Text Mining For Earnings Event Trading

Authors:

Jiahao(Steven) Wang, Chengxiaoyuan Wang

Executive Summary

In the U.S equity market, when any anticipated material information or report is about to be released, the related securities' prices tend to experience large reactions to such release. This reaction could be extreme, with precedents of more than a 50% increase or decrease in the stock price, or mild, that doesn't change much from the price level before the information release. Such price uncertainty and volatility create opportunities for profits if the market participants act correctly while bearing a significant amount of risks.

Intuitively, when a company reports positive earning performance that beats the consensus expectations, stock prices should benefit from this information. However, when the market expects more to justify or sustain the current valuation, the stock price of the company could still decrease significantly even if they have beaten the consensus expectations on its quarterly performance, and vice versa. Such circumstances aren't rare in practice, and raise a valuable question: how can we predict the direction of the price movement post a company's earnings announcement?

Business Goal Analysis

Under the assumption that there are pricing inefficiencies in the market, market participants use different forecasting models and have different expected price targets for public companies. This is the potential reason for why the stock price decreases(increase) on consensus expectation beat(miss): they don't all expect the same with the consensus and don't act according to the consensus. Situations like this make it difficult to only use a financial valuation model for predicting the earnings price movement because even if the fundamental financial ratios of a company look great, they can still drop in stock price due to priorly stated reasons. This is also why we think using text mining would help greatly. Because it does not look at the possibly misleading fundamental metrics. Text mining could potentially detect the patterns in the financial news articles during the quarter that might help market participants forecast how a company's stock price will move at the next earnings announcement objectively.

Our team constructed a method of prediction on stock price movements at earnings announcements using financial news articles. We proposed to use the financial articles of the past 3 years(2019 - 2021), separate them by the market sectors, and conduct text mining techniques with machine learning models to predict the potential changes in the stock price after earnings. Due to extreme difficulty in predicting the exact percentage changes in each instance, we decided to classify our supervised learning with 3 target outcomes: price increase larger than 3% (class 1), price movement between 3% and -3%(class 0), and price decrease more than -3%(class 2). The reason the threshold is 3% is that taking into account tax and trading costs based on the planned trading strategy, a price movement of more than 3% in either direction is deemed actionable for a large potential return.

In short, our goal is to 1. Predict the outcome of the stock price movement when the corresponding companies announce their quarterly earnings, and 2. Create actionable trading strategies based on our prediction results.

Data Description

The data set we used for our project consists of 3 years (2019-2021) worth of news articles of 150 companies from 5 different market sectors: Financial Services Sector, Consumer Discretionary Sector, Communication Services Sector**,** Technology Sector, and Healthcare Sector. We acquired this data by creating a specific web scraping process for the chosen news websites sources including Barron's, Market Watch, and WSJ**.** We indicated specific various market sectors because we suspect that the important words and topics would potentially be different across different market sectors. Therefore, it would be beneficial to sample multiple companies from different sectors.

Upon completion of our data acquisition process, we have obtained approximately 40,000 pieces of raw articles, as well as the article date value and the corresponding company's stock ticker. In addition to our textual data, we also used yahoo finance API to obtain the earnings dates, and the necessary historical stock pricing data for the use of the trading backtest.

System Design

The system design of our project is complex, and consists of 3 major components: Data acquisition, transformation, and structure, Text preprocessing and representation, and Machine learning predictions with trading backtest.

1. Data Acquisition, Transformation, and Structure

This is the foundational step of our system. The primary function of the first component is to acquire targeted data according to multiple indicated factors, such as selected company names, stock ticker symbols, and each earnings date for the chosen periods. As the flow chart below illustrates that the process begins with identifying the target companies across 5 different sectors, then these companies' tickers were used as searching input for web data scraping on the selected news sources. Companies' stock price and historical earnings announcement schedule data are also retrieved simultaneously through related APIs. When news data were successfully collected, we conducted dictionary-based sentiment analysis on each article that provides 5 different sentiment element scores. After completing all previous processes, the data set is aggregated quarterly according to the earnings date for each company, and sentiment element scores are averaged by the total number of articles of the corresponding quarter. Now, we have completed the structuring of our data set.

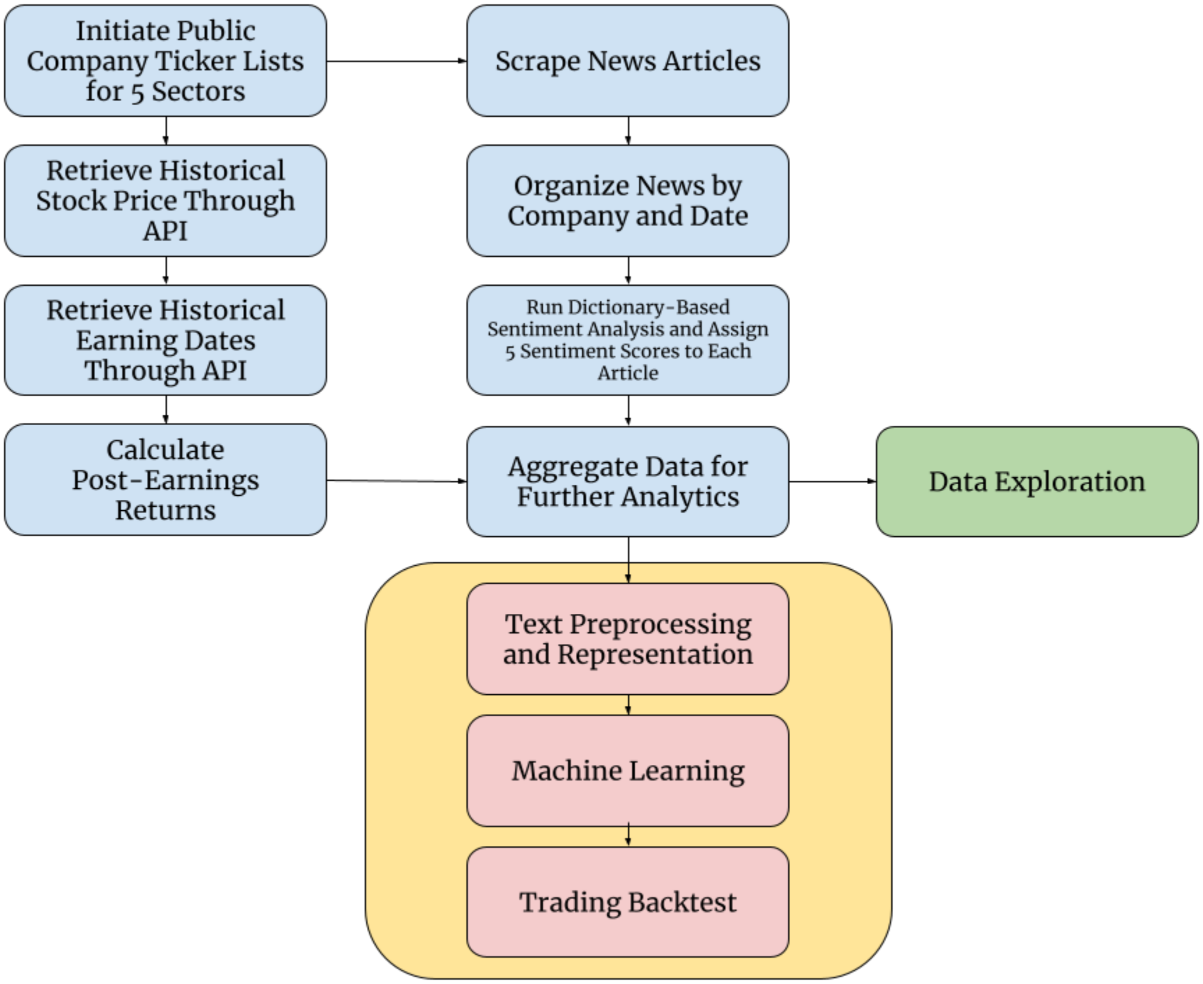

Initiate Public Company Ticker Lists for 5 Sectors
Scrape News Articles
Retrieve Historical Stock Price Through API
Organize News by Company and Date
Retrieve Historical Earning Dates Through API
Run Dictionary-Based Sentiment Analysis and Assign 5 Sentiment Scores to Each Article
Calculate Post-Earnings Returns
Aggregate Data for Further Analytics
Data Exploration
Text Preprocessing and Representation
Machine Learning
Trading Backtest

Figure 1 - Data Parsing, Transforming, and Structuring Flow Map

2. Text preprocessing and representation

The second component of our system design focuses mainly on pre-processing the textual data with different techniques and putting the data set into a machine-learning-ready format. Continuing with the data set after structuring, we process the textual data by using tokenization, stop-words removal, and lemmatization. Next, the preprocessed data is further transformed by using the bag-of-words and part-of-speech approach in combination with TF-IDF and LDA topic model. Each of these combinations is trained and tested separately to compare the difference in prediction performance. The flow chart below demonstrates the detailed workflow of the second component.

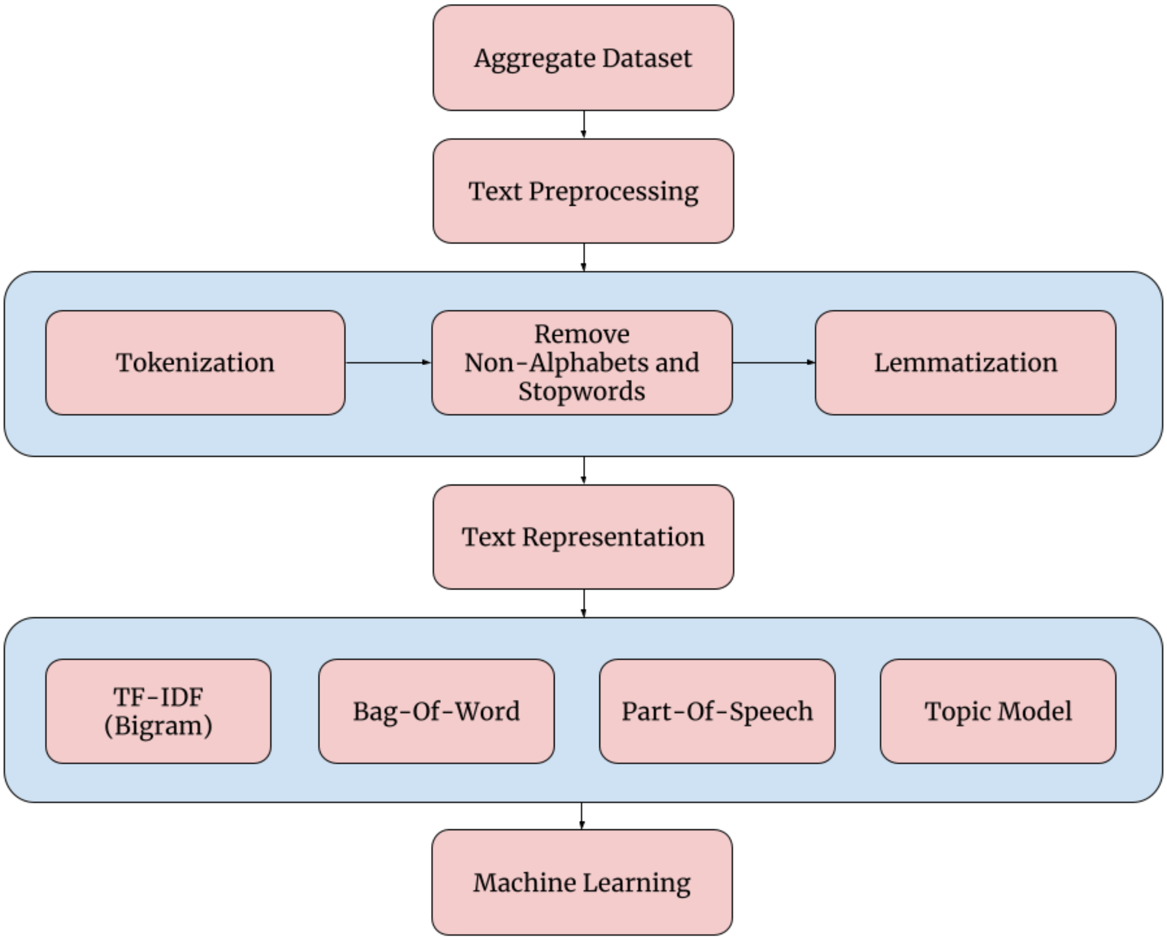


Figure 1 - Text Pre-processing & Representation Flow Map

3. Machine Learning Predictions with Trading Back Test

The last component of our design is using various machine learning models to predict the stock price movements at the earnings announcement. We use the prepared data set as input and implemented multiple models for the supervised classification. The 5 different market sectors are first trained and tested individually as well as trained and tested as a whole.

Upon completion of predictions, we join the necessary factors with the prediction results and created a function to calculate the percentages of return based on the pre-defined trading strategy, which is to buy/sell the stock that we predicted as class 1/class 2 before market close on the earnings date, then sell at the next day's market open to capture the volatile pricing movement due to pricing inefficiency.

These results are also shown both individually by sector and together as a whole.
The flow chart below demonstrates the detailed process for our third component.

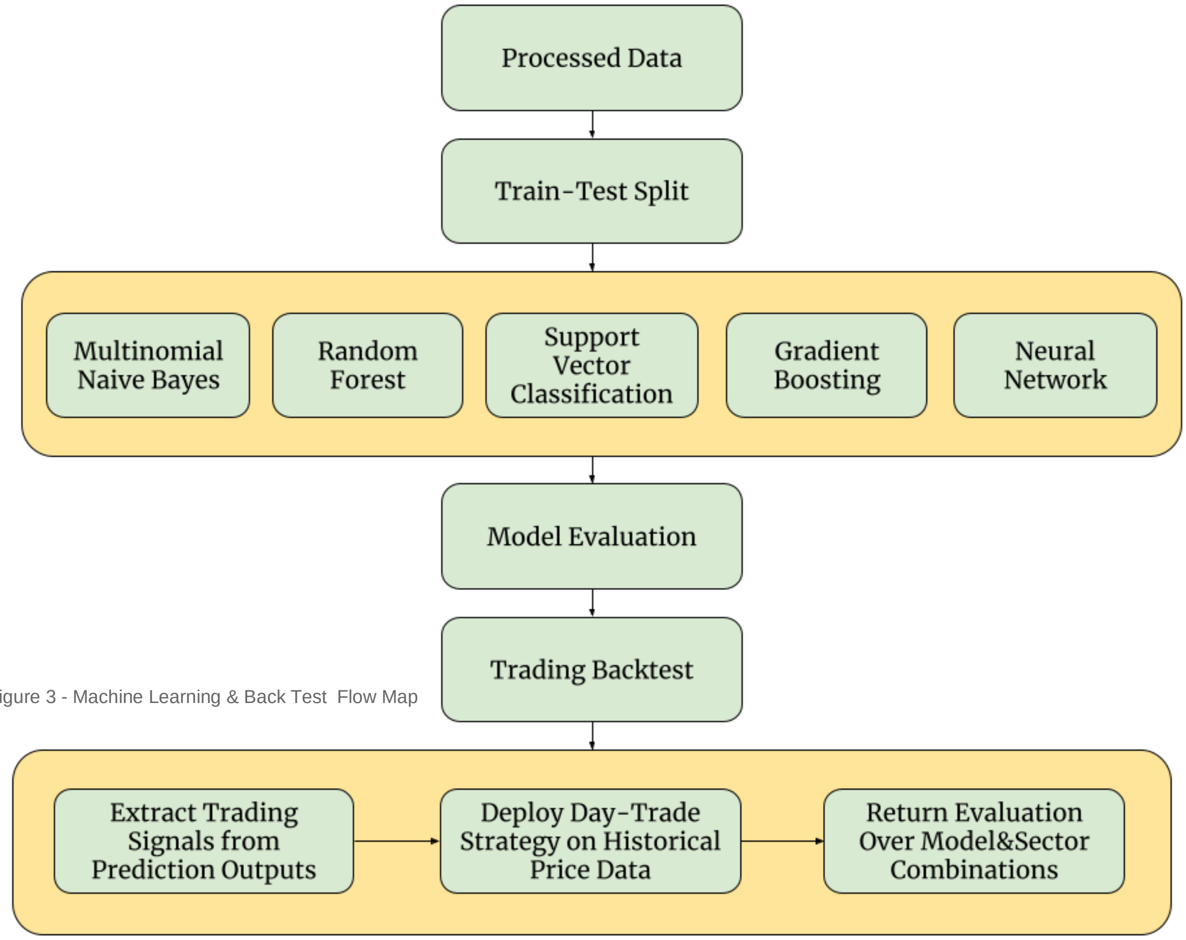

Processed Data
Train-Test Split
Multinomial Naive Bayes
Random Forest
Support Vector Classification
Gradient Boosting
Neural Network
Model Evaluation
Trading Backtest
Extract Trading Signals from Prediction Outputs
Deploy Day-Trade Strategy on Historical Price Data
Return Evaluation Over Model&Sector Combinations

Figure 3 - Machine Learning & Back Test Flow Map

System Implementation

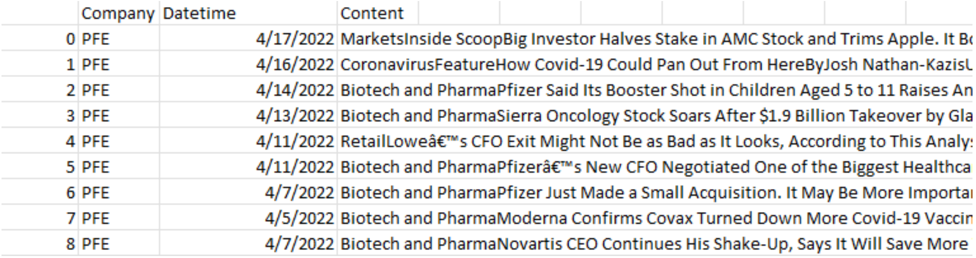
Part I – Web Data Crawling

1. Selenium: In our data crawling process, we utilized Selenium to navigate through different pages of the news website in order to get the most relevant articles about the target company. It supported us with functions such as input values into the search bar, click, and scroll. News website like Barron’s has a various structure on each page, which makes it difficult to pinpoint desired information without the use of Selenium

2. Beautiful Soup: After surfacing a specific part of the website by using Selenium, we used Beautiful Soup to extract the HTML code to obtain the information that the code contains. This information includes article links, article dates, and the article's main content.

Figure 4 - Sample Output

Part II – Data



	Company	Datetime	Content
0	PFE	4/17/2022	MarketsInside ScoopBig Investor Halves Stake in AMC Stock and Trims Apple. It B
1	PFE	4/16/2022	CoronavirusFeatureHow Covid-19 Could Pan Out From HereByJosh Nathan-Kazis
2	PFE	4/14/2022	Biotech and PharmaPfizer Said Its Booster Shot in Children Aged 5 to 11 Raises An
3	PFE	4/13/2022	Biotech and PharmaSierra Oncology Stock Soars After $1.9 Billion Takeover by Gla
4	PFE	4/11/2022	RetailLoweâ€™s CFO Exit Might Not Be as Bad as It Looks, According to This Analy
5	PFE	4/11/2022	Biotech and PharmaPfizerâ€™s New CFO Negotiated One of the Biggest Healthca
6	PFE	4/7/2022	Biotech and PharmaPfizer Just Made a Small Acquisition. It May Be More Importa
7	PFE	4/5/2022	Biotech and PharmaModerna Confirms Covax Turned Down More Covid-19 Vaccin
8	PFE	4/7/2022	Biotech and PharmaNovartis CEO Continues His Shake-Up, Says It Will Save More

Transformation and Structuring

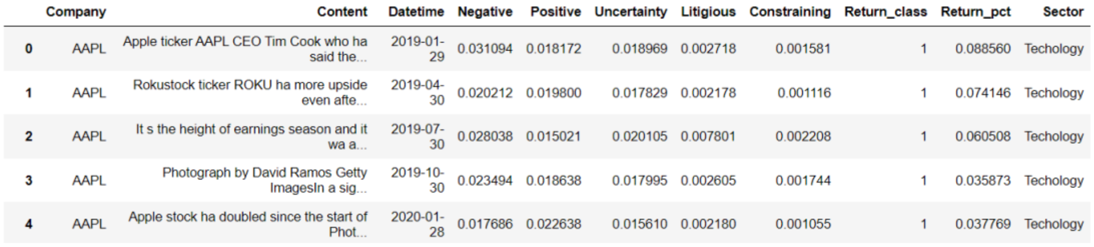
1. Datetime: We need to use the date as the main guidance to structure our data set as it is essential to the concept of our project. Therefore, the package Datetime is been heavily used in organizing the article in chronological order. We aggregate the contents of articles on a quarterly basis, and using DateTime allowed us to use calculation operators in creating conditions for our data structure. For example, when we aggregate the articles that are before a certain earnings date, we simply used 'article date' < 'earnings date' as a condition to achieve the separation.

2. Yahoo Finance API: There were two types of Yahoo Finance API used: 'yahoofin' and 'yfinance'. 'yahoofin' was used to directly obtain the historical prices on included companies through stock ticker queries and periods, and outputs a data frame of open and close stock prices on each day with datetime as its index. 'yfinance' was used specifically for obtaining each companies earnings history in the selected period(2019 - 2021). These pieces of data were then used to aggregate the articles before each earnings date, and calculate the return percentages for each earnings event to transform them into our target classification outcome

3. Pandas: Pandas package is the main body of our data structuring process. We used data frame as our main data type for various parts of our data set, and Pandas supported us in slicing, filtering, structuring, and transforming our dataset.

Sub-Part II: Sentiment Analysis

4. Loughran-McDonald Master Dictionary: The Loughran-McDonald Dictionary is the dictionary-based tool for sentiment analysis specifically in the financial context. The dictionary covers a large number of terms with multiple sentiment elements: Positive, Negative, Uncertainty, Litigious, and Constraining. It allowed us to add additional features besides just 'Positive', and 'Negative' that potentially increase the predictability of earnings results.



	Company	Content	Datetime	Negative	Positive	Uncertainty	Litigious	Constraining	Return_class	Return_pct	Sector
0	AAPL	Apple ticker AAPL CEO Tim Cook who ha said the...	2019-01-29	0.031094	0.018172	0.018969	0.002718	0.001581	1	0.088560	Techology
1	AAPL	Rokustock ticker ROKU ha more upside even afte...	2019-04-30	0.020212	0.019800	0.017829	0.002178	0.001116	1	0.074146	Techology
2	AAPL	It s the height of earnings season and it wa a...	2019-07-30	0.028038	0.015021	0.020105	0.007801	0.002208	1	0.060508	Techology
3	AAPL	Photograph by David Ramos Getty ImagesIn a sig...	2019-10-30	0.023494	0.018638	0.017995	0.002605	0.001744	1	0.035873	Techology
4	AAPL	Apple stock ha doubled since the start of Phot...	2020-01-28	0.017686	0.022638	0.015610	0.002180	0.001055	1	0.037769	Techology

Figure 5 - Sample Output

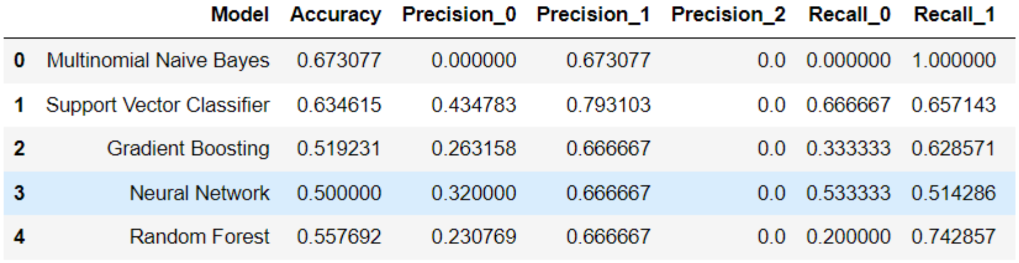
Part III – Text Preprocessing

1. NLTK: NLTK package is the primary preprocessing tool used for our textual data. This package provided our project with crucial functions such as tokenization, stop-words removal, part-of-speech tagging, and stemming (lemmatization).

Part IV – Modeling and Predictions

1. Gensim: Gensim package was mainly used for creating a term-document matrix and implementing the Latent Dirichlet Allocation (LDA) topic model to reduce the feature dimensionality.

2. Scikit Learn: Various models were taken out of the box from Scikit Learn for our supervised learning. TF-IDF vectorizer is also used for feature extraction and dimensionality reduction. The machine learning models we implemented include multinomial Naive Bayes, support vector classification, gradient boosting, neural network, and random forest. Lastly, the model evaluation 'metrics' function was also supported by Scikitlearn for us to evaluate the prediction outcomes.



	Model	Accuracy	Precision_0	Precision_1	Precision_2	Recall_0	Recall_1
0	Multinomial Naive Bayes	0.673077	0.000000	0.673077	0.0	0.000000	1.000000
1	Support Vector Classifier	0.634615	0.434783	0.793103	0.0	0.666667	0.657143
2	Gradient Boosting	0.519231	0.263158	0.666667	0.0	0.333333	0.628571
3	Neural Network	0.500000	0.320000	0.666667	0.0	0.533333	0.514286
4	Random Forest	0.557692	0.230769	0.666667	0.0	0.200000	0.742857

Figure 6 - Sample Output

Evaluation

For evaluation, we divide it into three parts: Descriptive Analysis, Machine Learning Performance, and Trading Backtest Performance.

Part I. Descriptive Analysis

In this part, we aim to grasp a high-level picture of the massive text dataset and derive meaningful insights from it.

1. Unbalanced dataset

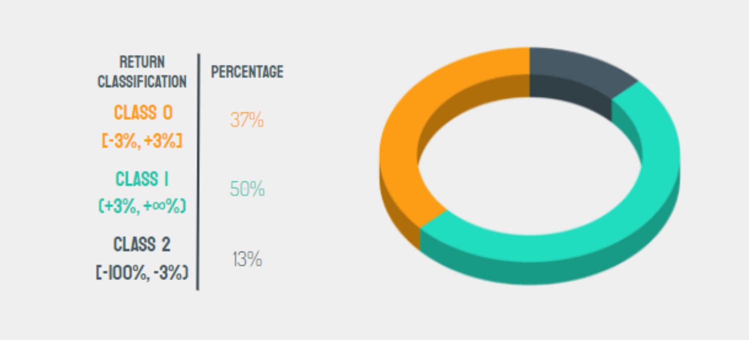


Figure 7 - Target Outcome Ring Chart

As mentioned in the previous section, we defined three classes for the dependent variable (post-earnings stock return) with each class representing a range of price changes. Through studying the distribution of these three classes, we found that Class 1 covers half of the data, followed by Class 0 covering 37%, and Class 2 covers merely 13%.

We concluded that imbalance in post-earning return is almost inevitable given the facts that a) the sample companies are dominantly large public corporations that benefited from the aggressive quantitative easing and the expansionary fiscal policies in 2020 and 2021; b) the U.S. equity market during the sample years, 2019-2021, was wildly bullish (except for March 2020 when COVID caught the economy off-guard). The initial pessimistic earning estimates from the Wall Street analysts offered more room for earning beats after rounds of economic incentives.

2. Counterintuitive correlation

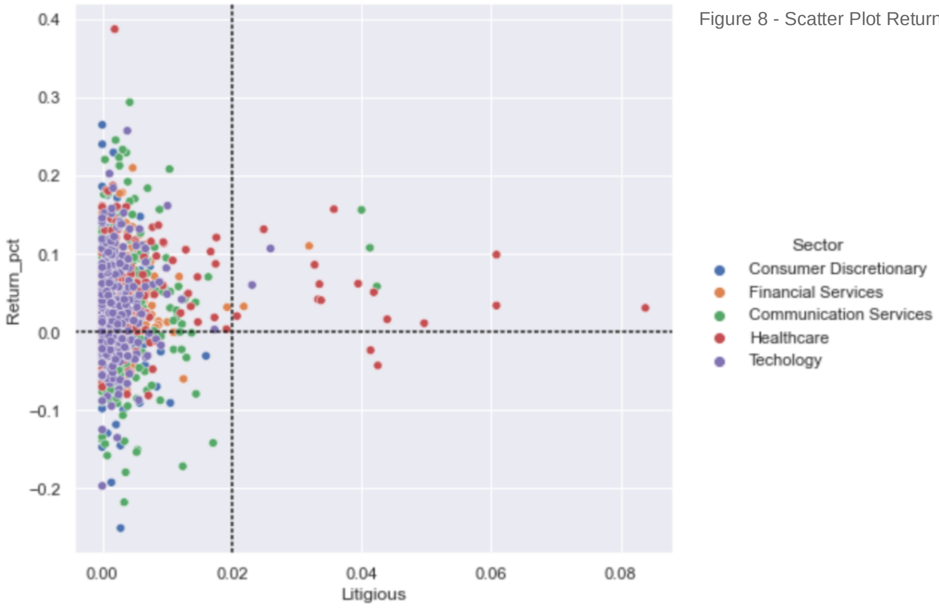


Figure 8 - Scatter Plot Return percentage vs. Litigious Score

Intuitively, articles classified with much higher litigious/negative scores should imply a higher likelihood of price decline after earnings announcements. However, our data demonstrate the opposite.

We came up with two reasons to explain the counterintuitive correlation:

a) Many financial articles over exaggerate the sentiment of certain events that won't fundamentally impact the stock performance.

b) Dictionary-based sentiment analysis would fail to differentiate negation or context, therefore, labeling false sentiment to certain texts.

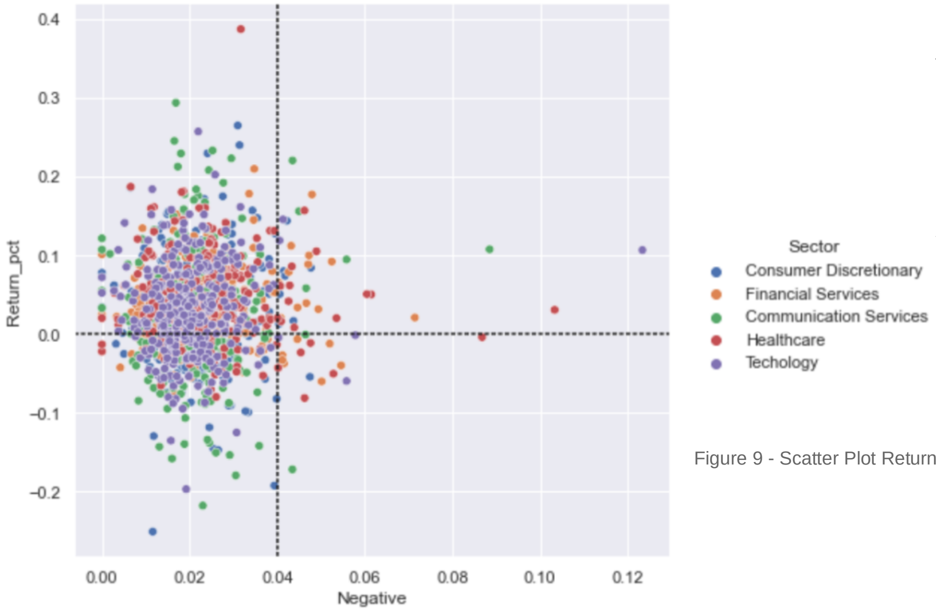
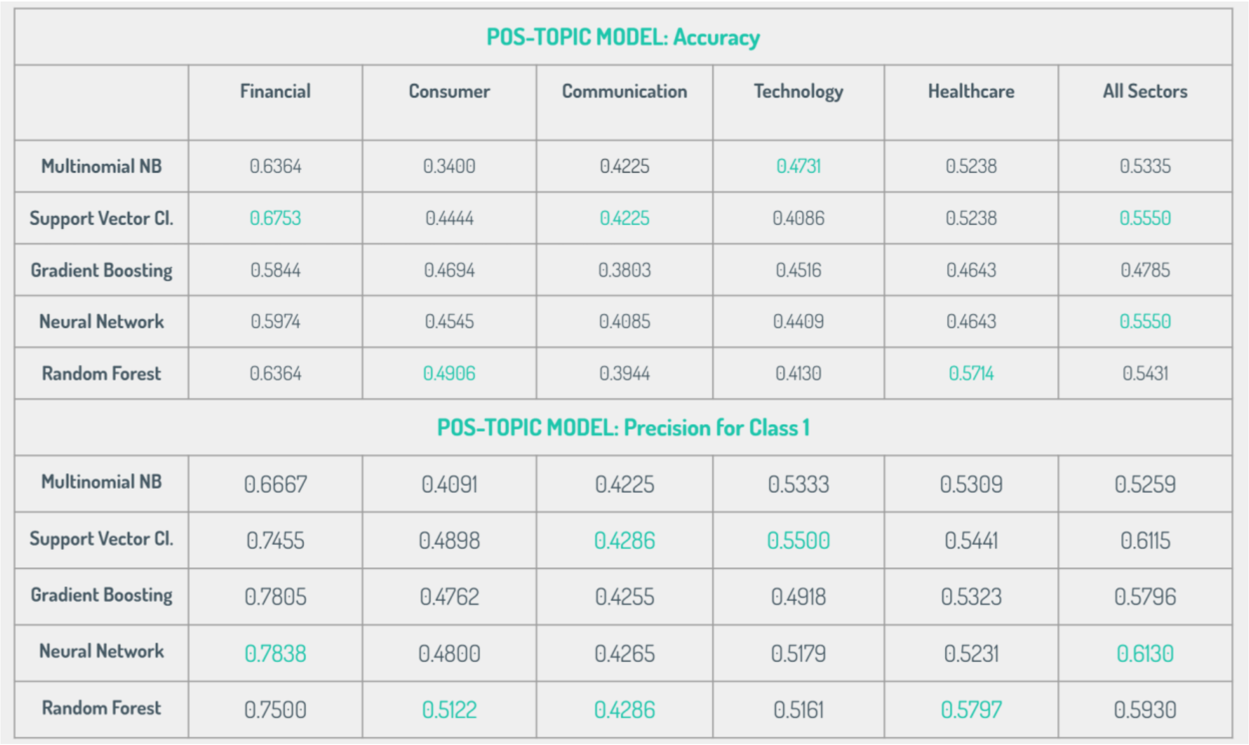


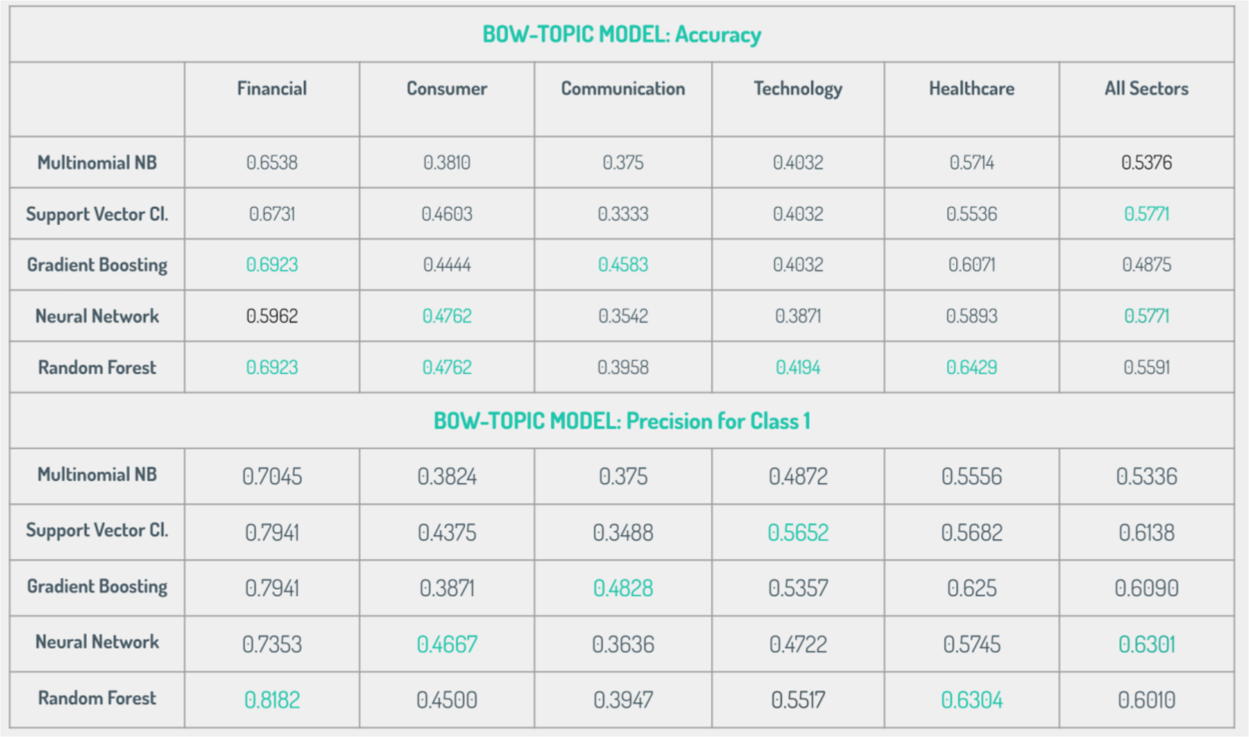
Figure 9 - Scatter Plot Return percentage vs. Negative Score

Part II. Machine Learning Performance

Figure 10 - POS-Topic Model Prediction results

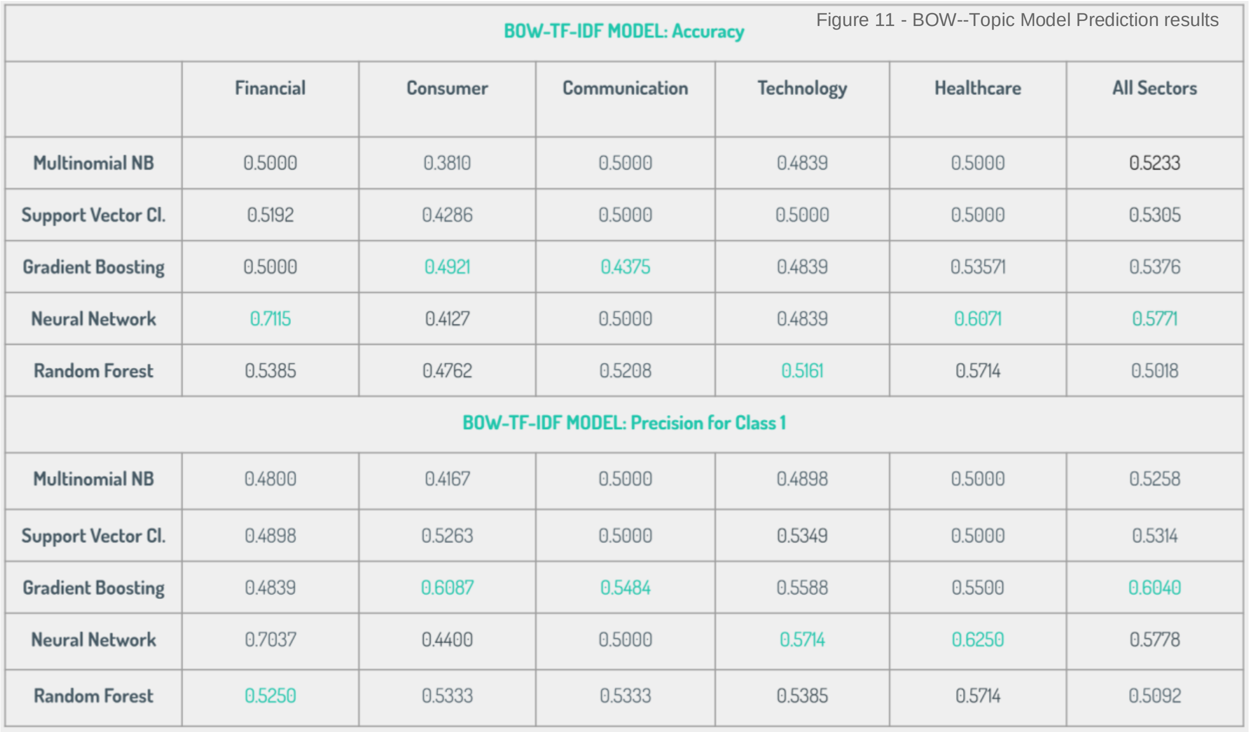


POS-TOPIC MODEL: Accuracy						
	Financial	Consumer	Communication	Technology	Healthcare	All Sectors
Multinomial NB	0.6364	0.3400	0.4225	0.4731	0.5238	0.5335
Support Vector Cl.	0.6753	0.4444	0.4225	0.4086	0.5238	0.5550
Gradient Boosting	0.5844	0.4694	0.3803	0.4516	0.4643	0.4785
Neural Network	0.5974	0.4545	0.4085	0.4409	0.4643	0.5550
Random Forest	0.6364	0.4906	0.3944	0.4130	0.5714	0.5431
POS-TOPIC MODEL: Precision for Class 1						
Multinomial NB	0.6667	0.4091	0.4225	0.5333	0.5309	0.5259
Support Vector Cl.	0.7455	0.4898	0.4286	0.5500	0.5441	0.6115
Gradient Boosting	0.7805	0.4762	0.4255	0.4918	0.5323	0.5796
Neural Network	0.7838	0.4800	0.4265	0.5179	0.5231	0.6130
Random Forest	0.7500	0.5122	0.4286	0.5161	0.5797	0.5930



BOW-TOPIC MODEL: Accuracy						
	Financial	Consumer	Communication	Technology	Healthcare	All Sectors
Multinomial NB	0.6538	0.3810	0.375	0.4032	0.5714	0.5376
Support Vector Cl.	0.6731	0.4603	0.3333	0.4032	0.5536	0.5771
Gradient Boosting	0.6923	0.4444	0.4583	0.4032	0.6071	0.4875
Neural Network	0.5962	0.4762	0.3542	0.3871	0.5893	0.5771
Random Forest	0.6923	0.4762	0.3958	0.4194	0.6429	0.5591
BOW-TOPIC MODEL: Precision for Class 1						
Multinomial NB	0.7045	0.3824	0.375	0.4872	0.5556	0.5336
Support Vector Cl.	0.7941	0.4375	0.3488	0.5652	0.5682	0.6138
Gradient Boosting	0.7941	0.3871	0.4828	0.5357	0.625	0.6090
Neural Network	0.7353	0.4667	0.3636	0.4722	0.5745	0.6301
Random Forest	0.8182	0.4500	0.3947	0.5517	0.6304	0.6010

Figure 11 - BOW--Topic Model Prediction results



BOW-TF-IDF MODEL: Accuracy						
	Financial	Consumer	Communication	Technology	Healthcare	All Sectors
Multinomial NB	0.5000	0.3810	0.5000	0.4839	0.5000	0.5233
Support Vector Cl.	0.5192	0.4286	0.5000	0.5000	0.5000	0.5305
Gradient Boosting	0.5000	0.4921	0.4375	0.4839	0.53571	0.5376
Neural Network	0.7115	0.4127	0.5000	0.4839	0.6071	0.5771
Random Forest	0.5385	0.4762	0.5208	0.5161	0.5714	0.5018
BOW-TF-IDF MODEL: Precision for Class 1						
Multinomial NB	0.4800	0.4167	0.5000	0.4898	0.5000	0.5258
Support Vector Cl.	0.4898	0.5263	0.5000	0.5349	0.5000	0.5314
Gradient Boosting	0.4839	0.6087	0.5484	0.5588	0.5500	0.6040
Neural Network	0.7037	0.4400	0.5000	0.5714	0.6250	0.5778
Random Forest	0.5250	0.5333	0.5333	0.5385	0.5714	0.5092

Figure 12 - BOW-TFIDF Prediction results

For text representation, we applied three ensemble methods:

a) Part-Of-Speech & Topic Model;
b) Bag-Of-Words & Topic Model;
c) Bag-Of-Words & TF-IDF.

The transformed data from each of the three ensemble methods were then fed into five machine learning algorithms:

a) Multinomial Naive Bayes;
b) Support Vector Classification;
c) Gradient Boosting;
d) Natural Network;
e) Random Forest.

The above three charts are the performance matrices for all text representation and model combinations. We only demonstrated the precision score for Class 1 because it counts for 50% of the data. The highlighted numbers represent the highest within each category.

Here are some interesting findings from the performance matrices:

a). Random Forest performs the best with the topic model while Gradient Boosting performs the best with TF-IDF models.
b). Financial sector and healthcare sector perform significantly better than the other sectors.

We think the reason behind the financial and healthcare sectors' outperformance is that the indicators of success for these two sectors are more commonly shared among the firms within the sector. For example, typical phrases like 'FDA approved', 'BLA submitted', 'phase III clinical trial concluded' are highly correlated with a healthcare company's commercial success and hence make the model more predictive. In comparison, companies in the communication services sector vary vastly in terms of their indicators of success; publication, online streaming, and social media companies are all categorized in this sector but the measures of their commercial success have little overlap.

Part III. Trading Backtest Performance

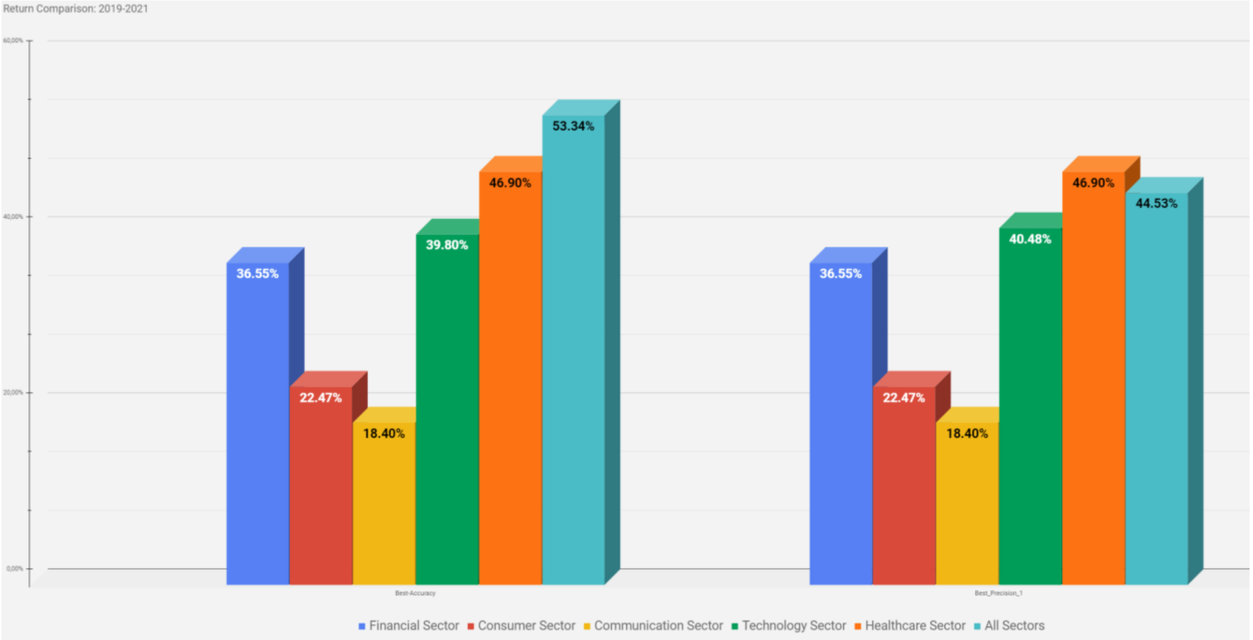


Figure 13 - Quarterly-Compound Total Return by Sector (Long/Short)

Based on model performance, we extracted trading signals from the best models for each sector as well as the aggregated all-sector. Then, we performed the backtest with the day trade strategy. Here are some important highlights from the backtest:

a). The All Sector model yielded the highest quarterly compound return of 55%, which surpassed the highest 3-year return of 32% from the S&P500 ETF.

b). Apart from the All Sector model, Financial Sector and Healthcare Sector still outperforms the rest as expected, while Communication Sector and Consumer Discretionary Sectors delivered the lowest returns.

Conclusion and Future Directions

1. Proof of Concept

Our text mining project proved the concept that running machine learning models on news articles with the help of text representation methods may be leveraged for event-driven trading in the equity market.

2. Potential Tool for Speculative Trading as an Addition to Conventional Portfolios

With the help of sentiment analysis, investors may use text mining as a scope for understanding market sentiment toward certain companies or industries so that they may gain a more holistic picture of the market outlook. This sort of analytics could be built into their conventional portfolio management toolkits.

3. Challenges & Limitations:

a) Since news coverage is largely concentrated on big or popular firms, our model might not perform optimally in predicting the price movement of smaller firms.
b) Our model was built on data from the past three years, during which we have witnessed several extreme events which may bring distortion or bias into our model.
c) A sector-based model may not perform well on all companies within the sector given that industry-level variances may greatly impact the text mining performance.

4. Future Direction:

a) We could expand the time range of the news coverage to include more market conditions.
b) We could expand the number of companies that we cover and build an industry-based model rather than a sector-based model.
c) We could include indicators from the fundamental analysis in our model training.
d) We could consider text mining on financial fillings such as 10-Q, 10-K, and 8-K.

References

1. Loughran, Tim, and McDonald, Bill, *When is a Liability, Not a Liability? Textual Analysis, Dictionaries, and 10-Ks* (March 4, 2010). Journal of Finance, Forthcoming, Available at SSRN: https://ssrn.com/abstract=1331573

2. Multiple Authors, Multiple Articles, Barron's, https://www.barrons.com/

3. Multiple Authors, Multiple Articles, Market Watch, https://www.marketwatch.com/

4. Multiple Authors, Multiple Articles, Wall Street Journal, https://www.wsj.com/