Stock Trend Prediction Web App in Python!

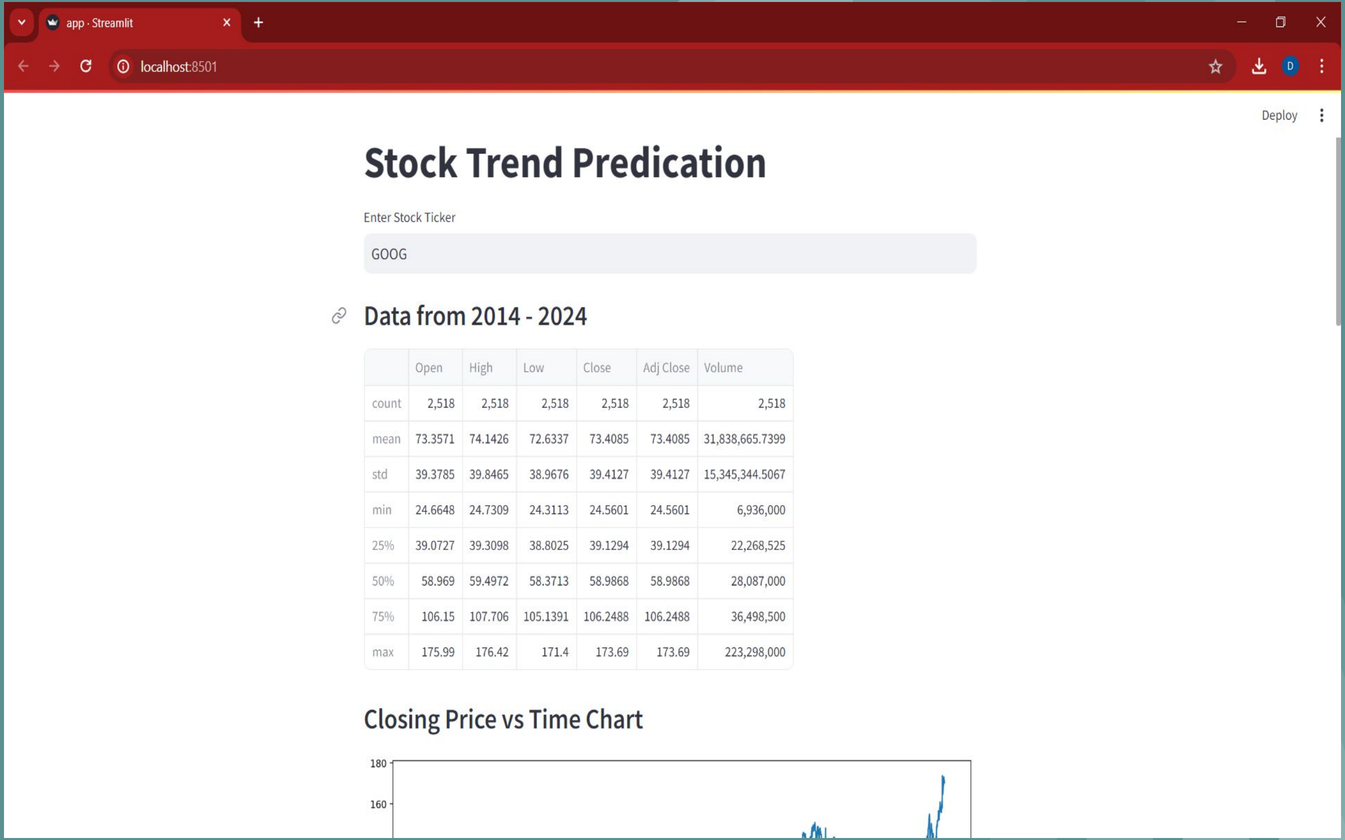
By: D’Ronze Malcom

WHY?

Everyday the stock market opens up and people make it their livelihoods to accurately predict how the market flows. It sounded cool to automate that a little and use it as a tool.

How?

Reading stock data from a csv, Cleaning and pruning the data, splitting, testing and training the data, using a deep LSTM-based neural network, making graphs then using stream lit to to make a web app.



Stock Trend Predication

Enter Stock Ticker

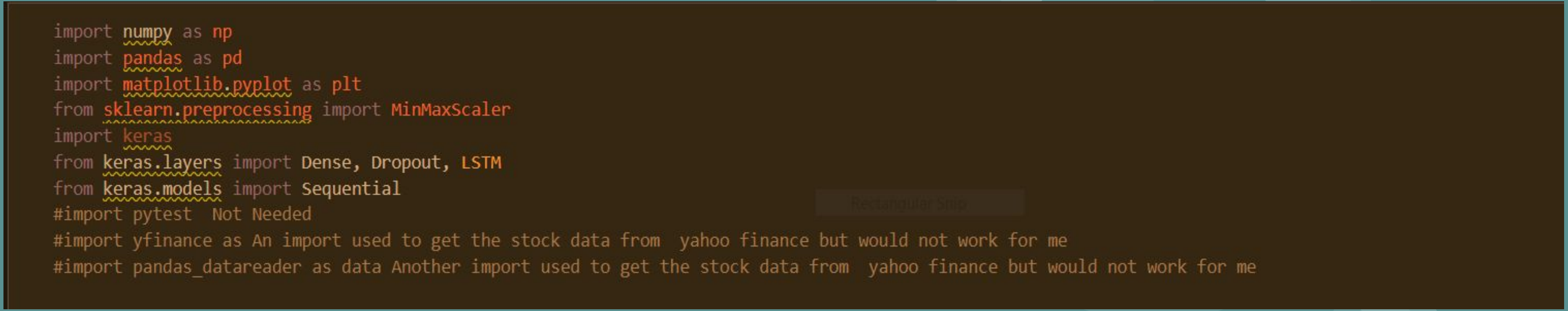
GOOG

Data from 2014 - 2024

	Open	High	Low	Close	Adj Close	Volume
count	2,518	2,518	2,518	2,518	2,518	2,518
mean	73.3571	74.1426	72.6337	73.4085	73.4085	31,838,665.7399
std	39.3785	39.8465	38.9676	39.4127	39.4127	15,345,344.5067
min	24.6648	24.7309	24.3113	24.5601	24.5601	6,936,000
25%	39.0727	39.3098	38.8025	39.1294	39.1294	22,268,525
50%	58.969	59.4972	58.3713	58.9868	58.9868	28,087,000
75%	106.15	107.706	105.1391	106.2488	106.2488	36,498,500
max	175.99	176.42	171.4	173.69	173.69	223,298,000

Closing Price vs Time Chart

Imports



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import keras
from keras.layers import Dense, Dropout, LSTM
from keras.models import Sequential
#import pytest  Not Needed
#import yfinance as An import used to get the stock data from  yahoo finance but would not work for me
#import pandas_datareader as data Another import used to get the stock data from  yahoo finance but would not work for me
```

I used familiar imports for the basics such as pandas, matplotlib, and sklearn. I also used keras for deep LSTM-based neural network. Typically you can use imports that collect stock data from online but my case was different.

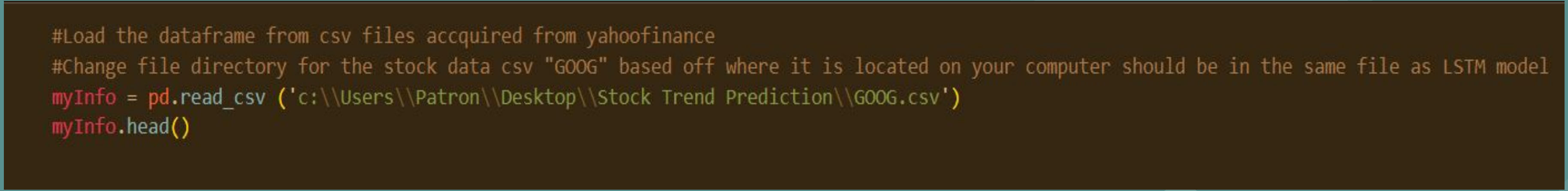
Challenges

- The hard part was setting up the imports, downloading the numerous amount of pip install packages, and on top of that I could not get pandas_datareader or even finance to work. I believe it has something to do with conflict between my python version and the latest version of pandas getreader.

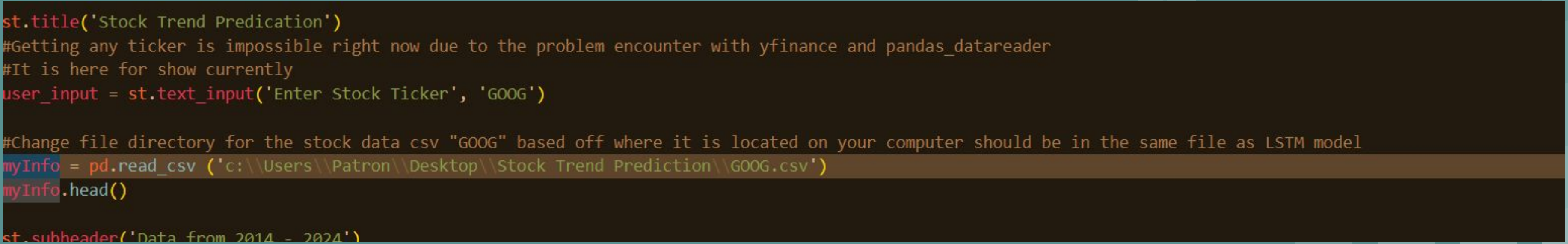
- The easier part surprisingly was wrapping my head around the algorithms and implementations. The graphs and even the web server sounds a lot more intimidating than it really is.

Running Code + Web Server

Step 1: Make sure location of the “GOOG” stock data csv matches up to what's in the program (LSTM model1) and the python app (app.py) for example:



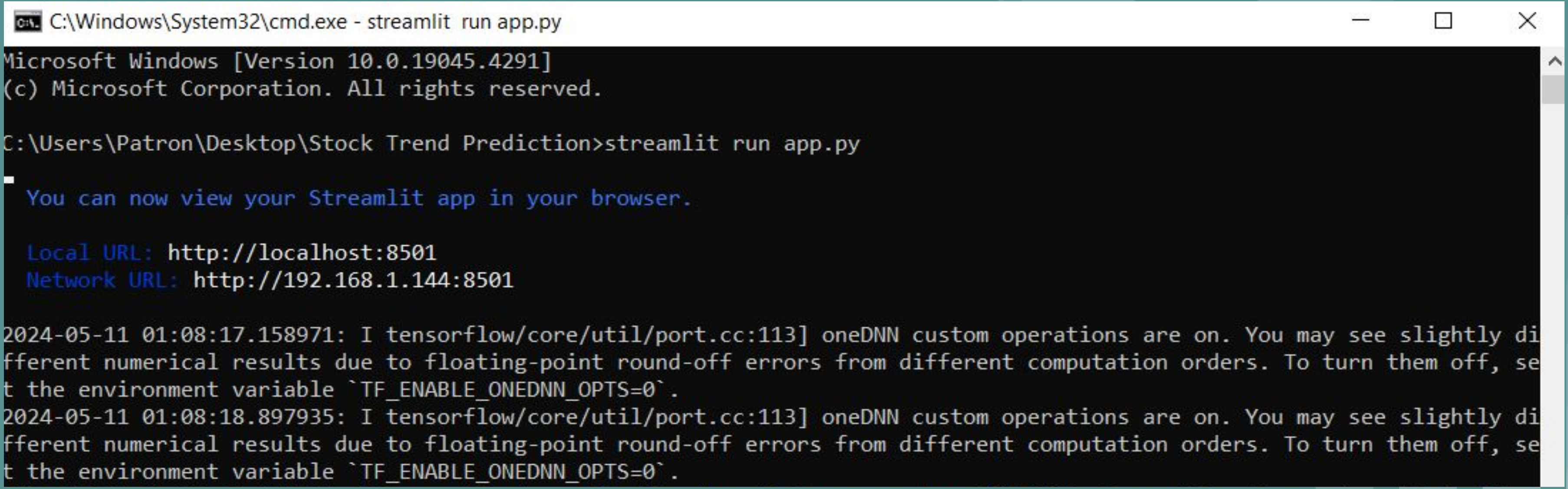
```
#Load the dataframe from csv files accquired from yahoofinance
#Change file directory for the stock data csv "GOOG" based off where it is located on your computer should be in the same file as LSTM model
myInfo = pd.read_csv ('c:\\Users\\Patron\\Desktop\\Stock Trend Prediction\\GOOG.csv')
myInfo.head()
```



```
st.title('Stock Trend Predication')
#Getting any ticker is impossible right now due to the problem encounter with yfinance and pandas_datareader
#It is here for show currently
user_input = st.text_input('Enter Stock Ticker', 'GOOG')

#Change file directory for the stock data csv "GOOG" based off where it is located on your computer should be in the same file as LSTM model
myInfo = pd.read_csv ('c:\\Users\\Patron\\Desktop\\Stock Trend Prediction\\GOOG.csv')
myInfo.head()

st.subheader('Data from 2014 - 2024')
```

Running Code + Web Server

Step 2: Opening cmd terminal and running the web app through stream lit (may need stream lit module) "streamlit run app.py":

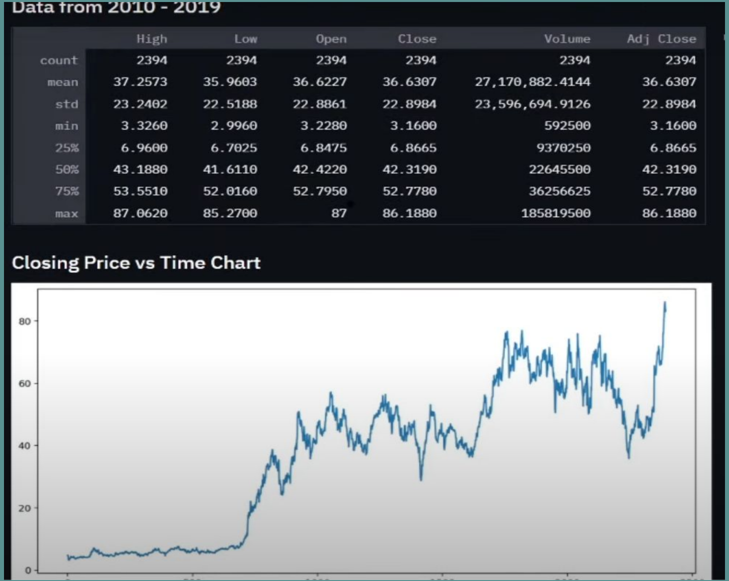


C:\Windows\System32\cmd.exe - streamlit run app.py

```
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Patron\Desktop\Stock Trend Prediction>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.144:8501

2024-05-11 01:08:17.158971: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-05-11 01:08:18.897935: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly di
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off, se
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
```

Running Code + Web Server

Step 3: View the fruits of your labor. The local server with all your graphs and predicted data should pop up on a web browser:



Data from 2010 - 2019

	High	Low	Open	Close	Volume	Adj Close
count	2394	2394	2394	2394	2394	2394
mean	37.2573	35.9603	36.6227	36.6307	27,170,882.4144	36.6307
std	23.2402	22.5188	22.8861	22.8984	23,596,694.9126	22.8984
min	3.3260	2.9960	3.2280	3.1600	592500	3.1600
25%	6.9600	6.7025	6.8475	6.8665	9370250	6.8665
50%	43.1880	41.6110	42.4220	42.3190	22645500	42.3190
75%	53.5510	52.0160	52.7950	52.7780	36256625	52.7780
max	87.0620	85.2700	87	86.1880	185819500	86.1880

Closing Price vs Time Chart

Sources:

1: Geeks For Geek: Building a stock prediction

2: How to use finance to get stock data + download csv

3: How to use pandas_datareader