
```
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 111,451
Trainable params: 111,451
Non-trainable params: 0
_________________________________________________________________
Epoch 1/100
97/97 [==============================] - 22s 148ms/step - loss: 0.1127 - MSE: 0.2420 - val_loss: 0.0266 - val_MSE: 0.1018
Epoch 2/100
97/97 [==============================] - 14s 139ms/step - loss: 0.0238 - MSE: 0.0753 - val_loss: 0.0167 - val_MSE: 0.0601
Epoch 3/100
97/97 [==============================] - 13s 132ms/step - loss: 0.0181 - MSE: 0.0516 - val_loss: 0.0174 - val_MSE: 0.0454
Epoch 4/100
97/97 [==============================] - 13s 133ms/step - loss: 0.0149 - MSE: 0.0412 - val_loss: 0.0169 - val_MSE: 0.0377
Epoch 5/100
97/97 [==============================] - 13s 131ms/step - loss: 0.0137 - MSE: 0.0352 - val_loss: 0.0176 - val_MSE: 0.0330
Epoch 5: early stopping
saving weights
plotting loss
```



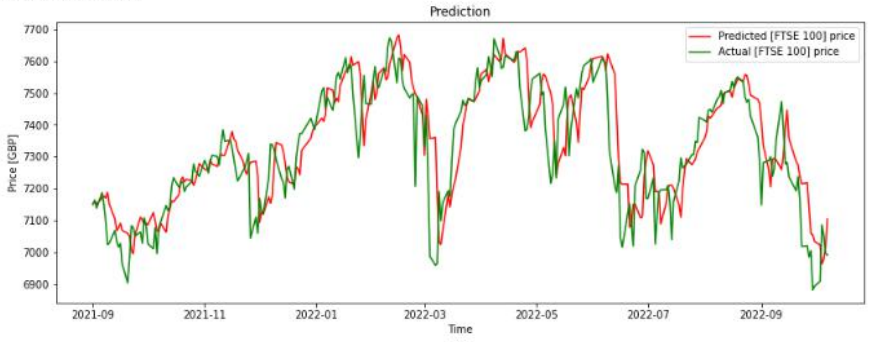

```
plotting MSE
```

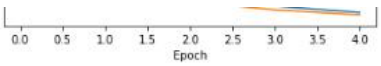
Epoch

plotting MSE

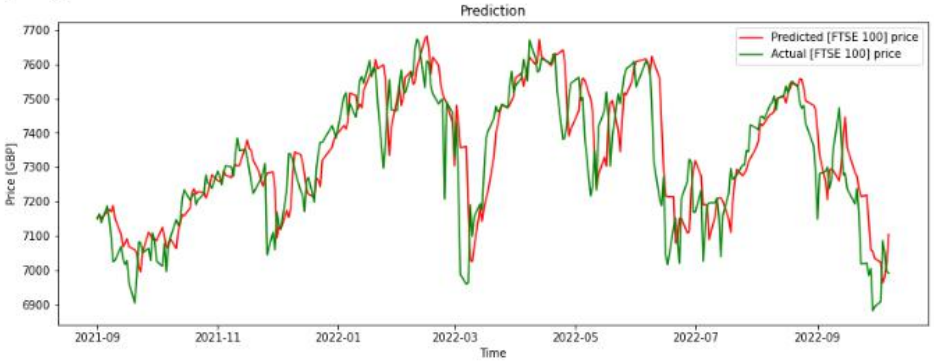



```
display the content of the model
9/9 - 0s - loss: 0.0176 - MSE: 0.0322
loss :  0.017598537667274475
MSE :  0.03224984462695122

plotting prediction results
plotting predictions
```






```
display the content of the model
9/9 - 0s - loss: 0.0176 - MSE: 0.0322
loss :  0.017598537667274475
MSE :  0.03224984462695122

plotting prediction results
plotting predictions
```




```
prediction is finished
```

In [ ]: