Natural Language Processing (NLP), Artificial Neural Network, Naïve Bayes

The Secret Behind Warren Buffet's Words

"Using an NLP algorithm to predict the performance of Berkshire using the words in the Annual Report"

Volak Sin, Oct 3, 2017

I Introduction

Warren Buffet is arguably the greatest investor of all time. His company, Berkshire Hathaway, has completely outperformed the S&P 500. If you had put $10,000 into the S&P 500 in 1965, today that would worth $1.3 million. If you had given that amount instead to Warren Buffet, you would have $88 million dollars.

Warren Buffet is so revered among investors and shareholders that his annual stockholders' meeting is filled up with stadiums of people; all waiting to hear what the chairman has to say.

We took that interest in Warren's words one step further. We will analyze every word that Warren has written in Berkshire's annual report and try to make future predictions off them.

Once we have created our prediction algorithms, we will take this year's annual report and predict the year-end stock performance. In particular, we want to answer these two questions:

Will Berkshire Hathaway (BRK) outperform the S&P 500?

What is the predicted stock performance of Berkshire?

II Methodology

The reports are available on Berkshire's website: http://www.berkshirehathaway.com/letters. Ideally, we would like to have the annual reports starting from 1965, but the website only provides reports starting from 1977. The annual reports from 1977 to 2007 are in HTML format and can be easily "scraped" from the website. The reports from 2007 up until the present are in PDF format. In order to process these reports, we had to download them and then convert them to text using Adobe Reader. Once converted, we append these reports to the previously scraped ones.

Removing Unnecessary Characters

One of the first steps of processing text data is to remove unnecessary characters and strings. There is a Python package called Beautiful soup that will help with this process. As with most annual reports, we must remove all the tables with numbers. We will also remove non-essential "stop words" like "the" and "to."

Stemming Words

After removing all the unnecessary characters, we will stem words together. Stemming is a process in which we take words like "loved" and convert it to "love." This helps remove redundancies of certain words. It also has the added effect of drastically reducing the total number of words used in the model.

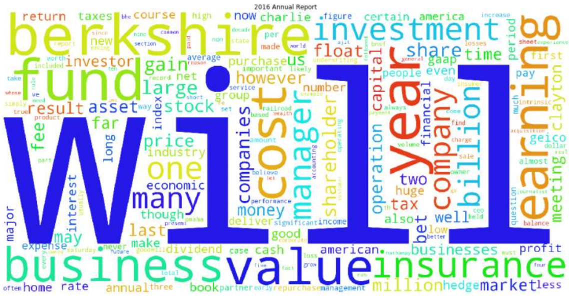
Creating a vector of Words

The final step in processing the text is to convert the words into "vectors." This process turns each word into an explanatory variable with a binary outcome; either the word was used in a particular annual report or it was not.

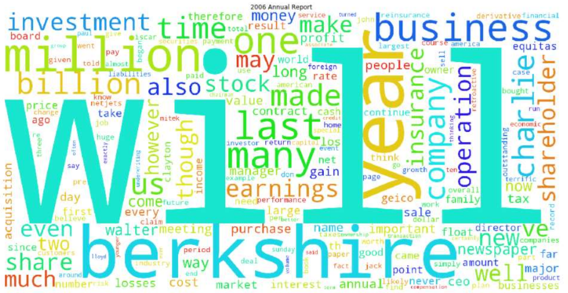
It is apparent by using 1s and 0s to represent each word, we are creating a dataset where the words are simple categorical variables. The consequence of transforming these words into features / explanatory variables is that you now have over 8000 variables.

Word Cloud

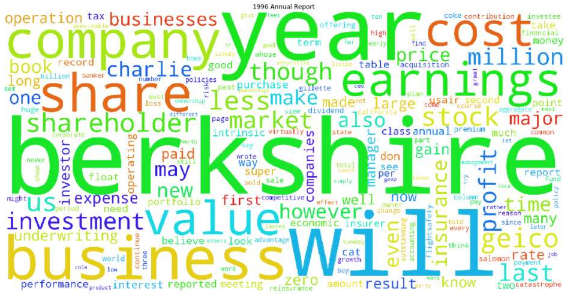
Below are word clouds generated from a selected number of annual reports. The images of the word cloud are to just give a quick glance of the types of words used in the annual report and to show if those words have changed through the years.



We see how the above word cloud from the 2016 annual report and the word cloud from 2006 below share similar word usage.



For the ten years prior to that, the two word clouds are a little more varied. It seems the word "Berkshire" had more emphases in 1996 than in the above 2006 word cloud. Again, these word clouds are purely for illustrative purposes, and we would not want to draw any conclusions from the validity of certain words being used. Using text to predict investment performance is already an impossible task, let alone trying to understand trends by looking at commonly used words.



III Machine Learning Algorithms

There will be two types of algorithms used; one for each question. The first question requires a classification algorithm while the second requires a regression algorithm. For the classification algorithm, we will use the Logistic Regression as the baseline model for comparison. We then followed with the K-Nearest Neighbors, Support Vector Machines, Naive Bayes, Decision Tree, Random Forest, and XgBoost. For our regression model, we will use the linear regression as the baseline model. Given the nature of Natural Language Processing, there is virtually no chance of this model being linear, but we will do it just to see now nonlinear the model is. We then will run a Support Vector Regression, Random Forest Regression, XgBoost, and a Neural Net on the NLP dataset.

i. Logistic Regression

The Logistic regression will be the baseline model for our classification algorithms. Since most of the variables/ features have a binary outcome, the model should be able to make pretty powerful interpretations.

ii. K Nearest Neighbors

For our K Nearest Neighbor algorithm, we chose the number of neighbors to be 5 and set the distance parameter to be Euclidean. After running the algorithm, we notice that the model predicted that Berkshire will always beat the S&P 500. This prediction does not add any insight to our dataset, so we do not include it in the Jupyter notebook.

iii. Support Vector Machine

Support Vector Machines requires the features to be scaled, but the algorithm does not automatically do so. Since the bag of words creates values of 1s and 0s, we do not need to scale our features.

iv. Naive Bayes

For the Naive Bayes classifier, we used the default parameters. The classifier is also nonlinear. The classifier works on the principles of Bayes theorem, but without the requirement of feature independence.

v. Random Forest

For the first ensemble method, we will choose to run a Random Forest. We chose our number of estimators to be 2000 as this will help compensate for the 8000 features in the dataset.

vii. Artifical Neural Network

For our Neural Network, we choose a three-layer architecture with 1000 nodes in each of the hidden layers. We tried various combinations of layers and nodes, and this one seems to produce the optimal result. We used a rectified linear unit for the hidden layers for both the classification and regression case. For the output layer of the classification, we chose a sigmoid function with "accuracy" being the loss metric being measured.

vii. XgBoost

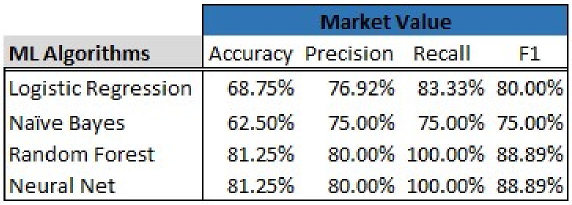
XgBoost is one of the most popular models in machine learning. It is also the most powerful implementations of gradient boosting. One of the major advantages of Xgboost besides having high performance and fast execution speed is that you can keep the interpretation of the original problem.

IV Results

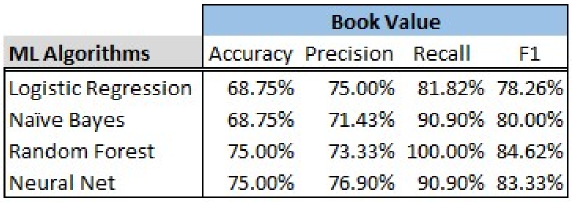
i. Classification

Looking at the Market value table, we see that the random forest algorithm and neural network algorithms produce the best results when comparing all 4 metrics of accuracy, precision, recall, and F1 score. They, in fact, came to the same conclusion.

Classification Results:



	Market Value			
ML Algorithms	Accuracy	Precision	Recall	F1
Logistic Regression	68.75%	76.92%	83.33%	80.00%
Naïve Bayes	62.50%	75.00%	75.00%	75.00%
Random Forest	81.25%	80.00%	100.00%	88.89%
Neural Net	81.25%	80.00%	100.00%	88.89%



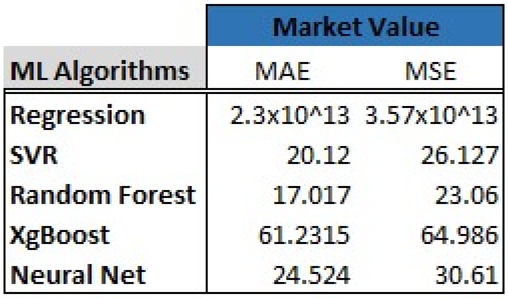
	Book Value			
ML Algorithms	Accuracy	Precision	Recall	F1
Logistic Regression	68.75%	75.00%	81.82%	78.26%
Naïve Bayes	68.75%	71.43%	90.90%	80.00%
Random Forest	75.00%	73.33%	100.00%	84.62%
Neural Net	75.00%	76.90%	90.90%	83.33%

An accuracy above 80% does not necessarily mean our model has worked in offering better predictions. For one thing, Berkshire already beats the S&P 500 index 72% of the time, and a 9% increase is not really "stepping out on a limb" when it comes to predictions. Our biggest concern is an "over fit" of our algorithm. Even though we did a train / test split, the size of the data coupled with our inability to perform a k-fold cross validation due to computing constraints puts an asterisk on our model. Our model is not worthless since it is quite above 50%, a random prediction made by a coin flip.

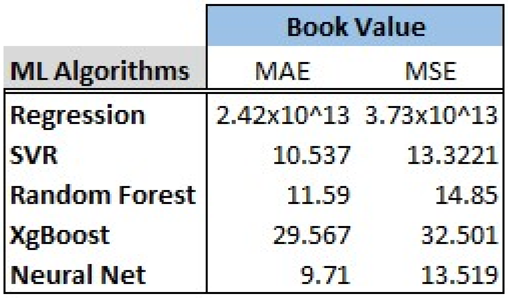
ii. Regression

We originally did not plan to run a regression model as it seemed pointless; regardless of the results, there would be no serious interpretation of the outcome. The regression models were performed just to see what outcomes would be produced.

Regression Results:



	Market Value	
ML Algorithms	MAE	MSE
Regression	2.3x10^13	3.57x10^13
SVR	20.12	26.127
Random Forest	17.017	23.06
XgBoost	61.2315	64.986
Neural Net	24.524	30.61



	Book Value	
ML Algorithms	MAE	MSE
Regression	2.42x10^13	3.73x10^13
SVR	10.537	13.3221
Random Forest	11.59	14.85
XgBoost	29.567	32.501
Neural Net	9.71	13.519

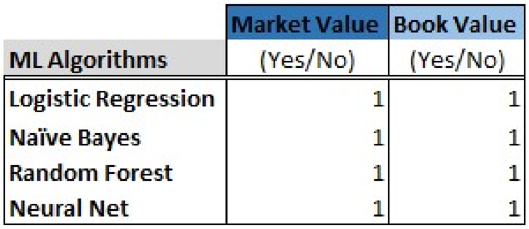
As expected, our baseline Linear Regression model produced outrageous values. This verified the fact that the dataset is not a linear model. It was surprising to see the models having relatively small mean absolute errors and mean squared errors. Of course, these values are useless for all practical purposes; what good is an error value that can fluctuate between a negative and positive return. It is interesting to note that all the Mean Squared Errors values are greater than the Mean Absolute Errors. This is probably due to a greater variation in errors among larger numbers.

V Predictions

i. Classification

All of our algorithms made the same prediction when answering the question, "will Berkshire outperform the S&P 500." Again, this prediction is hardly surprising given the fact that it had done so 72% of the time.

Classification Prediction: Will Berkshire beat the S&P 500?

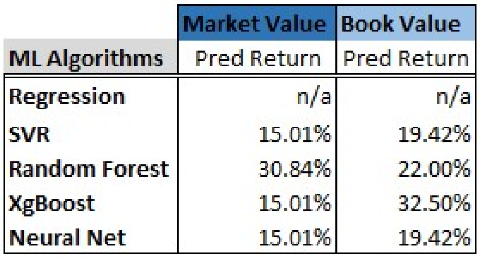


	Market Value	Book Value
ML Algorithms	(Yes/No)	(Yes/No)
Logistic Regression	1	1
Naïve Bayes	1	1
Random Forest	1	1
Neural Net	1	1

ii. Regression

The results of our prediction models were surprising, especially when comparing market value to book value. The market value has greater yearly variance than the book value. As such, we would expect there to be greater variance in prediction value. When we look at the results table below, we see that the opposite happened. That our Market value prediction had more modeling consensus than our Book Value prediction.

Regression Prediction: How will Berkshire perform next year



	Market Value	Book Value
ML Algorithms	Pred Return	Pred Return
Regression	n/a	n/a
SVR	15.01%	19.42%
Random Forest	30.84%	22.00%
XgBoost	15.01%	32.50%
Neural Net	15.01%	19.42%

Besides that, it was also surprising to see that our models predicted that next year's return on Book Value will be greater than the market value. These prediction values seem fairly reasonable until you take into account the Mean Square Error of the model from the previous section. Even under the best model scenario a prediction of 15%, we would still get a prediction range of -6.98% to 36.98% with a 95% confidence level. Again, this range is not very useful for all intents and purposes.

VI Conclusions

This analysis was done purely for educational purposes. Our goal was to get familiar with the methodology behind Natural Language Processing. There was no expectation for a model created using the words written in an annual report to have any prediction validity. Though our models did produce results that seem to indicate some value being added by analyzing the words of an annual report, our computational limitation did not allow us to remove overfitting as a possible culprit for model accuracy.

Though it is quite possible that the chairman may use words that seem to indict a more optimistic outlook on the prospects of Berkshire Hathaway, it would still be virtually impossible to use those words to extrapolate a regression model. For a classification model, we would need a near-perfect accuracy rate before we could even begin to assume that modeling the success compared to the S&P 500 is possible.

VII Improving The Model

On the 50th anniversary of Berkshire Hathaway, Charlie Munger also wrote his thoughts and opinions in that year's annual report. Manually removing the words written by Charlie Munger could help improve the model.

The best improvement from the model would probably come from selecting parts of the annual report that only pertain to the future prospects of the company. Since we initially did not believe the model could produce sensible results, we did not take the time to manual read the annual reports and only included those parts.

Including the annual reports from 1965 to 1977 would also strengthen the accuracy and validity of our models. It would be quite a search to find these annual reports. There is a book available that include these annual reports, but an OCR device would be needed to convert the book image into usable text.

VI Alternative Strategies.

An alternative strategy would be to include transcripts of Warren Buffet's conversations throughout the year. The chairman of Berkshire Hathaway routinely gives interviews to various media outlets. He answers a wide range questions, including the prospects of the company.

These added transcripts would help add to the sparse dataset. There is a max number of 50 annual reports, and we would need at least 1000 instances of the Warren Buffett's words. This new model would run into difficulty if the chairman's speaking vocabulary is quite different from his written vocabulary.