# **Final Project: Stock Price Forecasting Using LSTM**

## Sitare Arslantürk

# Introduction

Predicting stock returns gives crucial insight on market efficieny. The dynamic nature of stock market makes it difficult for the investors to predict the prices considering the external factors such as political situations or the public image of the company. The adjusted closing price shows the stock's value after posting a dividend to accurately reflect that stock's value after accounting for any corporate actions. In order to predict the adjusted closing prices of a company, for example Apple Inc.’s, long short-term memory (LSTM) was used. This artificial neural network was used to find if a relationship existed between the adjusted closing price of stocks with the open, close, high and low values.

# Hypothesis

The efficient market hypothesis implies that it is impossible to beat the market consistently on a risk-adjusted basis since market prices should only react to new information. In conformance to this theory, the random walk hypothesis states that stock market prices evolve according to a random walk and thus cannot be predicted using past data. This project attempts predict future adjusted closing prices and tries to see if there is no relationship between the adjusted closing price and the historical stock price data. The main null hypothesis states that there is no relationship between the open, high, low, close variables and the output adjusted close values. The alternative hypothesis states that dependent variable close has a relationship to the independent variables of open, high, low and close.

# Data Collection and Pre-processing

I have used Yahoo Finance to retrieve the stock price values of the Apple Inc. company from 2010 to 2020. The data consisted of Open, High, Low, Close and Adjusted Close attributes, each having its own value. Total data of 2516 entries was collected as input for the linear regression model and saved into a csv file. The data collected was modified by selecting only the required columns and removing the unwanted data. Afterwards, the data was checked for invalid entries such as NaN values or null (empty) values. The invalid entries were removed by list-wise deletion. The data is then manually split into 80/20 for training and testing. In addition, the stock prices are normalized to values between 0 and 1, using MinMaxScaler from the sklearn library.

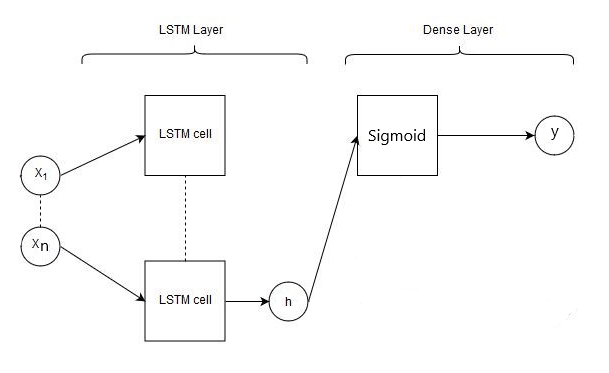
# LSTM Model

Tensorflow.Keras is used to implement the LSTM model. In Keras, neural networks are as sequence of layers which is held by Sequential class. After creating the Sequential class, the layers are added on top of each other to be fully connected. The LSTM recurrent layer comprised of LSTM memory units is used in order to implement the long short term memory model. Following LSTM, the Dense layer is used in order to output a prediction.

The first step is to create an instance of the Sequential class. Then you can create your layers and add them in the order that they should be connected. The LSTM recurrent layer comprised of memory units is called LSTM(). A fully connected layer, Dense() follows the LSTM layers and is used for outputting a prediction.

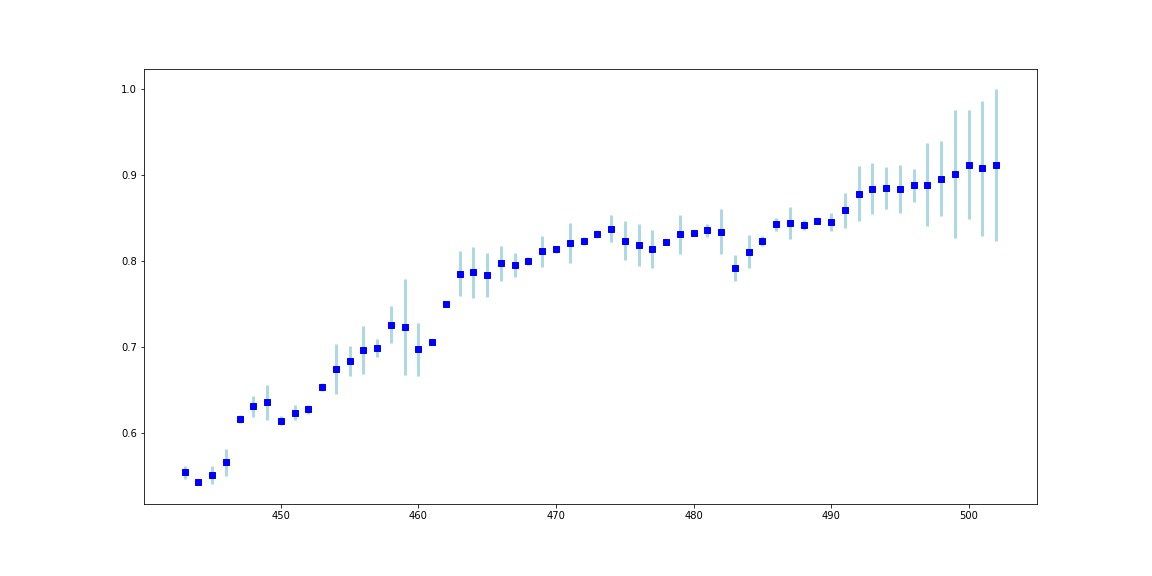
The first layer in the network defines the number of inputs to expect. The input must be three-dimensional, comprised of samples which are the rows of the data, timesteps which are the past observations for a feature, and features which are the columns of the data. The choice of activation function is most important for the output layer as it will define the format that predictions will take. I observed that choosing ‘linear’ instead of ‘sigmoid’ yields a better prediction in future adjusted closing stock prices. However to avoid overfitting, I preferred to use the sigmoid activation function.

Compilation transforms the simple sequence of layers defined into a highly efficient series of matrix. Compilation requires an optimization algorithm to use to train the network and a loss function used to evaluate the network that is minimized by the optimization algorithm. My choices were to use the mean squared error (mean\_squared\_error) loss function and the ‘adam’ optimizer which are used for regression purposes. After fitting the model, I predicted the adjusted closing prices with the test set. Figure 1 depicts the visual representation of the LSTM model.



**Figure 1:** LSTM Model

# Conclusion

Graph 1 depicts the results and the absolute errors of the predictions and test values. The results reject the null hypothesis that there are no relationship between stock prices. LSTM seems to be able to predict adjusted closing stock price behavior. Further investigation shows that there is even a correlation between the overall SP500 prices. This encourages the use of better tuned models, specifications and processing can be implemented to predict the future stock market adjusted closing prices.

**Graph 1**: Error Bar Plot