Capstone Project 2

Twitter Sentiment Analysis and Stock Prediction

Introduction

Problem Statement: Analyze stock market movements using Twitter sentiment analysis to find the correlation between “public sentiment” and “market sentiment”

The price stocks trade at seem to be determined more by the human perception of the stock. Behavioral economics states that the emotions and moods of individuals affect their decision making process. Twitter can be used to gauge the public sentiment and possibly predict stock price movements. This study will focus on 4 individual stocks: Netflix ($NFLX), Disney ($DIS), Amazon ($AMZN), and Google ($GOOGL).

Data Collection

Stored all my data into a SQLite Database

Twitter Data

Used Twitter API Standard Search to collect tweets from 5-15-19 to 6-26-19. The following are the queries used:

- Netflix: '@Netflix OR $NFLX OR Netflix'
- Disney: ‘@Disney OR @ESPN OR @ABCnetwork OR @Pixar OR @Marvel OR $DIS’
- Amazon: ‘@Amazon OR @PrimeVideo OR @awscloud OR @TwitchPrime OR @Alexa OR @WholeFoods OR $AMZN’
- Google: ‘@Google OR @Android OR @Waymo OR $GOOGL’

Tweets Dataframe
created_at
tweet
follower_count
pos_sent
neu_sent
neg_sent
compound_sent
sentiment
Company

Twitter Sentiment Analysis

1. Clean the tweet so that it processes more accurately
   a. Regular Expression library use to locate text strings and remove them
   b. String to be removed:
      i. User mention
      ii. Hyperlinks
      iii. Hashtag sign
      iv. “RT”

```
'RT @Google: Toy Story is back. See the latest Toy Story 4 trailer #WithALittleHelp from Google → https://t.co/np6XbygVvi ht
tps://t.co/Hnpmy…'
```

```
'  Toy Story is back See the latest Toy Story 4 trailer WithALittleHelp from Google   '
```

2. VADER (Valence Aware Dictionary and sEntiment Reasoner) as sentiment tool
   a. Lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media.
   b. Provides positive, negative, and neutral score. Then computes a compound score which sums the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive).
      i. Compound Score > 0.05 = Positive Tweet
      ii. Compound Score < -0.05 = Negative Tweet
      iii. -0.05 < Compound Score < 0.05 = Neutral Tweet

Data Collection

<u>Stock Data</u>

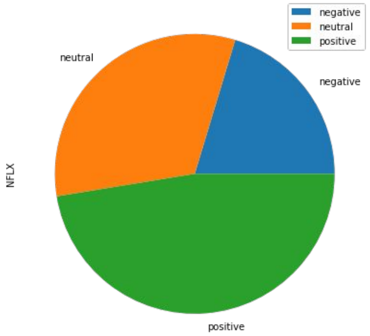
Used Alpha Vantage API to retrieve stock data for all 4 stocks from the date range 5-15-19 to 6-26-19. The data is missing weekends and holidays since the market is not open on those days.

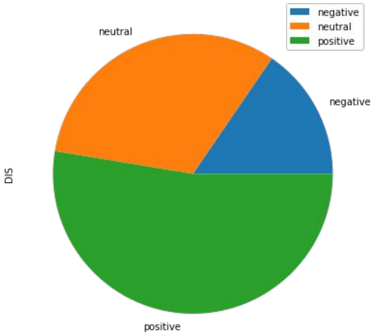
However, to see the trend in the time series, I opted to have continuous data. The information for the weekends and holidays were filled in using forward-filling linear interpolation. It estimates a new value by connecting two adjacent known values with a straight line.

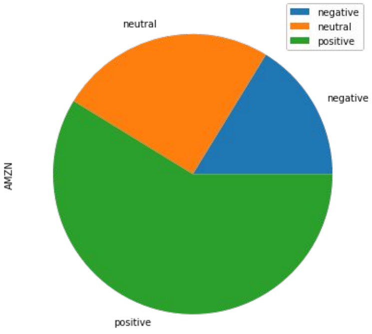
Stock Dataframe
date
open
high
low
close
volume
company

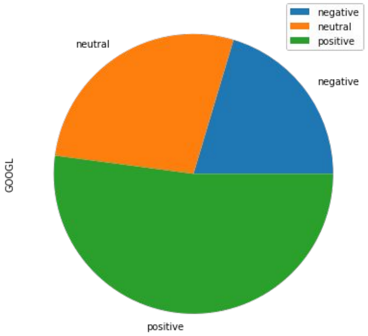
Exploring Twitter Data

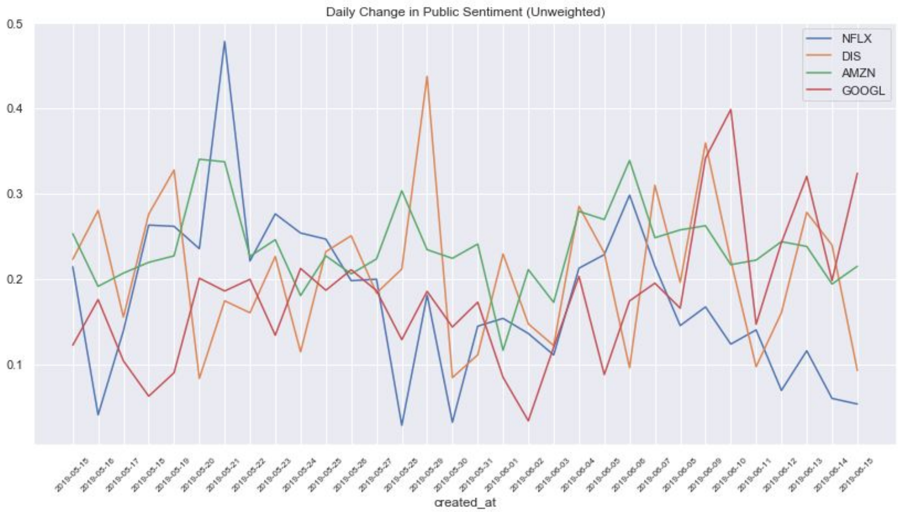
Company	Total Tweets	Avg Daily Tweets	Average Compound Sent
Netflix	64,055	2,002	0.18
Disney	78,404	2,450	0.21
Amazon	79,707	2,491	0.24
Google	89,728	2,804	0.18

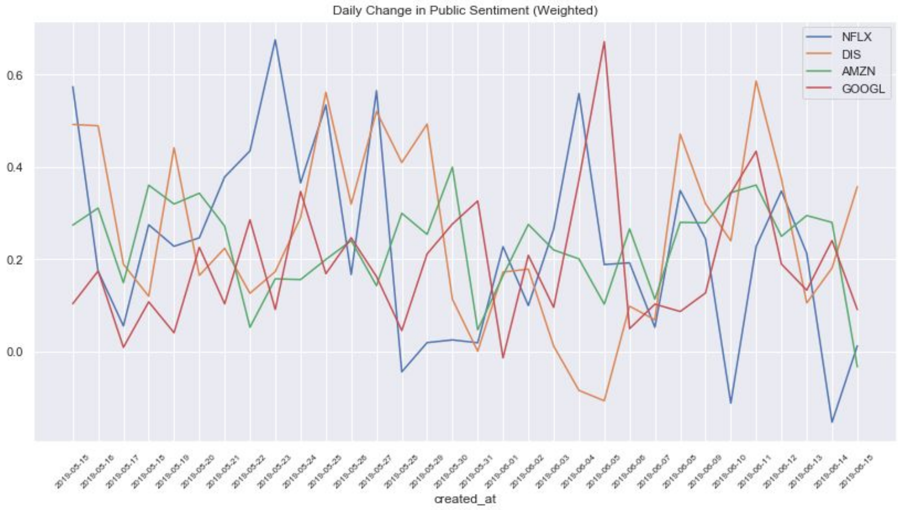












Daily Change in Public Sentiment

Twitter accounts that have more followers have more influence on the community since their tweet reaches more people. To incorporate this, we have given weight to each tweet according to proportion of followers they have compared to the total amount of followers for each day.

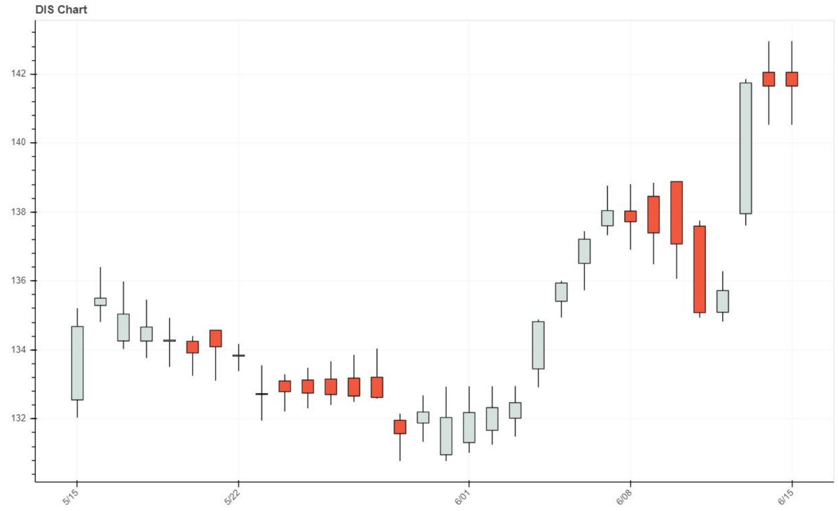
Exploring Stock Data

Netflix

- Average Close Price: $350.97
- Average Daily Volume: 5,380,332

Disney

- Average Close Price: $134.97
- Average Daily Volume: 7,934,529



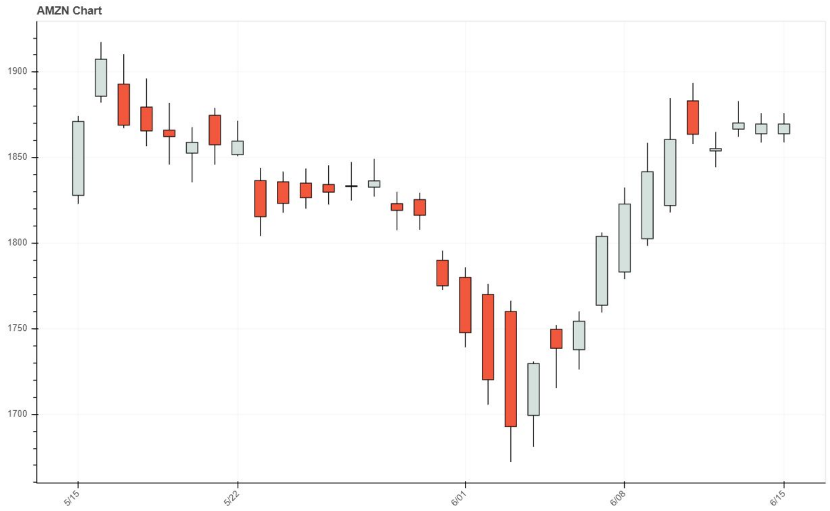
Exploring Stock Data

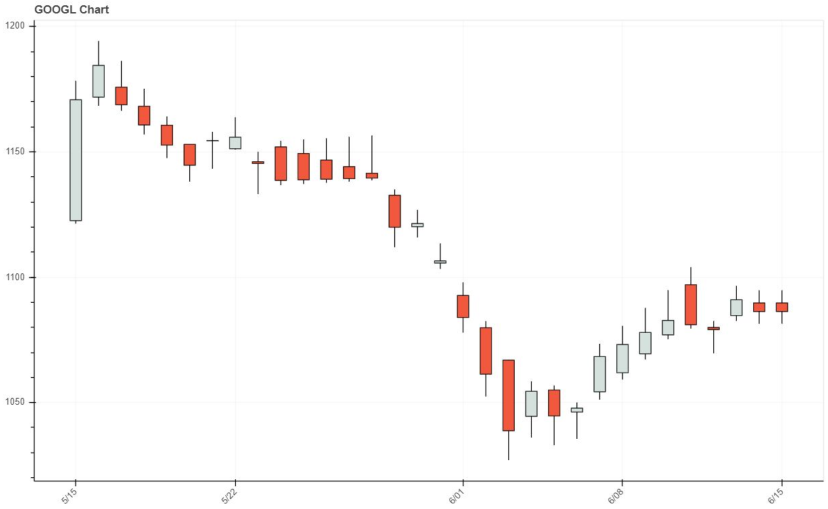
Amazon

- Average Close Price: $1,824.00
- Average Daily Volume: 4,322,092

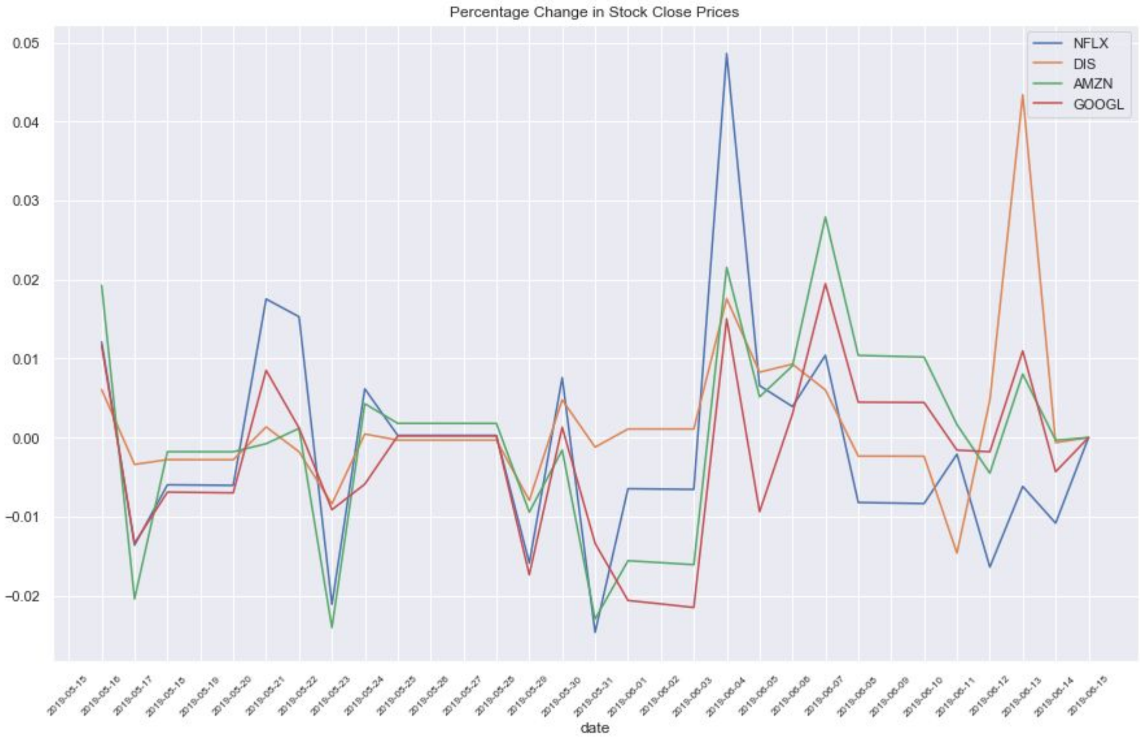
Google

- Average Close Price: $1,110.56
- Average Daily Volume: 1,702,416



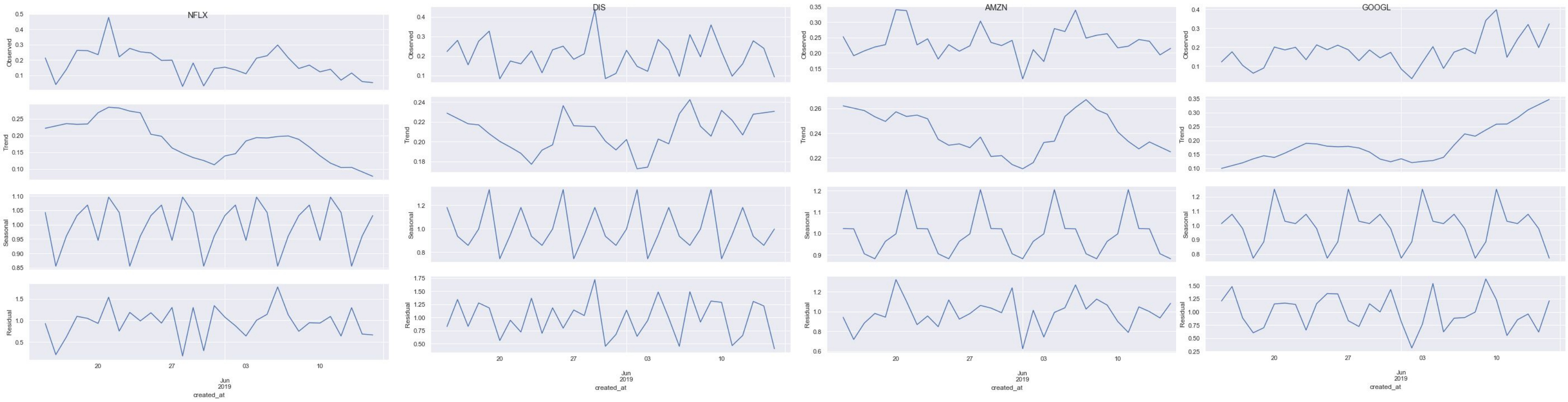


Daily Percentage Change in Stock Close Prices



Netflix and Disney are sold at way lower prices than Amazon and Google. It would be hard to see the trend in all the stock prices on one graph. Therefore, the prices must be standardized so that price movements would be comparable. I have computed the percentage change in price via difference in natural log of stock close prices between the current price and the previous price.

Time Series Analysis: Seasonal Decomposition



Multiplicative Time Series: **Value = Base Level x Trend x Seasonality x Error**

After decomposing the seasonality from the time series, it is apparent that the public sentiment for Netflix has decreased over time. Google, on the other, has an upward trend in public sentiment. The other two have a more balanced trendline.

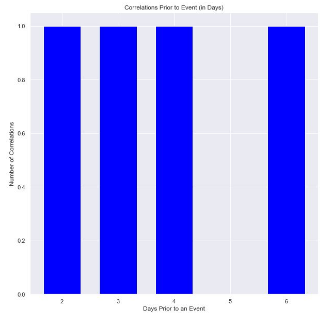
Time Series Analysis: Granger Causality

This tests whether the time series in the second column Granger causes the time series in the first column.

H0: The public sentiment for the company does NOT Granger cause the movement in stock price for that company.
H1: The public sentiment for the company Granger causes the movement in stock price for that company.

```
============================================================

No Causality

NFLX closing stock price and NFLX twitter sentiment showed NO causality, count: 31
DIS closing stock price and DIS twitter sentiment showed NO causality, count: 31
AMZN closing stock price and AMZN twitter sentiment showed NO causality, count: 31

--------------------------------------------------------

Causality

GOOGL closing stock price and GOOGL twitter sentiment showed causality, count: 31

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Causality Count: 1
No Causality Count: 3

std 1.479019945774904
avg 3.75
Percent showing causality: 0.25
```

The only Google's public sentiment and market sentiment time series rejected the null. There was a correlation shown between the public sentiment and market sentiment for 2,3,4 and 6 days prior to the stock price movements.



Machine Learning: Feature Engineering

Twitter Data Set

Original Dataframe

created_at	tweet	follower_count	neg_sent	neu_sent	pos_sent	compound_sent	sentiment	Company
2019-05-15	RT @MileyCyrus: Black Mirror Out June 5th @net...	537	0.000	1.000	0.000	0.0000	neutral	NFLX
2019-05-15	RT @LaurenGerman: Let's all give a HUGE ROARIN...	2	0.000	0.733	0.267	0.7027	positive	NFLX
2019-05-15	@netflix This shit got me weak💀😂 talkin thru t...	259	0.535	0.465	0.000	-0.8481	negative	NFLX
2019-05-15	RT @MileyCyrus: Black Mirror Out June 5th @net...	762	0.000	1.000	0.000	0.0000	neutral	NFLX
2019-05-15	RT @LaurenGerman: Let's all give a HUGE ROARIN...	6	0.000	0.733	0.267	0.7027	positive	NFLX

Each row in the twitter data set represents one tweet. As mentioned earlier, each company approximately received 2,500 tweets per day and the training set covers the span of 31 days. Therefore, the twitter data must be condensed into daily summary statistics.

- Weighted average compound sentiment score (weight = number of followers for the tweet / total number of followers for the day
- Proportion of tweets classified as negative, positive, and neutral. To prevent multicollinearity, the proportion of tweets classified as neutral has been left out.

Resulting Dataframe

date	w_avg_sent	percent_neg	percent_pos
2019-05-15	0.573347	0.083511	0.454787
2019-05-16	0.173842	0.409941	0.400855
2019-05-17	0.055269	0.231081	0.442793
2019-05-18	0.274295	0.152960	0.565753
2019-05-19	0.227649	0.173720	0.584442

Machine Learning: Feature Engineering

Stock Data Set

Original Dataframe

date	open	high	low	close	volume	Company
2019-05-15	343.34	356.500000	341.390	354.990000	6.340118e+06	NFLX
2019-05-16	356.37	364.000000	353.935	359.310000	6.441463e+06	NFLX
2019-05-17	356.39	359.620000	353.785	354.450000	4.725448e+06	NFLX
2019-05-18	354.67	357.219333	350.990	352.336667	4.690804e+06	NFLX
2019-05-19	352.95	354.818667	348.195	350.223333	4.656159e+06	NFLX

To perform time series forecasting, time lags must be introduced into the model. The stock data from the previous day will be used to predict the stock growth of the current day.

Fundamental Analysis

- Percentage change in S&P index price

Standardizing Variables

- Volume
- Percentage change in stock growth instead of close price

Resulting Dataframe

date	volume_l1	stock_growth	stock_growth_l1	sp_growth
2019-05-15	NaN	NaN	NaN	NaN
2019-05-16	0.791431	0.012169	NaN	NaN
2019-05-17	0.884759	-0.013526	0.012169	0.008895
2019-05-18	-0.695522	-0.005962	-0.013526	-0.005837
2019-05-19	-0.727426	-0.005998	-0.005962	-0.002250

Machine Learning: Building Predictive Model

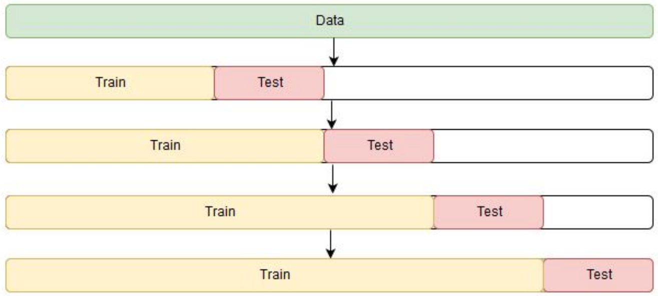
Final Dataset: Merging Twitter Data and Stock Data

Target Variable	Predictor Variables
• *Stock Percentage Change*	• *Stock Percentage Change (1 day lag)* • *Standardized Volume (1 day lag)* • *S&P 500 Percentage Change (1 day lag)* • *Weighted Compound Score (2 day lag)* • *Proportion of Negative Tweet (2 day lag)* • *Proportion of Positive Tweet (2 day lag)*

Training Data:
05/15/2019 - 06/15/2019 → 30 rows

Testing Data:
06/16/2019 - 06/26/2019 → 10 rows

Algorithm: Regression
Cross Validation: TimeSeriesSplit Method (5-fold)

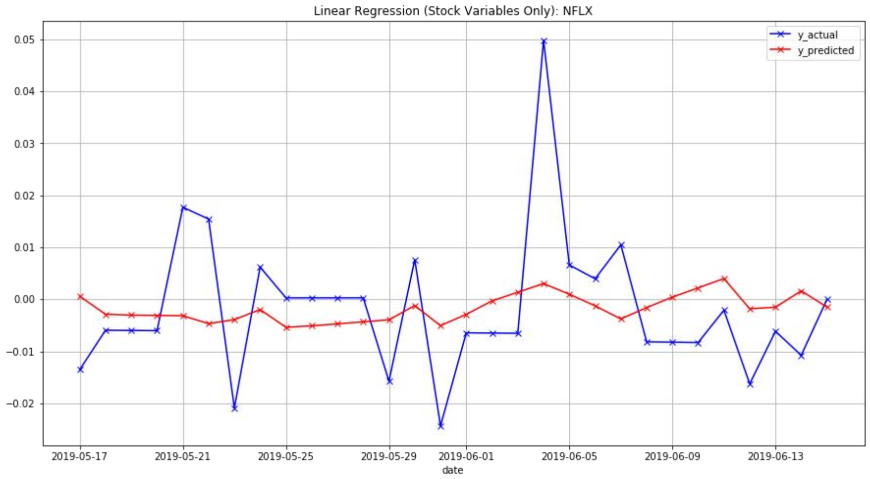


Machine Learning: Building Predictive Model

Linear Regression Stock Variables Only (finds the best fit line with the smallest prediction error throughout)

Predictor Variables: Stock Growth (1 day lag), Standardized Volume (1 day lag), S&P 500 Percentage Change (1 day lag)

Target Variable: Stock Growth

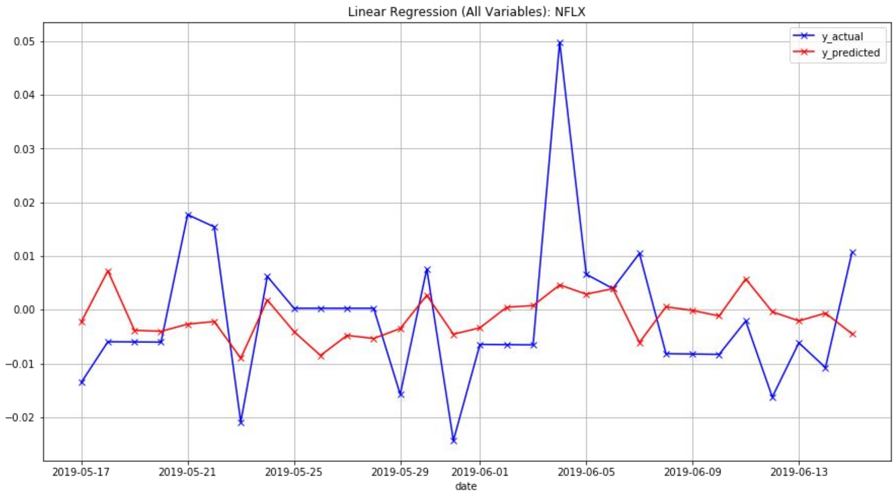
NFLX	• RMSE: 0.258 • MAPE: 54.94 % • Direction: 44 %	

Machine Learning: Building Predictive Model

Linear Regression All Variables

Predictor Variables: Stock Percentage Change (1 day lag), Standardized Volume (1 day lag), S&P 500 Percentage Change (1 day lag), Weighted Compound Score (2 day lag), Proportion of Negative Tweet (2 day lag), Proportion of Positive Tweet (2 day lag)

Target Variable: Stock Growth

NFLX	• RMSE:0.040 • MAPE: 32.93 % • Direction: 32.0 %	

Machine Learning: Building Predictive Model

The model performed fairly well on the test data. Growth in S&P 500 for the previous day seemed to have the highest positive effect while stock growth for the previous day seemed to have the most negative effect on stock growth. All the sentiment related variables have a positive effect on stock growth. Interestingly, the percentage of tweets classified as negative had the highest positive effect out of the 3 sentiment variables.

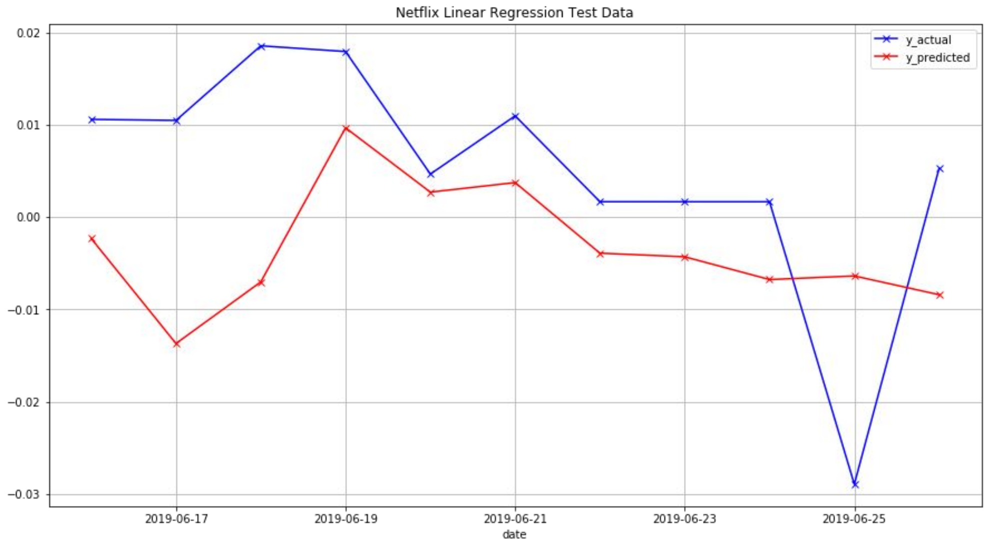
NFLX

Metrics

- RMSE: 0.014
- MAPE: 1.096 %
- Direction: 18.18 %

Coefficients

- sp_growth_l1: 0.716
- percent_neg: 0.063
- percent_pos: 0.025
- w_avg_sent: 0.008
- volume_l1: 0.003
- stock_growth_l1: -0.305

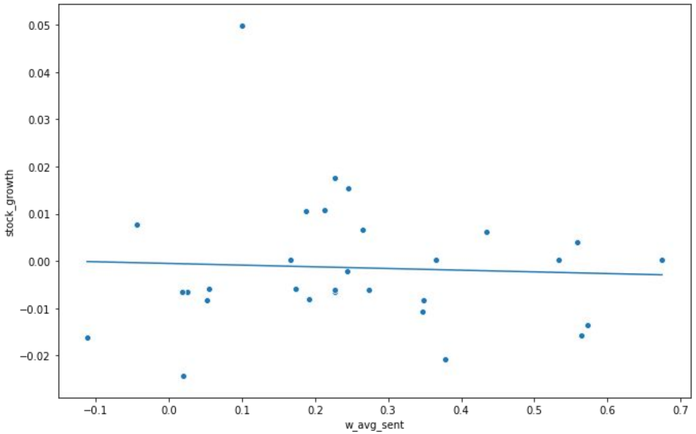


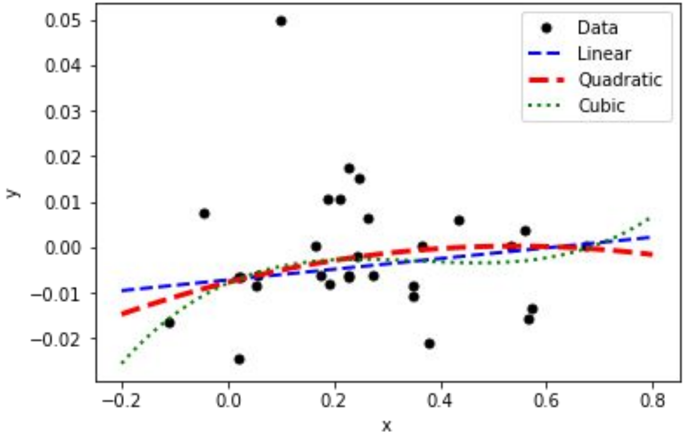
Machine Learning: Building Predictive Model

Polynomial Regression

Linear regression captures the patterns in the data better when the relation between the dependent variable and the independent variable is linear. However, the relationship between the target variable and predictor variables is not linear.

To illustrate this, I have graphed the relationship between stock growth and average sentiment value. There is more variance towards the lower values in sentiment value than higher values. The cubic function seems to capture the relationship better.



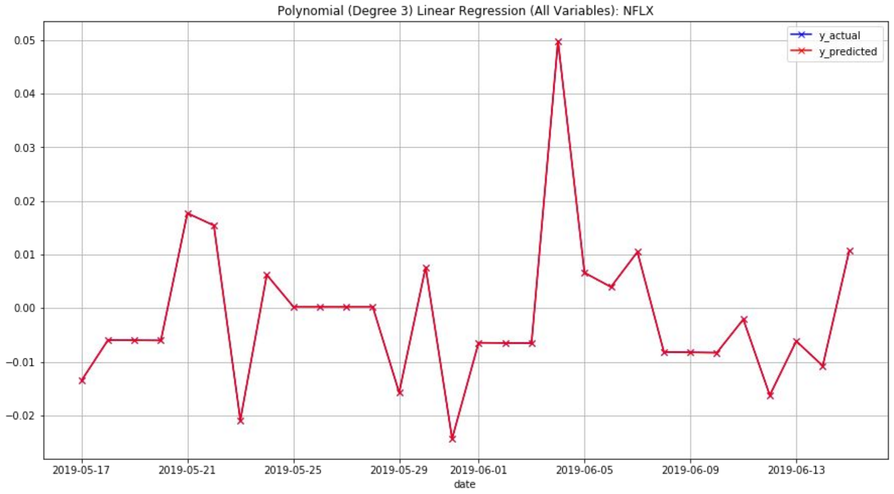


Machine Learning: Building Predictive Model

Polynomial Linear Regression (Degree of 3) All Variables

Predictor Variables: Stock Percentage Change (1 day lag), Standardized Volume (1 day lag), S&P 500 Percentage Change (1 day lag), Weighted Compound Score (2 day lag), Proportion of Negative Tweet (2 day lag), Proportion of Positive Tweet (2 day lag) + Polynomial of all variables + Interaction between all variables

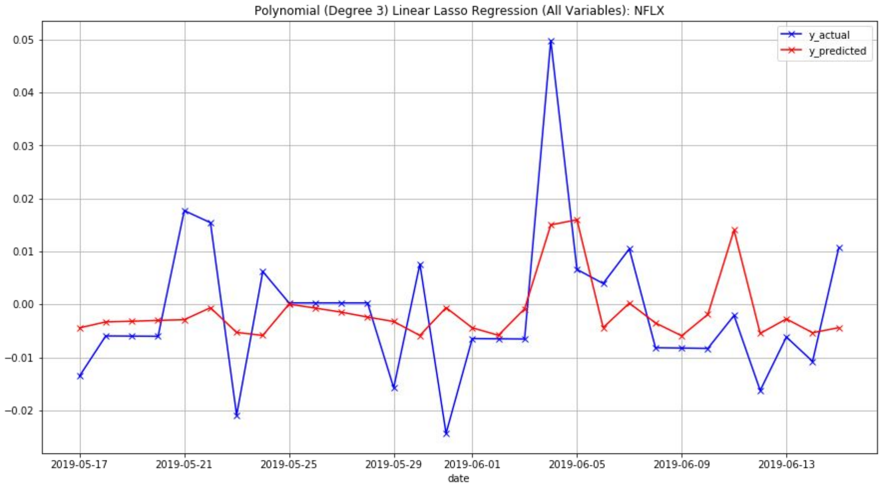
Target Variable: Stock Growth

NFLX	• RMSE: 0.11 • MAPE: 79.8 % • Direction: 52.0 % OVERFITTING!!	

Machine Learning: Building Predictive Model

Polynomial Lasso Regression (Degree of 3) All Variables (“least absolute shrinkage and selection operator”)

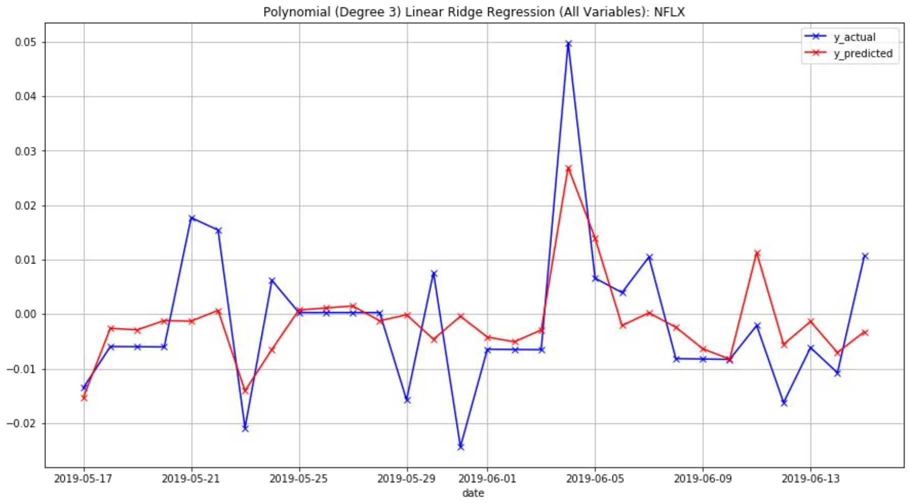
Shrinks the coefficients and it helps to reduce the model complexity and multi-collinearity. It adds a penalty (absolute value of the magnitude of the coefficients) to the cost function (sum of squared prediction error). It can lead to zero coefficients. Therefore, Lasso regression not only helps in reducing overfitting, but it also helps in **feature selection**.

NFLX	• RMSE: 0.015 • MAPE: 2.61 % • Direction: 64.0 % UNDERFITTING!!	

Machine Learning: Building Predictive Model

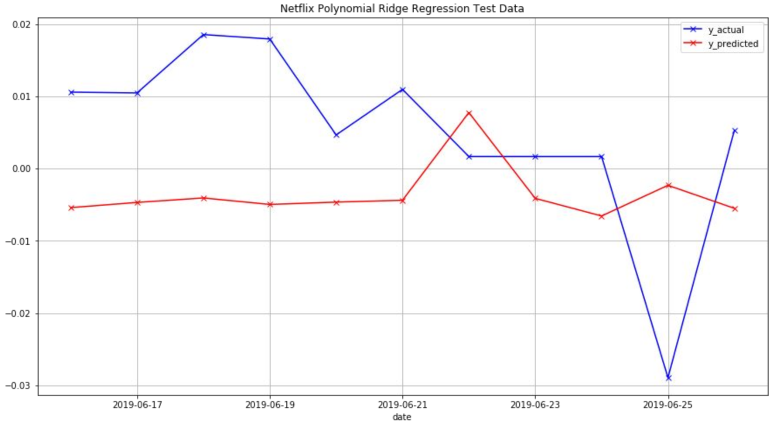
Polynomial Ridge Regression (Degree of 3) All Variables

Shrinks the coefficients and it helps to reduce the model complexity and multi-collinearity. It adds a penalty (square of the magnitude of the coefficients) to the cost function (sum of squared prediction error). The coefficients in Ridge regression converge towards 0, but never 0. Ridge regression only helps in reducing overfitting.

NFLX	• RMSE: 0.021 • MAPE: 5.5 % • Direction: 60.0 %	

Machine Learning: Building Predictive Model

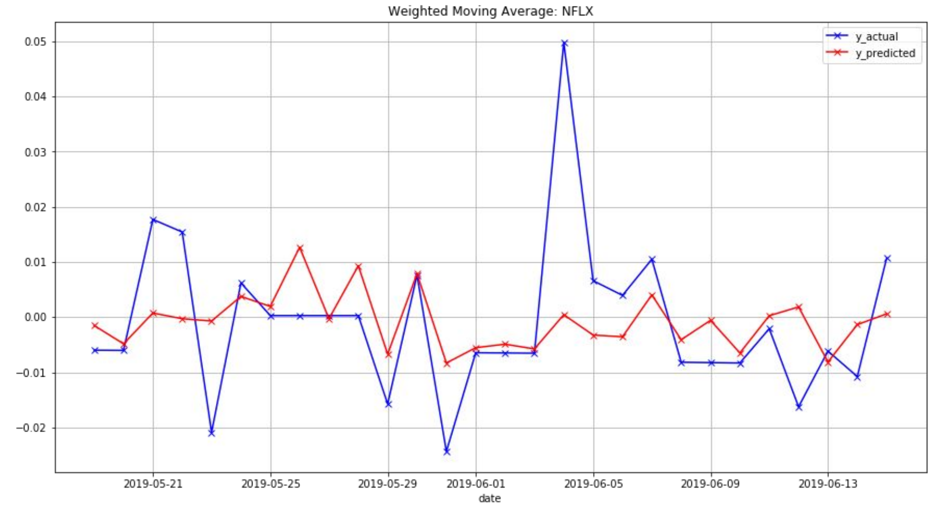
Out of 3 models, polynomial regression with lasso performed the best in terms of all 3 metrics. It seems to perform better in predicting the stock growth values when it is close to 0. The model does not capture large fluctuations well. This is most likely due to the shrinkage of the coefficients which results in a model that loosely follows the trend. However, it is more important that we capture the fluctuations. Ridge regression seems to be a good middle ground between overfitting and underfitting. This model performs similarly to the linear regression without the polynomial regression when applied to the test data. The RMSE is only a little worse (0.001 difference) and the direction accuracy remained the same. Both models did not anticipate the drop in growth on 6/25/2019.

NFLX	• RMSE: 0.015 • MAPE: 2.19 % • Direction: 18.18 %	

Machine Learning: Building Predictive Model

Weighted Moving Average (Stock Growth and Compound Sentiment Value)

I run a linear regression on historical stock growth and sentiment value data. The model determines the value of the weights given to each lag in the variables. For example, if I am running the model with a 3 day lag, the model will be trained on the stock growth and the sentiment value for the past 3 days. I tested the performance of models trained on historical data ranging from the past 2 days to the past 6 days. 3 day lag did well in terms of both RMSE and direction.

<table>
<tr>
<td>NFLX</td>
<td>• RMSE: 0.018
• MAPE: 9.91 %
• Direction: 50 %</td>
<td>
</td>
</tr>
</table>

Results

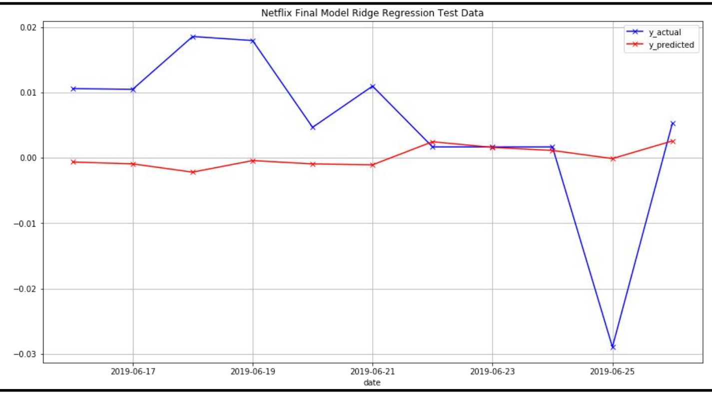
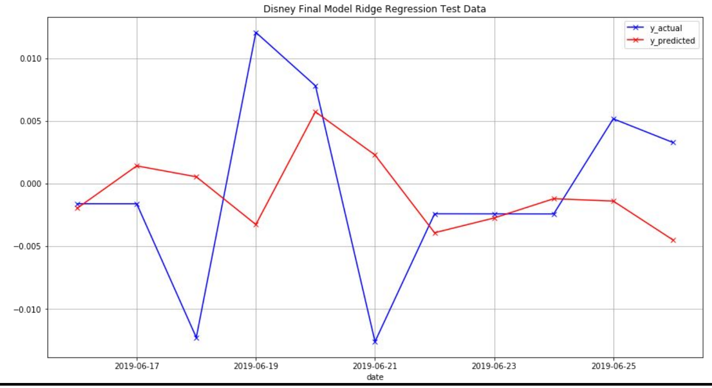
Company	Algorithm	RMSE	MAPE	Direction
AMZN	Polynomial Ridge Regression (Degree of 3)	0.012996	3.865511	36.0
AMZN	Linear Regression (Only Stock)	0.014983	3.236855	36.0
AMZN	Polynomial Lasso Regression (Degree of 3, Alph...	0.016009	2.809316	32.0
AMZN	Linear Regression (All Variables)	0.016407	3.628019	36.0
AMZN	Weighted Moving Average	0.023974	6.787748	55.0
AMZN	Polynomial Regression (Degree of 3)	1.228785	66.054658	44.0
DIS	Polynomial Lasso Regression (Degree of 3, Alph...	0.008717	2.145738	48.0
DIS	Polynomial Ridge Regression (Degree of 3)	0.009714	3.906355	48.0
DIS	Linear Regression (Only Stock)	0.013188	9.748320	44.0
DIS	Weighted Moving Average	0.013329	3.962521	25.0
DIS	Linear Regression (All Variables)	0.015153	11.118734	36.0
DIS	Polynomial Regression (Degree of 3)	0.236995	202.881103	40.0
GOOGL	Polynomial Ridge Regression (Degree of 3)	0.010487	4.297911	44.0
GOOGL	Polynomial Lasso Regression (Degree of 3, Alph...	0.010577	4.680170	44.0
GOOGL	Weighted Moving Average	0.018437	15.827871	30.0
GOOGL	Linear Regression (All Variables)	0.023313	10.140069	44.0
GOOGL	Linear Regression (Only Stock)	0.025380	30.702864	28.0
GOOGL	Polynomial Regression (Degree of 3)	2.895230	164.115478	36.0
NFLX	Polynomial Lasso Regression (Degree of 3, Alph...	0.015140	2.619799	64.0
NFLX	Weighted Moving Average	0.017684	9.912616	50.0
NFLX	Polynomial Ridge Regression (Degree of 3)	0.021401	5.495615	60.0
NFLX	Linear Regression (All Variables)	0.040265	32.931784	32.0
NFLX	Polynomial Regression (Degree of 3)	0.106495	79.800306	52.0
NFLX	Linear Regression (Only Stock)	0.258143	54.941599	44.0

The model had improved given the public sentiment information as opposed to only stock variables for all the companies. Overall, it seems as if the polynomial regression with some sort of regularization modeled the data best.

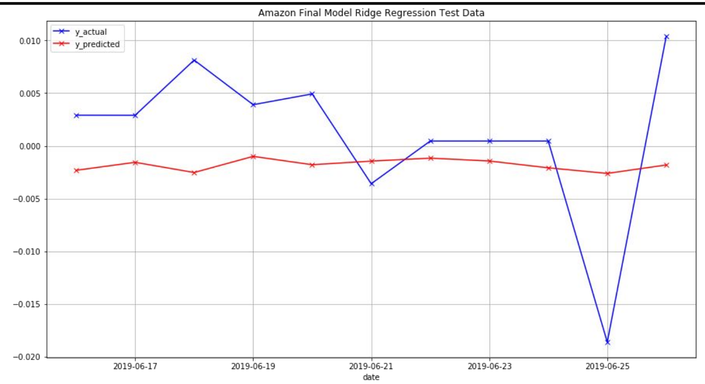
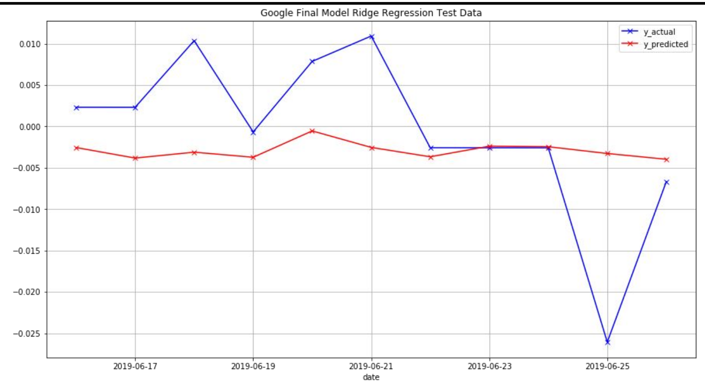
To pick the final model, I average the RMSE across the different algorithm to see which algorithm has the lowest RMSE score. Polynomial regression with lasso regularization seems to perform the best in all 3 metrics. However, as stated before, it underfits the model with it's extreme regularization and does not capture the fluctuations that investor would typically want to know. Investors do not care if the stock grows by 0.01%; they care more if the stock would grow by 5%. That's when an investor would make the most money. Therefore, the final model will be the polynomial regression with ridge regularization.

Algorithm	RMSE	MAPE	Direction
Polynomial Lasso Regression (Degree of 3, Alpha 0.001)	0.012611	3.063756	47.0
Polynomial Ridge Regression (Degree of 3)	0.013650	4.391348	47.0
Weighted Moving Average	0.018356	9.122689	40.0
Linear Regression (All Variables)	0.023785	14.454652	37.0
Linear Regression (Only Stock)	0.077923	24.657409	38.0
Polynomial Regression (Degree of 3)	1.116876	128.212886	43.0

Results

Company	Metrics	Visualization
Netflix	● RMSE: 0.014 ● MAPE: 0.844 % ● Direction: 45.45 %	
Disney	● RMSE: 0.008 ● MAPE: 0.841 % ● Direction: 45.45 %	

Results

Company	Metrics	Visualization
Amazon	● RMSE: 0.008 ● MAPE: 2.207% ● Direction: 18.18 %	
Google	● RMSE: 0.0097 ● MAPE: 1.442 % ● Direction: 54.54 %	

Limitations

Although the RMSE scores are low, the models did not capture the drastic fluctuations as I hoped it would (e.g. drop on 6/25/2019). This may be due to the following:

- Insufficient training data (only 30 rows)
- Better sentiment analyzer (VADER misclassified neutral tweets)
- More predictor variables (Unexplained variance in stock growth)
- Explore more regression algorithms

Coefficients of the weighted average model illustrated that there was a need for more training data.

The coefficients of this model are expected to diminish in magnitude towards 0 as lag from the current day increases. However, the coefficients for the models seems to fluctuate over the 3 day lag. This may mean that the number of lag days need to be increased, but there's not enough training data.

Coefficients	**Netflix**	**Disney**	**Amazon**	**Google**
Stock_growth_l1	-0.028	0.1464	0.1915	0.2215
Stock_growth_l2	-0.026	-0.1823	0.2210	0.1260
Stock_growth_l3	-0.058	0.0590	0.1622	0.0819
W_avg_sent_l1	0.007	-0.0033	0.0009	-0.0037
W_avg_sent_l2	-0.006	0.00823	-0.0145	0.0204
W_avg_sent_l3	0.026	-0.0111	-0.0339	-0.0081