Stock Market Price Prediction

CS 178: Machine Learning and Data Mining
Final Project Report

Final project group #29
Darren Lim
Jerson Villanueva
Yingda Tao

Problem Statement

Our project will predict stock price using linear regression and long short term memory (LSTM) recurrent neural network. We will also compare the two data models using the quantitative results showing the differences between the two models, and ultimately, the better predictor.

As a general side note, stock market price predictions have too many variables to keep track. There are technical variables such as statistical figures and charts, as well as fundamental variables such as the company's environment and financial performance [2]. Although company sentiment and how people feel about their value is important in stock market prediction, we did not have the time nor the research methods to collect and use that kind of data. Therefore, this project solely focused on the technical side using previous market prices and "predicting" how its trends will affect future prices.

Project Decomposition

We split up the progression of this project into different milestones / goals.

1. Finding the data for the project (Everyone)
    - This includes researching methods and articles about stock prediction. Also, finding appropriate stock market data; we chose two companies, Amazon and Google, because we wanted more than one graph of each model.
2. Coding the linear regression model (Darren, Jerson)
    - The code for the linear regression was coded by the group.
3. Coding the Long Short Term Memory (LSTM) Recurrent Neural Network model (Yingda)
    - The code for the LSTM model was taken from Singh's model and adapted to our dataset. [2].

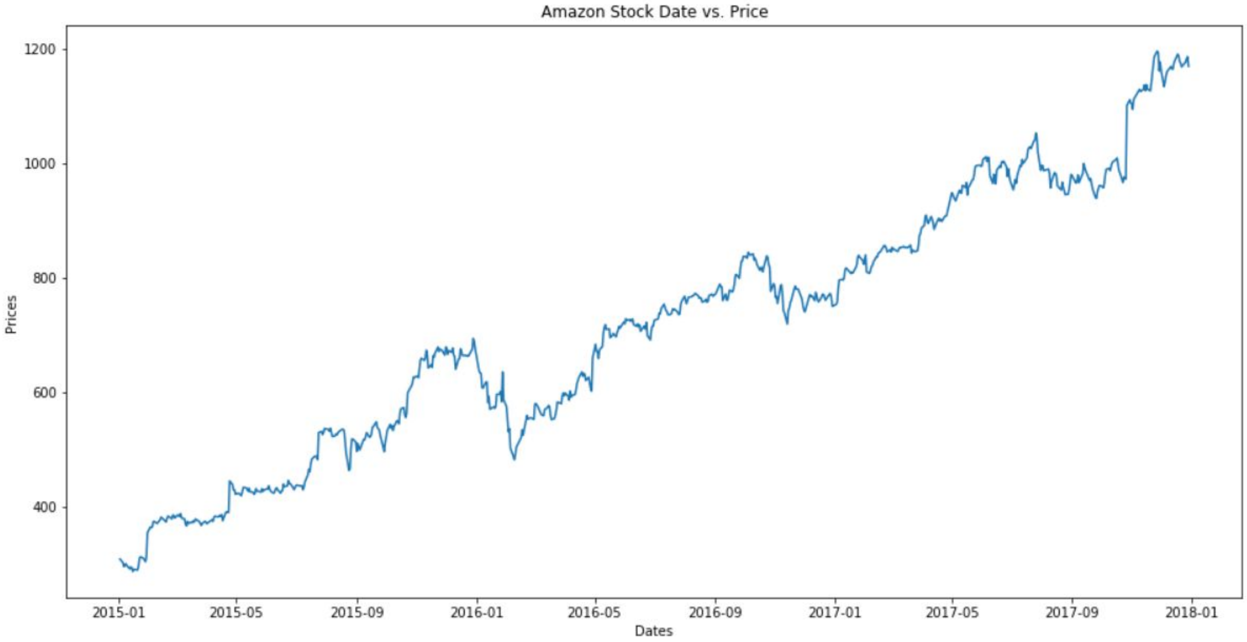
4. Comparing the experiments / results (Everyone)
5. Conclusion (Everyone)
6. Putting together the poster (Everyone)

We divided up some of the work to model the dataset, but everyone worked on this paper as well as comparing each model.

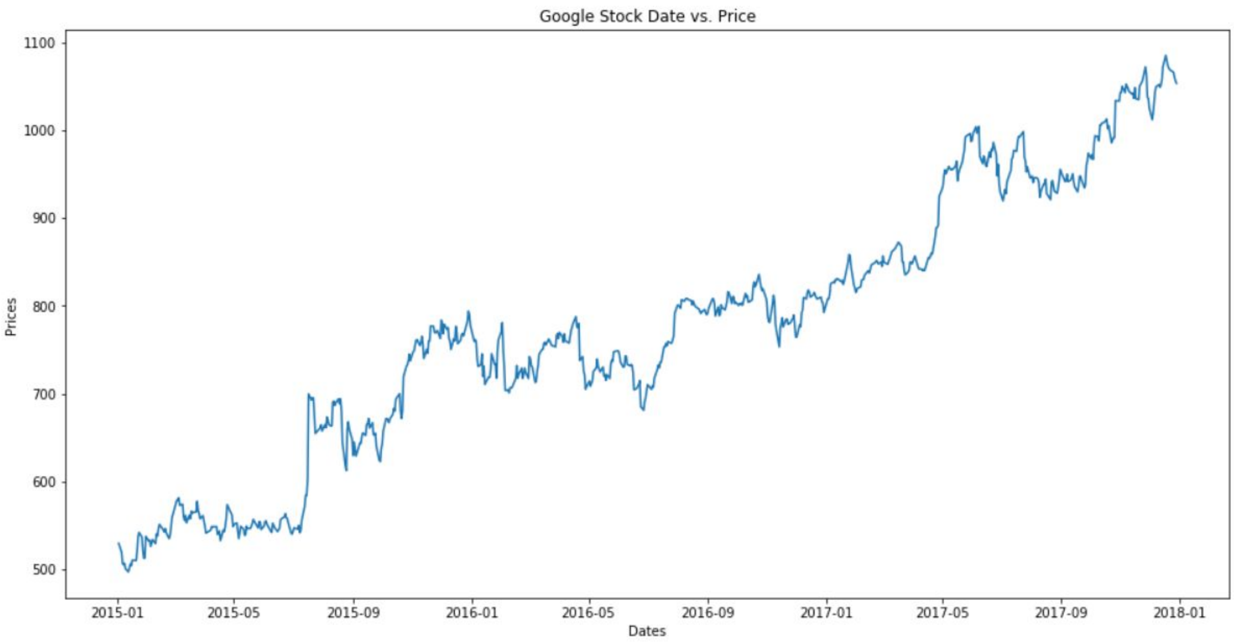
Experience Coding the Experiment

We decided to use quandl because it seemed the easiest way to get stock market data. Pulling the dataset from quandl was easy, all we had to do was install the quandl package, make an account on the quandl website, and use the account api key to grab the Amazon and Google stock data. However, to keep getting the data every time we run the code, an api key is required. We used the closing prices over a three year period for the dates 2015-1-1 to 2018-1-1 as our dataset to train, test, and graph our models. However, some dates will be missing from the graph because the stock market is not open on weekends or holidays. Taking the prices and data, we had to reshape them into a list that made it easier to plot [5,9]. The closing prices and dates consisted of 753 different points.

Here is the Amazon's closing price stock graph from 2015-1-1 to 2018-1-1



And here is Google's closing price stock graph from 2015-1-1 to 2018-1-1

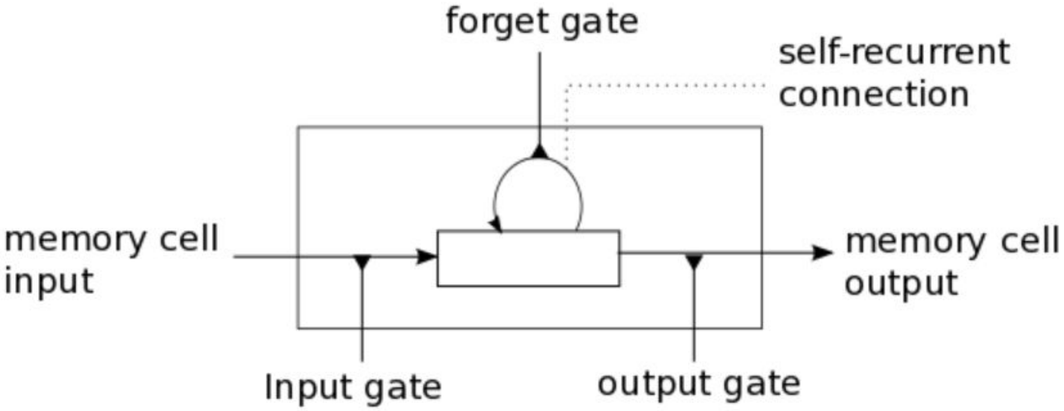


Linear regression is a supervised learning model that determines the relationship between the independent variables and the dependent variable. It is a very common and baseline model used in various problems [4]. In our experiment, the independent variable is the dates and the dependent variable is the price.

Since linear regression was part of homework 2, we needed a refresher to remember how to use it. So we just looked back over homework 2 to figure out how to fit and graph the data. This part of the project was done by ourselves. We decided to split the data into 70% testing and 30% training. The training data was then fit onto a linear regression model and we graphed the predicted values. Surprisingly, the error rate was high, but the graph fit the predictions as expected. Looking at Sultangulova's article, it seems our method and her initial method are similar [4].

From Mwiti and Singh's article, we get to know how to predict the stock market price using Long Short Term Memory (LSTM). LSTM is an artificial recurrent neural network architecture used in the field of deep learning. It can not only process single data point like images, but also entire sequence of data like videos[6]. A common LSTM consists of three gates: the input gate which adds information to the cell state, the forget gate which removes the information that is no longer required by the model and the output gate which LSTM selects the information to be shown as output[2].

Here's a simplified illustration for LSTM memory cells:



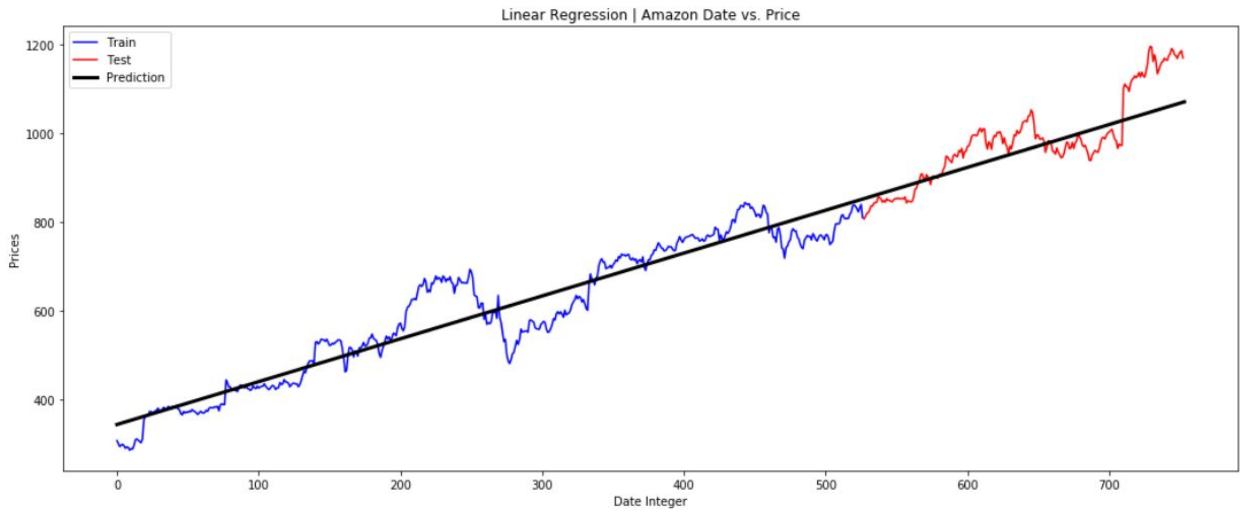
Adapted from “LSTM Networks for Sentiment Analysis”[11]

We finally decided to implement it because LSTM networks are well-suited to processing, classifying and making predictions[6] based on time series data (which works well for the purpose of predicting stock market). Given that it’s able to store past information that is important, and forget the information that is not[2], it fits perfectly to our subject. The accuracy of stock price prediction is heavily based on the previous data.
After importing python library Keras which uses a Tensorflow backend, we used “Sequential” for initializing the neural network, “Dense” for adding a densely connected neural network layer and “LSTM” for adding the Long Short-Term Memory layer[3]. We also imported MinMaxScaler to transform the dataset. Like what we did using linear regression, we let the first 75% of the data become our training data and the latter 25% become test data. With MinMaxScaler, we converted the dataset into x_train and y_train. After reshaped it, We add the LSTM layer and later add the Dense layer that specifies output of 1 unit. After that, we compile our model using the adam optimizer and set the loss as the mean_squarred_error[3].

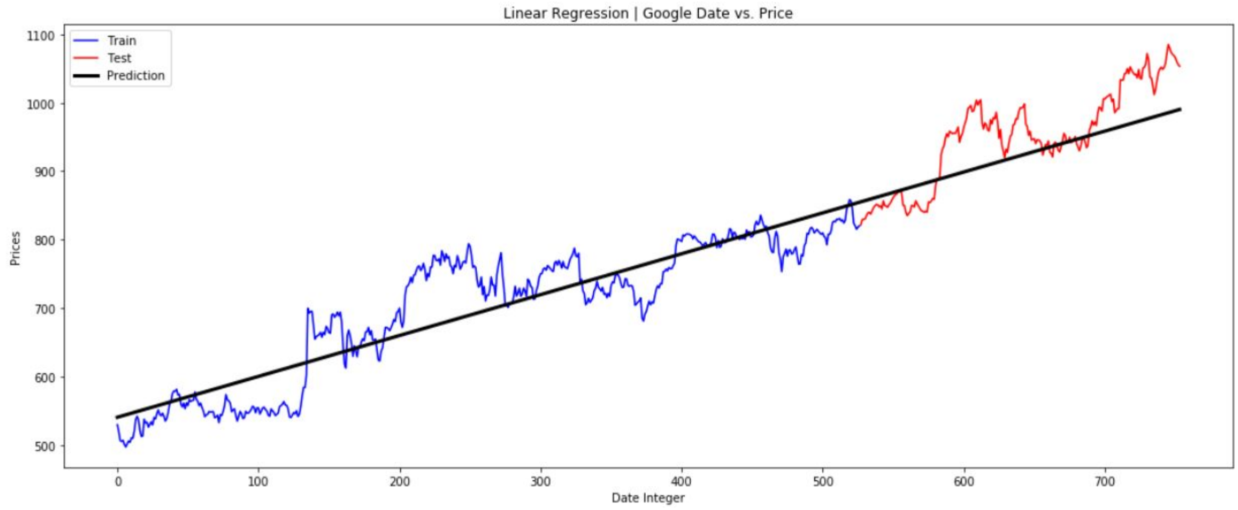
Also from our experience, the LSTM model will predict different prices each new time it is run.

Comparing Each Model / Results

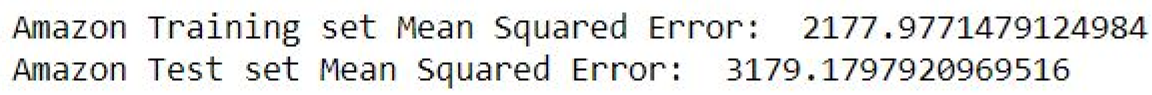
Linear Regression graph for Amazon’s data points:



Linear Regression graph for Google's data points:

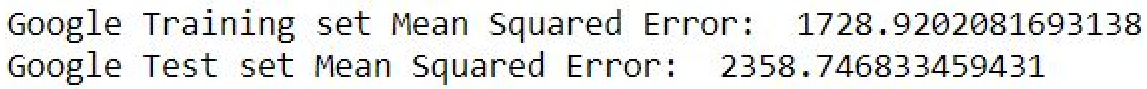


Firstly, although the results of the linear regression graph showed a best fit line for the training data, we can immediately tell it is not a good model for price prediction. Most of the test data has a high variance around the predicted regression line, which yielded a very high error rate. The test and training set error rate is as shows for Amazon:



```
Amazon Training set Mean Squared Error:  2177.9771479124984
Amazon Test set Mean Squared Error:  3179.1797920969516
```

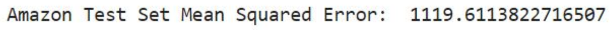
And the error for Google:



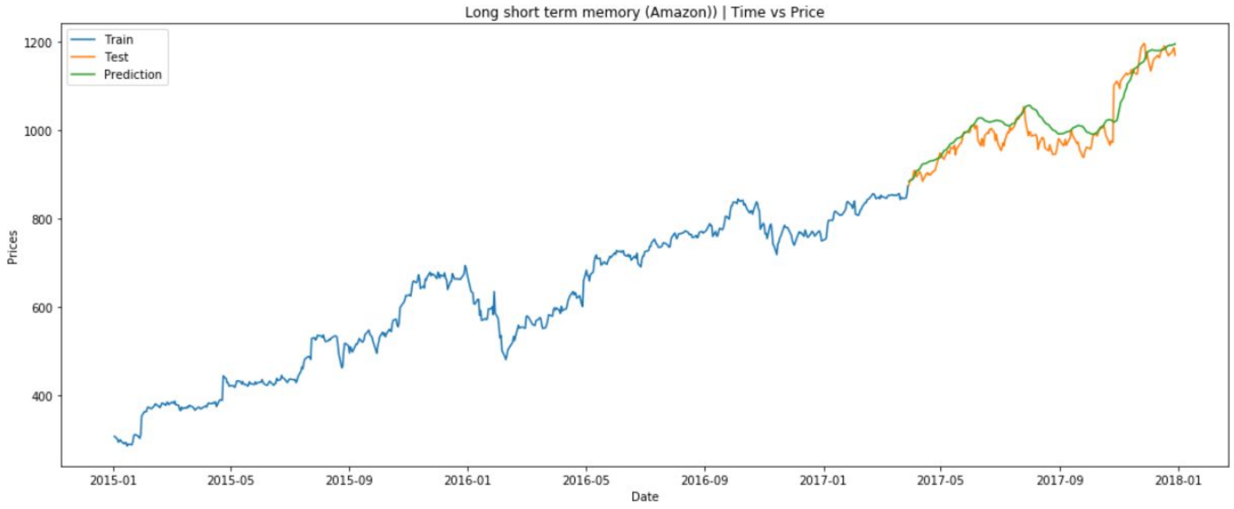
```
Google Training set Mean Squared Error:  1728.9202081693138
Google Test set Mean Squared Error:  2358.746833459431
```

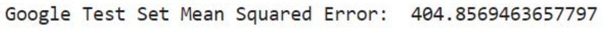
However, because it fit the trend of the rising stock prices, linear regression is a good way to predict a general rise or fall of prices over time.

As for the LSTM model, here are the results for Amazon and Google:

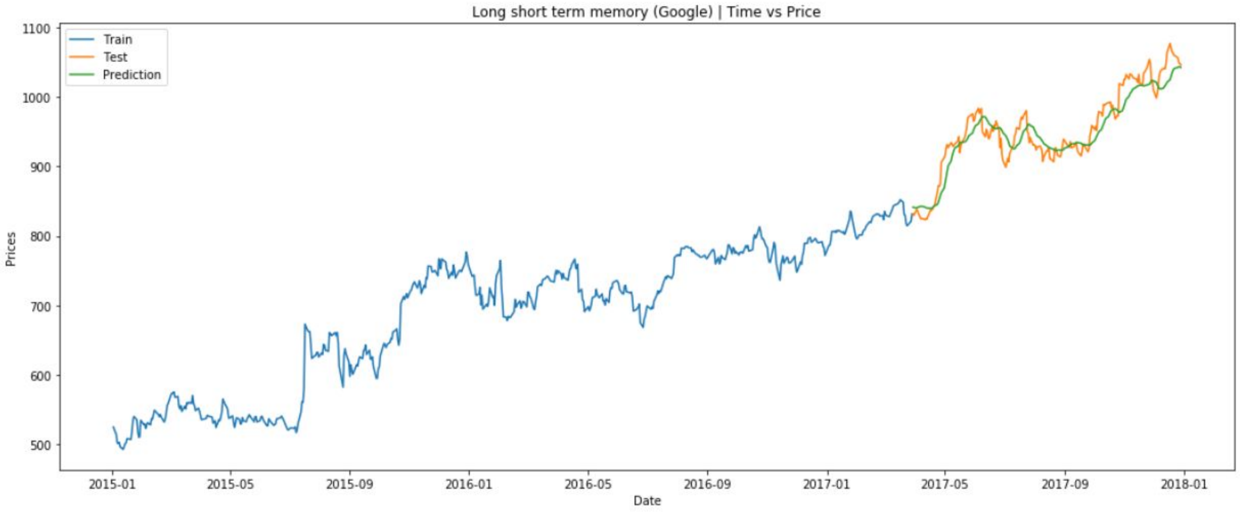


```
Amazon Test Set Mean Squared Error:  1119.6113822716507
```





```
Google Test Set Mean Squared Error:  404.8569463657797
```



As we discussed earlier, LSTM has been proved to be extremely effective on sequence prediction problems. Compared to the result from linear regression, the mean squared error on predicting Amazon's stock reduced from 3000 to 1000. For Google's database, it's reduced from 2300 to 400 which shows a large gap.

Comparing the two graphs visually and the error rate, the long short term model fits a lot better than the linear regression model. As stated before, the error rate for LSTM is also smaller than that of linear regression. However, linear regression can be a usable model for long term stock price predictions [4], so we shouldn't disregard the model. Also from our experience, linear regression is easier to understand than LSTM, due to LSTM having complicated mathematical theory versus the simplicity of linear regression.

Conclusion

From our comparisons, we can surely say the LSTM model is the better choice. The error rate is lower and it better fits the test cases, versus linear regression.

Although LSTM is a good model, our real world is way more complex than what machine learning can cope with. As stated at the beginning of this paper, the actual stock price has many factors that computer are not able to analyze. Political atmosphere, news, merger/demerger of the companies could all be aspects affecting the price. Our data shows using closing prices is not nearly enough to properly predict changing prices. Thus, our methods used in this experiment are not very useful in accurately predicting stock prices.

Bibliography

[1] Alexandre Xavier. “Predicting stock prices with LSTM”, January 2018
https://medium.com/infosimples/predicting-stock-prices-with-lstm-349f5a0974d4

[2] Aishwarya Singh. “Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)” October 2018.
https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/

[3] Derrick Mwiti. “Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices” KDnuggets, November 2018.
https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html

[4] Dinara Sultangulova. “Future stock prices prediction based on historical data using simplified linear regression” Github, https://oftomorrow.github.io/linear-regression.html

[5] Frank. “Prediction Stock Prices with Linear Regression” Programming For Finance
https://programmingforfinance.com/2018/01/predicting-stock-prices-with-linear-regression/

[6] “Long short-term memory” Wikipedia.
https://en.wikipedia.org/wiki/Long_short-term_memory

[7] “nyilmaz1: Google-stock-price-prediction” Github,
https://github.com/nyilmaz1/Google-stock-price-prediction/blob/master/Google%20Stock%20Price%20-RNN%20LSTM.ipynb

[8] Pierre Luc Carrier, Kyunghyun Cho, “LSTM Networks for Sentiment Analysis”
http://deeplearning.net/tutorial/lstm.html

[9] Samay Shamdasani. “Build a Stock Prediction Algorithm” Enlight, December 2017.
https://enlight.nyc/projects/stock-market-prediction/

[10] Vivek Palaniappan. “Neural Networks to Predict the Market” Towards Data Science, September 2018.
https://towardsdatascience.com/neural-networks-to-predict-the-market-c4861b649371

[11] Will Koehrsen. “Stock Prediction in Python” Towards Data Science, January 2018. https://towardsdatascience.com/stock-prediction-in-python-b66555171a2