Predicting Stock Performance Using Quarterly Analyst’s Call Transcripts and Natural Language Processing (NLP): An Exploration

by

Maureen Brennan

A Capstone Project Submitted to the Faculty of

Utica College

August 2021

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in
Data Science

Abstract

This research uses publicly available data to predict stock price performance using Natural Language Processing (NLP) approaches. The text data is five years of transcripts from the quarterly earnings calls of 10 publicly traded, NASDAQ-listed companies as the basis to compare five different text-based predictive analysis approaches over two different forecast horizons, one business day after the earnings call and five business days after the earnings call. Stock performance is assessed by comparing the change in the price of the company's stock relative to the change in NASDAQ index over the same time period. The methodology leverages the following NLP processes to vectorize the transcript text: TF-IDF, Word2Vec, Doc2Vec, and Recurring Neural Networks (RNN). This paper also explores a novel word sentiment feature extraction process using the Word2Vec most_similar() function to generate a list of words that could help extract industry-specific insights. Another set of forecasts were generated using stock price data for the 10 companies on the dates of the earnings calls. With the exception of the RNN models, the prepared data was split into training and test sets and used to fit and test a Random Forest Classification machine learning model to compare each data source's predictive strength. The RNN model also used the same train/test data, but the RNN process encapsulated the text tokenization and model fitting and testing process into an integrated process, so a Random Forest analysis was not required. The baseline NLP models show very promising predictive capabilities. The NLP data sources performed very competitively against the stock price data, especially in the 5-day forecast horizon. The novel sentiment approach also showed surprisingly high recall, indicating that it may provide information gains of value to investors.

Keywords: Data Science, Dr. Michael McCarthy, NLP methods, industry-specific text sentiment

Acknowledgments

The author extends sincere thanks and appreciation for the support of her capstone committee, Dr. Michael McCarthy and Gunther Scherer, for their time, feedback, and insights. She also thanks her husband for his continued support and encouragement.

Table of Contents

List of Illustrative Materials

Introduction

Machine learning literature reviews show that standard machine learning models can predict short-term directional movements in stock prices with enough accuracy to be profitable in some mature financial markets (Hsu et al., 2016). Predicting changes in stock prices is important because traders can profit from anticipating the directional movement of stock prices. As machine learning models demonstrate their ability to out perform traditional econometric stock movement forecasts (Hsu et al., 2016; Lv et al., 2019), market research reveals that approximately 1,360 hedge funds (about 9% of all hedge funds) have adopted computer-based models to prepare their trades (Metz, 2016).

In their review of stock price predictive machine learning models, Hsu et al. indicated that standard machine models perform well and excluding typical econometric details from models actually produces more accurate predictions (2016). Two academic surveys of machine learning and stock predictions outline the success of traditional machine learning algorithms to create profitable stock trading predictions (Hsu et al. 2016, Lv et al., 2019). Similarly, a blog posting published by Microsoft outlines a team's efforts to leverage natural language deep learning to make similar predictions of stock price movements with results that show real promise (Ryan, 2017).

In addition to considering markets and models that generate effective stock price predictions, the Hsu et al. team also analyzed the length of the prediction horizon and a model's profitability and accuracy. A short forecast horizon, in hours for example, may have more profitable results, especially in markets with high degrees of volatility. However, longer horizons (starting at one day), show increases in the accuracy of predictions (Hsu et al., 2016). That team suggests that differences in intraday volatility and trading opportunities, as compared to volatility and trading opportunities between days, may account for these effects (Hsu et al., 2016). Because text-based approaches intend to accelerate the rate at which investors can glean insights that may disseminate more slowly than financial data, this analysis uses two different forecast horizons to test this theory. The Day +1 forecast horizon predicts the performance of a company's stock when there are only a few hours for the public to assimilate this information. The Day +5 forecast horizon provides a prediction of a stock's performance after industry experts have had an opportunity to absorb the complex public disclosures.

This paper uses five years of transcripts from the quarterly earnings calls of 10 public, NASDAQ-listed companies between 2016 - 2020 to analyze the potential information gain made possible by Natural Language Processing (NLP) and sentiment analysis, compared with stock-price approaches. To explore the value of word context and sentiment analysis, the author incorporated five different types of NLP analysis, each with a unique approach to calculate word context and sentiment to prepare data for machine learning predictive analysis. All models in this study were measured according to precision, recall, and accuracy.

In contrast to many models that predict simple positive/negative stock price changes over time (Hsu et al., 2016; Ryan, 2017; Lv at al., 2019), this study seeks to consider opportunity costs and background risk. For each company in the study, the company's opening stock price and the NASDAQ's opening price on the day of the earning's call is compared to the opening price of the company's stock and the NASDAQ index on day of the forecast horizon period. So, for the one-day forecast horizon ("Day +1"), this means the opening prices on the day of the earning's call are compared to the opening stock prices on the consecutive business day. In this scenario, a prediction would be calculated at the end of the reporting day and the stock could be bought, sold, or held based on the prediction before the following market day. To determine whether the stock performed “positively,” the change in price percent of the company stock is compared with the change in price percent of the NASDAQ index. Therefore, a company’s stock may have a slight increase over the forecast horizon but would still have a “negative” performance if it lags behind the NASDAQ performance over the same time period. This reflects the underlying reality that an investor would be better off holding a NASDAQ index fund over the forecast horizon than the stock in question.

Stock performance over the five-day forecast horizon (“Day +5”) follows the same logic, using the opening stock prices on the day of the earnings call and comparing them to the opening day prices on the fifth business day of the forecast horizon, and comparing the relative performance of each company’s stock to the NASDAQ performance in the same time period. Because companies vary in financial calendars and dates for earnings calls, the start and end of each prediction covers a unique timeframe for every company and quarter.

For baseline comparison of the models, the author also prepared a non-text based Random Forest classifier, similar in concept to the machine learning data inputs described by Hsu et al., though in this analysis a simple Random Forest classifier models the predictions related to stock movements (as opposed to the SVM and ANN networks) (Hsu et al., 2016). The intention here is not to find the best machine learning prediction model possible, but to compare baseline results across the various modeling approaches. Because the performance of machine learning models using stock price details has already been demonstrated (Hsu et al., 2016), this comparison indicates whether NLP data sources can perform competitively with stock price data sources. This is especially pertinent because of the increased resource requirements required to gather the vast amounts of data that would be required to build more mature and implementation-worthy NLP-based predictive models.

Among the NLP data source preparation alternatives, the author introduces a novel approach to sentiment analysis, not seen in any academic research found. This unique approach seeks to determine whether word sentiment can leverage industry-specific insights about foundational company performance. If this is possible, there may be an opportunity for investors to tune and direct NLP data modeling to reflect industry insights that individuals might accumulate through years of research and experience. By comparing this novel approach against established NLP methods, the author seeks to determine whether this approach has statistical validity and potential to guide the use of data sources in larger studies.

For this novel word sentiment approach, the author considers expertise about how individual companies leverage data, and its relationship to its overall success. The importance of data strategy, architecture, management, and monetization are all increasingly important factors to companies' success (Iansiti & Lakhani, 2020). Because the 10 companies included in this analysis are all in high-technology, this is especially true because high-tech companies must efficiently offer data products and services in addition to managing their own internal data. The "Data Word Sentiment" data preparation uses word sentiment functions available in the Word2Vec text data processing package to extract words from the earnings call transcripts that may reflect underlying reality related to how each company approaches this extremely important subject of data management. This "manual" word sentiment approach extracts a list of the top 25 words with positive associations with the word "data" from the corpus of earnings transcripts using a random sample of 80% of the transcript documents. It then counts the instance of each of these target words in every individual call transcript in the study. The resulting matrix is used to fit a Random Forest model to predict the Day +1 and Day +5 stock performance for each company.

Using the inputs described above, this study design intends to test the following hypotheses:

> H_1: A RNN natural language deep learning neural network will outperform more traditional NLP machine learning models in accurately predicting the performance of stock prices relative to the NASDAQ index.
>
> H_2: Extracting words with positive sentiment associations to the word "data" in the earnings call transcripts and calculating features to increase the weight of those words will improve the performance of the text-based prediction.
>
> H_3: The stock price data (non-text-based) Random Forest model will perform at least as well as the text-based models on the data.
>
> H_4: Text-based prediction models will generate better precision, recall, and accuracy using the Day +5 forecast horizon than the Day +1 forecast horizon.

Literature Review

Predicting the movements of stock prices is important from the perspective of managing investment portfolio risk and earning returns for financial assets. Both institutional investors and individual investors rely on consistent stock market returns to help grow assets and plan for future activities. For these reasons, the history of stock return predictions is long and contentious.

Some economists subscribe to the hypothesis that markets are, at their core, efficient in distributing new information quickly, which in turn is quickly reflected in their stock prices. The scarcity of arbitrage opportunities, imbalances in markets that can be exploited by savvy traders, serves as proof to the "Efficient Market Hypothesis" crowd that markets do not have predictable, repeatable inefficiencies that can be leveraged for profit (Malkiel, 2003). The Efficient Market Hypothesis school of thought believes that a stock's price reflects the consensus of investor opinion reflecting the long-term value of the underlying company (Malkiel, 2003).

More recently, a new class of investor has leveraged emerging technology and available computing power to focus on finding very short-term inefficiencies that create arbitrage opportunities that are more difficult to assess, not only in information about a company's future performance, but also in the technologies used to trade stocks, collect and communicate economic information, and more (Jansen, 2020). Contemporary traders leverage machine learning and artificial intelligence models to find and exploit these changes whenever possible.

Little is written about a hybrid approach of these two schools of thought. Namely, can machine learning be used to determine which companies are best managed for future growth and success? The review of the Efficient Market Hypothesis approach and the machine learning/artificial intelligence approaches to stock prediction below provide background for the author's proposal that the two concepts could, hypothetically, be combined to assess the longer-term growth potential of a stock portfolio.

Efficient Market Hypothesis (EMH)

In his 2003 paper, "The Efficient Market Hypothesis and Its Critics," Burton G. Malkiel explains the efficient market perspective on the movement of stock prices. Because this group believes that information travels quickly and access to trade stocks is unencumbered, stock prices reflect the consensus opinion about a company's long-term value. The inability of experienced portfolio managers to beat the performance of stock indexes suggests that there are no arbitrage opportunities available for dependable profit-making (Malkiel, 2003). History has shown multiple examples where short-term, positive associations could be found; Malkiel points out that they do not negate the Efficient Market Hypothesis for three key reasons: 1) the cost of executing trades to take advantage of short-term movements in the stock market typically outweigh any potential profit, 2) because stocks will revert to the mean over the long run, traders are likely to lose profits made related to short-term trading strategies, and 3) when the public catches on to any short-term arbitration potential, the collective behaviors of the crowd quickly subsume any predictive potential (2003).

The Efficient Market Hypothesis says that stock market movements are unpredictable, as all the known information about companies, interest rates, politics, economic trends, and more are already calculated into the price of the stock. Only new events can effectively change the perceived value of a company and its stock (Malkiel, 2003). Although bubbles occur, investors make mistakes, and data mining can produce short-term insights into predictable patterns, these all exist within the context of the market, and the prices will always reflect the collective judgment of investors. Malkiel summarizes this point of view with a standard reference for this field, “If $100 bills are lying around the stock exchanges of the world, they will not be there for long” (Malkiel, 2003, p. 80).

Machine Learning and Stock Predictions

Contrasting with the long-established Efficient Market Hypothesis, popular news sources percolate tantalizing headlines suggesting that the application of machine learning and neural networks may uncover some “$100 bills” hidden in stock exchanges and only wait to be interpreted correctly. Wired magazine reported on hedge funds developed specifically around artificial intelligence strategies, which appear to beat human-based stock trading returns in some cases (Metz, 2016). Similarly, the Financial Times reported in 2017: “Computer-based hedge funds have been admitted to a list of all-time top 20 best performers for the first time in a sign that the dominance of traditional human investing is being radically challenged by technology” (p.13).

The astute reader will note that in neither case do these articles indicate that these models return profitable margins, nor do they compare them to an index fund of random stocks to compare results to a baseline. Empirical academic papers help illuminate the potential of machine learning and related technologies to improve traditional stock forecasting processes. Studies evaluating the performance and profitability of various machine learning technologies (including deep learning neural networks) shows promise (Hsu, et al., 2016). Support Vector Machine and similar "traditional" machine learning models appear to perform better than deep learning neural networks, for example (Hsu, et al., 2016, and LV et al., 2019). The paper "Bridging the Divide in Financial Market Forecasting: Machine Learners vs Financial Economists" specifically addresses the question, does the improved performance of machine learning models contradict the Efficient Market Hypothesis? Should these academic findings change our fundamental understanding of efficiency of financial markets (Hsu, et al., 2016)?

After careful analytical comparison of multiple forecasting techniques across a range of financial markets, Hsu and coauthors determine that machine learning does not fundamentally change our understanding of market behavior, mostly because many interacting factors determine how well a model can predict stock price direction and how profitable it can be in the process (Hsu et al., 2016). Rather, this group of authors suggests that inefficiency may exist in some markets. Importantly, the authors also note that a model does not have to have accuracy far above a 50-50 coin flip predicting the direction a stock will move to be profitable. The ability of an algorithm to extract even small information gains, even just beating the 50% luck rate, can yield a profitable stock trading strategy (Hsu et al., 2016).

Available research suggests that a number of factors impact the success of a machine learning model in both prediction and profitability outputs, including the selection and tuning of the model, the amount of training data, the selection of the market, the forecasting timeline, and more (Lv et al., 2019, Hsu et al., 2016). It is the unique, and comparatively novel approaches available to data scientists that find apparent “inefficiencies” to be exploited, as Hsu et al. explain:

> Our results, therefore, show that ML techniques, such as SVM and ANN [artificial neural networks], are useful techniques for detecting market anomalies. The conventional approach in the financial economics literature (Fama, 1970; Fama & French, 1993) is to use autocorrelation and linear regression models to examine the relation between explanatory factors and stock prices (e.g., Keim, 1983; French 1980). ML methods work a different way. They are trained to recognize patterns in a data-driven manner and do not require human intervention. (2016)

The success of academic models in predicting the direction of stock price movements suggests that there is potential in this field for the application of machine learning to not only predict changes based on history, but also to identify variables for predicting stock performance that may currently either fall outside traditional econometric models, or that require more analytical processing time so information is more slowly assimilated into traditional stock pricing and forecasting approaches.

Natural Language Processing and Stock Movement Predictions

The empirical studies analyzing machine learning models’ accuracy in predicting stock market movements (typically a one-day forecast) did not exceed 60% in the most favorable conditions in a review of methods, variables, and markets (Hsu et al., 2016). This team’s analysis included the opening, closing, and change in stock prices, but determined that other technical econometric measures did not improve their models’ predictive accuracy or profitability (Hsu et al. 2016). The team noted their surprise that technical economic covariates did not improve models’ accuracy or return on investment, and they recommended excluding technical economic indicators, like moving averages, from machine learning forecasting models

(Hsu et al., 2016). This raises an interesting question, if traditional econometric calculations are not effective variables for predicting stock price movements, what are effective variables?

A Microsoft.com blog post by Patty Ryan titled, “Stock Market Predictions with Natural Language Deep Learning” provides an interesting perspective on alternative data sources that can yield effective insights. In her post, Ryan describes an NLP machine learning pipeline intended to provide insights to a person who will make final decisions related to stock purchases and sales. The baseline accuracy she calculates is 33.33% (Ryan, 2017), presumably allowing that in each case a stock price can decline, stay the same, or increase, though Ryan does not explicitly provide the rationale for this baseline assumption.

Ryan and team used text from two years of quarterly SEC filings of thousands of publicly held companies, focusing on the following sections of each company’s 10-k reports: “the business discussion, management overview, and disclosure of risks and market risks” (p. 1, Ryan, 2017). They used that text input to predict the direction of the stock price five business days after the 10-k disclosures were made public. While the author notes that their sample size was limited, and the results of their “prototype” would require additional data and subsequent analysis, the results were still impressive. The team targeted stocks that would decline, as they determined that those instances were of the greatest importance to their audience, who naturally would prefer to avoid investments that were likely to lose value. They were able to predict the accuracy of these price declines with 62% accuracy (Ryan, 2017). Comparing those results to the stock-price machine learning methods indicates that NLP has substantial potential to add significant insight into stock trading workflows, which aligns with the 60% target machine learning accuracy for stock price performance defined by Hsu et al. in their review of profitable stock performance machine learning algorithms (2016).

In her results discussion, Ryan suggests that better results may emerge by separating text analysis by industry (2017). This suggests that extracting context of the jargon, strategies, and operations specific-words within an industry may provide more insight than general text that lacks that industry-specific context.

Methodology

The analysis in this paper uses transcript text from earnings calls for 10 public companies, all of which were listed on the NASDAQ stock market, from 2016 – 2020. Earnings calls are quarterly meetings hosted by companies to complement their financial disclosures. They are distinct from SEC reporting text in that they are much shorter and that they also include questions posed by industry analysts and exectuives' replies (Jansen, 2020). 188 earnings calls text transcripts were curated and made publicly available by Roozen, D. and Lelli, F., along with the related stock prices for each company in the data set over the selected period; the site also posts the NASDAQ index prices over the same period (Roozen & Lelli, 2021). The ten companies included in the sample are:

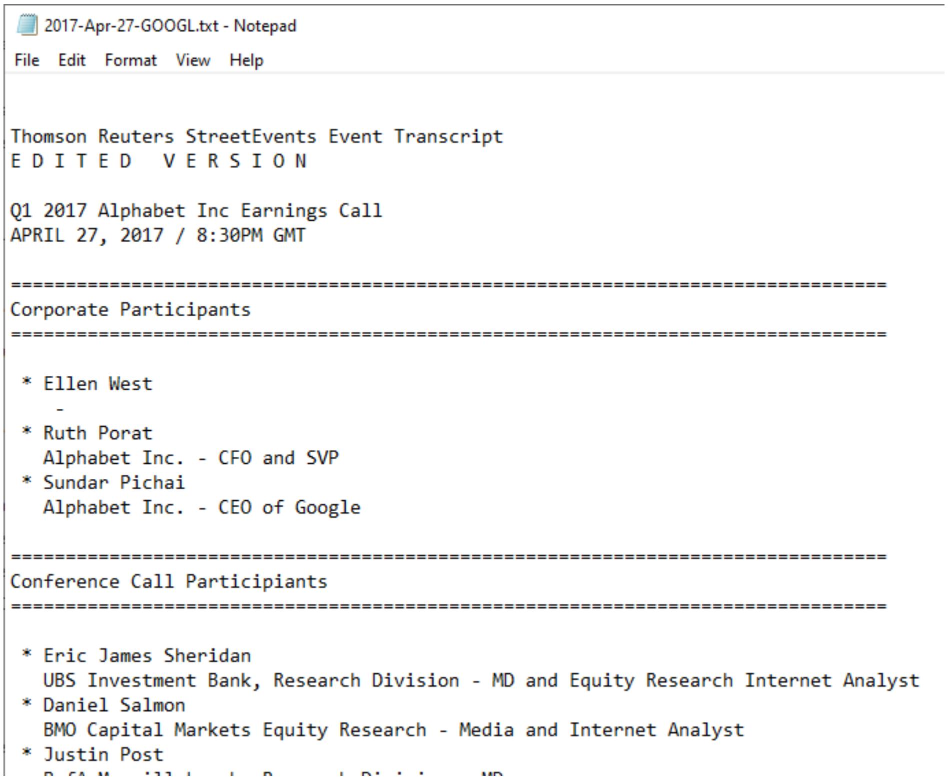
Table 1: Companies included in NLP stock price movement analysis

Stock Ticker Symbol	**Company Name**	**Number of Transcripts**
AAPL	Apple, Inc.	19
AMD	Advanced Micro Devices, Inc.	19
ASML	ASML Holding N.V.	19
AMZN	Amazon.com, Inc.	19
CSCO	Cisco Systems, Inc.	19
GOOGL	Alphabet, Inc.	19
INTC	Intel Corporation	19
MSFT	Microsoft Corporation	19
MU	Micron Technology	17
NVDA	NVIDIA Corporation	19

In addition to the analyst call transcripts, the stock price details, including daily opening price, high price, low price, closing price, adjusted close, and volume are contained in the stock price reports curated for this data set by Roozen and Lelli (2021).

Transcript Text Preparation

In the data curated by Roozen and Lelli, the transcript for each earnings call was its own text file. The file name contained the company ticker symbol and the date of the earnings call, and the document header included the reporting quarter and the date of the earnings call, as well as other details about the time and participants on the call (Figure 1).



```
Thomson Reuters StreetEvents Event Transcript
E D I T E D   V E R S I O N

Q1 2017 Alphabet Inc Earnings Call
APRIL 27, 2017 / 8:30PM GMT

===============================================================================
Corporate Participants
===============================================================================

 * Ellen West
    -
 * Ruth Porat
   Alphabet Inc. - CFO and SVP
 * Sundar Pichai
   Alphabet Inc. - CEO of Google

===============================================================================
Conference Call Participiants
===============================================================================

 * Eric James Sheridan
   UBS Investment Bank, Research Division - MD and Equity Research Internet Analyst
 * Daniel Salmon
   BMO Capital Markets Equity Research - Media and Internet Analyst
 * Justin Post
```

Figure 1: 2017 Q1 Google Earnings Call Transcript (Roozen & Lelli, 2021)

A series of Alteryx workflows extracted the following data from each transcript text file: company name, ticker symbol, earnings call date, reporting quarter. Subsequent workflows joined the opening stock prices associated with each earnings call using the combination of call date and ticker symbol. Formulas calculated the date for the Day +1 and Day +5 by excluding Saturdays and Sundays, for which stock data would not be available. Lastly, the Stock_lag attribute was calculated by comparing the Stock_ratio and the Sector_ratio for the two forecast horizons, as outlined in Table 2:

Table 2: Stock_Lag Attribute Calculation

Day +1 Stock Ratio	Open stock price the day after earnings call/ Open stock price on the day of the earnings call
Day +1 Sector Ratio	Open NASDAQ index price on the day after the earnings call/ Open NASDAQ index price on the day of the earnings call
Day +5 Stock Ratio	Open stock price the fifth business day after earnings call/ Open stock price on the day of the earnings call
Day +5 Sector Ratio	Open NASDAQ index price on the fifth business day after the earnings call/ Open NASDAQ index price on the day of the earnings call
Stock_Lag	This attribute displays “1” if the Sector Ratio > Stock Ratio

By associating each Stock_lag calculated attribute with the source earnings call for each, the results could be joined back to the text data records. After the Stock_lag labels were added to the data set, an “output” file could be prepared using only the Stock_Lag and Text attributes, ready for text processing and model building.

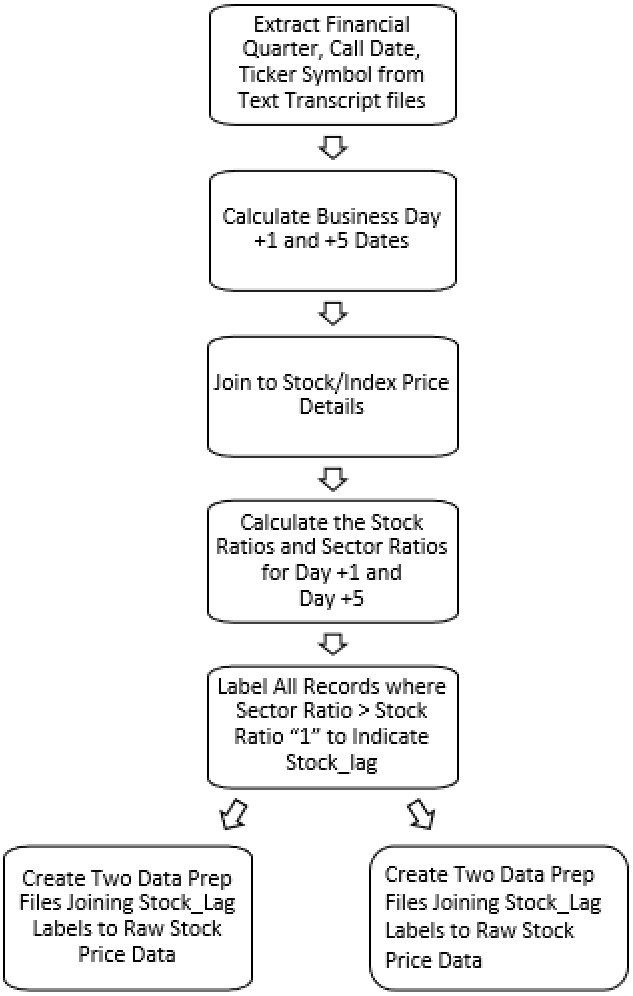

Extract Financial Quarter, Call Date, Ticker Symbol from Text Transcript files
Calculate Business Day +1 and +5 Dates
Join to Stock/Index Price Details
Calculate the Stock Ratios and Sector Ratios for Day +1 and Day +5
Label All Records where Sector Ratio > Stock Ratio "1" to Indicate Stock_lag
Create Two Data Prep Files Joining Stock_Lag Labels to Raw Stock Price Data
Create Two Data Prep Files Joining Stock_Lag Labels to Raw Stock Price Data

Figure 2: Transcript Data Pre-processing Flow

The pre-processed files became the input data for the NLP text-based models for Day +1 and Day +5 predictive analysis. After the data was uploaded to a Juptyer notebook, the data was split into training (80%) and test (20%) sets and saved as separate files for the training and test dependent variables (Stock_lag labels) and independent variables (text inputs). Splitting the data once and using the same training and testing sets allowed the best comparison across models by removing the chance that some splits would more naturally generate more or less advantageous results than other random data splits.

Because the stock data model (the non-text-based control comparison analysis) did not use transcript text as an input, an Alteryx workflow joined the Stock_lag labels directly to the stock details, and no further pre-preparation was required to ingest the data for that model. The stock data details for Day +1 and Day +5 were each split into training (80%) and test (20%) data sets for ingestion by a Random Forest model for each scenario. Only one predictive model was built for each, so no files were exported for reuse in these cases.

To complete the preparation of the stock price data, one-hot-encoding was used to capture the company ticker symbol (AAPL, GOOGL, etc). Encoding the company into the non-text data set gives the model the benefit of potentially identifying trends or patterns specific to the company without including any text-based input. The open, high, low, close, adjusted close, and volume metrics were all standardized using the sklearn.pre-processing MinMaxScaler function, to fit those values between 0 and 1 (Pedregosa, F. et al., 2011). Standardizing the variables mitigates the risk that natural variability among the companies' stock prices and volumes do not skew the results of the prediction model.

Word2Vec and Doc2Vec Text Preparation

After the initial text pre-processing and labeling, the data was ready to be ingested for the Word2Vec and Doc2Vec data processing functions. In a Jupyter notebook, the transcript records were processed using NLTK model functions (Bird, Klein, & Loper, 2009) to clean the data, remove punctuation, and remove standard stop words. The list of standard NLTK stop words was amended with the name of each company to mitigate potential bias related to the repetition of the company names in the transcripts. No other potential identifiers were removed, like names of executives, key products, or competitors. A regular expression removed numbers from the transcripts. The cleaning process results in a list of words for each record.

There is an additional step that applies a stemming or lemmatizing function to reduce the total number of words included in the analysis. These functions combine the same words that appear in different forms (“earnings” and “earning” are transformed to “earning”, for example) based on pre-programmed logic. However, some experts opine that lemmatizing and stemming may degrade results where “highly-inflectional languages” are involved (Hassani, n.d.). Because analyst calls have nuances particular to themselves and considering the relatively small sample size of texts included in the present study, the author opted to leave out lemmatizing for this analysis. Future studies that include optimizing text-based models may consider comparing results with and without lemmatizing, the more sophisticated of the two methods.

The cleansed data was vectorized using the appropriate function for each text-based model (described below).

Text Vectorization

Each text-based model has its own vectorization process, which reflects how it converts words into numeric vectors, based on their context. In the simplest “bag of words” NLP models, vectors are created by assigning a word from the corpus (the entire body of text included in the analysis) to a column and increasing that number by one for each additional instance it appears in the text (Muller and Guido, 2017). The models selected for this study apply more sophisticated approaches that assimilate the context of each word to assign it a calculated score.

TF-IDF (Term Frequency - Inverse Document Frequency)

The TF-IDF vectorizer considers the number of times that a word appears in the entire corpus and the number of times the word appears in each document that makes up the corpus. Words that are very common across the corpus will have a lower rating than words that are unique to a specific document, for example (Scott, 2019).

Word2Vec

Word2Vec is a collection of technologies designed to learn word embeddings. It uses a shallow neural network to create word embeddings (TensorFlow Core, 2021b). The result, according to the Gensim Word2Vec documentation, creates a mathematical representation of words given the context in which they appear in the corpus. This vectorized representation can be used to compare the meaning of words, using the generated vectors to quantifiably compare the difference and similarity between words based on their context within the corpus. For example, the Word2Vec Embeddings page on the *Gensim Topic Modeling for Humans* website explains:

> A set of word-vectors where vectors close together in vector space have similar meanings based on context, and word-vectors distant to each other have differing meanings. For example, “strong” and “powerful” would be close together and “strong” and “Paris” would be relatively far. (Section 3)

Doc2Vec

Gensim documentation compares Doc2Vec with Word2Vec, except the Doc2Vec algorithm has another layer of embedding specific to each document (Gensim Topic Modelling for Humans, 2021a). The end result works better in some situations, assuming the specific documents have context that adds value above and beyond the embeddings generated by the entire corpus in the Word2Vec approach.

Recurring Neural Network (RNN)

This analysis uses Keras on top of TensorFlow to build an RNN deep learning model. Keras has its own tokenization function that cleans and tokenizes or vectorizes the text data. According to the TensorFlow documentation, the tokenization process in Keras does not include the complex context embedding performed by the Word2Vec and Doc2Vec utilities (TensorFlow Core, 2021a). This aligns with the fact that the data in this scenario will feed into a neural network that may determine its own context based on how it processes the text.

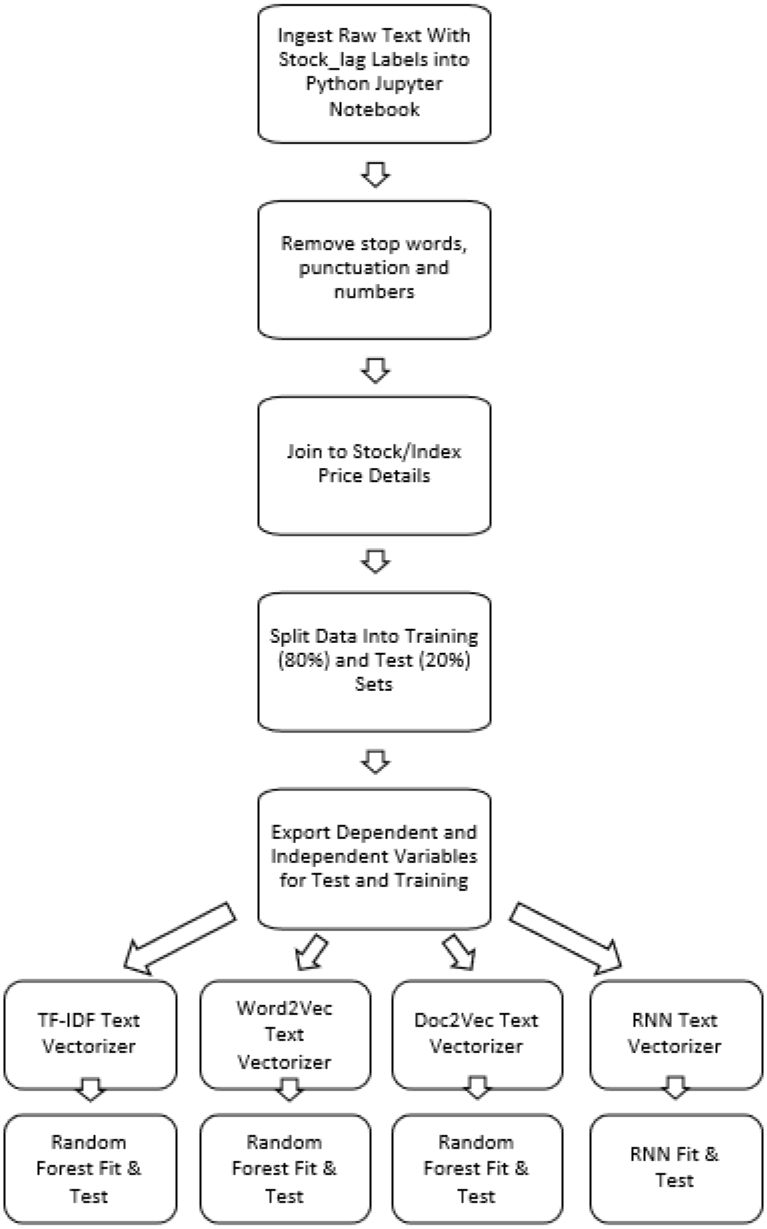

Ingest Raw Text With Stock_lag Labels into Python Jupyter Notebook
Remove stop words, punctuation and numbers
Join to Stock/Index Price Details
Split Data Into Training (80%) and Test (20%) Sets
Export Dependent and Independent Variables for Test and Training
TF-IDF Text Vectorizer
Word2Vec Text Vectorizer
Doc2Vec Text Vectorizer
RNN Text Vectorizer
Random Forest Fit & Test
Random Forest Fit & Test
Random Forest Fit & Test
RNN Fit & Test

Figure 3: Standard NLP Data Processing Flow

Sentiment Analysis

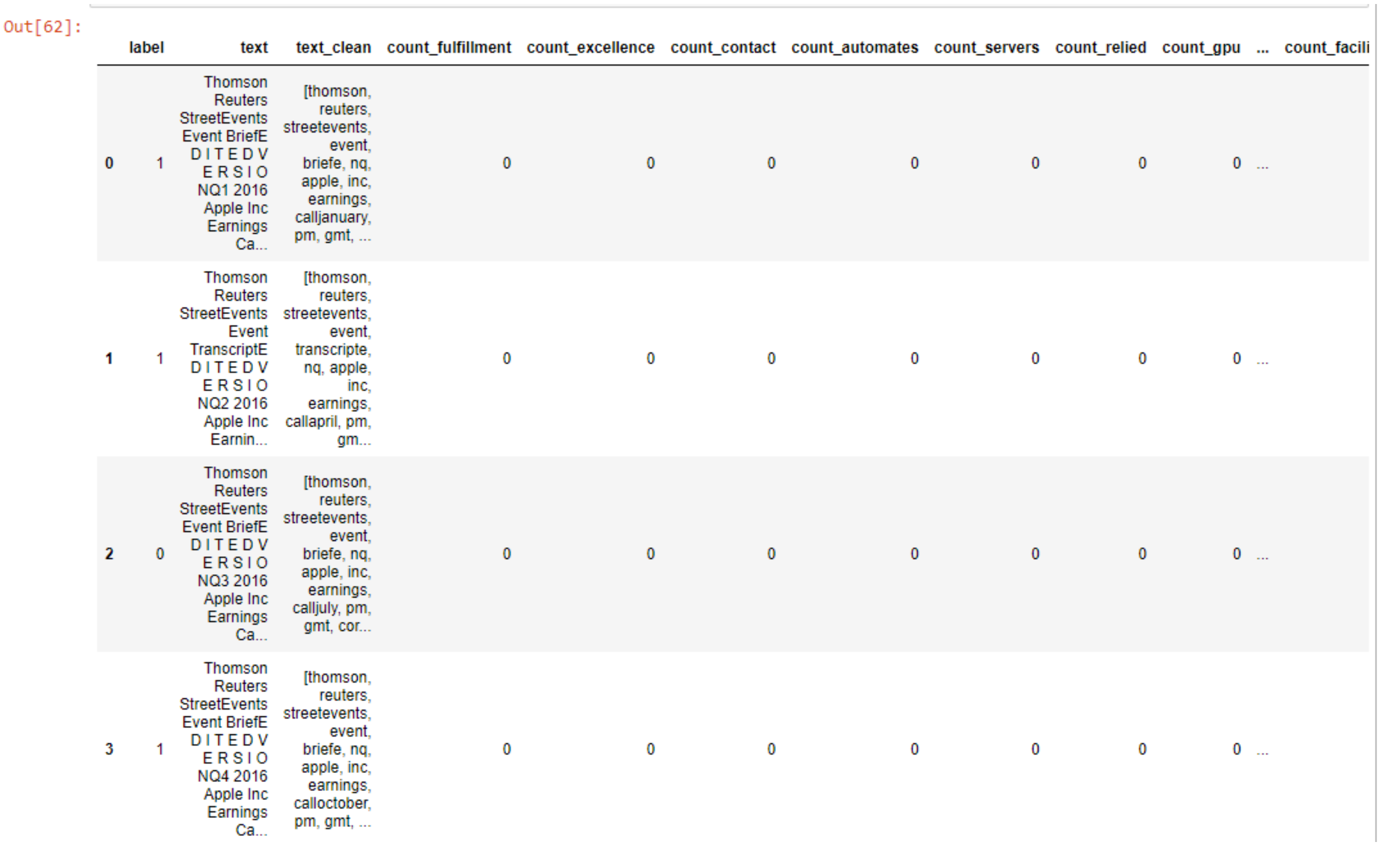
A training subset of data was used to generate a list of words with positive sentiment associations to the word “data.” A subset was used to avoid the potential to generate a representative sample that could be used on both training and test data. The author used the “most_similar()” function within Word2Vec to create a matrix of the top 25 words in the corpus that have a positive sentiment relationship with the word “data.” The most_similar() method “computes cosine similarity between a simple mean of the projection weight vectors of the given keys and the vectors for each key in the model” (Gensim Topic Modeling for Humans, 2021b). In other words, this method compares the vector attributes of the target word, in this case “data,” and finds the top words that generated the most similar vectors. Table 3, below, shows the words in the corpus of earnings call transcripts that generated vectors most similar to the vector generated by the word “data:”

Table 3: Most similar words to “data”

Similar Word from Corpus	Word2Vec Similarity Score
'fulfillment'	0.830
'excellence'	0.639
'pittsburgh'	0.601
'uc'	0.540
'contact'	0.540
'computing'	0.536
'ambient'	0.532
'supercomputing'	0.505
'collaborative'	0.496
'sortation'	0.495
'imec'	0.486
'servers'	0.483
'pc'	0.474
'fractionalize'	0.466
'dna'	0.460
'gpu'	0.452
'cloud'	0.449
'client'	0.448
'switching'	0.436

'hannover'	0.432
'facilities'	0.431
'space'	0.429
'graphics'	0.423
'nvlink'	0.421
'hyperscale'	0.416

A python script counted the instances of each of these new target words for every record in the corpus. The count of target words from the list above generated a new set of features to use for predictive sentiment modeling. Note that because the pre-processed Day +1 or Day +5 files were used for this analysis, the Stock_lag labels, raw text, and cleaned text also appear in the file (Figure 4).



Out[62]:

	label	text	text_clean	count_fulfillment	count_excellence	count_contact	count_automates	count_servers	count_relied	count_gpu	...	count_facili
0	1	Thomson Reuters StreetEvents Event BriefEDITED VERSIONQ1 2016 Apple Inc Earnings Ca...	[thomson, reuters, streetevents, event, briefe, nq, apple, inc, earnings, calljanuary, pm, gmt, ...	0	0	0	0	0	0	0	...	
1	1	Thomson Reuters StreetEvents Event TranscriptEDITED VERSIONQ2 2016 Apple Inc Earnin...	[thomson, reuters, streetevents, event, transcripte, nq, apple, inc, earnings, callapril, pm, gm...	0	0	0	0	0	0	0	...	
2	0	Thomson Reuters StreetEvents Event BriefEDITED VERSIONQ3 2016 Apple Inc Earnings Ca...	[thomson, reuters, streetevents, event, briefe, nq, apple, inc, earnings, calljuly, pm, gmt, cor...	0	0	0	0	0	0	0	...	
3	1	Thomson Reuters StreetEvents Event BriefEDITED VERSIONQ4 2016 Apple Inc Earnings Ca...	[thomson, reuters, streetevents, event, briefe, nq, apple, inc, earnings, calloctober, pm, gmt, ...	0	0	0	0	0	0	0	...	

Figure 4: Counting Words with Positive Associations to "Data" in Each Transcript

Removing the text columns from the file created a numeric matrix that represents positive word associations with the word “data” from each record in the Day +1 and Day +2 data sets.

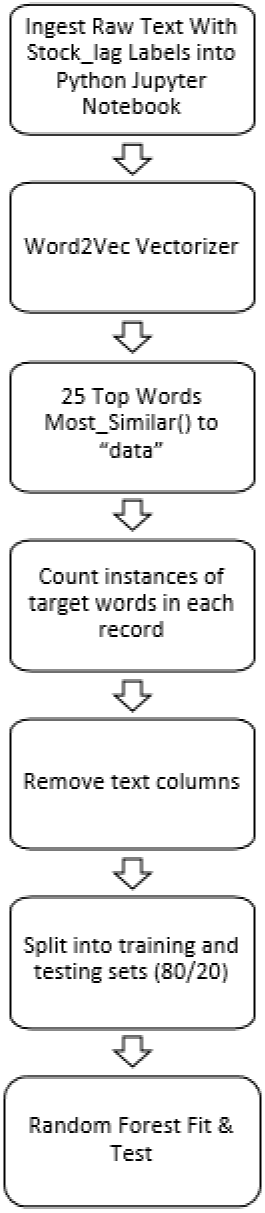

Ingest Raw Text With Stock_lag Labels into Python Jupyter Notebook
Word2Vec Vectorizer
25 Top Words Most_Similar() to “data”
Count instances of target words in each record
Remove text columns
Split into training and testing sets (80/20)
Random Forest Fit & Test

Figure 5: 'Data' Word Sentiment Process Flow

Dependent Variable Balance

The balance between the Stock_lag positive and negative records were fairly balanced in each model, so no extra steps were needed to balance these out. The prepared and split training data for Day +5, for example, showed 73 records that exceeded the NASDAQ performance, and 77 that lagged. The training data for Day +1 contained 80 records that outperformed the NASDAQ index, and 70 that lagged.

Machine Learning Models

The prepared and vectorized data for Day +1 and Day +5 preparation for TF-IDF, Word2Vec, Doc2Vec, and Word Sentiment analysis was fit to a Random Forest classification model. The precision, recall, and accuracy was calculated for each model's ability to predict the Stock_lag label in the withheld test data set.

For the RNN model, a framework was constructed of an embedding layer, an LSTM layer, and two dense layers. 11 epochs were run to generate and analyze the predictions against the withheld test dataset. A screen shot of the RNN model configuration appears in Appendix A of this paper.

The data set for the non-text data was also fit against a Random Forest Classifier. The intention of using the same Random Forest model for each (with the exception of the RNN model) is to attempt to compare the data inputs to each other, to evaluate the insights that can be gleaned from each compared to the others. Because this is an exploratory analysis, future analysis may take the most promising approaches and optimize these using a variety of models and techniques.

Each model generates predictions regarding whether a company's stock is likely to decline, relative to the NASDAQ, on the first or fifth business day following the earnings call.

This insight could help investors determine when to make stock trading decisions, especially stock selling decisions, in light of recent research related to these activities. According to the Planet Money newsletter, the average hedge fund manager makes worse decisions related to when to sell stocks than random chance (Rosalsky, 2021), so this insight is very germane to investment managers.

The model fits the data from the training data set, then tests its prediction against the reserved test set. A formal, optimized, and tuned data set would require the addition of a validation data set, but this initial exploration intends to determine whether collecting the vast amounts of data required to support a more robust analysis are worthwhile. A much larger set of data would be required to support separate training, validation, and testing splits.

As Hsu et al. point out, models do not need to approach 100% accuracy to have value in the real world (2016). That team also points out that because of the innumerable factors contributing to stock price movements, such high accuracy is not realistic. The same team also indicated that an accuracy of about 60% in stock movement predictions can result in profitable trading strategy (Hsu et al., 2016). Additionally, because investors are motivated by loss aversion (Ryan, 2017), it is reasonable to focus on identifying stocks that will decline in value and avoid predictions that a stock will increase in value when its actual performance decreases. For these reasons, the analysis focuses on the "recall" measure, because it will be an important factor to achieve the lowest possible number of false negatives. Accuracy is a helpful measure to compare the results of these models to existing academic literature related to machine learning and stock movement predictions. The following graphic (Figure 7) helps provide context for the precision and recall measures (Riggio, 2019), indicating that precision focuses on how many "true" predictions were true in reality (i.e., the model predicted a stock would decrease in value and it did), and indicating that recall concerns false negatives (i.e., the model predicted a stock's value would increase, but it lagged instead).

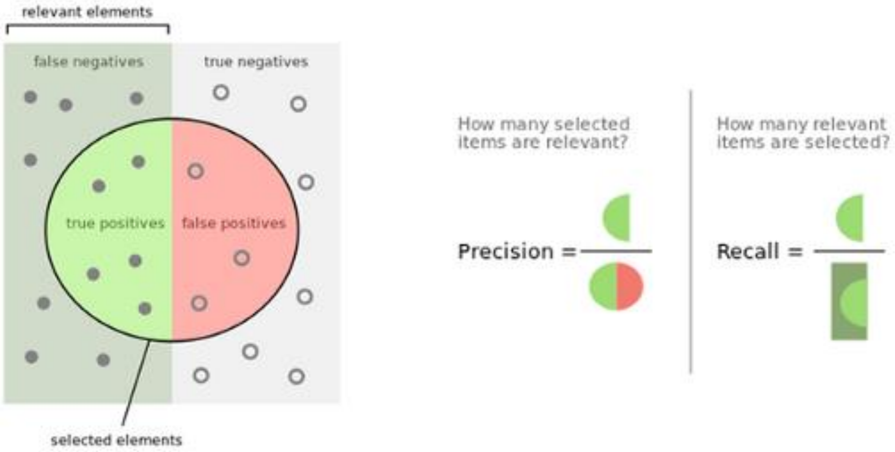


Figure 6: Accuracy, precision, and recall (Riggo, 2019)

Findings

The tables below illustrate the outcomes of each of the data source and forecast horizon combinations:

Table 4: Stock Performance Machine Learning Predictions

Data preparation and Modeling		**Day +1 Forecast Horizon**			**Day +5 Forecast Horizon**		
DV Prep	**ML Model**	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**	**Accuracy**
TF-IDF	**Random forest**	0.267	0.308	0.474	0.464	0.684	0.447
Word2Vec	**Random Forest**	0.316	0.462	0.474	0.550	0.579	0.553
Doc2Vec	**Random Forest**	0.312	0.385	0.500	0.368	0.368	0.368
RNN (11 epochs)	**RNN**	0.250	0.350	0.605	0.719	0.742	0.711
"Data" Word Sentiment	**Random Forest**	0.571	0.600	0.553	0.647	0.550	0.605
Stock Price Details	**Random Forest**	0.467	0.350	0.447	0.409	0.643	0.526

The RNN model outperforms the other models in this Day +5 dataset by a wide margin. Considering the fact that the RNN model only ingests text data, it is very noteworthy that it outperformed the stock price data source in recall, precision, and accuracy. The TF-IDF model also outperforms the stock price detail data in recall and precision. The "Data" Word Sentiment modeling outperforms both the Word2Vec and Doc2Vec models on every measure, indicating that this novel approach has potential to identify industry-specific sentiments that may help predict stock performance over a 5-day forecast horizon. While the Word2Vec and Doc2Vec models show little promise of enhancing stock performance prediction capabilities over a lucky guess, the performance of the other text-based approaches indicates valuable potential to generate powerful predictive insights based on NLP text analysis.

The Day +1 results are most striking in how poorly they perform compared to the Day +5 results. The other striking feature of these results is the high performance of the “Data” Word Sentiment modeling results, especially related to recall. The fact that it outperforms all the other data preparation techniques seems to indicate that it has some potential value in this methodology. On the other hand, it may also reflect issues related to the small sample size.

The results of the modeling activities led the author to the following observations regarding the study hypotheses:

Table 5: Hypotheses Analysis

H_1: A RNN natural language deep learning neural network will outperform more traditional NLP machine learning models in accurately predicting stock performance relative to NASDAQ index.	Accept. The RNN model outperformed all other approaches by every measure in the Day +5 outcomes. It also achieved the highest accuracy of all approaches in the Day +1 outcomes, though it had a relatively low recall score.
H_2: Extracting words with positive sentiment associations to the word “data” in the earnings call transcripts and calculating features to increase the weight of those words will improve the performance of the text-based prediction.	QUALIFIED HOLD. This model demonstrated promise, especially in the Day +1 performance. It did not perform as well in the Day +5 performance, though it outperformed the TF-IDF, Word2Vec and Doc2Vec models.
H_3: The stock price data (non-text-based) Random Forest model will perform at least as well as the text-based models on the data.	Reject. The stock price data lower recall in the Day +5 set compared to TF-IDF and RNN approaches. While it had better accuracy than most models in Day +5, it still underperformed the RNN approach in this measure. In the Day +1 data set, it did not have better than average outcomes.
H_4: Text-based prediction models will generate better precision, recall, and accuracy using the Day +5 forecast horizon than the Day +1 forecast horizon.	Accept. The average scores for each measure improved for the text-based models in the Day +5 predictions compared to the Day +1 predictions, as shown below. This indicates that the text-based models may work by accelerating insights that are more slowly disseminated by traditional means.

Conclusion

The findings indicate that text-based machine learning models have the potential to improve the predictive capabilities of machine learning models to predict stock performance at the five-day forecast horizon.

These results indicate that the time and expense to collect data from SEC filings and analyst calls and other sources published by companies themselves could achieve improvements over existing methods. At the same time, it is important to consider the fact that the sample of companies is very small (10), and all relate to the high-tech industry. Although these results are very promising, the author hastens to note that a larger data population and formal tuning and validation would need to be performed before implementing this or any other machine learning model in a real-live market.

Ethics and Social Responsibility

All the data described in this study is publicly available, so there is no obvious ethical concern about the collection or analysis of the data. Regarding social responsibility, any reader needs to be aware that any stock trading should be undertaken in the context of an informed stock management portfolio that considers risk, and whose participants understand the risk related to losing their invested capital. The author emphasizes again, these results are exploratory and should not be construed as investment advice, especially considering the small sample size and inherent bias in the data.

Bias

The curated data used in this analysis focused on a very small number of high-tech companies over the years 2016 - 2020. This is a very small sample size, especially compared to other machine learning models that consider the daily movement of stock prices for a larger number of companies. This analysis should be considered exploratory and in no way definitive of the strength of NLP as a stock movement predictor compared to other machine learning approaches.

Future Analysis

In addition to using data sets representing texts from a wider variety of companies, more text data sources could also be incorporated into future models. For example, data from SEC reports or news items would be interesting to assess and based on these preliminary results are probably worth exploring. Further experimentation related to industry-specific sentiment analysis is also warranted, and domain experts in various industries could be consulted to test different words for their performance. Similarly, negative word sentiment analysis could be examined to see if negatively correlated words predict lagging stock performance.

There are virtually countless ways in which machine learning models could be further explored, including reproducing the SVM and ANN models described by Hsu et al. in their machine learning survey (2016). Further tuning and optimization of various steps would also be required for mature model development, including testing the use of lemmatization in text preparation. The performance of the NLP data compared to stock price data, which has already shown high levels of performance when optimized in machine learning models, indicates that thorough exploration of this area may pay dividends.

Clearly, there is fertile ground for building robust machine learning predictions combining both historical data stock price data with NLP analysis. Optimizing a real-world trading strategy would almost certainly require a sophisticated blend of approaches of financial and text data sources.

References

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit.* O'Reilly Media, Inc.

Fortado, L., & Johnson, M. (2017, Febrary). Computer-driven hedge funds join industry top performers. *Financial Times,* 13.

Gensim Topic Modelling for Humans. (n.d.) *Doc2Vec Model*. Retrieved August 8, 2021 (a) from https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html.

Gensim Topic Modelling for Humans. (n.d.) *Word2Vec Embeddings*. Retrieved on August 8, 2021(b) from https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html.

Hsu, M., Lessmann, S., Sung, M., Ma, T., & Johnson, J. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications,* 61, 215 – 234, 2016, https://doi.org/10.1016/j.eswa.2016.05.033

Hassani, H. (n.d.) *Is it normal to get better accuracy without stemming and lemmatization than using them in NLP text classification?* Quora.com. Retrieved on August 8, 2021 from https://www.quora.com/Is-it-normal-to-get-better-accuracy-without-stemming-and-lemmatization-than-using-them-in-NLP-text-classification.

Iansiti, M., & Lakhani, K. (2020). *Competing in the Age of AI: Strategy and Leadership When Algorithms and Networks Run the World*. Harvard Business Review Press, Boston, MA.

Jansen, S. (2020). *Machine Learning for Algorithmic Trading*. Packt.

Lv, D., Yuan, S., Li, M., & Xiang, Y. (2019). An empirical study of machine learning algorithms for stock daily trading strategy. *Mathematical Problems in Engineering* v 2019. https: //doi.org/10.11555/2019/7816154.

Malkiel, B. (2003). The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, 17(1), 59-82.

Metz, C. (Jan. 25, 2016). The Rise of the Artificially Intelligent Hedge Fund. *Wired*. https://www.wired.com/2016/01/the-rise-of-the-artificially-intelligent-hedge-fund/#comments (accessed July 18, 2021).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research,* 12 (Oct), 2825–2830.

Riggio, C. (2019, November 1). "What's the deal with accuracy, precision, recall, and F1?" *Towards Data Science*. Retrieved on August 8, 2021 from https://towardsdatascience.com/whats-the-deal-with-accuracy-precision-recall-and-f1-f5d8b4db1021

Roozen, D., & Lelli, F. (2021, February 18). *Stock Values and Earnings Call Transcripts: a Dataset Suitable for Sentiment Analysis*. Preprints. (doi: 10.20944/preprints202102.0424.v1).

Rosalsky, G. (2021, August 3). *There's a way you can beat the best investors. You've just got to know when to sell.* Planet Money. Retrieved on August 4, 2021 from https://www.npr.org/sections/money/2021/08/03/1022840229/why-even-the-most-elite-investors-do-dumb-things-when-investing.

Ryan, P. (2017, December 4). *Stock Market Predictions with Natural Language Deep Learning.* Microsoft.com. https://devblogs.microsoft.com/cse/2017/12/04/predicting-stock-performance-deep-learning/ (Retrieved on July 18, 2021).

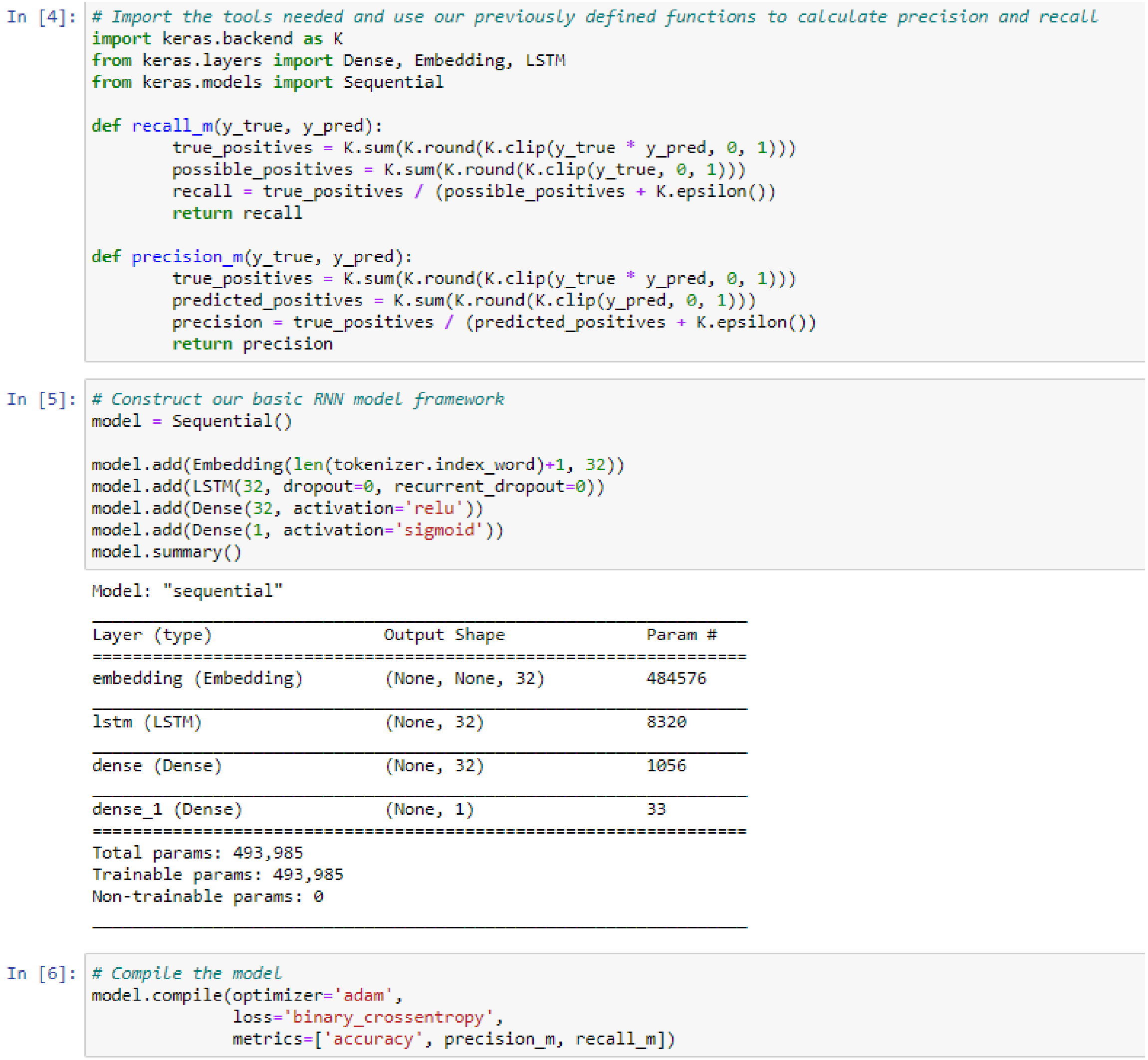
Scott, W. (2019, February 15). *TF-IDF from scratch in python on real world dataset.* Towards Data Science. Retrieved on August 8, 2021 from https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089.

TensorFlow Core. (n.d.) *tf.keras.preprocessing.text.Tokenizer*. Retrieved on August 8, 2021 (a) from https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer.

TensorFlow Core. (n.d.) *Word2Vec Tutorial*. Retrieved on August 8, 2021 (b) from https://www.tensorflow.org/tutorials/text/word2vec

Appendix A

RNN Model Configuration Screen Shot



```
In [4]: # Import the tools needed and use our previously defined functions to calculate precision and recall
        import keras.backend as K
        from keras.layers import Dense, Embedding, LSTM
        from keras.models import Sequential

        def recall_m(y_true, y_pred):
                true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
                possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
                recall = true_positives / (possible_positives + K.epsilon())
                return recall

        def precision_m(y_true, y_pred):
                true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
                predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
                precision = true_positives / (predicted_positives + K.epsilon())
                return precision
```

```
In [5]: # Construct our basic RNN model framework
        model = Sequential()

        model.add(Embedding(len(tokenizer.index_word)+1, 32))
        model.add(LSTM(32, dropout=0, recurrent_dropout=0))
        model.add(Dense(32, activation='relu'))
        model.add(Dense(1, activation='sigmoid'))
        model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 32)          484576
_________________________________________________________________
lstm (LSTM)                  (None, 32)                8320
_________________________________________________________________
dense (Dense)                (None, 32)                1056
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 493,985
Trainable params: 493,985
Non-trainable params: 0
_________________________________________________________________
```

```
In [6]: # Compile the model
        model.compile(optimizer='adam',
                      loss='binary_crossentropy',
                      metrics=['accuracy', precision_m, recall_m])
```

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA