**Section 1: Case Study Report**

**What does the analysis/model building tell you?**  
The model is contingent upon identifying key text within the earnings call, in order to be given a sentiment. This sentiment would suggest a trading signal is expected shortly after the earning call. The classification report suggests the key selection of the transcript I initially identified coupled with the price action reviewed after the earnings call, would suggest those key words tend to tap into a more ‘positive’ sentiment earnings call.

**What are your recommendations?**

Not having to use the entire transcript, my recommendation would be to run the script after the earnings call. Currently, there is a 86% accuracy rate for the key words used, in which we were able to identify a positive trading signal – that would suggest buying into the company after earnings which would produce a positive % improvement. Or, if not providing the purchase yourself, providing a positive signal recommendation to others. This also reinforces that the introduction and exit of the earnings call may not produce as much important signals, but the middle of the call specifically within 3,000 – 3,500 word counts, may strategically hold the critical forward looking view.

**How would you pitch this business problem to a group of stakeholders to gain buy-in to proceed?**

Big banks and financial institutions have analysts monitor hundreds of companies, and reviewing every earnings call can be time-consuming and challenging. Instead of hiring new analysts, we can develop a script that mines earnings transcripts in order to provide a cliff-notes version of the call and/or to provide insights based on proprietary knowledge that the institution of bank carries from their financial experience.

**Why should someone in the business care about this solution?**

Analysts can spend more time finding the *next* best company or spend time on understanding what determines a ‘good’ company. The script would be able to not just mine current transcripts, but any. This provides a level of detail that would be unparalleled, as you’d be able to review years, if not decades, worth of transcripts and gleam insights that could impact trading signals. The way this could be done is when you start to investigate the 500 or so words, you can build your own library or even various libraries contingent on industry and/or companies that would continuously evolve as new CEO’s emerge.

**What are some of the potential challenges or additional opportunities that need to be explored?**

On a student/academic scale, there were challenges connecting to the motleyfool, and procuring the transcripts. Often, the request would time out, or the API I was using would not be able to find certain tickers. Of course, in a professional setting, transcripts would be easier to obtain, if not, the connection speeds would be substantially better and access to API’s to pull pricing data would be more reliable.

The biggest opportunity next would be to review the 500-1000 words within the transcript and separate the most common words between the positive and negative deltas. From there, one would be able to generate their own sentiment library, or at least words and terms that tend to suggest one way or the other. By continuing to pull transcripts, one can develop a model that is continuously learning – and we have the actual price action of what is occurring to act as a validation point.

**Section 2:**

**Milestone 1**

People look for edges when it comes to stock trading all the time. With the advent of ML, there are new areas that we can mine data in order to determine and edge in our understanding of wallstreet, and how to creep more consistent profits. That idea is to mine the earning transcripts that are released after the earnings call in order to gleam any type of information that may suggest if a stock may have a positive signal, after the earnings.

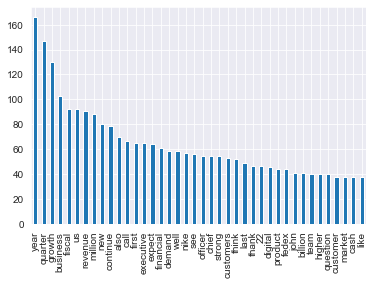
Milestone 1 was the test, and was not done automatically. The transcripts were manually pulled, and the stock price ‘feature’ had not been identified through the use of the API yet. Many of these features were not included in the final milestone, as the project evolved from looking for sentiment to identifying if there event exists a relationship between the text and price difference. In addition, we also discovered perform sentiment on the entire transcript, despite using the stop words library is not enough – we have to shrink the text down and further break down the stop words.

Bar Chart 1: The bar chart represents common words that exist within the original transcripts reviewed.

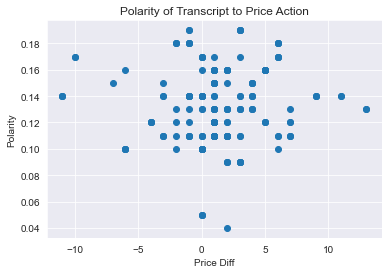
Bar Chart 2: The multiple bar charts indicate transcript length

Scatter Plot 1: Polarity of Transcript to Price Action

Scatter Plot 2: Transcript Length to Polarity

Chart, bar chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Word Cloud Charts: Visualize the most common words for the initial tickers

The word clouds definitely, in the beginning, were suggestive that something *could* exist. Meaning, there could exist a relationship that suggests or could hint the way the price would move after the call.



A picture containing logo

Description automatically generated Text

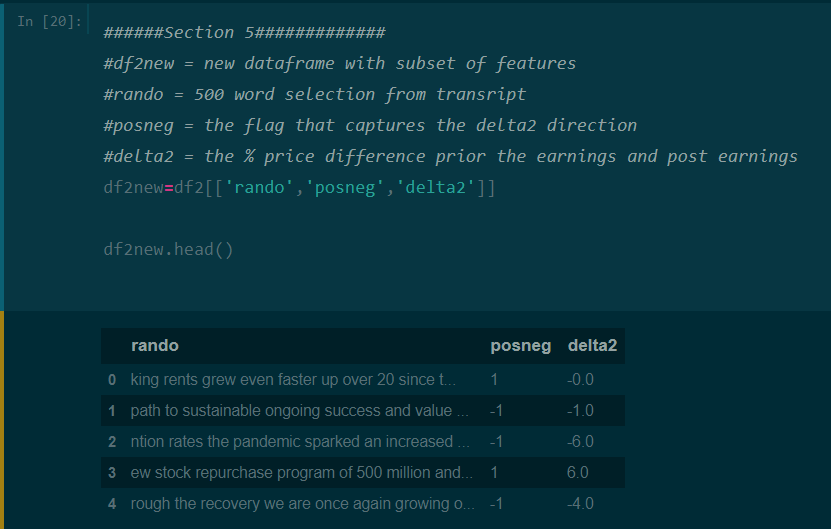
Description automatically generated with medium confidence

**Milestone 2:** Dimensionality & Feature Reduction and Feature Engineering

In the final model that was produced, I no longer used textblob and the sentiment analysis existing libraries. The polarity and subjectivity was too generic, and the original transcript was too large and contained too many unnecessary words, even with removing stopwords, and adding to the stop word library.

I performed data-wrangling, and reduced the final model into 3 features. The short selection of the transcript in the ‘middle’, the delta of the price action (% diff) prior/after the call, and a positive/negative flag to act as a signal for the % diff if it was positive or negative.

For the transcript itself, I used the stop words library in order to eliminate as many wasteful words as possible, and then added many to the list, such as Operator","quarter","analyst","welcome","president","year","earnings","earning","call","vice","chief","financial","officer","conference"



**Milestone 3:** Model Selection & Evaluation

For my model, I used logistic regression. Logistic regression is a classification algorithm which is how I structured my dataset. We had a random set of text (from the earnings call) coupled with the price difference. We were trying to determine if the sub-string of earnings call would signal a positive or negative flag. In that regard, our dependent variable is binary (0/1 – negative or positive)

**Classification Table:**

precision recall f1-score support

-1 0.62 0.89 0.73 9

1 0.97 0.85 0.90 33

accuracy 0.86 42

macro avg 0.79 0.87 0.82 42

weighted avg 0.89 0.86 0.87 42

**Conclusion:**

In conclusion, the 500 or so words that were selected are indicative of how the ticker will move shortly after the earnings call. This is to suggest the meat of the earnings call, exists well after the introduction and well before the analysts q&a.

precision recall f1-score support

-1 0.62 0.89 0.73 9

1 0.97 0.85 0.90 33

accuracy 0.86 42

macro avg 0.79 0.87 0.82 42

weighted avg 0.89 0.86 0.87 42

Trying to use existing polarity and subjectivity sentiment tools didn’t act as the trigger. Instead, performing analysis on just selecting a sub-string of the transcript, and then coupling that small blurb with how the stock performed acted as a much more powerful and suggestive measure, than using existing sentiment tools. For more complex and nuanced analysis, since it’s been determined, with the classification matrix there exists some relationship one would need to data-mine the transcripts even further, especially in the key 3,000 – 3,500 areas in order to assess what are some of the suggestive words and vocabulary used.

Its been understood that earnings calls could be devastating, if bad news is delivered poorly. Much work goes into how earnings calls are delivered to wall street, so it’s rather fascinating that there exists 500 or so words within an earnings call that tend to suggest more than what a common listener may suggest. Further analysis should be conducted on the 500 or so words, and what words are they exactly. Instead of trying to use an existing library, one should look at tickers that perform well after the earnings and then data-mine and extract the most common words within the 500 – 1000 word count in each call. By doing so, instead of use textblob, one can generate their own sentiment counter that evolves from each call.

For someone in the industry, such as banking, in which they perform these reviews and analysis on companies they cover and track – it’s another tool in the analysts chest to try and determine nuances within the officer suite, and how they communicate good/bad/neutral news to wallstreet. It’s going to be developing these nuanced methods and ways that help the analysts and/or investment firm gain an edge, however small, to help drive profits.