
# Part 3: Predict Stock Prices

## Group 5

Emmanuel Amoah Kwame - 10464434
Amanfo Oforiwaa Abena - 10714326
Dumenu Ernest Mawuse - 10726400
Kayang Edwin Pelpuo - 10728521
El-Karece Amoakoa Asiedu - 10735358

# Introduction

Artificial intelligence is a new technological paradigm, which is slowly but efficiently transforming the logic behind the computerization of mundane tasks. The concept denotes the simulation of human intelligence processes by machines, especially computer systems. Artificial intelligence is the brains behind a lot of technological innovations such as natural language processing, speech recognition, computer vision, machine translation as well as pose detection.

Recently, python has become one of the primary languages, equipped by programmers as they venture into the world of developing intelligent machines. This is due to its readability as well as its numerous libraries, which allow the programmer to focus more on the logic on the artificial intelligence algorithm, rather than the correct procedures and syntaxes for implementing lower level functions. Moreover, renowned corporations have displayed their approval of the language by developing tools in it, to speed up their AI workflows. Among these are Google, which owns the Tensorflow library and its variant as well as Facebook which owns PyTorch.

The task which was accomplished in this project, involved developing a machine learning algorithm which was efficient at predicting stock prices. This was done using the python programming language, and a graphical interface was to be developed to serve as a medium for user input as well as visualisation of results.

# Stocks: An overview

A stock or equity is a general term used to describe a security representing ownership of part of a corporation. Thus, a person owning stocks in a company is entitled to a proportion of the company's assets. This also means a person who owns stocks, or a shareholder, profits based on how much stock they own. Generally, after a company goes public via an Initial Public Offering, investors are allowed to buy and sell the company's stock on the exchange. This buying and selling influences the value of the stock, based on the principles of supply and demand.

Gaining profit as a shareholder usually deals with selling stocks at a higher price than they were bought. Thus, the net amount becomes the profit. The stock market is usually unpredictable to untrained people, and many may think stock prices fluctuate randomly and

profit can only be made through sheer luck. However, there are individuals trained to predict rises and falls of stock prices, and take advantage of this to make profit. These are known as stock brokers.

# Machine Learning Workflow

## Data collection

The data for stocks is tracked by a number of websites on the internet. For the purpose of this project, the target site to scrape stock prices from, was Yahoo finance. This was achieved using pandas. The information was downloaded 1st January 2010 as a start date and the current date as the end date. The company stock to be retrieved was also represented using an abbreviation known as a ticker. Examples of these were AAPL for Apple, MSFT for microsoft, GOOG for Google and AMZN for Amazon. The information was then stored in a dataframe. Apple was used as the target stock during the experimentation phase.
The data was then viewed to study its form, and it was found to have 6 features. These were Open, High, Low, Close, Adj Close and Volume

```
df.head()
```

| Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| 2010-01-04 | 7.622500 | 7.660714 | 7.585000 | 7.643214 | 6.535084 | 493729600 |
| 2010-01-05 | 7.664286 | 7.699643 | 7.616071 | 7.656429 | 6.546383 | 601904800 |
| 2010-01-06 | 7.656429 | 7.686786 | 7.526786 | 7.534643 | 6.442257 | 552160000 |
| 2010-01-07 | 7.562500 | 7.571429 | 7.466071 | 7.520714 | 6.430346 | 477131200 |
| 2010-01-08 | 7.510714 | 7.571429 | 7.466429 | 7.570714 | 6.473096 | 447610800 |

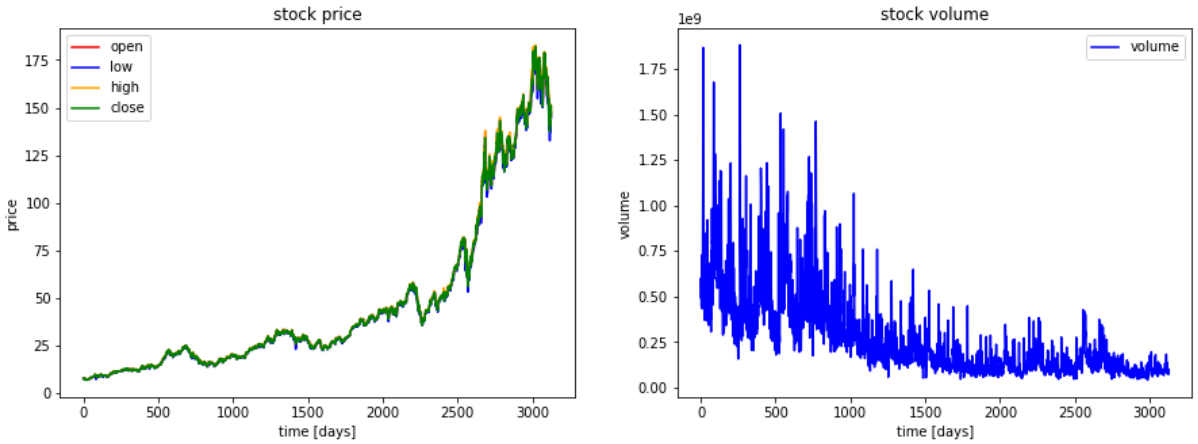
Features of stock dataframe

## Data Visualisation

Using the matplotlib library, the data was visualised for 3 main reasons:

- To check for any overall trend in the data
- To scan for outliers
- To check for any noticeable trend in the features

The features were plotted and it was established that the Open, High, Low and Close parameters grew over time, while the stock volume reduced.




Visualisation of features against time

## Data preparation

This step mainly involved creating the dataset for training the model. Since the data was time series, it was decided that the training set would be represented as a time slice, with the value immediately after the time slice representing the label. The window slice was a variable parameter and was initialised at 50.

An 80-20 split was used to separate the dataset into training and test sets. The training values were also not randomised, since the data was time based.

```
train_percentage = 0.8
train_end = int(len(df)*train_percentage)

train_split = df['Close'][:train_end]
test_split = df['Close'][train_end:]

training_data = pd.DataFrame(train_split)
test_data = pd.DataFrame(test_split)
```

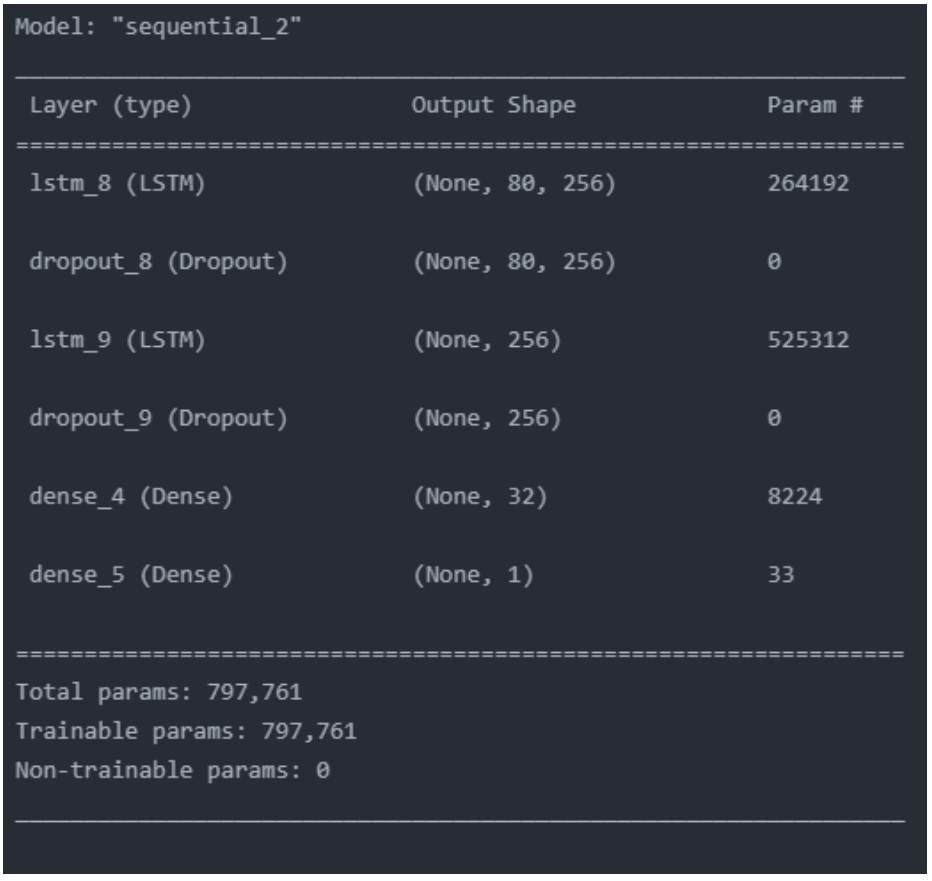
Train-test split

Normalisation of the dataset was also done using a MinMaxScaler. This function transformed the initial range of the dataset to a range between 0 and 1 based on the largest and smallest values in the dataset. The datasets were then represented as numpy arrays.

## Model creation and training

Due to the nature of data, an LSTM(Long, Short Term Memory) was used as the main architecture of the machine learning model. LSTMs are a type of Recurrent Neural Network capable of learning both long term trends and short term patterns in time series data. This type of model would be beneficial to use for stock prediction since it would be able to efficiently study the nature of the rises and falls in the stock patterns and effectively predict future patterns.

The model was made up of 2 LSTM layers each with a dropout layer. The dropout layer was used to prevent the model from relying too much on a particular set of neurons for predictions. The LSTM layers were followed by 2 dense layers and an output was produced.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_8 (LSTM)               (None, 80, 256)           264192

 dropout_8 (Dropout)         (None, 80, 256)           0

 lstm_9 (LSTM)               (None, 256)               525312

 dropout_9 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 32)                8224

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 797,761
Trainable params: 797,761
Non-trainable params: 0
_________________________________________________________________
```

Model Summary

An Adaptive Moment Estimation (Adam) optimizer was used, since it would be able to tweak the learning rate efficiently to minimize the loss of the model. A loss of mean squared error

was also used, since it effectively measures the deviation of a set of predicted values from the actual values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

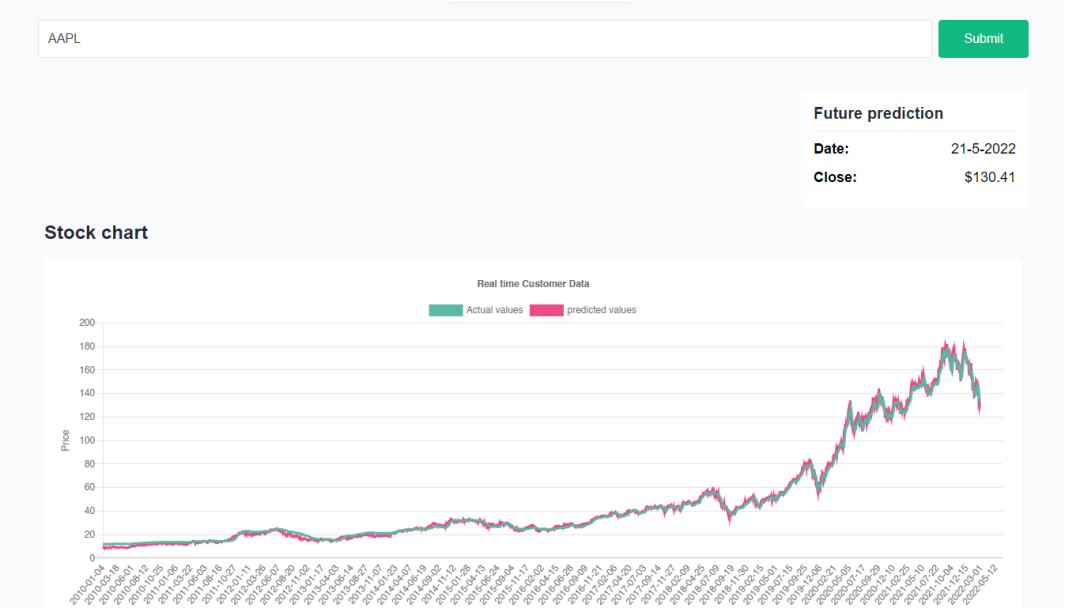
Mean Squared Error

Training was done over 50 epochs. The window was then tweaked to minimise the loss, and in the end, the loss reduced to 0.001.

## Web App Development

The web app was built with python, using Flask as a backend framework, as well as a combination of HTML, CSS and TailwindCSS for the frontend. The app was developed in a virtual environment, in order to have control over the library versions used. The necessary packages were then installed in this environment. The workflow for making predictions was then created as a function in the app, and called in one of the routes.

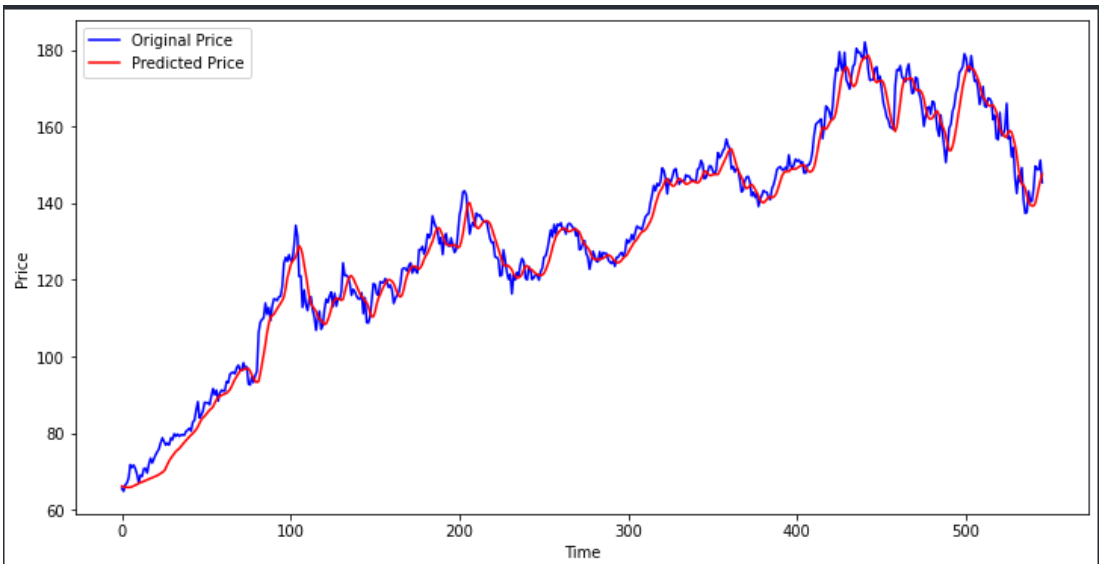
On the frontend, there was an input for the stock ticker, followed by a card displaying future predictions as well as a chart to show the trend over time. The query for the stock prediction was made in javascript, since there was a need to perform DOM(Document Object Model) manipulations after the  query was complete.




Site Interface

# Results

The model produced a loss of 0.001 with a final window of 80 values and 50 epochs for training.




Graph of actual price against predicted prices

The app was also able to effectively display these predictions from the model.

# Conclusion

To sum up, Artificial intelligence is a very efficient tool for computerising processes. Stock prediction has proved to be one of them, since the artificial intelligence algorithm was able to learn the trends of the data and closely predict future values.

However, the values produced from this model should be taken with a pinch of salt, as there are many other variables that influence stocks such as social media, celebrities, natural disasters and others. In order to get a far more trustworthy machine learning model, these values would have to be accounted for.

# References


1. E. Burns, “What is Artificial Intelligence (AI)”, What is Artificial Intelligence (AI)? Definition, Benefits and Use Cases (techtarget.com), 2019
2. A. Hayes, “Stock Definition”, Stock Definition (investopedia.com), 2022
3. Colah, “Understanding LSTM Networks”, Understanding LSTM Networks -- colah's blog, 2015
4. J. Frost, “Mean Squared Error(MSE)”, Mean Squared Error (MSE) - Statistics By Jim, 2018