Earnings Report Project - Final Report

Mary Ardoin, Ngoc Le, Laney Rzeppa

Development Process

Initial Development:

For the initial development of the project, we knew that we wanted to use a sentiment analysis. We felt that this would produce the best results as there could be personal bias or error with a lexicon approach. Additionally, we used a lexicon approach in the practice project and wanted to try something different this time. The sentiment analysis would provide insight into if the text has a negative or positive tone. This would be the biggest factor of the project and then we wanted to add additional factors so it could be more accurate.

Our group wanted to include the fog index of the MD&A because we believe that the difficulty in reading the document will have a correlation with the earnings outcome. Similar to the article, "Annual report readability, current earnings, and earnings persistence," our thought process is that the higher the fog index, the greater the chance of a negative market reaction.

Other ideas that we explored were using previous market reaction numbers as an indicator of the future. We were not sure how exactly we could do this, but the thought process included looking at the market from previous years in terms of how it was reacting before the reports came out.

We used the article "Examining the Effect of Linguistic Style in an MD&A on Stock Market Reaction" as an academic source for our approach. The study in the article did not find that linguistic style has a significant impact on market reaction. We kept this in mind when building our model and its coefficients.

Adjustments:

After brainstorming ideas for the initial development, we realized that most of these ideas could be implemented into the actual project. We developed code that included factoring in sentiment score, fog index, prior 5-day market performance, prior 5-day firm performance, prior year market cap, py/pe ratio, and number of tokens. Initially, the sentiment score and fog index held equal weights, but based on the research from the article and our own testing, we ultimately made the sentiment score count significantly more than the fog index. Additionally, after testing the code, we realized that the prior year market cap, py/pe ratio, and number of tokens did not affect the outcome enough to be included in the final coding model. We changed their coefficients to 0. Prior 5-day market performance and prior 5-day firm performance had more of a significant effect on the performance of our model, so we increased their coefficients a lot to reflect this.

The Final Model

Code Overview:

The primary code consists of a Python script using the Natural Language Toolkit (NLTK) library and the VADER sentiment analysis tool. The script is organized into several functions, the first being the *predict_return* function. In total and in order it initializes a score to predict the stock return, calculates the sentiment score for the entire document using VADER, applies a word tokenizer to the document, and performs feature engineering by combining sentiment score (fog index, prior 5-day market performance, prior 5-day firm performance, prior year market cap, prior year EPS, and the number of tokens), weights the features and sums them up to get the final score, and scales the score and keeps it within an appropriate range. These features were chosen based on Tailab and Marshall's research combined with our efforts.

The next chunk of code consists of auxiliary or supplemental functions. This includes *sentiment* (uses VADER to calculate sentiment scores), *syl* (counts syllables in a word), *word_tokenize* (tokenizes words in a text), *sentence_tokenize* (tokenizes sentences in a text), and fog (computes the Fog Index, a measure of textual complexity). We chose these methods based on Li's research which covered the complexity and readability of MD&As, primarily the importance of FOG.

The final function is simply a test function that collects and runs our data for us. In total, it tests the model, reads financial data, gets the predicted return, calculates an ROI based on predicted and actual returns, and aggregates the results for all firm-years. This function simply pulls together all the other functions written to do the big job we ask it to.

Feature Selection and Coefficient Tuning:

The predictive model employs a weighted sum of features, and coefficients are adjusted to optimize the model's performance. The current coefficients have been manually set, and the script includes a commented-out section for potentially adding a feature related to the highest Fog Index of a sentence.

Forward-Looking Predictions:

We expect the model to perform well with the additional unknown MD&As. Our group performed many tests to see how each aspect of the model affected the outcome. Our extensive testing makes us confident that the final numbers chosen for the model will perform well with all of the additional tests and continue to yield positive returns. Additionally, we believe that the model will perform well with all types of companies. If we had to choose, we think that the model might do better with bigger/more established companies. Their stock is usually pretty stable and with the way that the model is returning either -1 or 1 in prediction, it should work pretty well with those companies.

Sources

Li, Feng. “Annual report readability, current earnings, and earnings persistence.” *Journal of Accounting and Economics*, vol. 45, no. 2–3, 4 Mar. 2008, pp. 221–247, https://doi.org/10.1016/j.jacceco.2008.02.003.

Tailab, Mohamed M., and Marshall J. Burak. “Examining the effect of linguistic style in an MD&A on stock market reaction.” *International Journal of Business Communication*, vol. 58, no. 3, 2018, pp. 430–458, https://doi.org/10.1177/2329488418762293.