Springboard Data Science Career Track
Jesse Mailhot
Mentored by Tony Paek

Capstone Project 2
Final Report
Predicting Microsoft Stock Price Movement With LSTM

Introduction

Forecasting stock price movement is an important part of financial management. Using predictions from a machine learning model can help financial advisors make more informed decisions, thereby creating more value for the company and the client. The goal of this project is to build a Long Short Term Memory (LSTM) recurrent neural network to predict the daily, weekly, monthly, and quarterly movement in Microsoft stock price, then compare it to yearly predictions using Facebook's Prophet additive regression model.

Goals

1. Gather data using the Alphavantage API
2. Clean and prepare the data for use in the LSTM and Prophet models
3. Train the models and generate predictions
4. Compare model performance

Dataset

The dataset will be comprised of Microsoft adjusted closing price, and daily high obtained using the Alphavantage API.

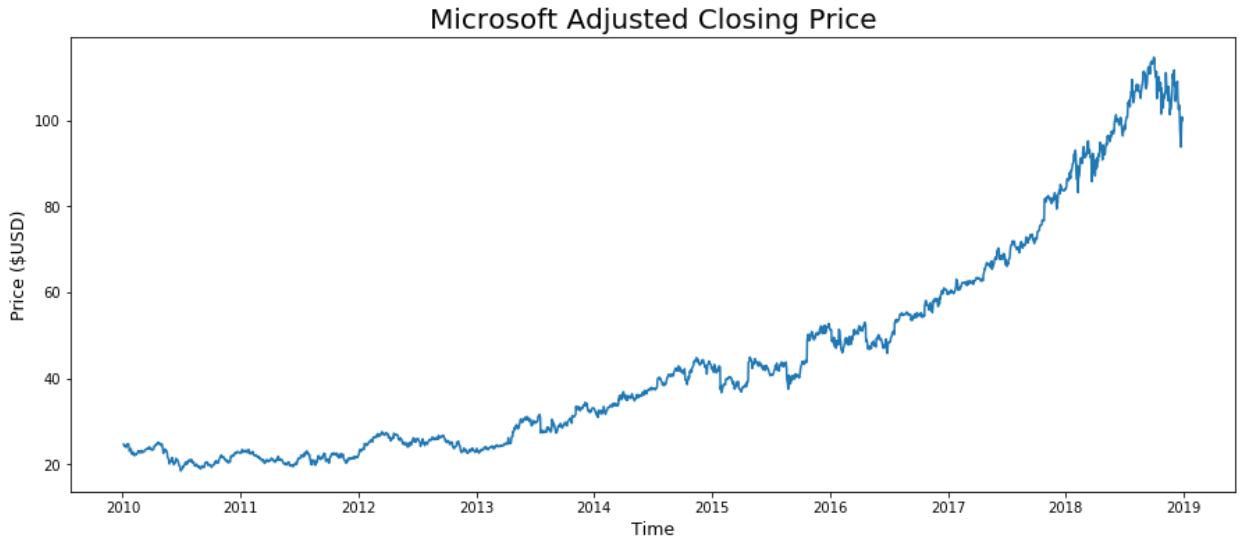
Data Wrangling

A custom function was created to obtain the stock price data using the Alphavantage API. These functions use the requests package to extract the data in json format, load it into a pandas dataframe and return the data for the specified time frame. Once the data has been sliced into the desired time frame, it is loaded into a CSV file and stored in the raw data folder for exploratory data analysis. The dataset contains 2263 rows of Microsoft data from 1/4/2010 - 12/28/2018.

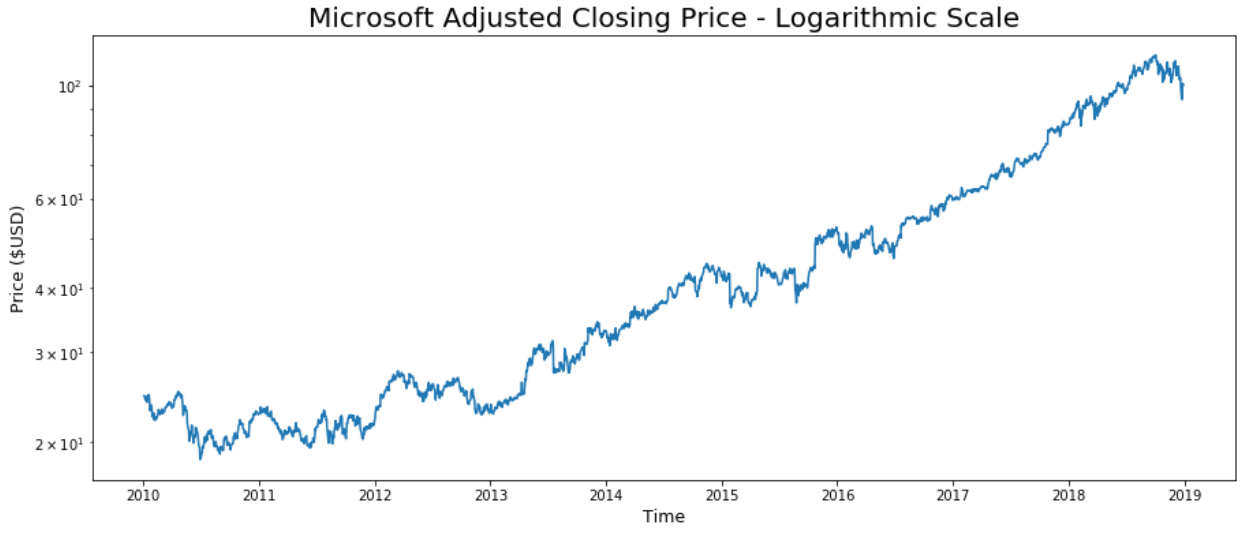
Data Analysis

Raw Data

The raw price data for Microsoft adjusted closing price has a clear long term upwards trend, but short term trends such as daily and weekly movement are much more random. Such movement is normal for stock prices as the markets fluctuate.



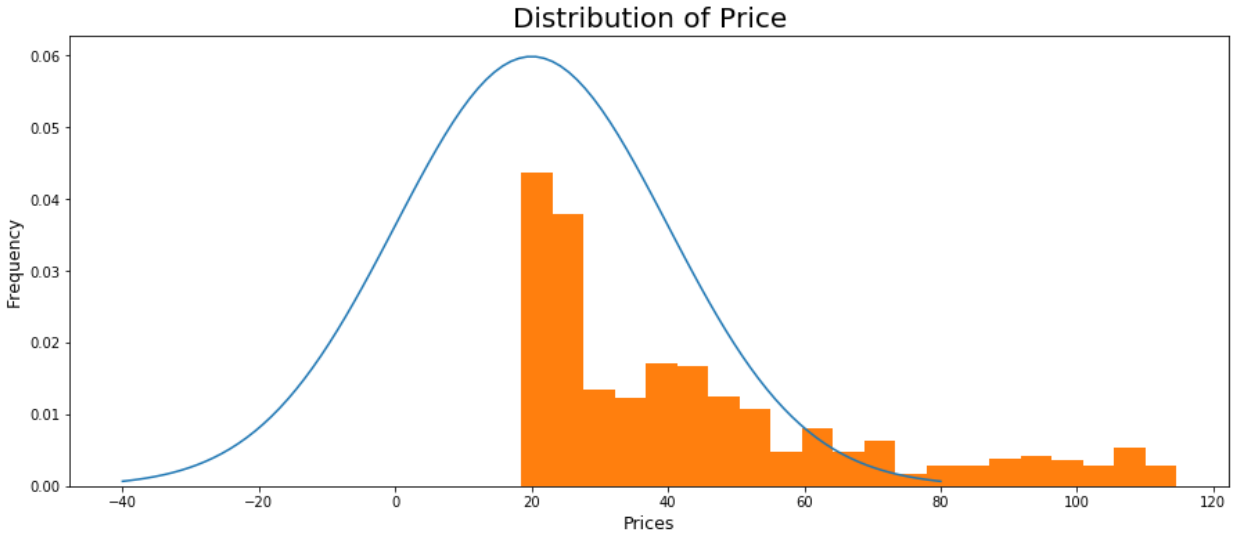
To get a clearer picture of the long term trend, it is advantageous to view the data on a logarithmic scale.



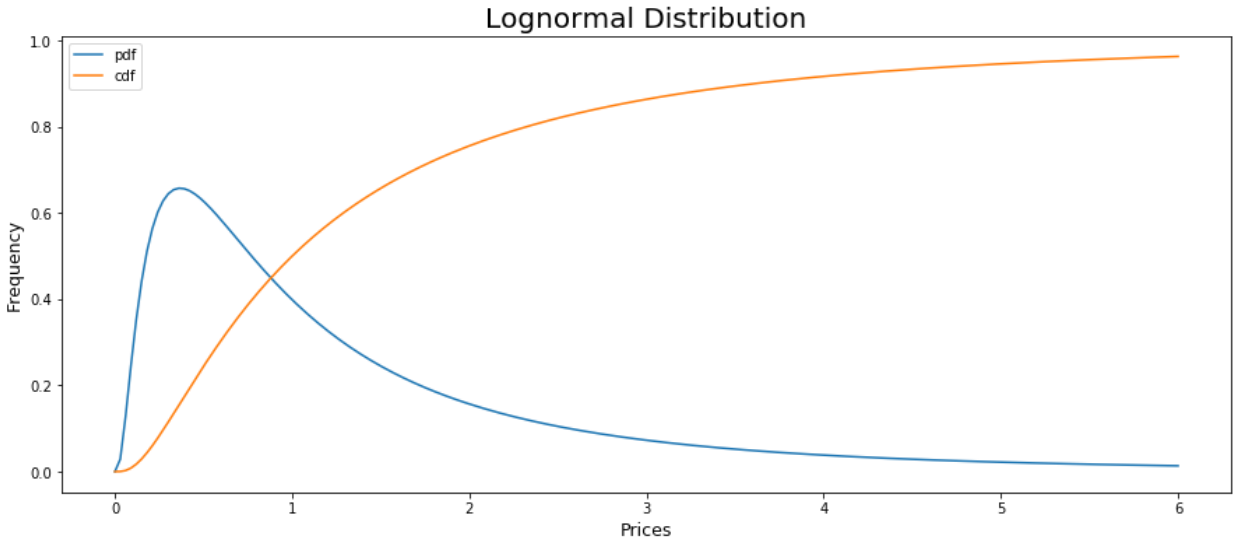
The long term trend of price increase is clearer; however, the short term movement is still very jagged.

Distribution of Price

Examining the distribution of prices, it is clear the data is not normally distributed.



Additionally, price alone is meaningless because it derives its current value from the value of prior periods. Instead of using historical prices to predict future prices, I will use return periods; however, return periods are also not normally distributed. To solve this problem, I will use logarithmic returns instead of traditional holding period returns.



Using logarithmic returns means the data will resemble a lognormal distribution, which more accurately describes daily price movement for financial time series since most daily movement is relatively small. The intuition behind using logarithmic returns is that it is equivalent to infinitely compounding returns. The logarithmic return also represents the instantaneous rate of change of return, which describes the current direction the stock is moving in, which is ultimately what the model will try to predict.

Preprocessing

LSTM

The first step in preprocessing is to transform the data into a supervised learning problem. To do this, each row of data will contain the previous 60 days and the future 60 days from each data point. Next, the data will be split into randomized testing and training sets to avoid any bias from time series. This is a very important step; without it, the model would be biased towards the historical value of the stock, and would not be adverse to future changes in the stock price differing from the historical trend. After the data has been split into testing and training sets, each set is then reshaped into the LSTM format: actual data, number of periods (steps), and number of features.

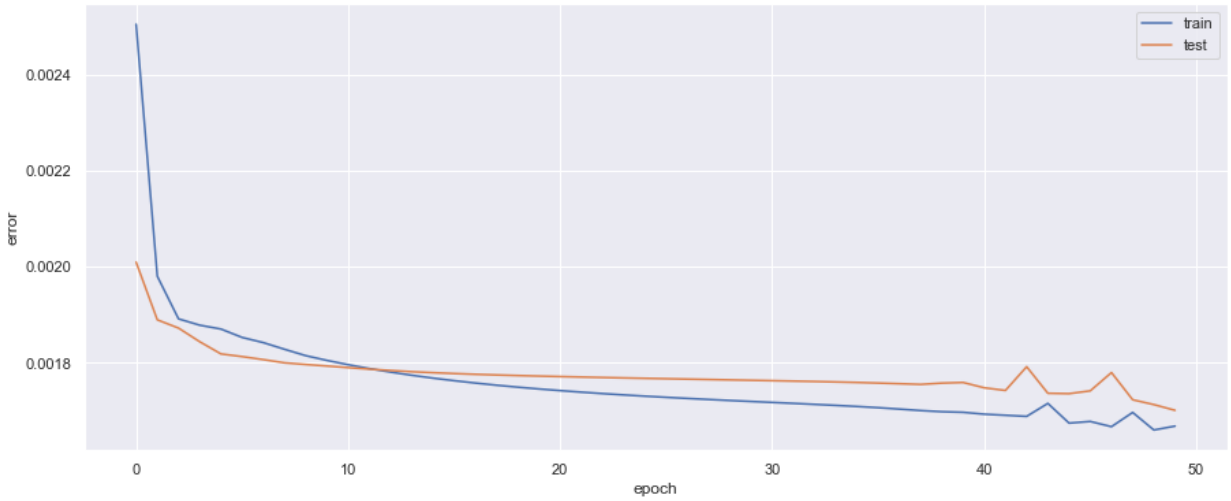
Prophet

For the prophet model, the data preparation is much simpler. Prophet takes as input a Pandas dataframe with 2 columns: ‘ds’ (datestamp) and ‘y’ (historical data). To get comparable results for the 2 models, the prophet model will train on 1 year of data, then make a prediction for the following year. Each iteration will call a new instance of prophet in order to avoid historical bias.

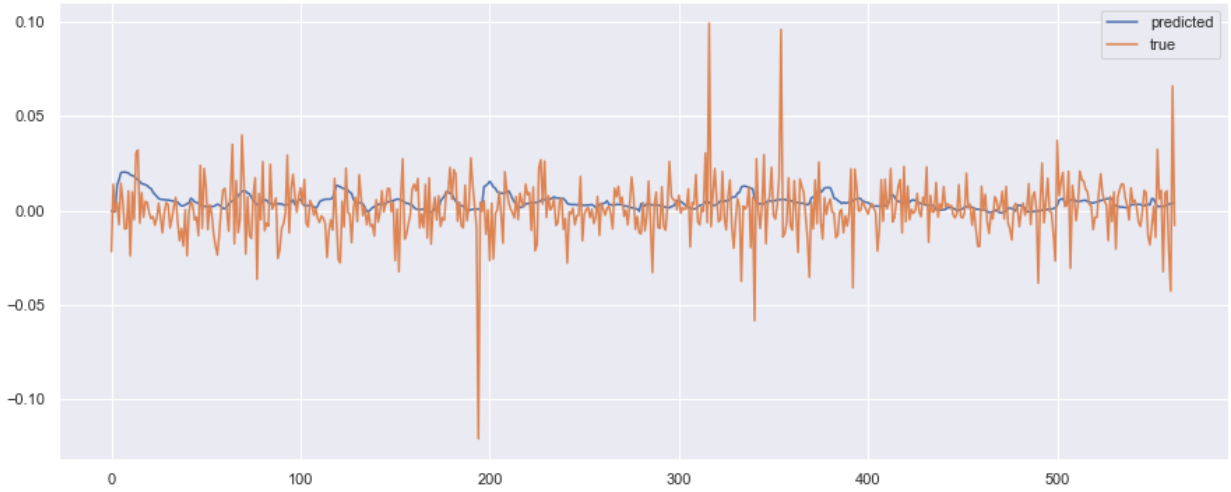
Model Analysis

LSTM

The LSTM model for this project will have four layers: three 50 node LSTM layers and one 4 node dense layer for output. The will use mean squared error as the loss function with the ADAM optimizer. The model will be trained over 50 epochs with a batch size of 72. Below is a plot of the training and validation loss.



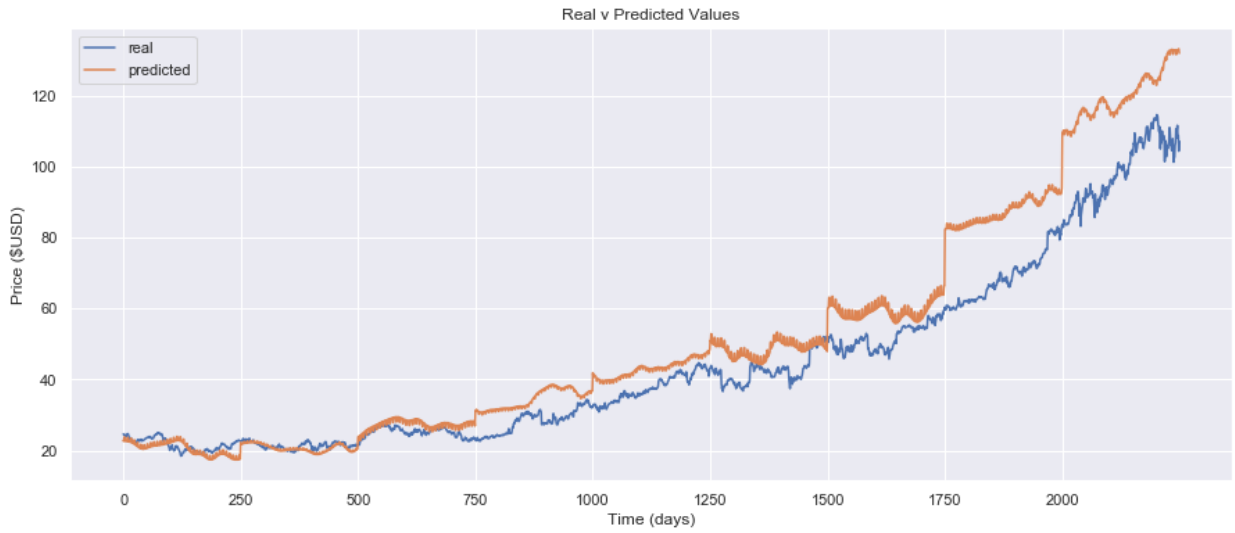
After compiling and training the model, it is run on the test set. The results for daily returns are visualized below.



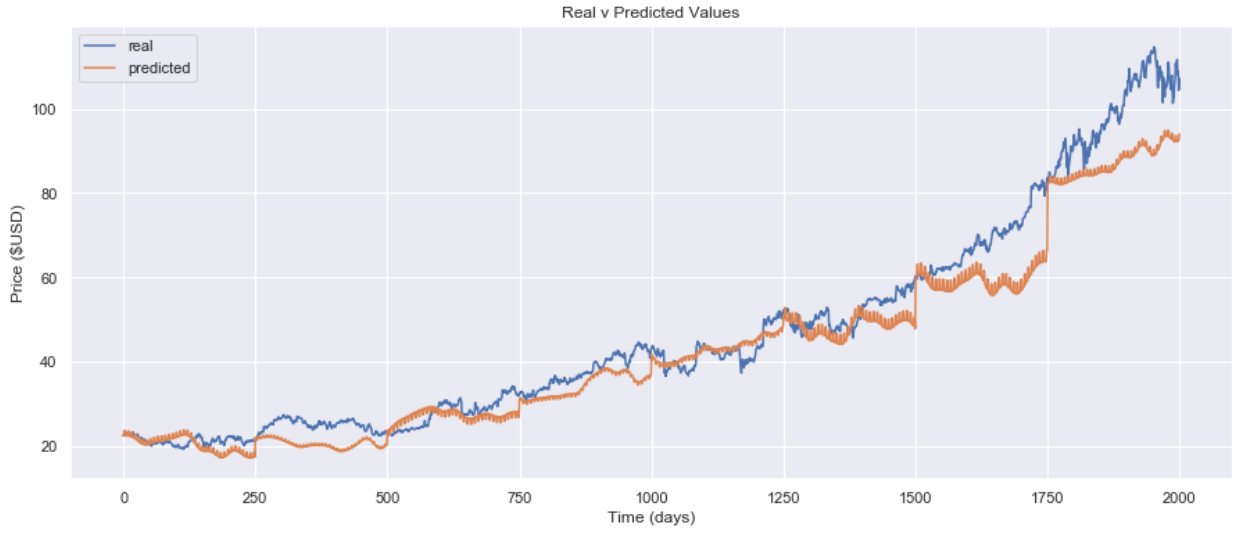
The LSTM model correctly predicted the direction of the daily stock price movement 53% of the time; whereas it correctly predicted the direction of the quarterly stock price movement 74% of the time.

Prophet

The prophet model will be the standard out-of-the-box model, using the yearly seasonality. Running the model in a for loop for each year produced the following results.



From the plot it is interesting to note that as time progresses, the model starts making large jumps from the previous endpoint, especially at 1750 and 2000. Running each model separately (not in a for loop) produces significantly different results, as shown below.



There is a clear difference in the way the model reacts when run in a loop compared to being run individually. In a production setting, this model would likely only be run on one year at a time, so it is not necessarily a problem.

Model Comparison

The prophet model correctly predicted the direction of the stock price movement for each of the 7 years it predicted on. This should be taken with a grain of salt; however, due to the biased nature of training a regression model on historical stock data. Given that the underlying goal of the stock market is to create value and increase prices, simply predicting that the stock price would increase every year would yield a correct prediction 6 out of 7 times.

The LSTM model is designed to be averse to this kind of bias, and therefore I assert that it is the better of these two models for the purpose of predicting stock price movement. The LSTM model performed better on longer term data, 74% correct quarterly direction prediction compared to only 53% correct daily direction prediction.

Next Steps

Further evaluation could be performed by testing these models on other stock prices, both related to Microsoft or in other sectors such as retail or stock price indices like the S&P 500. Cross validation could also be expanded upon, such as testing the prophet model with quarterly predictions instead of yearly. This would generate a more comparable dataset to the LSTM model and provide more insight into how well the models perform. Other features could also be incorporated into the dataset as well, such as a rating of articles relating to the stock for each day. Collecting and analyzing this data would likely take a significant investment of time, but could potentially improve the overall model performance of the LSTM. Alternatively, other deep learning frameworks could be applied such as CNN, or other types of RNN models such as GRU.