Capstone Project 2 Overall Report
Submitted by Khanh Ho
4-9-2020

Project Idea

The idea for Capstone Two originates from my interest in investing in the stock market. Investing and trading is governed by supply, demand and human behavior. There are certain patterns in human behavior that occur over time. My interest was to see if there was some connection between news headlines and how it affects stock demand.

Proposal Summary

News headlines can drive a stock value up or down. Human behavior is predictable for stock as history has shown. Can we predict how much and how the stock market will change with the impact of a news headline? If a pattern can be modeled we would like to have a model that will take a news headline and predict how the price action will change on the day of the news release.

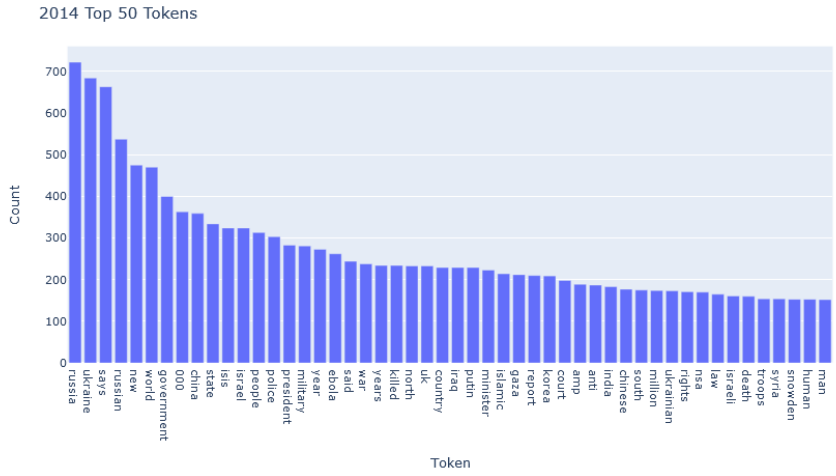
Data Wrangling

All the files were downloaded from Kaggle.com. 'RedditNews.csv', file contains the Date column and then another column named News. The News column contains the news headline for the given date. 'Combined_News_DJIA.csv' file has the Date, a Label and the top news headlines for the column headers. This is a post processed file from the author. The Label header is 0 for when the stock value decreased and 1 when the stock value rose or stayed the same. The first news column is named Top1 and is the hottest news item, there are 25 news headlines for a date. 'upload_DJIA_table.csv' file has a date and then columns titled Open, High, Low, Close, Volume and Adj Close for the Dow Jones Industrial Average(DJIA).

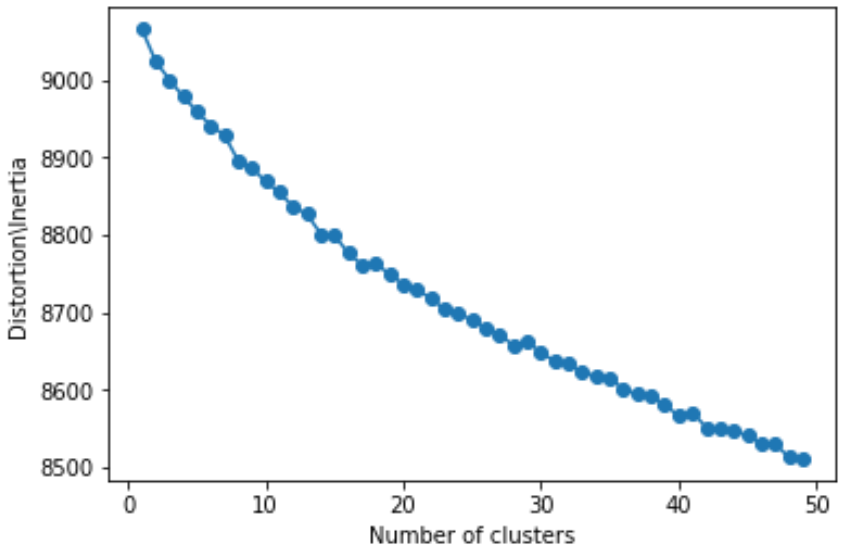
There was a lot of preprocessing of the data for the NLP. There was tokenization, stemming and lemmatization of the various headlines from the 'RedditNews.csv' file. There was the usage of Part of Speech(POS) tagging that was done on the tokens, stems and lemma.

Exploratory Data Analysis, Data Story and Milestone Report

The project initially started with a NLP analysis. The Reddit news headlines was tokenized and the top tokens were discovered for every year. Below is a graph of the top tokens for 2014.



The tokens were then lemmatized and then pushed through various NLP algorithms. One of the algorithms was the K means clustering of the 2014 news headlines. We ran the K Means clustering and graphed the number of clusters by Inertia. We incremented the cluster count until we could see a slower drop in the inertia over the number of clusters. Below is the chart and the point that was chosen was between 42-44 clusters. 44 was chosen as there are three points that are pretty horizontal.



We cluster the headlines into 44 clusters and below is an example of some of the clusters.

Cluster 0:	Cluster 2:	Cluster 4:	Cluster 6:	Cluster 8:	Cluster 10:	Cluster 12:	Cluster 14:
china	russia	snowden	isra	chines	ban	germani	0
say	sanction	nsa	palestinian	china	turkey	spi	year
year	crimea	edward	gaza	vietnam	school	nsa	100
new	say	spi	teen	ship	court	war	10
report	gas	surveil	hama	factori	youtub	solar	old
death	deal	german	kill	mh370	govern	merkel	peopl
court	europ	document	death	hacker	china	electr	30
presid	eu	leak	bank	philippin	russia	nazi	500
million	impos	say	settlement	vessel	say	energi	china
peopl	china	greenwald	netanyahu	say	twitter	say	million
Cluster 1:	**Cluster 3:**	**Cluster 5:**	**Cluster 7:**	**Cluster 9:**	**Cluster 11:**	**Cluster 13:**	**Cluster 15:**
canada	ebola	australia	protest	japan	uk	kill	russian
canadian	outbreak	reef	thousand	minist	govern	attack	ukrainian
marijuana	case	barrier	ukrain	prime	polic	milit	pro
court	liberia	great	polic	whale	surveil	afghan	crimea
legal	africa	dump	anti	say	british	bomb	separatist
govern	virus	australian	govern	hunt	use	pakistan	moscow
countri	infect	coal	venezuela	abe	year	peopl	forc
suprem	sierra	dredg	march	toni	ban	al	jet
sand	leon	abbott	student	new	say	yemen	militari
soldier	spread	toni	istanbul	resign	tortur	drone	nato

We also applied Latent Dirichlet Allocation for topic modeling to see if we can discover any trends in the topics for 2014. Both these exercises were interesting to do but they did not provide much insight unfortunately in this scenario.

After the NLP analysis we can see if we can do any Regression analysis on the text and the TFIDF specifically. The first thing to do was look at the 2014 data, splitting that into a train and test set and do a test run to gauge the length of time to do the analysis. Using logistic regression with the x variables as the TFIDFs and the y variables as the 0 or 1 stock change labels. The first line in the table below shows the regression returned an accuracy of 53%, which isn't satisfactory. We thought maybe 1 year of data wasn't enough so we went with 2 years of data so we used the years 2013-2014 and splitting the data into a train and test set. The results this time returned a max accuracy of 57%, which is not satisfactory.

Yearly Data	Regression	Accuracy
2014	Logistic	53%
2013-2014	Logistic	57%
2013-2014	MultinomialNB	57%
2013-2014	RandomForest	55%
2013-2014	SVM	56%

We tried a GridSearchCV with a Pipeline next to see if we can fine tune the models to get a higher accuracy. We again started with a small data set to test out the Pipeline and find some of the best parameters. Each finding helped us fine tune the model. A useful finding was the max features maxed out at 250 so that kind of obsoleted the K means clustering and elbow method we did earlier. We even changed the accuracy score to blank but the accuracy score did not change in the end. The final best regression was using Logistic regression at 54% which is not satisfactory.

Yearly Data	Vectorizer	Vectorizer Parameters	Regression	Regression Parameters	Accuracy	Findings
2013-2014	StemTFIDF	max_df, max_features	Logistic	C, newton-cg, lbfgs,liblinear,sag,saga	57%	best C is .001, best solver is newton-cg
2013-2014	StemTFIDF	max_df, max_features	Naïve Bayes		55%	best max_df is .25
Full Data	StemTFIDF	max_features	Naïve Bayes		53%	best max_features is 250
Full Data	StemTFIDF	max_df	Logistic		54%	best max_df is .25

After some learnings, the accuracy scores obtained are not satisfactory enough to use the current model for stock price prediction. The RedditNews site is a global news site and may not be accurate of all news that goes on in the US. The DJIA is composed of American companies and not all global news has an impact on the DJIA. Furthermore the DJIA is a composite of several companies from different business sectors. The impact of the news could affect one business sector and not another but it is not traceable in the composite mix of stocks. It might prove more insightful to look at standalone stocks from a business sector, e.g. Coca Cola and see how the news headlines and stock price changes together. With Yahoo, another stock financial data set could be downloaded and we could re-evaluate our NLP and stock price changes. It also might be more insightful to get a news source that is more American news centric. We would have to find another data set or scrape the web for this data. Another option, because the NLP learnings proving to not be helpful we may look at stock prediction from a technical analysis. This would look at average prices, moving averages, returns and other numerical features to help predict what the stock price will be.

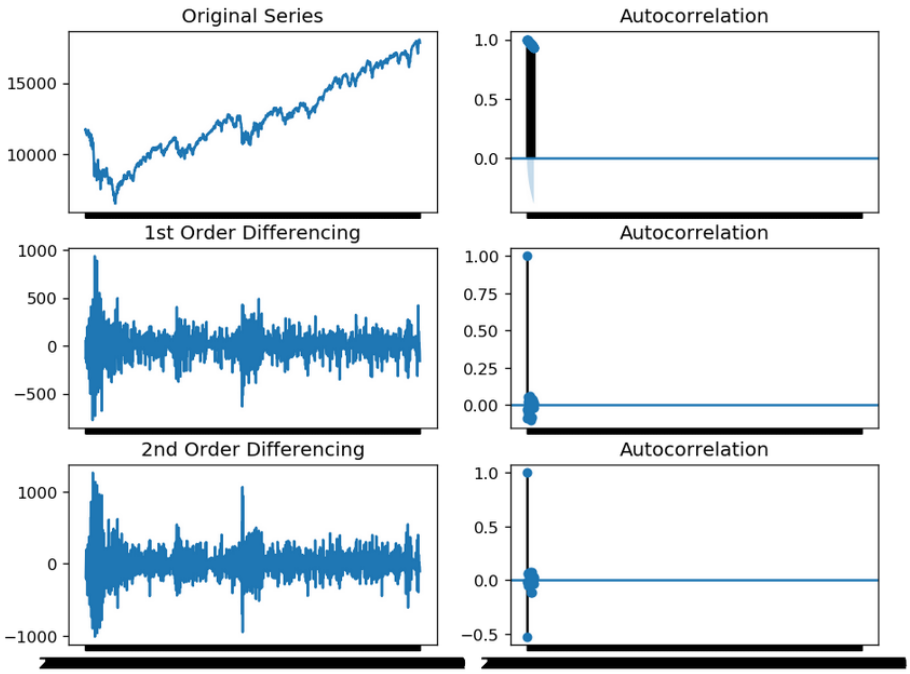
After determining Natural Language Processing of the Reddit News returned a poor prediction of the stock direction and some contemplating, I decided to go with a technical analysis of the current stock data. We will turn to technical analysis to determine the stock price action prediction as a different approach to hopefully get a better predictor.

Machine Learning In Depth Analysis

Preprocessing and Learning

In this new scenario we had to do some data wrangling with the stock data. For the stock data we had to do the regular splitting of the data into a training and test set. We also prepped the data with features using feature engineering. The feature engineering was pulled from the book, Python Machine Learning by Example. The features include a lot of previous day's prices, previous day's rolling averages for 5, 30 and 365 days. The features also included volume and used the previous day's rolling averages for 5, 30 and 365 days. The volume features also include ratios of the newly calculated volume features. The same was calculated using the standard deviation on the volume. The features also include the previous returns for the past 1, 5, 30 and 365 days. Lastly the features also includes the 5, 30 and 365 day rolling moving average for the previous day.

In addition to this we also want to explore Time Series forecasting and apply ARIMA to the data to see if we can make an accurate prediction with that tool. We had to do some preprocessing of the data in order to run it through ARIMA. One of the things we had to do was take the log base 10 of the close prices. This was done in order to help ensure the series was stationary. We calculated the ADF statistic to see if the normal close price and natural log close price were stationary. According to the ADF stat the data was non stationary so we difference the series and we go on to find the values of p and q. Below are charts showing the differencing on the DJIA close prices. We can see that at the first differencing the series looks more stationary.



We experimented with the close price and log close price to see if we get a better forecast prediction. There was also some manual exploration of the various p and q values to see what the AR and MA coefficients were valid according to their P-values. We also used the Auto ARIMA function to explore the best pdq values once we manually attempted to find the best pdq to validate our manual search. Below is a result of the log close price Auto ARIMA output. The pdq values are 1,0,3 and all the P-values are less than 0.05 so we can reject the null hypothesis.

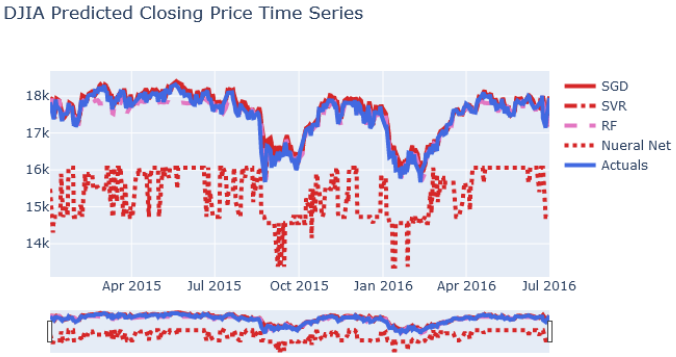
```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1611
Model:               SARIMAX(1, 0, 3)   Log Likelihood                6038.644
Date:                Tue, 07 Apr 2020   AIC                         -12065.287
Time:                        21:18:50   BIC                         -12032.980
Sample:                             0   HQIC                        -12053.295
                                - 1611
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0023      0.003      0.747      0.455      -0.004       0.008
ar.L1          0.9995      0.001   1348.612      0.000       0.998       1.001
ma.L1         -0.1150      0.015     -7.534      0.000      -0.145      -0.085
ma.L2         -0.0570      0.011     -5.079      0.000      -0.079      -0.035
ma.L3          0.0654      0.015      4.336      0.000       0.036       0.095
sigma2      3.237e-05   5.19e-07     62.404      0.000    3.14e-05    3.34e-05
===================================================================================
Ljung-Box (Q):                       89.65   Jarque-Bera (JB):              5909.50
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               0.13   Skew:                            -0.25
Prob(H) (two-sided):                  0.00   Kurtosis:                        12.37
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

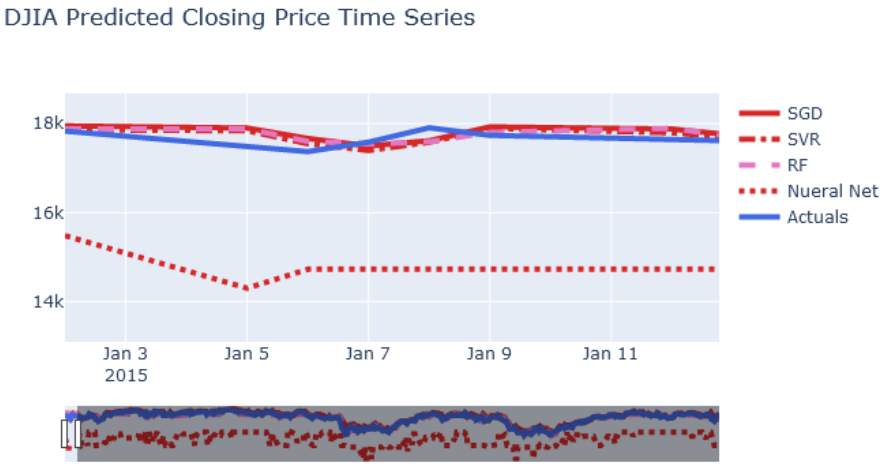
The data will go through 4 various Machine Learning algorithms and a grid search to see which one returns the best parameters to use in the prediction. The Machine Learning algorithms are Stochastic Gradient Descent, Random Forest, Support Vector and Neural Network.

Evaluation

We run the data first through the regression algorithms and we get the following results. The first graph is for the entire test set period. The SGD, SVR and RF models predicts the trend pretty well. The Nueral Net did have some weird error when it did the grid search so maybe the best parameters weren't reached. We didn't try to rerun the Nueral Net grid search as it took a very large amount of time to return and it returned with an error. Nevertheless we have three predictors to explore with.



Next we look at the 5 day forecast as this will be a lot more important as the model will lose accuracy over time and the market will be more susceptible to unpredictable news that could drive the market in a crazy direction. All models except NN predicted a higher value than the Actuals, then the Actuals rise and the model predictions are delayed and finally the models and Actuals come close to the same values on the fifth day. Though at a glance, seems good, there is a lot of error between the model and Actuals. The errors in this case are in the amount between $200-400, which in trading is a large room for error.



Next we have a table of various prediction accuracy measures. We compared all regression models based on their entire forecast and then for just the 5 day forecast. The cells highlighted in green indicate the lowest errors scores. Support Vector did the best for the entire forecast and for the 5 day forecast RF score better in MAE but SV overall doing the best out of the four regression models.

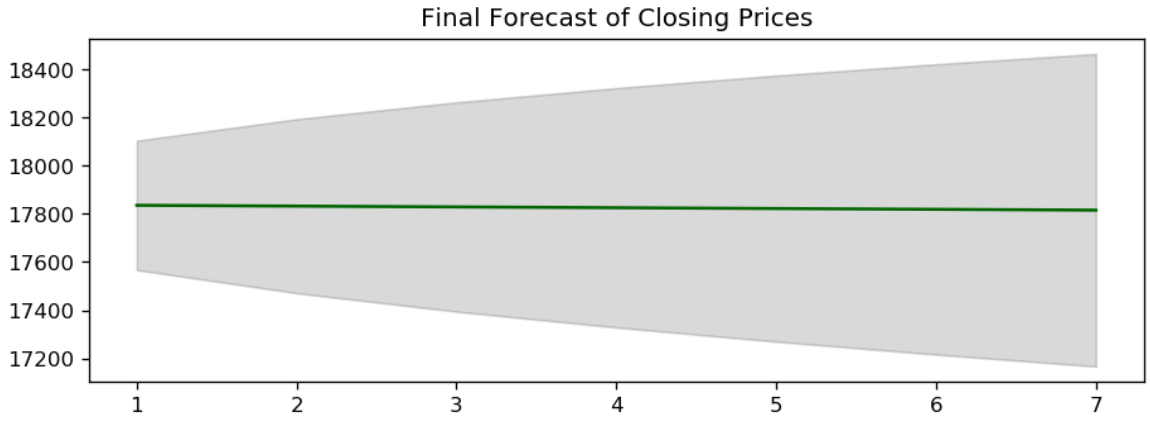
	Entire Forecast		5 Day Forecast			
Model	**MSE**	**MAE**	**MAE**	**MAPE**	**ME**	**RMSE**
SGD	41382	153	238	0.014	88	266
RF	42363	163	203	0.012	60	245
SV	31225	132	212	0.012	6	240
NN	4742582	2103	2845	0.161	-2845	2863

Below is a table of the model outputs and the variance from the Actuals. A negative value indicates the model predicted higher than the Actuals. It would be preferable to have the model predict lower than the Actuals so we could have a better chance of selling the stock if we went with the predicted value. We can see the magnitude of the variance of the model and Actual close prices. SV does well for the first three days, then RF does well on the fourth day and SGD does well on the fifth day. The units here are in dollars and the precision isn't close enough to trade as these variances could mean significant losses.

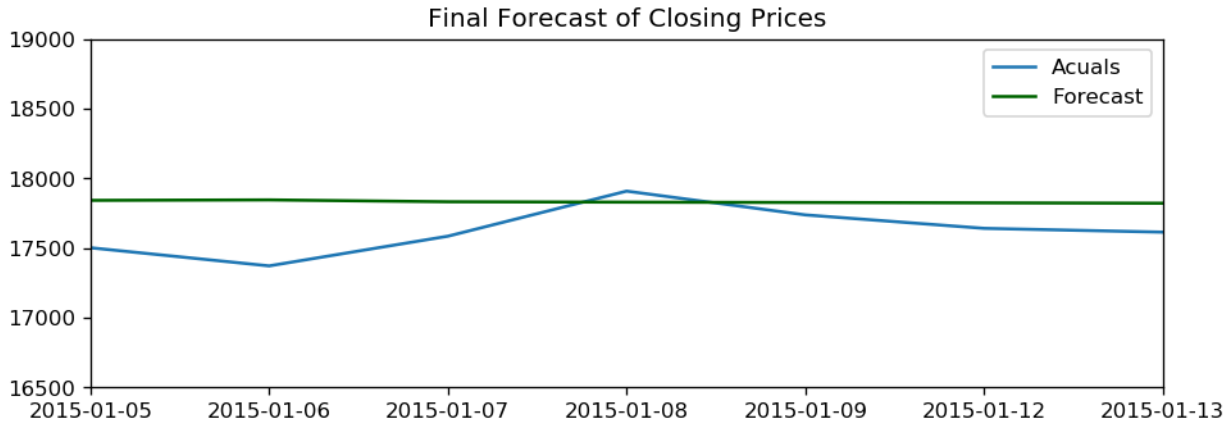
Date	SGD	SVR	RF	NN	Actuals	SGD_var	SVR_var	RF_var	NN_var
2015-01-02	17949.85	17862.69	17879.80	15484.47	17832.99	-116.86	-29.70	-46.81	2348.52
2015-01-05	17902.41	17841.56	17882.06	14303.67	17501.65	-400.76	-339.91	-380.41	3197.98
2015-01-06	17668.77	17548.52	17602.20	14729.30	17371.64	-297.13	-176.88	-230.56	2642.34
2015-01-07	17499.23	17397.05	17543.52	14729.30	17584.52	85.29	187.47	41.00	2855.22
2015-01-08	17617.39	17582.02	17592.61	14729.30	17907.87	290.48	325.85	315.26	3178.57

The regression methods are not bad actually but they do miss the precision that is desired. If the model was close within 10 units the model would be more usable. We will explore Time Series forecasting with ARIMA next to see if we can achieve a higher accuracy.

ARIMA needs very little data to work with, just the date and the historical values that we will use to predict into the future. We explored the close and log close prices. One of the things we did was look at the close price forecast that used the Auto ARIMA solution. That solution for the close model suggested a pdq of 1,0,1. Below is the forecast for the next 7 days. The green line is the forecast and the shaded grey area is the 95% confidence band.



It’s not a very useful forecast as the forecast is a straight line and from historical actuals, the price action rarely goes that way. We also explored how the forecast would look if we used log close and then using Auto ARIMA to find the best pdq. After running Auto ARIMA the optimal pdq was 1,0,3. Below is the forecast compared to the Actuals. The variance is too high and makes the forecast not very useful unfortunately.



Next Steps

This capstone project covered a lot. For the text data, we applied Natural Language Processing task such as tokenization and Stemming. We applied Unsupervised leaning algorithm such as K means clustering and Latent Dirichlet Allocation. We used the text data and four regression algorithms to predict how the stock price would move but we only got an accuracy in the 50%-57% range. Moving over to just the stock data, we applied feature engineering and four regression algorithms to predict the actual stock price. The models predicted the price trend well but the precision was not at a level that was usable. In the future we could explore more variables that could help us predict the price action. Perhaps there are many more variables we need to consider and that would bring the predicted price close to the actual price. We could also try modeling 15 minute closing prices as that might be a better range to predict. We could also try modeling another stock or another industry where the prices are not as high as the DJIA. I believe the key to improving the model prediction of the prices will come down to feature engineering and selecting the correct stock and time frame. The ARIMA forecasting proved to be unfruitful but we learned how to apply many steps of time series analysis, testing for stationarity and using ARIMA to predict forecasting. ARIMA was not useful in this case but it could have value in another scenario, perhaps another domain of business.