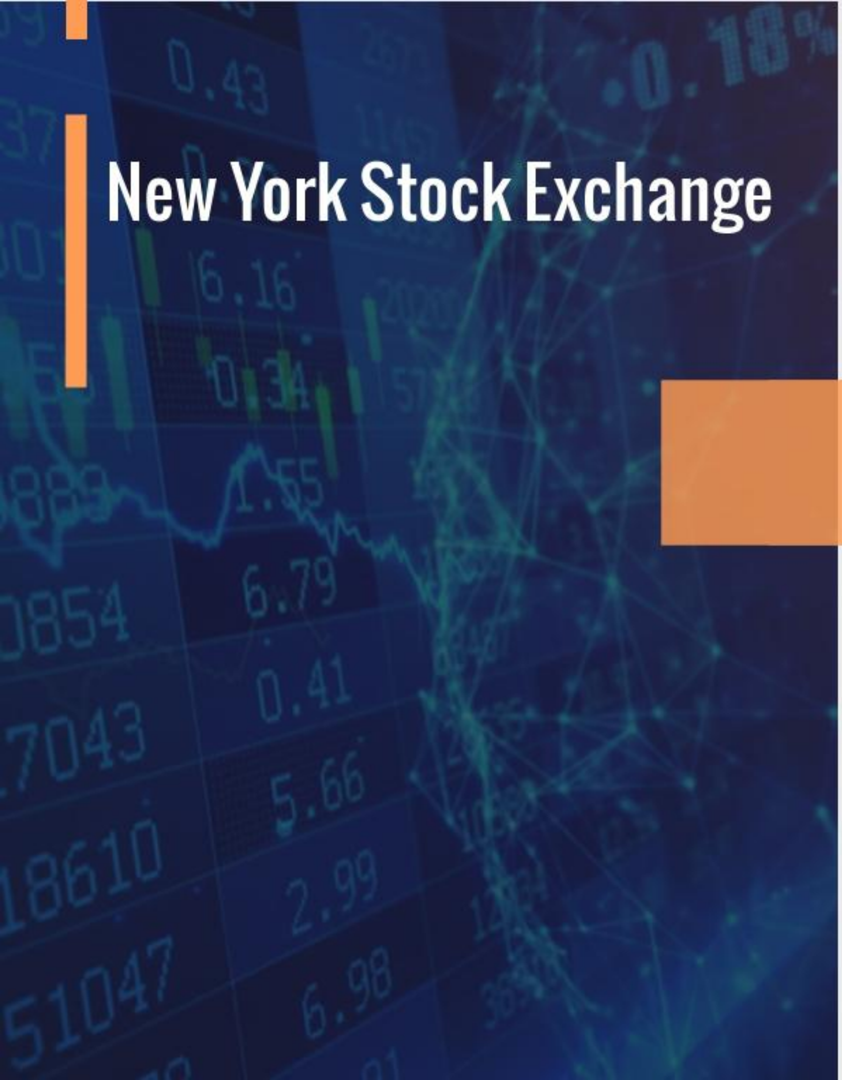
# Stock Price Prediction

Time Series Analysis

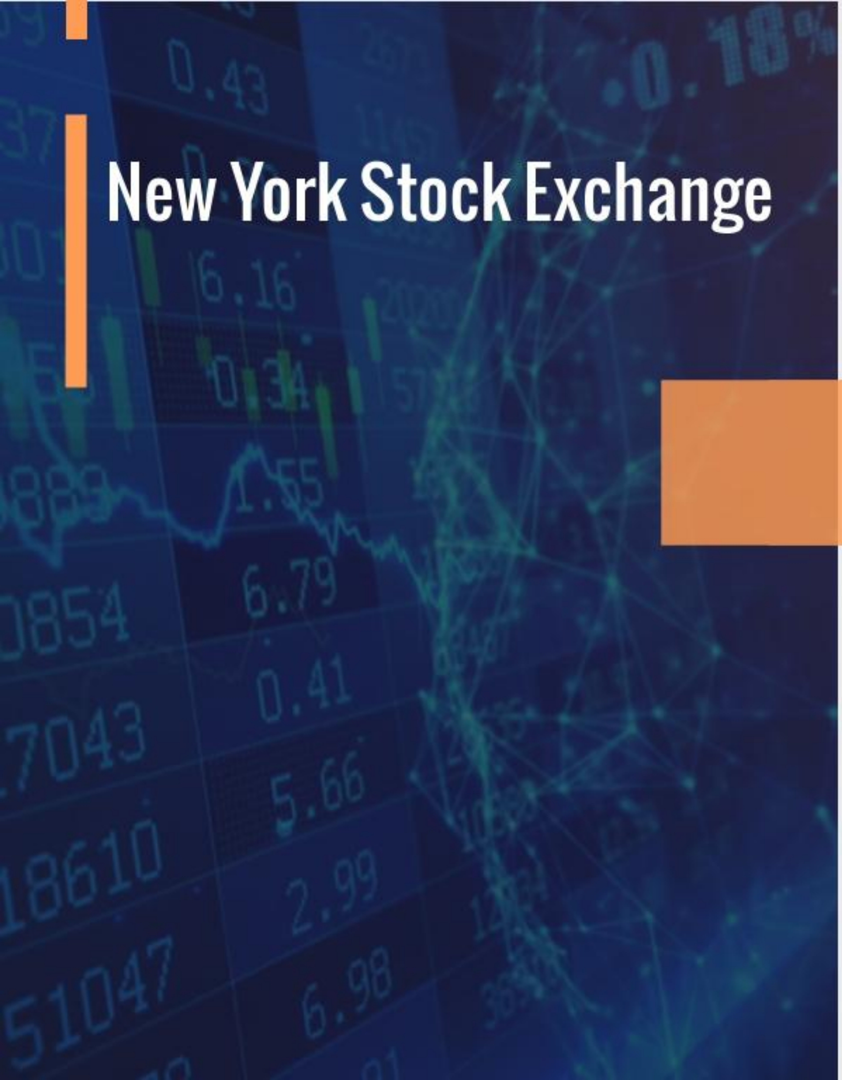
Yutong Wang

# New York Stock Exchange



## WHAT DO WE HAVE?

- Daily stock prices from S&P 500 company (fetched from Yahoo Finance), mostly spanning from 2010 to the end of 2016
- The open, close, high, low prices are recorded each stock each day
- 501 different stocks in total



# New York Stock Exchange

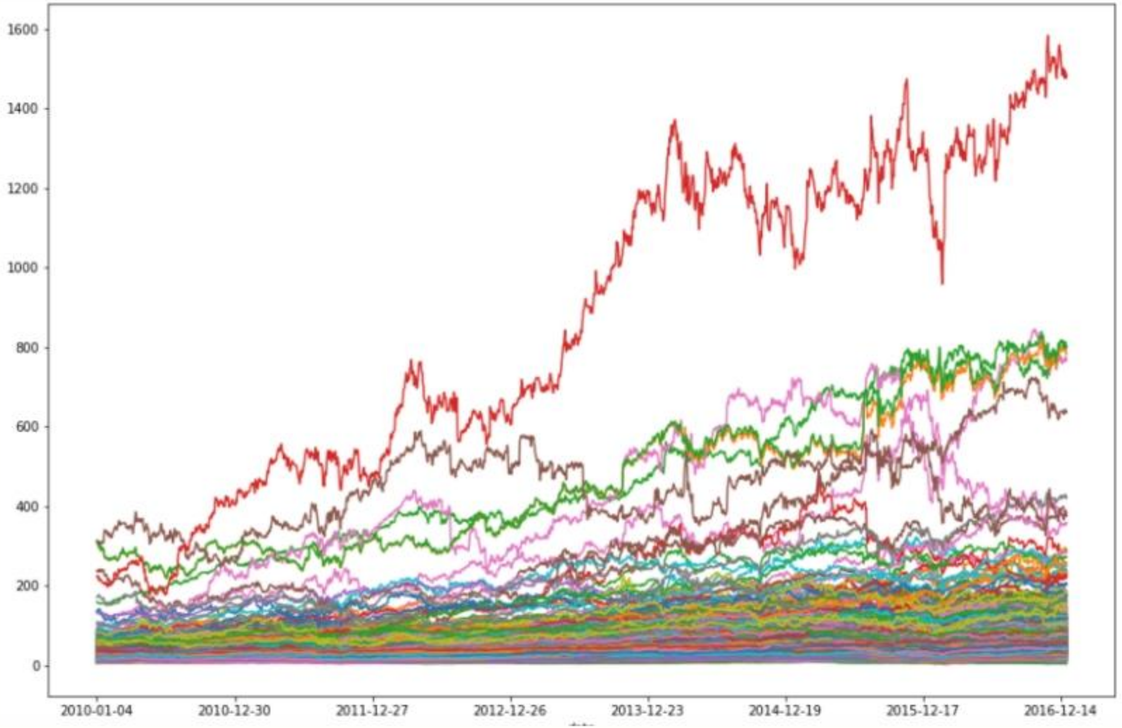
## Our Goal

- Make one-day ahead prediction on historical prices using Time Series Analysis (ARMA and LSTM)
- Find the most accurate model using MSE and MAPE as measurements

# The Process

1. Investigate the Data, and split into 70% train and 30% test
2. Check for Stationary
3. Define evaluations: MSE & MAPE
4. Create a simple model and make the prediction on the train data
5. Fit and refine ARMA model and make prediction on train data
6. Fit and refine LSTM model and make prediction on train data
7. Evaluate performance of all models and find the most accurate model
8. Make the final prediction

# Data Exploration



# Evaluations

## Mean Square Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

The MSE either assesses the quality of a predictor (i.e., a function mapping arbitrary inputs to a sample of values of some random variable), or of an estimator (i.e., a mathematical function mapping a sample of data to an estimate of a parameter of the population from which the data is sampled).

```
def mse(a, b):

    summation = 0  #variable to store the summation of differences
    n = len(a) #finding total number of items in list

    for i in range (0,n):  #looping through each element of the list
        difference = a[i] - b[i]  #finding the difference between observed and predicted value
        squared_difference = difference**2  #taking square of the differene
        summation = summation + squared_difference  #taking a sum of all the differences

    MSE = summation/n  #dividing summation by total values to obtain average

    return MSE
```

## Mean Absolute Percentage Error

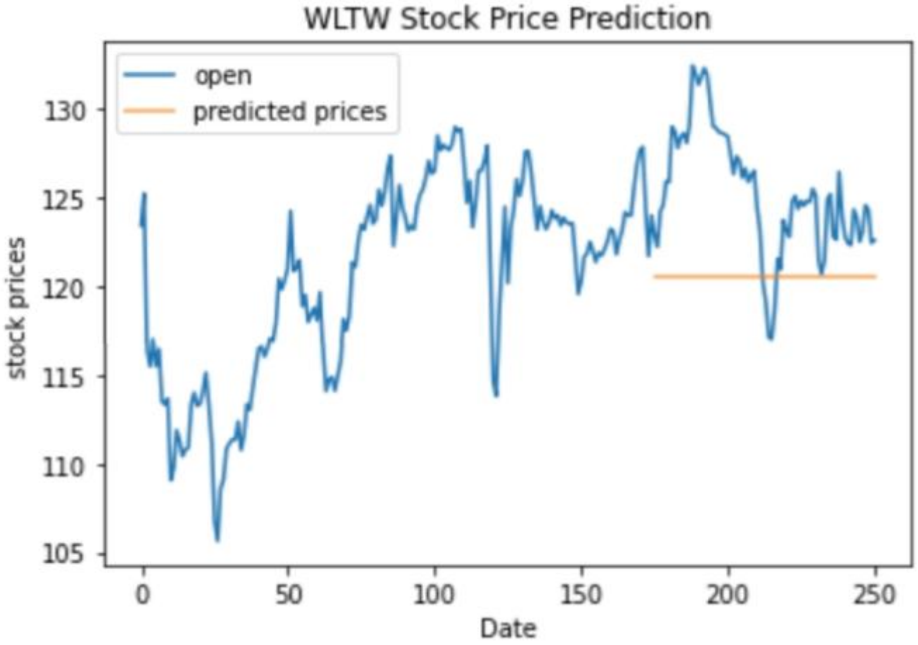
$$\mathrm{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

where At is the actual value and Ft is the forecast value. Their difference is divided by the actual value At. The absolute value in this ratio is summed for every forecasted point in time and divided by the number of fitted points n.

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

# The Simple Model

In this model, we predict the stock prices using their average prices. We will use the average price of the train set as out result and compare it to the real data to see the performance.



# ARMA Model

AR

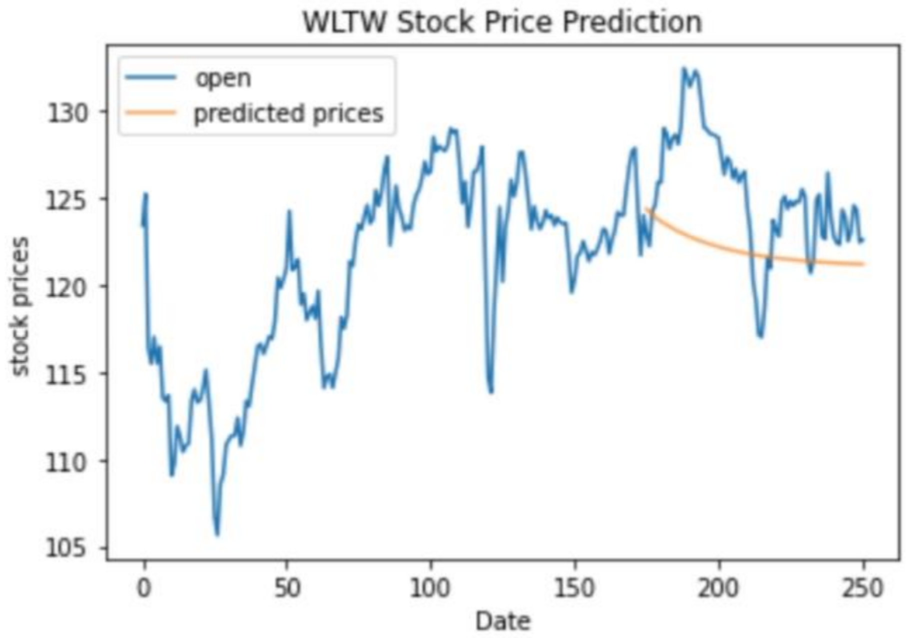
"calculates the regression of past time series and calculates the present or future values in the series " (Shetty, 2020)

MA

"calculates the residuals or errors of past time series and calculates the present or future values in the series" (Shetty, 2020)

# The ARMA Model

Here, we fit the test set into a ARMA model. To find the best parameters to use in the ARMA model, we start with fitting the model multiple times (loop through all combinations of parameters p and q from 0 to 5) and calculate the MSE (mean square error.) After comparing use, we store the best performing set of parameters. Then we fit the best model (the best parameter we got from the previous step) with our train set and calculate MSE and MAPE.



# LSTM Model

The road to success and the road to failure are almost exactly the sam

*Colin R. Davis*

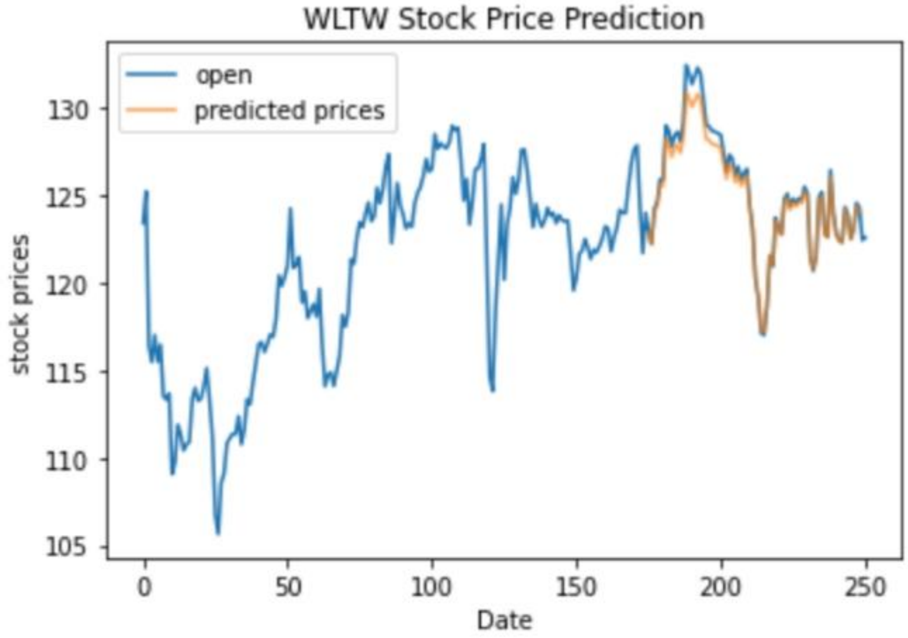
"The Long Short Term Memory architecture was motivated by an analysis of error flow in existing RNNs which found that long time lags were inaccessible to existing architectures, because backpropagated error either blows up or decays exponentially.

An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates – that provide continuous analogues of write, read and reset operations for the cells. … The net can only interact with the cells via the gates."

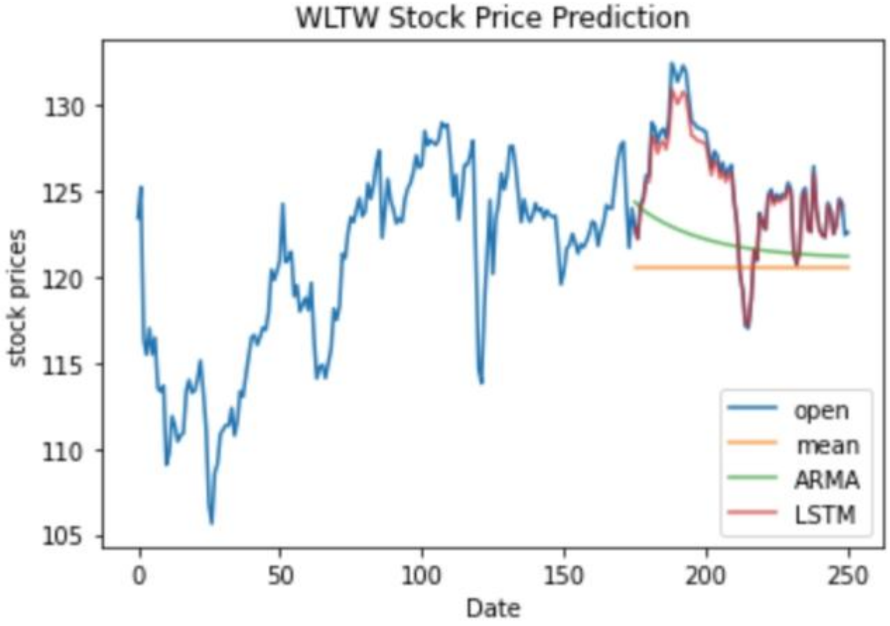
- Alex Graves, et al., Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005.

# The LSTM Model

In this Model, similar to ARMA, we calculate the MSE and MAPE for each stock.

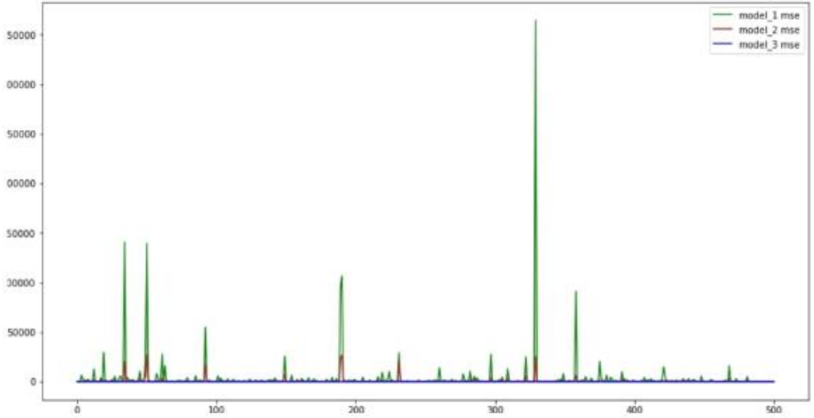


# Comparison

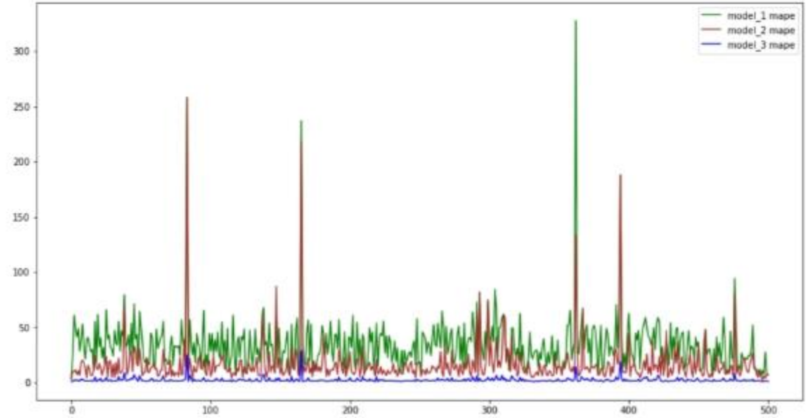

WLTW Stock Price Prediction
stock prices
130
125
120
115
110
105
0
50
100
150
200
250
Date
open
mean
ARMA
LSTM

# Comparison

MSE




MAPE



# Thank You

# Reference


Shetty, C. (2020, September 22). Time Series Models. Medium. https://towardsdatascience.com/time-series-models-d9266f8ac7b0.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 18(5-6), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

Brownlee, J. (2021, July 6). A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Machine Learning Mastery. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.