**Capstone: Economic Events**

Stephen Sigrist, Jorge de Leon Miranda and Richard Zhai

Abstract: Our project uses data science techniques to build up models for predicting the abnormal returns of publicly traded stocks using financial and economic news. We use Python to collect historical financial data and corresponding news and event data from a variety of sources, including SEC filings, Google Trends, and news articles published online by the Guardian UK, the New York Times, the Wall Street Journal, Blomberg, the Financial Times, and Seeking Alpha. After ingesting, wrangling, and merging the data, we analyze the statistical properties of the estimated abnormal returns themselves and implement a variety of machine learning models to see how well features extracted from the news data can predict directional binary classifications of abnormal returns. We find that the calculated abnormal returns exhibit covariance stationary statistical properties (white noise), as expected. Using TfidfVectorizer and other methods to create features based on n-grams, we find that models predicting binary classifications of abnormal returns top out with accuracy scores below 70% and exhibit F1 scores not much higher than 50%, which is consistent with prior literature. We also use sentiment indicators to predict “large” movements with supervised classification machine learning techniques. Random Forest models perform the best scores and F1 scores are consistent with the main analysis. However, when using per-class model evaluation techniques with Yellowbrick, we find that fitted models are the bestat predicting “small” movements but do a poor job in predicting “large” movements. The results are robust to cross-validation and expanding window cross-validation procedures. An additional finding is that there is little overlap between relevant features extracted with TfidfVectorizer and those predicted by sentiment dictionaries.

[[1]](#footnote-1) ‘

**Introduction**

Public Stocks are generally thought to priced in efficient markets and react quickly to both general market factors and company-specific news. A common company-specific event study methodology is to estimate a stock’s residual (or abnormal) return on a given day with a basic market model regression and then test the statistical significance of the residual at a time that corresponds with the release of company-specific information. Significantly high or low abnormal returns at times corresponding to the release of identified company-specific news generally support the inference that the piece of news analyzed had material effects on stock returns.

Our project sought to apply a generalization of this methodology by using data ingesting, wrangling, data processing, and machine learning techniques in Python.We collected, cleaned, and merged news data from myriad data sources and modeled their ability to predict abnormal stock returns. For most of our modeling exercises, we adopted binary classification of the abnormal returns (i.e. a coding of 0 if the returns decreased and 1 if they increased, an in the alternative case, 1 if the abnormal returns was +/- 1 standard deviation away from the mean or 0 if not). Consistent with prior literature, most of our model’s F1 scores were only a bit above 50%, although the overall accuracy scores topped out a bit below 70%. The results were robust to randomized cross-validation procedures. Additionally, when using per-class model evaluation techniques and expanding window cross-validation, it is determined that fitted models are the best at predicting “small” movements and do a poor job in predicting “large” movements.

**Ingestion, Wrangling, Processing:**

The most important Python ingestion packages were: yfinance, sec\_edgar\_downloader, requests, Pytrends, TextBlob and BeautifulSoup. The bulk of our ingestion process is done in the 1, 2, and 3 codes in the capstone subfolder[[2]](#footnote-2). For the Guardian News and Google Trends data, event text data and time series search frequency data were obtained through the Guardian UK API and the Google Trends API. The code for the latter is organized in additional folders.[[3]](#footnote-3)

The financial data for our main sample is downloaded in the 1 code. For most of our analysis, we focus on about 10 years of data for all 7 biotechnology stocks in the S&P500 index[[4]](#footnote-4). The dates and closing prices were downloaded with yfinance and stored in a csv. The 7 code in the capstone folder does essentially the same thing for a much larger sample of stocks. Our benchmark—used to remove market factors from the returns—is the S&P500 total return index and was downloaded manually.

Every publicly traded company regularly files financial statements that are stored in SEC’s online Edgar database. For each company in our sample and each distinct type of filing (e.g. 8-K, 10-K, 13-F, etc.) we downloaded the last 200 filings of the given type, which usually entailed a times span of more than 10 years. The filings were downloaded in the form of text documents, but often retained embedded html code that presented challenges when parsing for useable features. Analyzing SEC filings for feature analysis required storing them in large datasets in Python. In order to minimize the size of the required analysis dataset in Python and to make feature analysis extracting smoother, the versions of the text documents stored in the analysis Python DataFrames were stripped of punctuation and non-english words with nltk English library. The date and time of each filing were extracted from the text by parsing it with regular expressions.

Scraping Google to look for company-specific articles from known important disseminators of financial news[[5]](#footnote-5) was the most challenging part of the ingestion process. The 3 code uses the requests and BeautifulSoup libraries to query Google for articles pertaining to each stock on each day between 2010 and 2020. Returned links with website addresses indicating that they were: 1) from the aforementioned known disseminators of financial news and 2) Were determined by their web addresses to have been published in the respective month of the query were selected and stored. Effort was made to create an algorithm that could parse the returned link texts to be certain of their exact date of publication, but we were unable to achieve this with sufficient accuracy. Instead, the text data for returned links was coded and stored for analysis on a monthly basis.

For additional news data, we explored several sources including Factiva (news aggregator), The Guardian UK, New York Times, Wall Street Journal, Financial Times and News API. The goal was to obtain news data that could help us create more features to predict the value of abnormal returns or a binary target that is derived from those abnormal returns. However, only the Guardian UK allowed users to download news for free through its API. One of the challenges that we had to overcome was the limitation on the number of items that can be downloaded per request and the number of requests per day. A code was created to search and request all articles in the Guardian UK that contained the name of the company of interest and industry relevant words: “biotechnology” and “pharmaceutical” for each day since January 1st, 2000. The new articles text data was stored using .json files.

For Google Trends Data, we use pytrends to download the Google Trends’ monthly frequency search indicator in response to our search query. Our query consists of the name of each of the companies listed for this project. Data is available since January 2004 and is stored using .csv files.

A list of approximately 4000 words which the Harvard IV-4 psychosocial dictionary has coded as either indicating positive or negative was also downloaded. The text data was scanned for word counts of each word. Prior literature has included sentiment analyses of financial news based on such dictionaries[[6]](#footnote-6).

**Computation and Data Processing:**

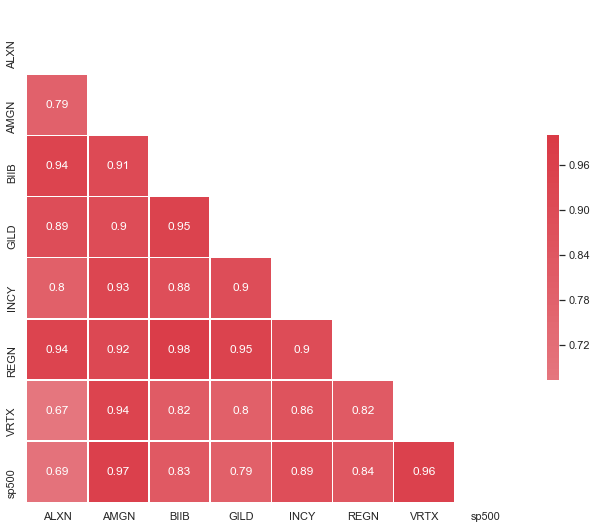
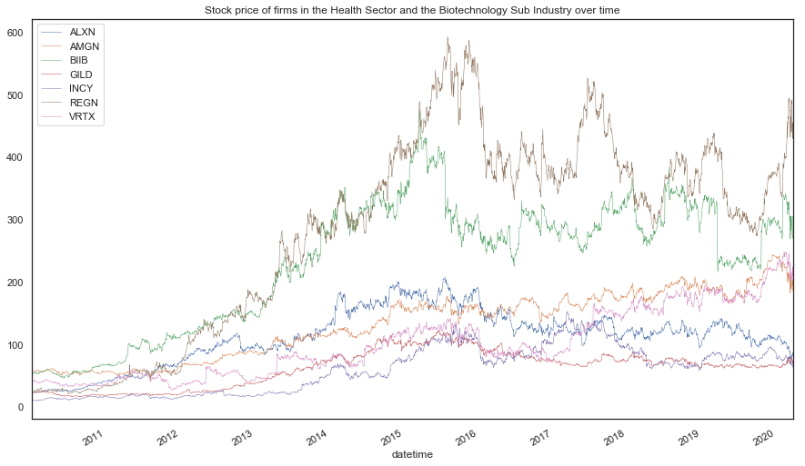
The capstone 4 code calculates the daily residual returns of each stock in the data samples with a function that regresses the daily log returns of the stocks on the log returns of the S&P 500 for a rolling 100 day window. The differences between the actual returns and those predicted by the rolling regressions constitute the logarithmic residual returns, and their exponentiated transformations the arithmetic or “abnormal returns.” The abnormal return should exclude the effects of market movements on the stock return, and hence only reflect the effects of company-specific information.

We also used text data from the Guardian UK to create a monthly average sentiment polarity and subjectivity indicator for each of search terms (i.e., the name of the companies), and for the words: “pharmaceutical” and “biotechnology”. We used NLTK to clean text data, i.e. transform to lower case, remove URLs, remove punctuation and remove stop words – and we used TextBlob to generate a sentiment polarity and a sentiment subjectivity indicator for each news article. We also created a tool using Counter from collections to count the frequency of specific words in the news articles using tokenized words produced by NLTK. This tool is not used for this project but can be used to generate frequency indicators based on terms of interest – this seems to be a widely used technique to quantify events using news data.[[7]](#footnote-7) These indicators and the number of articles per day were used to create monthly averages and create time series for the period 2000-2020. This data was used for the alternative modeling exercises that took place in this project.

**Exploratory Analysis of Residuals:**

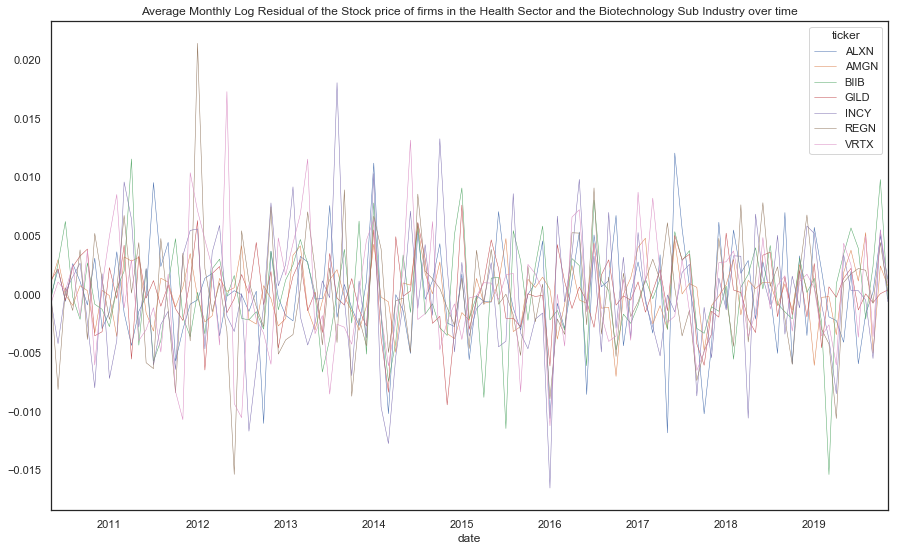
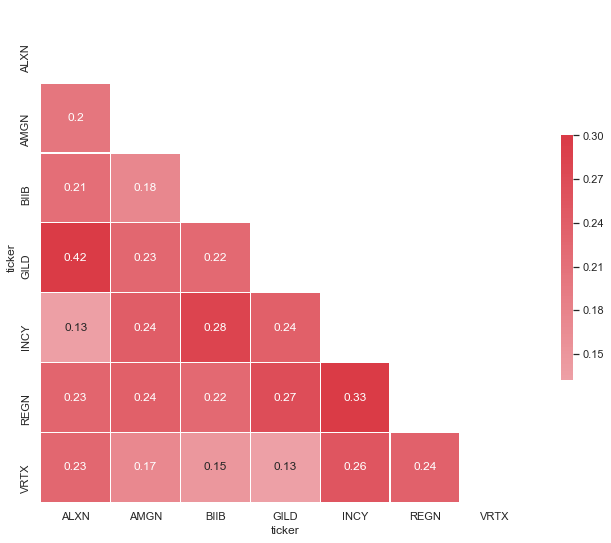
Once data was cleaned and merged into a monthly panel dataset that included all seven firms, several time series were analyzed[[8]](#footnote-8). All seven stock prices in the biotechnology sub industry seem to be correlated over time. These characteristics were confirmed using a Seaborn correlation matrix, positive and high correlations between the stock prices (figure 1). This evidence was taken into consideration when selecting features to improve model performance.

**Figure 1. Stock price of firms in the Health Sector and the Biotechnology Sub Industry and correlation between prices**



However, once abnormal returns were generated, the correlation between prices decreased in magnitude and visual evidence shows that once market-wide effects are removed from stock prices, industry-wide effects may play a smaller role in the co-movements of stock prices and co-movements may have little impact in the performance of predicting models (figure 2).

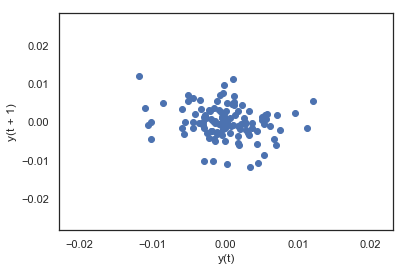
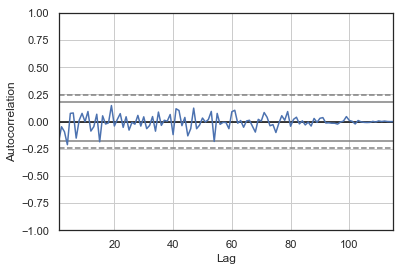
**Figure 2. Average monthly abnormal returns of firms in the Health Sector and the Biotechnology Sub Industry and correlation between abnormal returns**

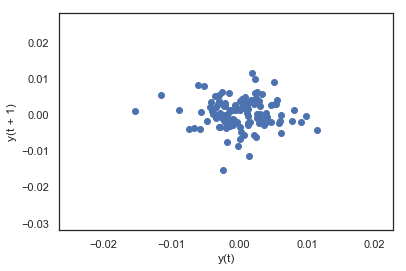
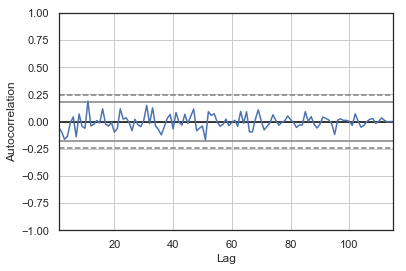
Autocorrelation was also analyzed for the abnormal returns using the Pandas autocorrelation plot. The objective was to determine if past information has predictive power and can improve the performance of the prediction model. Lag structure was also analyzed using the Pandas lag plot and confirmed the main result: average monthly abnormal returns are covariance stationary time series (white noise) which are considered to be time series processes that cannot be forecasted. For all seven firms, average monthly abnormal returns autocorrelation plots show evidence for which we conclude we fail to reject the null hypothesis that there is no autocorrelation (figure 3).

**Figure 3. Average monthly abnormal returns autocorrelation**

**ALXN**

** **

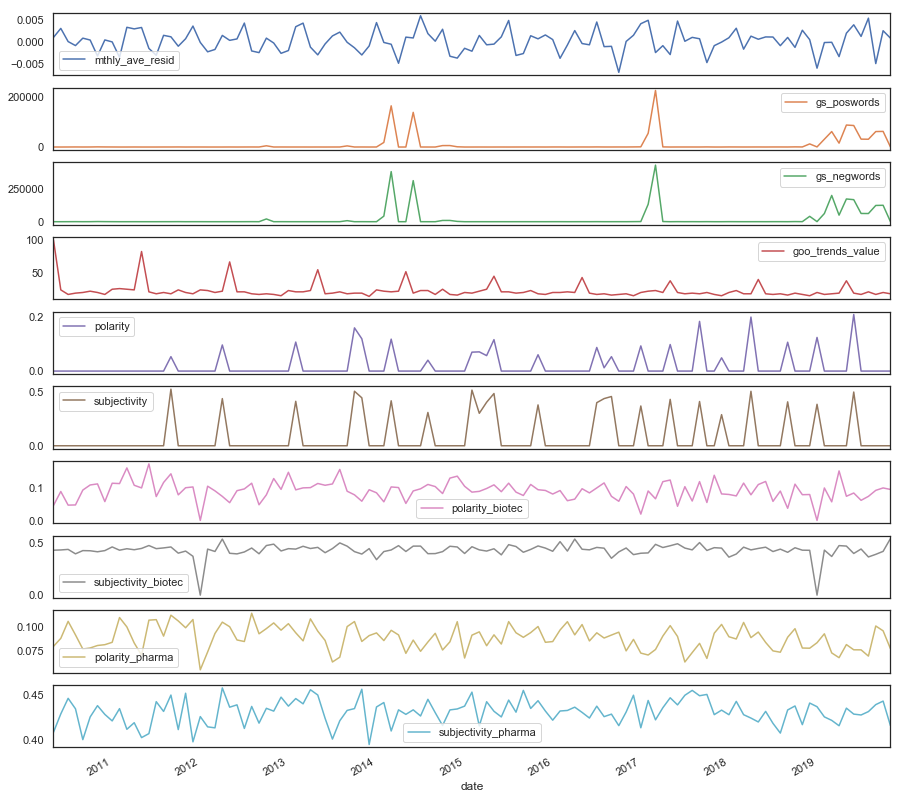
**BIIB**

** **

**Note: these plots are available for all firms, for conciseness only ALXN and BIIB are shown. The plots for other firms present similar evidence.**

Finally, to support feature selection for the alternative model, we examine the visual properties of the features with our target variable – abnormal returns (figure 4). For this particular case, we examined the following features: Google Trends search frequency indicator, firm’s monthly average polarity and subjectivity, monthly average polarity and subjectivity for articles containing the word “biotechnology”, and monthly average polarity and subjectivity for articles containing the word “pharmaceutical”.

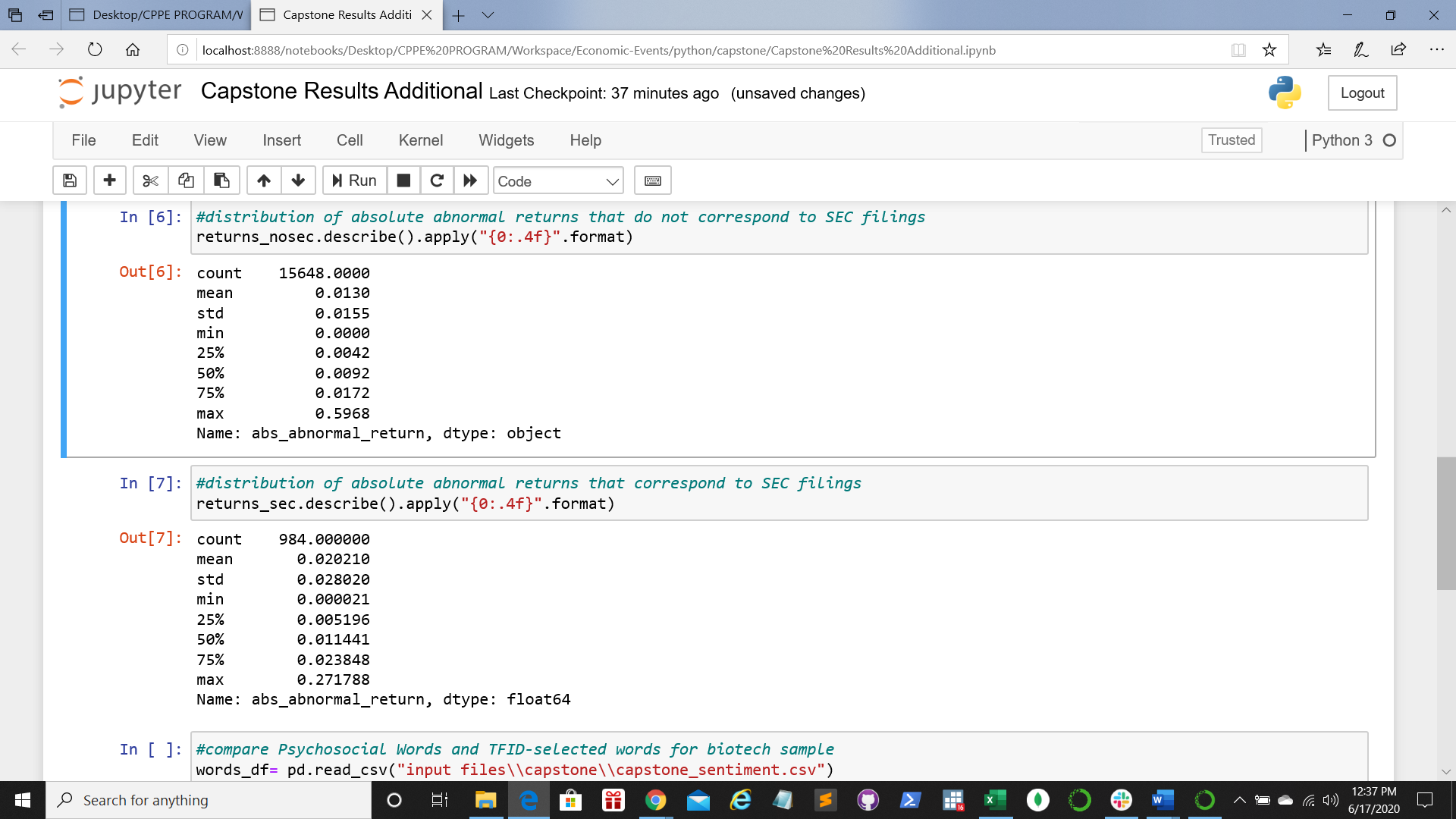
**Figure 4. Amgen, monthly abnormal returns and events data features**



As evidenced, events data does not do a good job at correlating with each abnormal return movement but there seems to be some usefulness in using these data to predict “large” movements, i.e., larger increases and decreases in the abnormal return over time.

Interestingly, the absolute value of abnormal returns are on average greater and exhibit greater variation on days that correspond to SEC filings than those dates do not.[[9]](#footnote-9) For the roughly 16,500 daily residuals of the sample of seven biotech firms, a t-test confirms that the 984 residuals that correspond to the days of SEC filings exhibit a greater absolute value of abnormal returns and also exhibit a higher standard deviation. This lends support to the null hypothesis that stock prices react to the information in financial disclosures.

**Figure 5. Abnormal absolute returns Distribution on days that do and do not correspond to SEC filings**



**Modeling and Results:**

We hypothesized that words and phrases (i.e. n-grams) could predict directional changes in stock prices, using both logistic regression and other models (RandomForests, SVCs, etc.). We used both the aforementioned sentiment-coded words from the Harvard-IV psychosocial dictionary and TfidVectorizer to create extract n-grams and create features based on them. This process entailed creating large, long data frames that paired daily residual returns with the text of associated SEC documents.

We initially intended to incorporate not just SEC filing data, but also news from other sources scraped from google searches. However, we did not feel we could determine the date of publication of the articles found on the web with sufficient accuracy for daily analyses. We instead collected the web-scraped articles into monthly datasets and evaluated several models based on the monthly datasets, but both the sample sizes and accuracy scores were low enough that we eventually decided that we had to abandon the google-scraped text data and just focus on the SEC filings. The models estimated with the monthly data can be found in the capstone folder under the 5 code.

The capstone 6 code uses TfidVectorizer to extract n-grams from the ‘text’ column of a Dataframe of the daily residuals that merged to SEC filings; it has 984 instances. Before running the Tfidvectorizer, we cleaned the text data by removing all punctuation and using the nltk English library to keep only English Words. Due to the small sample size, we limited the analysis to only 200 features and set the restriction that each feature correspond to words appearing in more than half and less than 1% of the texts[[10]](#footnote-10). We used a 50/50 test-train split. The results when the features were restricted to1-Grams gave us gave us an accuracy score of 48% on the test data, and class-specific F1 scores of 52% and 47%. These results did not change much with 4-fold cross-validation or the allowance of 2-grams and 3-grams (i.e. for multi-word phrases to count as vectors). The classification matrix for the model estimation without cross-validation displays below.

**Figure 6. Classification matrix**

A screenshot of a computer screen

Description automatically generated

Notably, many of the most positive and negative 1-gram features had intuitive plausibility. Some of the most positively-predictive 1-grams were: “innovative”, “technology”, “consolidated,” and some of the most negative were: “deferred” and “debt”. The “Capstone Additional Results” notebook compares the psychosocial sentiment-coded words with the coefficients on the 1-gram features. The sentiment dictionary does not appear to match the empirical estimates very closely. They agree about 53% of the time.

To see if more instances would change our findings, we also downloaded a large random sample of stocks and performed the same ingesting and wrangling exercises on the financial and SEC data, to create a merged sample of about 50k daily observations and associated SEC filings. To save computation time and select only the days most likely to have important company-specific news, we restricted the same to instances with absolute daily returns 1 standard deviation above the mean. The test 1 scores improved slightly, but overall accuracy did not improve much. There were some intuitively plausible n-grams: “share repurchase program” was positive was very positive and “cash restriction” was very negative.

An alternative approach was taken to examine the usefulness of news data in predicting movements of the monthly average abnormal returns[[11]](#footnote-11). We use the Google Trends search frequency indicator, firm’s monthly average polarity and subjectivity, monthly average polarity and subjectivity for articles containing the word “biotechnology”, number of the Guardian articles that include the word “biotechnology” per month, monthly average polarity and subjectivity for articles containing the word “pharmaceutical”, and number of the Guardian articles that include the word “pharmaceutical” per month. We started by using the original sample from 2010-19 using linear models to predict directly the monthly average residual returns. To improve models’ performance, we use a Pipeline() and a TransformedTargetRegressor() to normalize the target and features indicators using MinMaxScaler(). The mean Mean Absolute Error (MAE) and the R2 for the various linear models were recorded and evidence that model fit and prediction scores are poor, random cross-validation was performed using 10 splits (table 1).

**Table 1. Linear model scores**

|  |  |  |
| --- | --- | --- |
| Model | Mean MAE | R2 |
| LinearRegression | 0.003 | 0.031 |
| SGDRegressor | 0.016 | 14.216 |
| LassoRegression | 0.003 | 0.011 |
| ElasticNet | 0.003 | 0.011 |

To improve this analysis, 2 additional alternatives were explored: first, the sample was increased from 2010-2019 to 2000-2019 allowing for the implementation of an expanding window cross-validation; second, just like the previous analysis, this linear issue was transformed to a binary classification problem using the following rule applied to each firm separately:

* If a firm i’s residual return is greater or equal or lower or equal than +/- 1 standard deviation away from the firm i’s mean, the observation is equals to ‘1’ – “large” movements
* Any other observation is equal to ‘0’ – “short” movements

Using the entire panel data including the monthly average residual return and features for all firms, we fitted several classification models as listed below:

**Table 2. Classification model scores**

|  |  |
| --- | --- |
| **Model** | **F1 score** |
| SVC | 0.4812834224598931 |
| LinearSVC | 0.019656019656019656 |
| SGDClassifier | 0.0 |
| KneighborsClassifier | 0.37613019891500904 |
| LogisticRegression | 0.019704433497536946 |
| **BaggingClassifier** | **0.8086253369272237** |
| **ExtraTreesClassifier** | **0.8667563930013458** |
| **RandomForestClassifier** | **0.7895460797799175** |
| GaussianNB | 0.2823920265780731 |

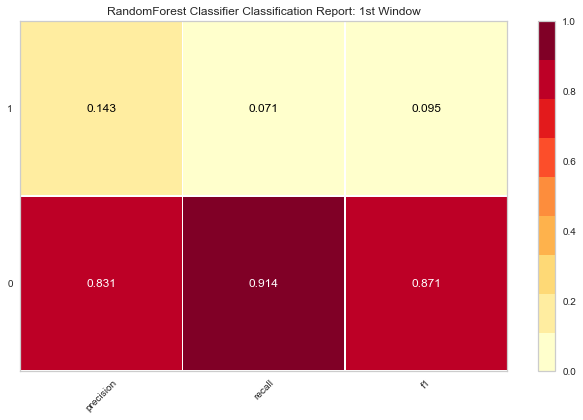
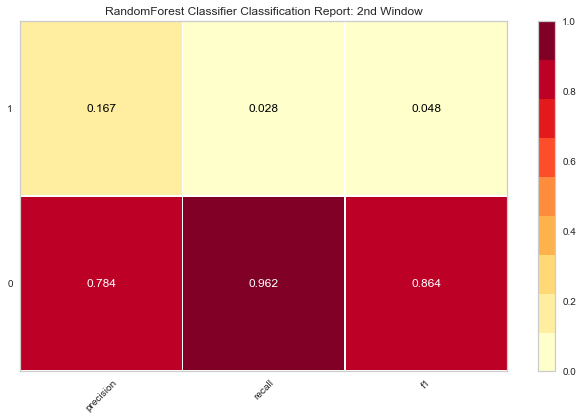
A particular issue we needed to overcome is the appropriate use of a non-random splits for cross-validation and the reason is for time series, we should take into consideration the importance of the time series one directional nature. We classified manually the extended panel dataset to create date appropriate splits and use expanding window cross-validation. The manual classification that took place is described in figure 7. Another important assessment we did was to examine per-class model evaluation using a classification report from Yellowbrick.

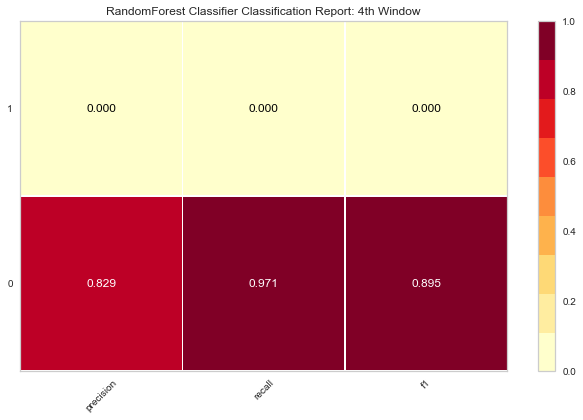
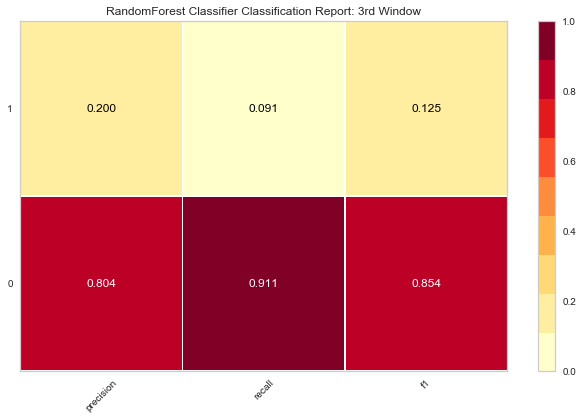
**Figure 7. Expanding window cross-validation**

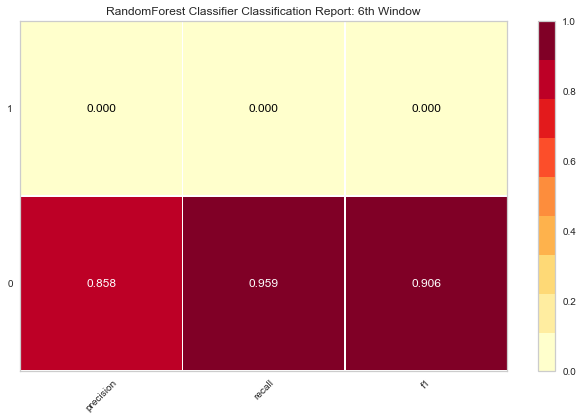
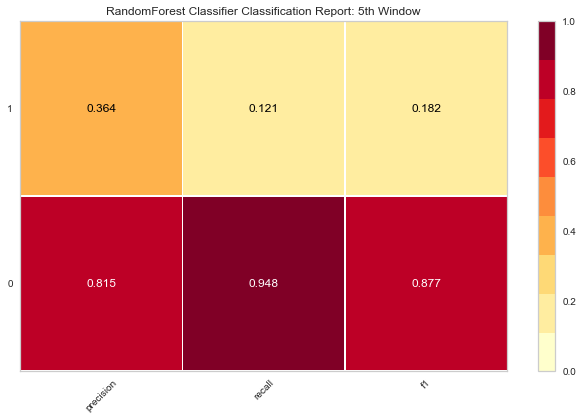


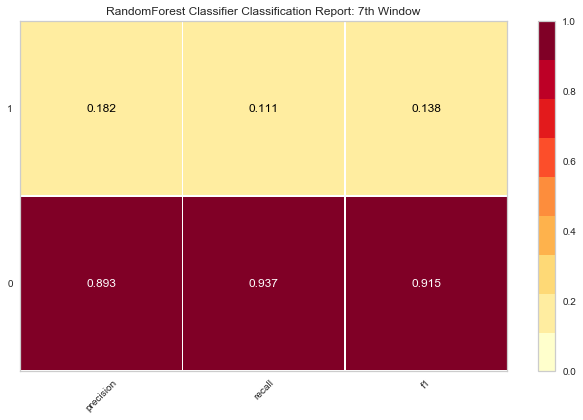
After evaluating performance, we determine that the best per-class performance was delivered by the Random Forest Classifier; however, when using an expanding window cross-validation and taking a look at the per-class model performance we determine that the Random Forest Classifier does a poor job at classifying “large” movements which in this case is most important for us given we find usefulness in this classification model by predicting large movements. The average f1 score for the Random Forest classifier for “large” movements using the 7 windows described above was 0.084 and the f1 score for “short” movements was 0.883. Window specific results are examined in figure 8.

**Figure 8. Random Forest Classifier Model performance evaluation using expanding window cross-validation and Yellowbrick’s classification report**







**Considerations for Enhancement** .

Overall, we feel that the accuracy scores are high enough to suggest that textual analysis with machine learning may be useful in designing trading algorithms and options pricing models. However, one issue that remains to be thoroughly investigated is that of leakage. Our unit of analysis is a day, but the general issue is independent of the unit of time of analysis. If the news analyzed were known and traded upon by a sufficient number of market participants before the time unit to which it’s assigned data, then some of the information will be reflected in the prior units price and the associated return will not fully reflect the effect of the news. This is likely to be an issue for any project trying to extract useful signals for pricing models.

**Conclusion**

The project has employed a variety of data science including data ingestion, wrangling and machine learning techniques in predicting the abnormal returns of publicly traded stocks based on financial and economic news. A few conclusions emerge on the use of these techniques in our projection process:

Frist, regardless of sample size, maximum scores for most models based on n-grams from SEC filings max out at under 70%; similarly, F1 scores are low, like in the published literature. Second, abnormal returns exhibit greater variation on the days of SEC filings; with findings more robust to cross-validation. Third, classification seems to perform better. However, when doing a per class model performance evaluation, classification results are not good for predicting “large” movements. Last, we found accuracy score is not necessarily increased by the increase of sample size, which may indicate more refined features associated with the event data are needed. Although data is not cheap, it is necessary to increase features with more events data.

1. Residual return is defined as actual return minus predicted normal return where predicted normal return for individual stock is modelled against broad market stock benchmark return [↑](#footnote-ref-1)
2. The capstone codes are simply labeled 1 through 8, providing a simple pipeline for replicating the process. Most ingestion is done in the 1 through 3 codes.

   <https://github.com/georgetown-analytics/Economic-Events/blob/master/python/capstone/1.%20Capstone%20import%20financial.py>

   <https://github.com/georgetown-analytics/Economic-Events/blob/master/python/capstone/2.%20Capstone%20SEC%20Import%20and%20Clean.py>

   <https://github.com/georgetown-analytics/Economic-Events/blob/master/python/capstone/3.%20Capstone%20Scrape%20Google.py> [↑](#footnote-ref-2)
3. <https://github.com/georgetown-analytics/Economic-Events/tree/master/python/Google%20Trends%20API%20Data%20ingestion>

   <https://github.com/georgetown-analytics/Economic-Events/tree/master/python/The%20Guardian%20Data%20ingestion%20and%20wrangling> [↑](#footnote-ref-3)
4. The firms that are analyzed in this project are: (i) Alexion Pharmaceuticals; (ii) Amgen Inc.; (iii) Biogen Inc.; (iv) Gilead Sciences; (v) Incyte; (vi) Regeneron Pharmaceuticals; and (vii) Vertex Pharmaceuticals Inc. Hereinafter, ‘ALXN’, ‘AMGN’, ‘BIOGEN’, ‘GILD’, ‘INCY’, ‘REGN’, and ‘VRTX’ respectively. [↑](#footnote-ref-4)
5. The filtering process searched for articles from the New York Times, Bloomberg, Financial Times, the Wall Street Journal, and Seeking Alpha. [↑](#footnote-ref-5)
6. Hagenau, M., Liebmann, M., & Neumann, D. (2013). Automated news reading: Stock price prediction based on financial news using context-capturing features. Decision Support Systems, 55. 685-697. [↑](#footnote-ref-6)
7. Ahir, H., Bloom, N., and D. Furceri, 2018, “The World Uncertainty Index”. International Monetary Fund, Washignton, D.C. For example. [↑](#footnote-ref-7)
8. The exploratory data analysis can be found here: <https://github.com/georgetown-analytics/Economic-Events/blob/master/python/Exploratory%20Data%20Analysis.ipynb> [↑](#footnote-ref-8)
9. This output can be replicated by running the Capstone Jupyter Notebbok “Capstone Additional Results,” located in the Capstone Folder. [↑](#footnote-ref-9)
10. These percentage-of-text restrctions are meant to simultaneously eliminate words related to the formatting the text files that may have survived the cleaning process and to prevent extremely company-specific words and phrases that may not generalize from affecting the analysis. [↑](#footnote-ref-10)
11. This analysis is saved in this folder:

    <https://github.com/georgetown-analytics/Economic-Events/blob/master/python/Alternative%20feature%20and%20model%20selectionevaluation.ipynb> [↑](#footnote-ref-11)