MSBA SECTION B

TEAM 10B

Basil Latif, Manodhar All, Xiaotong Liu, Rong Fang, Luyao Li

Stock Market Data Project - Data & Programming Analytics Final Report

Professor Tingting Nian

BANA 212

December 12 2019

Table of Contents

INTRODUCTION

The team had a meeting in early September 2019 to choose a topic for the “Data and Programming Analytics” Final Project at the MSB Building in the Paul Merage School of Business. At the meeting, each team member put up an idea for what the team should choose. After discussion, 5 items were up for vote and a decision was made that the team would look at individual stock data from the Nasdaq and Dow Jones exchanges for the project. In later meetings, we chose to study 4 stocks in particular—Amazon, Apple, Netflix, and Google.

We chose the topic of analyzing US stocks because we thought working with stock market data would be a real-world business example that would teach us about the inner workings of the stock market and is a real world use case of analytics in the financial industry also known as “Financial Analytics.” At a networking mixer at the Paul Merage School of Business, copies of the book “Everybody Lies: Big Data, New Data, and What the Internet Can Tell Us About Who We Really Are” by Seth Stephens-Davidowitz were given from the school to the students. In the book, Stephens-Davidowitz looks at many of the interesting ways Internet data can be analyzed using modern analytics and data science techniques. He shows that Internet data can be analyzed to understand the individual in a ultra-targeted way such as sentiment analysis on Twitter to predict stock prices. Traders and hedge funds were using Tweets to anticipate major moves in the US stock market. Such a financial strategy was the inspiration for the project.

We knew that the stock market is a rich and reliable data source that would be good to work on from a data mining and analysis perspective, so we chose to analyze Amazon, Apple, Netflix, and Google stock data to further understand the technology sector and what drives revenue for these Internet-based companies. We also wanted to analyze the effects of daily volatility in the stock market.

For the modeling part of our project, we want to understand different stock prediction models. We want to know what happened in the stock market for the last 5 years to build a predictive model for the following year. We want to understand how

stock prediction models work on the backend, what factors they consider, and how their underlying structure works. We also want to compare the performance of various models against each other.

The 2 questions our project aims to answer are listed below:

(1) Can we analyze what factors causes changes in stock price at large technology companies?

(2) Can we build a predictive model to predict future stock price with a high amount of accuracy?

DATA USED:

We used the Quandl API for the data for this project. Quandl is a financial data company based in Toronto founded in 2011. In order to access the Quandl stock market data, one needs to register for an API key to gain access to all free products. With the API key, one can access the Quandl API using a URL with the key embedded in it to download the data.

For our project, we used a Python script to download the data. We were able to download the data and save it to a CSV file. In order to use the API, the user has to specify the start and end date for when the data is needed. Quandl provides options to download up to 13 headers for the CSV file, including Data, Open, High, Low, Close, Volume, Ex-Dividend, Split Ratio, Adj. Open, Adj. High, Adj. Low, Adj. Close, Adj. Volume. We specified all of the headers to get a complete picture of each stock's data. We wrote the data to a CSV file with the headers. At first, we downloaded only 5 years of data for each stock. Since there are roughly 253 trading days in a year (5 weekdays a week minus federal holidays), we had roughly 1500 rows of data for each stock. As we had downloaded data for 5 stocks, we had in total about 7500 data points. Later, we realized that we could extend our analysis further, so we adjusted the Python script below to go back 10 years. A nice feature of stock market data is that it comes complete with no missing or erroneous values. Therefore, we had no data cleaning or processing to do.

Step 1 : Write a Python script to Download data from the Quandl API and store into CSV file

```
In [3]: fb_df = pd.read_csv("/Users/basillatif/Desktop/Stock Data Project/fb_data.csv")
        fb_df
```

Out[3]:

	Date	Open	High	Low	Close	Volume	Ex-Dividend	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close	Adj. Volume
0	2018-03-27	156.31	162.85	150.75	152.1900	76787884.0	0.0	1.0	156.31	162.85	150.75	152.1900	76787884.0
1	2018-03-26	160.82	161.10	149.02	160.0600	125438294.0	0.0	1.0	160.82	161.10	149.02	160.0600	125438294.0
2	2018-03-23	165.44	167.10	159.02	159.3900	52306891.0	0.0	1.0	165.44	167.10	159.02	159.3900	52306891.0
3	2018-03-22	166.13	170.27	163.72	164.8900	73389988.0	0.0	1.0	166.13	170.27	163.72	164.8900	73389988.0
4	2018-03-21	164.80	173.40	163.30	169.3900	105350867.0	0.0	1.0	164.80	173.40	163.30	169.3900	105350867.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1467	2012-05-24	32.95	33.21	31.77	33.0300	50237200.0	0.0	1.0	32.95	33.21	31.77	33.0300	50237200.0
1468	2012-05-23	31.37	32.50	31.36	32.0000	73600000.0	0.0	1.0	31.37	32.50	31.36	32.0000	73600000.0
1469	2012-05-22	32.61	33.59	30.94	31.0000	101786600.0	0.0	1.0	32.61	33.59	30.94	31.0000	101786600.0
1470	2012-05-21	36.53	36.66	33.00	34.0300	168192700.0	0.0	1.0	36.53	36.66	33.00	34.0300	168192700.0
1471	2012-05-18	42.05	45.00	38.00	38.2318	573576400.0	0.0	1.0	42.05	45.00	38.00	38.2318	573576400.0

Step 2: Load Data into Pandas Dataframe in Jupyter Notebook

Step 3: Version Control Jupyter Notebooks in GitHub repository

GitHub Repository

https://github.com/basillatif/Stock-API-Dev/blob/master/README.md.

Step 4: Exploratory Data Analysis

Step 5: Predictive Modeling

Step 6: Insights Learned

EXPLORATORY DATA ANALYSIS:

We realized the main numeric columns we would be analyzing would be as follows:

Ex-dividend: a cash payout to holders of a stock

Volume: number of shares that are traded daily

Split Ratio: a split in stock price indicates that the price of the stock goes down but the number of shares goes up by some ratio

Close: the daily closing price of each stock

Adj. Close: adjusted prices account for things like after-hours trading and changes to stock price caused by stock splits or dividend payouts

The first step we did in order to analyze stock price was do feature engineering on our csv data—using financial analytics. We created the following new columns:[1]

Daily Lag: stores the closing price from the day before

->fb_df['Daily Lag'] = fb_df['Close'].shift(1)

Daily Returns: tells you the returns you achieve each day

->df['Daily Returns'] = (df['Daily Lag']/df['Close']) -1

Moving Average: calculated by taking the average over a period of 10, 50, or 200 days

->mavg = close_px.rolling(window=10).mean()

Our first step in analyzing common predictors that affect the stock price. For each stock, we generated many different graphs to see what underlying trends we could detect. To that end, we generated a price line graph for each stock as well as a volume chart. We added further information to the price charts by including lines for the 10 days, 50 days, and 200 days moving average of each stock.

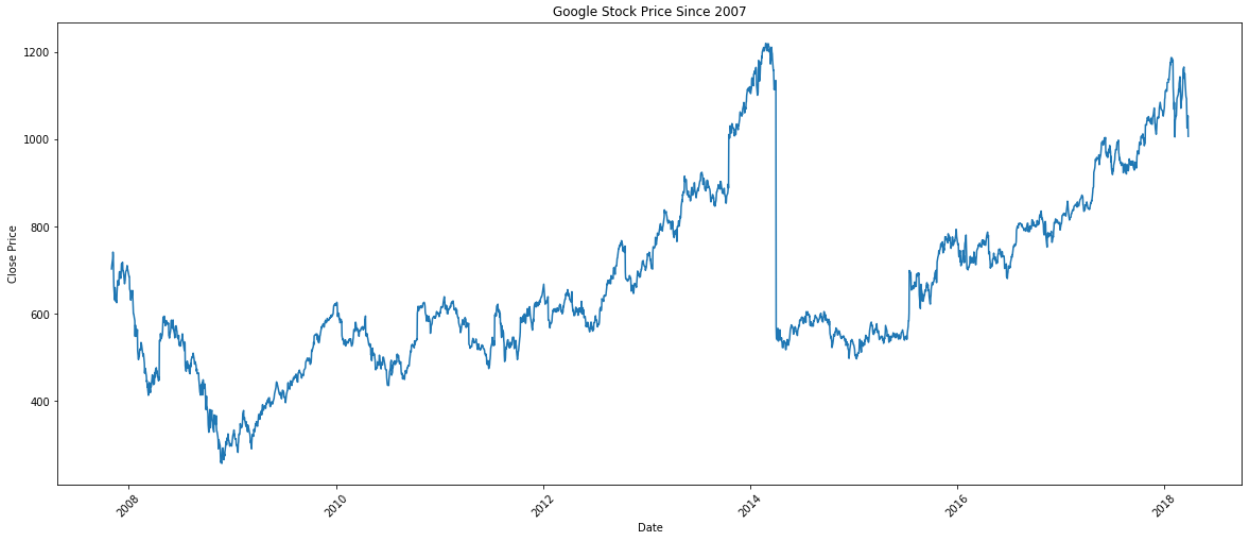
Another key part of our analysis involved knowing how to interpret volume charts. Generally, we used a 2-variable approach to do volume analysis. If the volume was increasing and the price is increasing, then we interpret a bullish market. If the volume is decreasing and the price is decreasing, then the we also interpret as bullish. For the other 2 cases, volume increasing with price decreasing or volume decreasing and price rising, then we interpret as bearish.

[1] Kiat, Kang Choon. "Financial Analytics - Exploratory Data Analysis of Stock Data." *Medium*, Towards Data Science, 3 July 2019, towardsdatascience.com/financial-analytics-exploratory-data-analysis-of-stock-data-d98cbadf98b9.

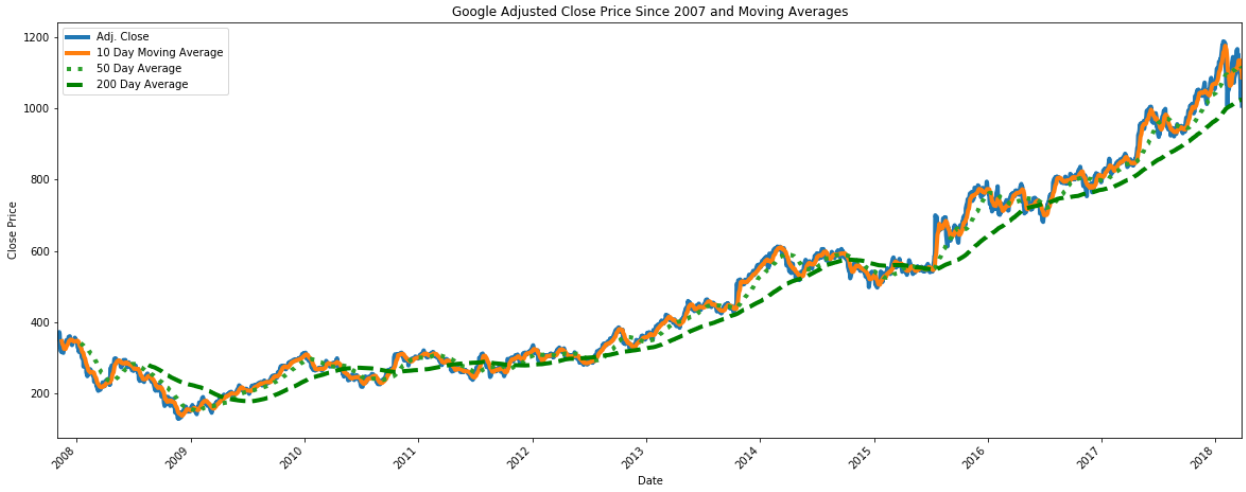
Google

For Google, we could see that the upward trajectory of the stock since the beginning of the dataset in 2007. When we initially plotted the data, we noticed a sharp decline in the price in 2014.

Visualization #1: Google Stock Price Since 2007



Visualization #2: Google Daily Stock Price Since 2007



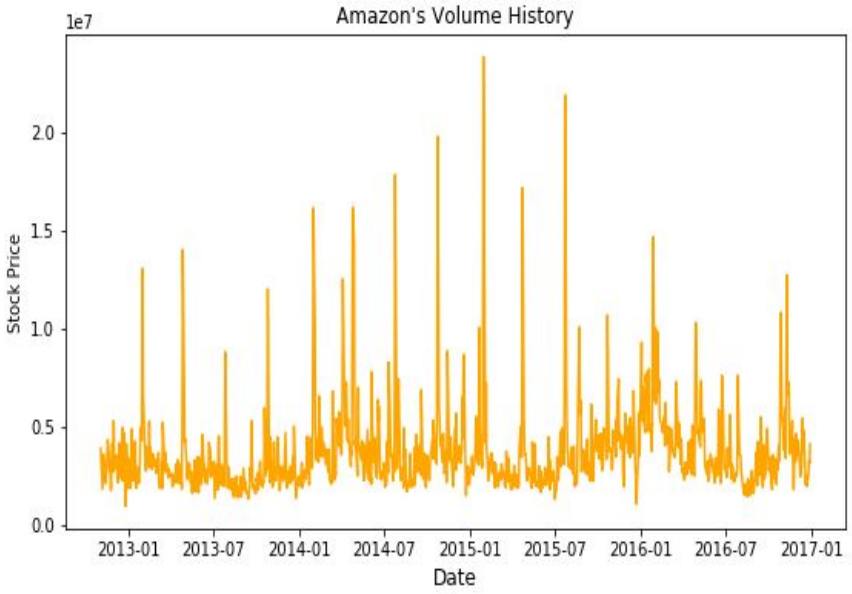
Upon investigation, it was easy to see a sharp price drop on April 3, 2014 in the first graph. On that day, Google made a dividend payment to its shareholders for the amount

of $567.97. What that means is that every owner of Google stock would receive a payment of ~$568 and double their number of shares. As a result, new investors can buy new Google stock at half price and investors who held Google stock before this would double their number of shares and get a payout for each stock owned and double their number of shares. Google did this, instead of a normal stock split, because they wanted to protect the voting power of their board members.[2]

Amazon

The graphs below show Amazon's price history and volume history between 2012 and 2017. There was a significant drop due to the company's disappointing earnings in 2016. However, the price increased right after the drop due to the launch of Alexa-powered devices. This suggests that the stock price can change rapidly by new product launches as analyzed from news sources.

Visualization #3: Amazon Volume Chart



The following graphs show Amazon's moving average price and historical price between 2012 and 2017. Stock price increased after Thanksgiving or Black Friday every

[2] "New Class of Shares Stock Dividend Google Spin Off April 2014." *TIMETOTRADE*, wiki.timetotrade.com/New_Class_of_Shares_Stock_Dividend_Google_Spin_Off_April_2014.

year. It was due to people's shopping routine or due to Amazon's BF sales-pushed strategies, which indicates that the stock price can also be driven by holidays or company's sales strategy. The Volume Analysis table helps to analyze trends in trading volume in conjunction with price movements, which can help to determine the significance of changes in price trend.

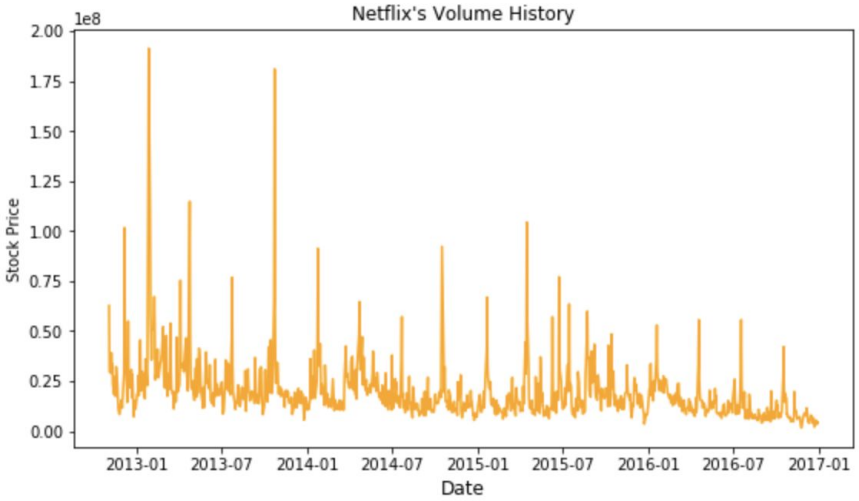
Visualization #4: Moving Average Chart for Amazon

Visualization #5: Amazon Historical Price

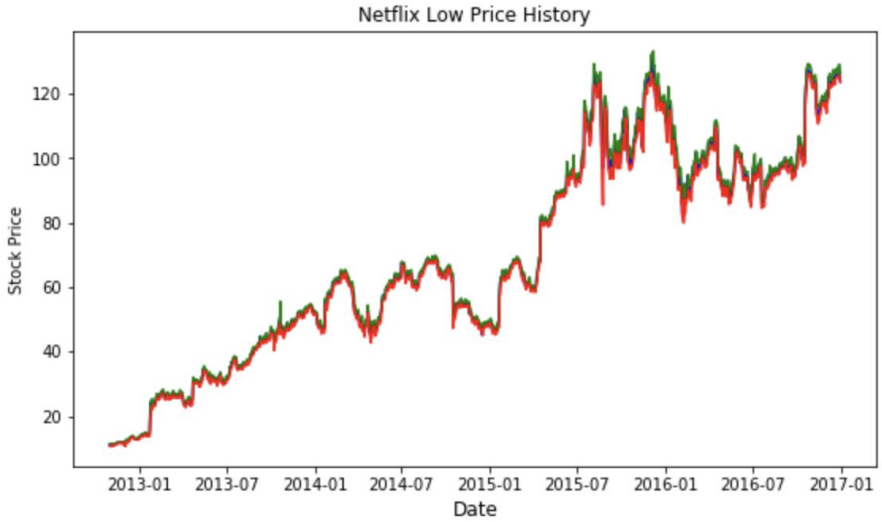
Netflix

The following graphs show Netflix's price and volume history between 2012 and 2017. There was a sharp drop in 2017, which is caused by Netflix raising the price of its monthly plan from $9.99 to $11.99. This indicates that stock price can be driven by product pricing change or new policy. In addition, Netflix's volume is trending downward as price is going up, which signals to investors that the bullish trend might soon end.

Visualization #6: Netflix Volume History



Visualization #7: Netflix Low Price History



MODELING:

We used 3 predictive models to forecast stock price: Facebook Prophet, LSTM, and Monte Carlo simulations.

#1 Facebook Prophet

Facebook Prophet is a new tool developed by the data science team at Facebook to make accurate predictions based on their own algorithm. Time series analysis is an approach to analyze time series data to extract meaningful characteristics of data and generate other useful insights applied in business situations. Generally, time-series data is a sequence of observations stored in time order.

To predict the movement of stock prices we used FB Prophet, a procedure for forecasting time series data based on an additive model where nonlinear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. The Prophet uses a decomposable time series model with three main model components: trend, seasonality,and holidays. They are combined in the following equation:

$$y(t)= g(t) + s(t) + h(t) + \varepsilon t$$

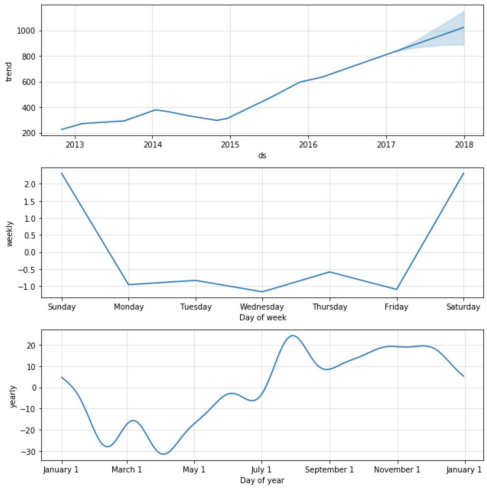
g(t): piecewise linear or logistic growth curve for modeling non-periodic changes in time series
s(t): periodic changes (e.g. weekly/yearly seasonality)
h(t): effects of holidays (user provided) with irregular schedules
εt: error term accounts for any unusual changes not accommodated by the model

Below are the graphs for the stock price movement for Amazon, Netflix & Google. Also below are the metric to check the prediction accuracy.
Note: We have considered data from 2012 till 2016 and predicted the stock price movement for 2017. Below are the results.

Amazon:

Figure #8: Facebook Prophet Amazon Stock Price Forecast

Figure #9: Daily, Monthly, and Yearly Stock Predictions



	R squared value	Mean squared error	Mean absolute error
Amazon	97.6%	1667.2	28.7

Netflix:

Figure #10: Facebook Prophet Netflix Prediction

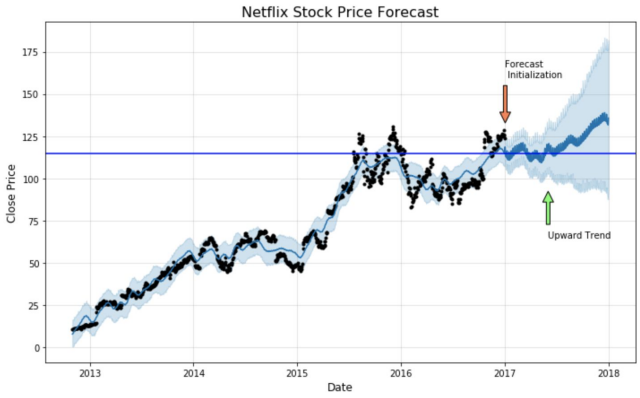
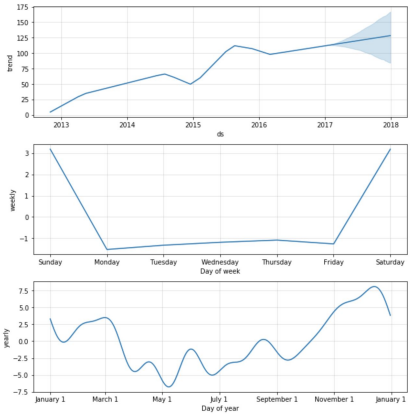


Figure #11: Daily, Monthly, Weekly Predictions for NFLX

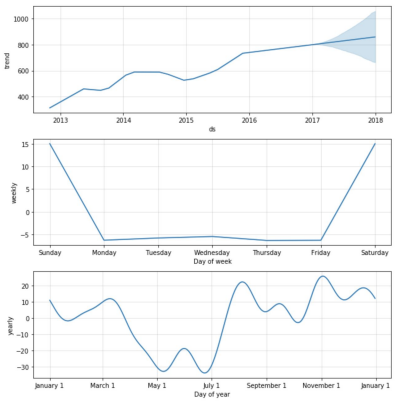


	R squared value	Mean squared error	Mean absolute error
Netflix	78.8%	494.7	12.9

Google:

Figure #12: Facebook Prophet Google Prediction

Figure #13: Daily, Monthly, Weekly Predictions for GOOGL



	R squared value	Mean squared error	Mean absolute error
Google	89.7%	3547.8	34.49

#2 LSTM

Long-Short-Term Memory(LSTM) models are a type of Recurrent Neural Networks(RNNs) which has the ability to learn and remember over long sequences of input data through the use of “gates” which regulate the information flow of the network.

LSTMs address the following limitations of RNNs.

- Short-term memory — Discarding information from earlier time steps when moving to later ones, which result in the loss of important information.
- Vanishing gradient — The gradient is the value used to update the weight used in a neural network. If a gradient value becomes extremely small, it doesn’t contribute too much to learn. In the vanishing gradient problem, gradient shrinks as it back propagates through time.
- Exploding gradient — This occurs when the network assigns unreasonably high importance to the weights.

Algorithms

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Forget Gate f_t

The remember vector is usually called the forget gate. The output of the forget gate tells the cell state which information to forget by multiplying 0 to a position in the matrix. If the output of the forget gate is 1, the information is kept in the cell state. From

equation, sigmoid function is applied to the weighted input/observation and previous hidden state.

Input Gate i_t

The save vector is usually called the input gate. These gates determine which information should enter the cell state / long-term memory. The important parts are the activation functions for each gates. The input gate is a sigmoid function and have a range of [0,1]. Because the equation of the cell state is a summation between the previous cell state, sigmoid function alone will only add memory and not be able to remove/forget memory.

Output Gate o_t

The focus vector is usually called the output gate. This is the place to determine out of all the possible values from the matrix, which should be moving forward to the next hidden state.

In our study, we applied this LSTM model to predict the stock price of Amazon, Netflix and Google. The results are shown as follows(We trained the model with the closing price data from 2012 to 2016, and then the closing price prediction started from 2017. The blue line and the orange line indicate the real closing price, and the red line shows the prediction of the closing price).

Amazon:

Figure #13: AMZN Stock Prediction Using LSTM

	R squared value	Mean squared error	Mean absolute error
Amazon	96.0%	1791.5	30.1

Netflix:

Figure #14: NFLX Stock Prediction Using LSTM

	R squared value	Mean squared error	Mean absolute error
Netflix	94.4%	133.6	7.0

Google:

Figure #15: GOOGL Stock Prediction Using LSTM

	R squared value	Mean squared error	Mean absolute error
Google	92.5%	710.5	20.5

#3 Monte Carlo

Monte Carlo simulations are used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty in prediction and forecasting models.

In this project, we applied basic MCS in R to a stock price using one of the most common models in finance: Geometric Brownian Motion (GBM). The stock price follows a random walk and is consistent with (at the very least) the weak form of the efficient market hypothesis (EMH)—past price information is already incorporated, and the next price movement is "conditionally independent" of past price movements.

The formula for GBM is found below:

$$\frac{\Delta S}{S} = \mu \Delta t + \sigma \epsilon \sqrt{\Delta t}$$

where:

$S = \text{the stock price}$

$\Delta S = \text{the change in stock price}$

$\mu = \text{the expected return}$

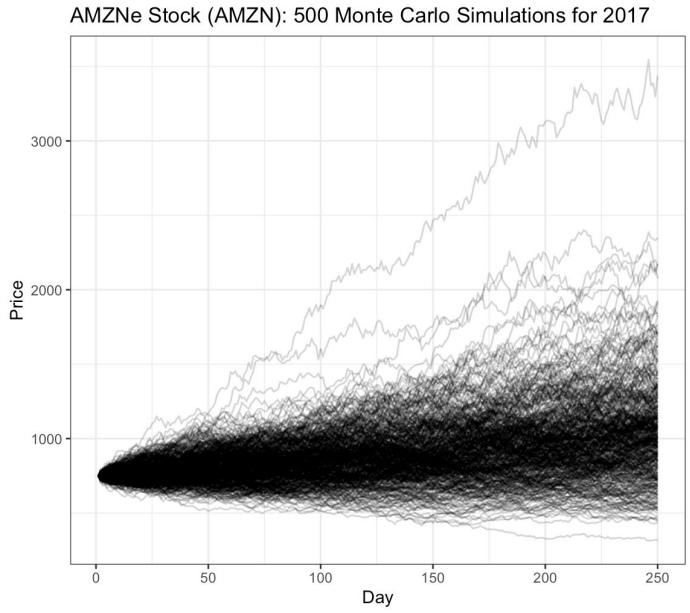
$\sigma = \text{the standard deviation of returns}$

$\epsilon = \text{the random variable}$

$\Delta t = \text{the elapsed time period}$

Amazon:

Figure #16: Monte Carlo Simulation for AMZN Stock



(500 times simulation model)

Figure #17: Confidence interval for 2017

```
> final_mat[250*1,-1]%>%as.numeric()%>%quantile(probs=probs)
     0.5%      2.5%       25%       50%       75%     97.5%     99.5%
 477.1849  533.5970  814.6345  996.8736 1235.8487 1895.2796 2101.3383
```

In the real stock data for 2017 Amazon, the closing price of last day was $1169, which was in the 95% confidence level ($533.5970-$1895.28)

Google

Figure #18: 500 Google Monte Carlo Simulations

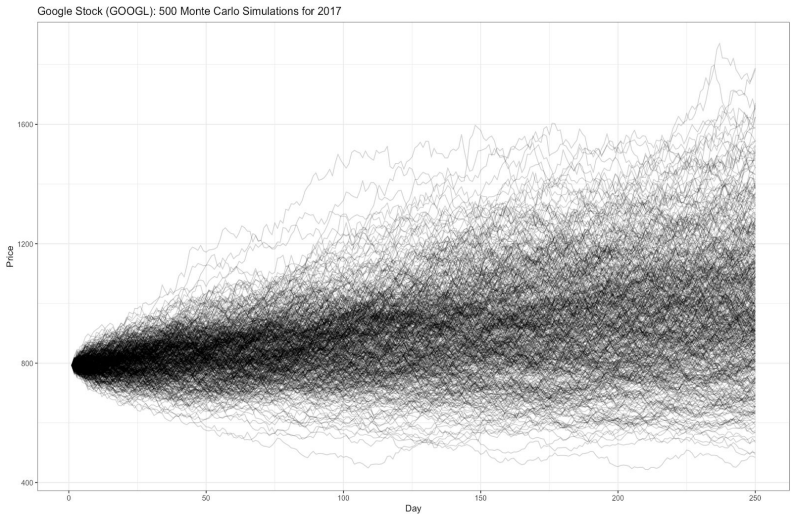


Figure #19: Confidence Interval for Google 2017

```
> final_mat[250*1,-1]%>%as.numeric()%>%quantile(probs=probs)
     0.5%      2.5%       25%       50%       75%     97.5%     99.5%
 517.6913  608.5756  832.0491  973.0922 1137.8398 1536.6943 1667.7100
```

In the real stock data for 2017 Netflix, the closing price of last day was $191.96, which was in the 95% confidence level ($84.26-$573.75)

Netflix

Figure #20: 500 NFLX Monte Carlo Simulations

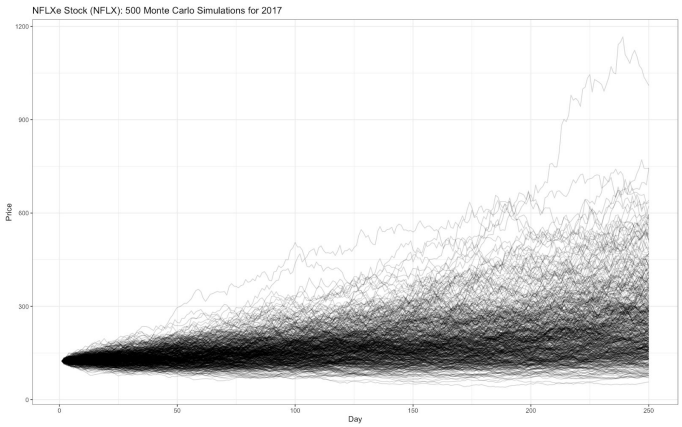


Figure #21: NFLX CI For 2017

```
> final_mat[250*1,-1]%>%as.numeric()%>%quantile(probs=probs)
     0.5%      2.5%       25%       50%       75%     97.5%     99.5%
 73.05754  84.26232 157.70549 215.64492 299.17454 573.72529 693.00043
```

In the real stock data for 2017 Netflix, the closing price of last day was $191.96, which was in the 95% confidence level ($84.26-$573.75)

The frequencies of different outcomes generated by Monte Carlo simulation will form a normal distribution, that is, a bell curve. The most likely return is at the middle of the curve, meaning there is an equal chance that the actual return will be higher or lower than that value. Still, there is no guarantee that the most expected outcome will occur, or that actual movements will not exceed the wildest projections.

INSIGHTS & CONCLUSION:

We learned that we could answer our 2 starting questions sufficiently:

1) Can we analyze what events drive revenue for large technology company stocks?
The answer to this question is that we found many significant correlations between news events happening and changes in stock price the following day. We know what some of those events are. For instance, after major earnings reports where the company exceeds earnings targets the stock price can go up. The stock price is sensitive to strong positive and negative news related to the company. Also, major changes to the company's core products can also change stock price. The market reacts to product price increases decreases as well as competitor moves.

2) Can we find a good predictive model to predict future stock prices with a high amount of accuracy?
We were able to build 2 highly accurate predictive models with Facebook Prophet and LSTM. Both models worked past 90% accuracy. We were able to test our findings with the actual data and were able to achieve high accuracy. We believe there is potential

individuals and companies to use predictive models for the purpose of making a good investment with lower risk than traditional methods.

Bibliography

Harper, D. R. (2019, October 28). How to use Monte Carlo simulation with GBM. Retrieved from https://www.investopedia.com/articles/07/montecarlo.asp

Kenton, W. (2019, January 10). Monte Carlo Simulation Definition. Retrieved from https://www.investopedia.com/terms/m/montecarlosimulation.asp

Kiat, Kang Choon. “Financial Analytics - Exploratory Data Analysis of Stock Data.” *Medium*, Towards Data Science, 3 July 2019, towardsdatascience.com/financial-analytics-exploratory-data-analysis-of-stock-data-d98cbadf98b9.

“New Class of Shares Stock Dividend Google Spin Off April 2014.” *TIMETOTRADE*, wiki.timetotrade.com/New_Class_of_Shares_Stock_Dividend_Google_Spin_Off_April_2014.

Stephens-Davidowitz, Seth. *Everybody Lies: Big Data, New Data, and What the Internet Can Tell Us about Who We Really Are*. HarperColllins, 2017.