Evan Helig, Dimetrius Johnson, and Sean Deery

[eahelig@syr.edu](mailto:eahelig@syr.edu), [djohns66@syr.edu](mailto:djohns66@syr.edu), [sdeery@syr.edu](mailto:sdeery@syr.edu)

Final Project

Topic and Sentiment Classification

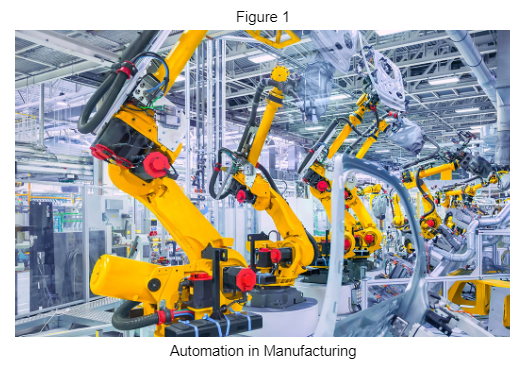
Topic and Sentiment Classification of New Articles

# Introduction

Financial companies and investors leverage stock analysis to help navigate through the ever-changing stock market. This analysis offers insight into market trends, guiding buy and sell decisions, and aiding companies in evaluating their own financial trajectories. While a current snapshot provides valuable information, the real art lies in predicting future market motivations.

Stock analysis involves comparing a company’s current position to its position in previous years. This information can be used to determine if the company is growing, stable, or deteriorating. The news is one source of information that guides many people on how they invest. The news captures topics and sentiment about major events, companies, and industries. One example of this, relatively recently, is how COVID-19 impacted different markets and the media’s dissemination of the information.

In each market, changes in laws, supply and demand, natural disasters, and innovation could cause significant changes in the economy. Agriculture is affected by natural disasters, climate change , and efforts toward sustainability. Sustainability trends in agriculture include a focus on transparency, circular economies, and reducing greenhouse gas emissions. Housing is affected by changes in Federal rates, mortgage rates, and supply and demand. Manufacturing is affected by supply chain issues, workforce disruption issues, and innovative advances in automation as shown in Figure 1 and AI. Technology continues to advance rapidly in fields such as AI, quantum computing, and internet-of-things as just a couple examples. The stock market is affected by market changes like tech stocks growing due to cloud computing, generative AI, and enterprise database software.



As these changes are communicated through news headlines, both companies and investors recalibrate their strategies in response. Companies look for opportunities or potential liabilities and investors want to maximize their profits. There are people who want to stay updated on the latest news, but because of the abundance of news, it can be time consuming to go through it all. It can also be hard to aggregate the different opinions about the different topics.

Therefore, this analysis seeks to accomplish two things. The first is to classify news articles by market topic and second is to predict the sentiment of news articles by topic. These are only two parts of what would be a much larger system that can make predictions on stock prices. The sentiment analysis would be combined with other analysis to make predictions on what is going to happen.

# Analysis

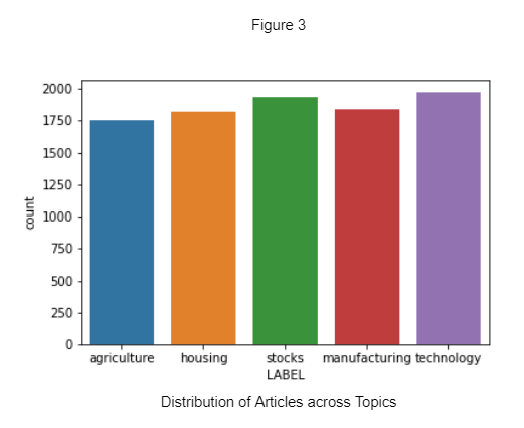
## Data

### News Articles Labeled By Topic

The news articles labeled by topic were acquired from News API through the newsapi.org website. News API allows developers to search through every article published by over 80,000 news sources and blogs in the last 5 years. The topics that were searched are "agriculture", "housing", "stocks", "manufacturing", and "technology". With 100 requests allowed per day, and 100 articles per request, 9,952 articles were retrieved between 8/26/2023 and 9/14/2023.

The data was then stored in a CSV file with columns for the topic name, the date, the source, the title, and the headline. The data was stored in a CSV to achieve consistent results during the development of the modeling. The CSV was then read into a data frame. 621 duplicate rows and 4 rows with missing data are removed. The final number of articles in the dataset was 9,328. Figure 2 shows an example of the label and raw text for each topic. Figure 3 shows the distribution of articles across the topics was fairly balanced, which helps the analysis avoid an issue with bias towards a particular topic.





### News Articles Labeled by Sentiment

The data for sentiment analysis was retrieved from Kaggle. The dataset is called Financial Sentiment Analysis and can be found at <https://www.kaggle.com/datasets/sbhatti/financial-sentiment-analysis>. The data includes financial text along with sentiment labels. It is a combination of two datasets: FIQA and Financial PhraseBank. The data includes 5,842 sentences. There were 6 duplicate rows that were removed which resulted in 5,836 sentences. Figure 4 shows an example label and raw text for each sentiment label.

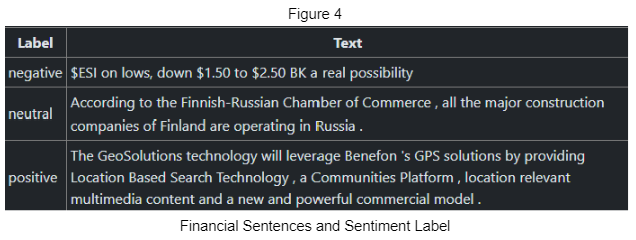
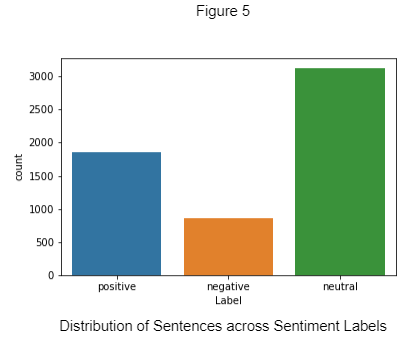
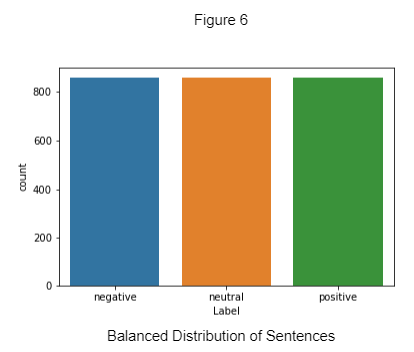


Figure 5 below shows the distribution of 5,836 sentences across the sentiment labels ‘positive’, ‘negative’, and ‘neutral’. The count plot shows the data is unbalanced and prediction models might end up biased towards the neutral label. The data was balanced and stored in a separate data frame to run as an experiment against using all the data. Figure 6 shows the balanced dataset containing 2,580 sentences.





## Vectorization

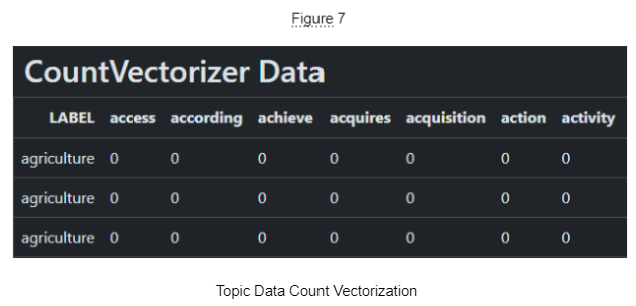
For the computer to be able to analyze the text data in a table, it needs to be put through a process called vectorization. In vectorization, each unique word in the corpus is set as a column, and each post is set as a row (Weiss, Indurkhya & Zhang, 2015). In a term frequency vectorizer, the cells indicate the frequency count of each word in each post. There are different methods of vectorization including term frequency and tf-idf which this analysis will focus on. For the term frequency, the cell values show the frequency count of each word in each review. For tf-idf, the cell values show the term frequency - inverse document frequency, which is the frequency of the word in each post, offset by the number of times the word appears in the entire collection of documents.

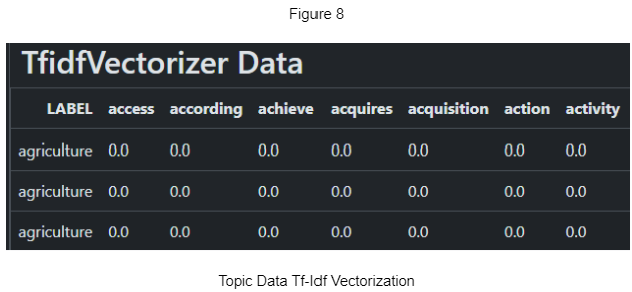
When vectorizing text, there are a couple of options to consider. The first option is whether or not each word should be changed to lowercase, uppercase, or kept the way they are. Leaving the words the way they are could allow the algorithm to pick up on differences between common nouns and proper nouns, but it would also force the algorithm to consider the same word as different words depending on the capitalization. A second option to consider in vectorization is whether to set a maximum number of features to include. Using all of the words in a corpus is computationally expensive and could lead to overfitting. A third option to consider is whether to remove stopwords. Words that occur in most documents are not useful in distinguishing documents. Stopwords are usually function words that bear no specific meaning, compared to content words (Weiss, Indurkhya & Zhang, p. 21-26, 2015).

There are a variety of tools to vectorize text data. This analysis focuses on the CountVectorizer and TfidfVectorizer tools from Python’s Scikit-Learn package. CountVectorizer contains multiple options that can help the analysis including setting all the words to lowercase, setting a maximum number of features, and removing stopwords. Since the goal is topic and sentiment analysis, the algorithm does not need to differentiate the proper nouns so adding the lowercase option is helpful. Setting the maximum number of features helps to avoid overfitting and helps reduce the computational expense. Lastly, removing the function words with the stopwords option would help remove noise from the data and allow the algorithm to focus on analyzing sentiment and classifying topics as accurately as possible.

### Vectorization For Topic Analysis

Therefore, the best mix of options for the topic analysis would be to apply the lowercase parameters, set max features to 1500, and remove English stop words. For the topic analysis, the topic words used to search for the articles were added to the stop words to be removed. The tokens were then combined with the labels into data frames, resulting in two data frames shown in Figures 7 and 8 (Weiss, Indurkhya & Zhang, p.156, 2015).





After vectorization, word clouds were created for articles as a whole and by topic. Figure 9 shows the overall words. It shows the words that appear the most over the 20 day period. It includes the countries China and India. It has some words that are used frequently in finance like stock, market, and dividend. It also has some company specific words like Apple, and iPhone.

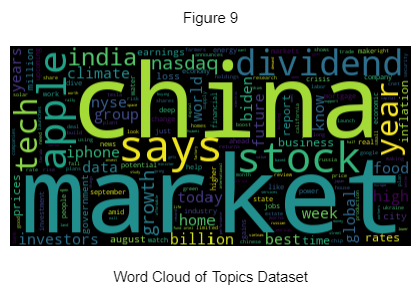


Figure 10 shows the word cloud for the agriculture topic. The words climate, farmers, food, water, global, and world appear the largest, which makes sense. There are also a number of places including India, Africa, Russia, and Florida.

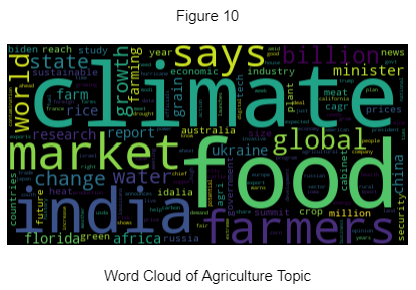


Figure 11 shows the word cloud for the housing topic.The words that come up the largest include home, market, mortgage, and rates which makes sense. The word crisis appears harem which is interesting because that word indicates negative sentiment.

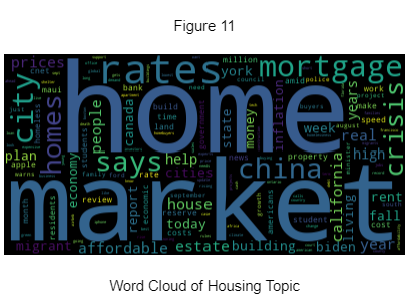


Figure 12 shows the word cloud for manufacturing. The words that come up largest are Apple, China, billion, and Biden. Many of the smaller words seem to be finance words like market, but there are also more places and companies like Tesla.

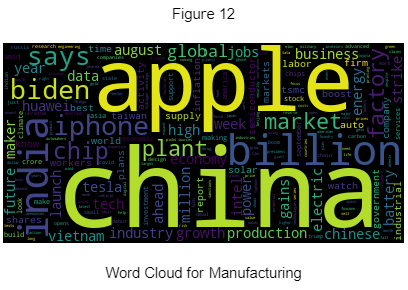


Figure 13 shows the word cloud for stocks. It shows more words that are related to stocks like dividend, stock, and market. This word cloud seems to be filled with finance and stock market buzzwords.

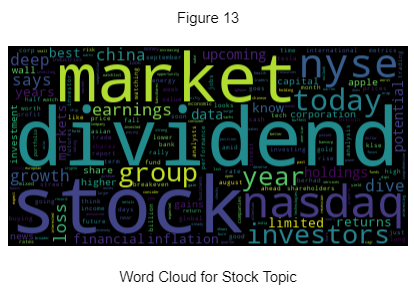
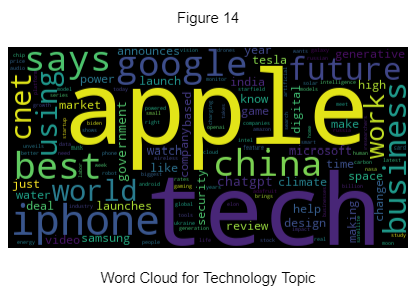
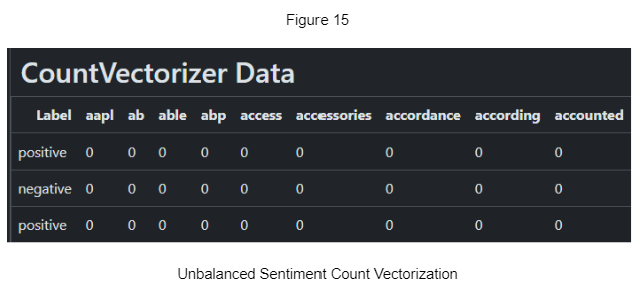


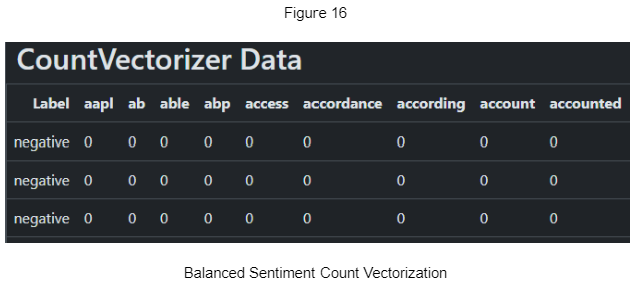
Figure 14 shows the word cloud for technology. Not surprisingly, Apple is at the top of the list along with Google and the country China. There are some products like iPhone and ChatGPT.

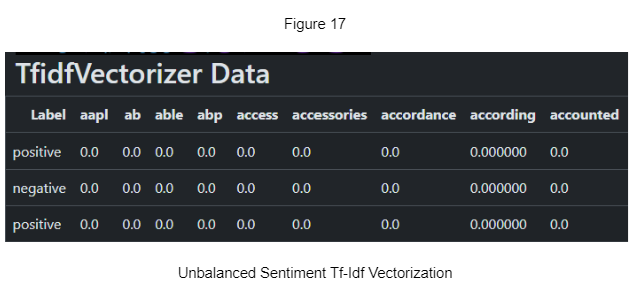


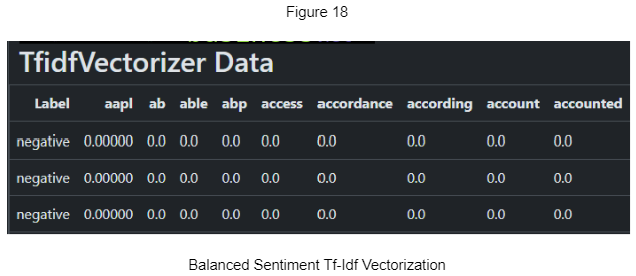
### Vectorization for Sentiment Analysis

The best mix of options for the sentiment analysis would be to apply the lowercase parameters, set max features to 1500, and remove English stop words. Once the text data was vectorized, tokens with numerical characters were removed. The tokens were then combined with the labels into data frames, resulting in four data frames shown in Figures 15 through 18. There are four data frames here because the sentiment data has already been split into balanced and unbalanced datasets (Weiss, Indurkhya & Zhang, p.156, 2015).

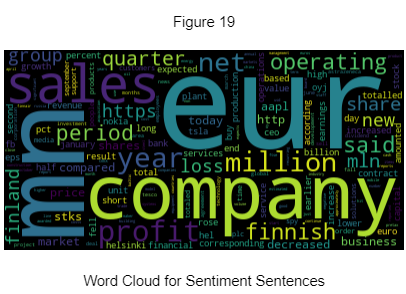




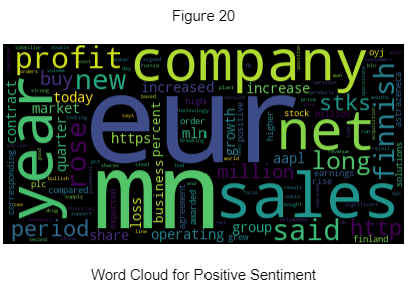


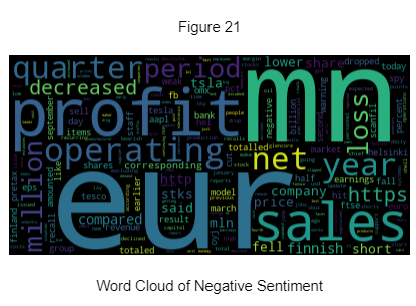


After vectorization, the word clouds for the sentiment data were created.Figure 19 shows the word cloud for the overall words in the sentiment data set. It shows eur and mn which would commonly be used in financial news. It seems like most of the words are function words used in finance.

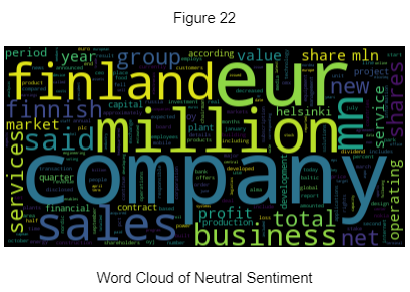


Figures 20 and 21 show the positive and negative word clouds. They appear to be almost identical to the overall words.





The word cloud in Figure 22 shows the words for neutral sentiment. This one is more interesting because it shows Finland and Finnish. This indicates there must have been a lot of subjective news about Finland in this dataset.



## Models

### Naive Bayes

Naïve Bayes is a classification algorithm that uses Bayes’ theorem to classify objects. The word “naïve” comes from the fact that this model assumes each word is independent of each other. This assumption is not true in text data, but the model can still perform accurately enough to give useful results. The way the algorithm works is that it calculates probabilities based on word frequencies (Weiss, Indurkhya & Zhang, p.58, 2015). One problem that comes up is that some words end up with a probability of zero. Since it is more realistic to have a very small probability than it is to have zero probability, a method called smoothing is used to make sure words have at least a very small probability assigned to them. This analysis uses the ScitKit Learn package MultinomialNB to create the Naïve Bayes model. The model is tuned with the alpha parameter which affects the level of smoothing the algorithm uses.

### Support Vector Machine

A Support Vector Machine, or SVM, is a classification algorithm that finds a linear hyperplane that will separate the data, thus creating a decision boundary. It finds the hyperplane that maximizes the margin between the data points of the labels. It does this by identifying the support vectors, which are training examples that are located on the margins. The rest of the training examples do not participate in the prediction process. The number of support vectors is an indicator of the complexity of the trained SVM model (Weiss, Indurkhya & Zhang, p.64, 2015). In prediction, the distance between the example and the decision boundary is an indicator of prediction confidence; the farther the better. If the labels are not linearly separable, a soft margin can be imposed which allows some training data points to be on the wrong side of the decision boundary. A soft margin can be tuned with the parameter C, which represents the penalty for misclassification. Another way to handle data that is not linearly separable is to apply a kernel function. Setting a different kernel results in the algorithm creating a new dimension to plot the training data across which can make it linearly separable. Picture a bed sheet with an archery target design, and the goal is to cut out the center circle. One might pinch the center of the sheet and pull the sheet up (creating a new dimension) to cut out the center circle (create a hyperplane). Typical kernel functions include a Polynomial kernel, a Gaussian radial basis function kernel, and a Sigmoid kernel. The most common kernel to use for text data is the linear kernel because most text data is linearly separable and there are a large number of features after vectorization. Other kernels are more likely to overfit the data when dealing with text.

### Decision Tree

“Decision Trees are special decision rules that are organized into a tree structure. It divides the document space into non-overlapping regions as its leaves, and predictions are made at each leaf” (Weiss, Indurkhya & Zhang, p.56, 2015). The major difference between Decision Trees and Naïve Bayes is that Decision Trees are non-linear and Naïve Bayes is a linear model. Decision Trees can also be visualized as an upside-down tree, where branch nodes are splits and the leaves are the predictions. Parameters that can be tuned include the criterion it uses to determine how to split the data, and how many splits or leaves there should be. The SciKit Learn package used is the DecisionTreeClassifier. It is worth noting Decision Trees are generally not used in text classification because of their weaknesses in handling mixed sentiment and a high number of dimensions. This analysis seeks to compare the results with Naïve Bayes and SVM which are used more commonly with text classification.

### LDA

LDA is an algorithm that uses Bayesian probabilities to find the hidden topics in a corpus of text. A topic is defined as a distribution over a vocabulary (Blei, Ng & Jordan, p.996, 2003). The algorithm assumes each document is a mixture of topics, and each topic is a distribution of word weights. The number of topics is a parameter that the algorithm assumes is the total amount of topics in the text. LDA’s generative process involves first randomly choosing a distribution over topics. Then, for each word in the document, the algorithm randomly chooses a topic from the distribution over topics in step 1, and then randomly chooses a word from the corresponding distribution over the vocabulary. For this analysis, the algorithm was run with the number-of-topics parameter set to 5, 10, 20, 30, 40, and 50. The number of actual topics is not known, so this range is run to get a variety of results that can be chosen from.

# Results

## Topic Classification

The results of the topic classification show the model with the highest 3-fold cross validation and test accuracy was the Naive Bayes model run with the data vectorized by count. Figure 23 shows the results of all the models run in order from most accurate to least.

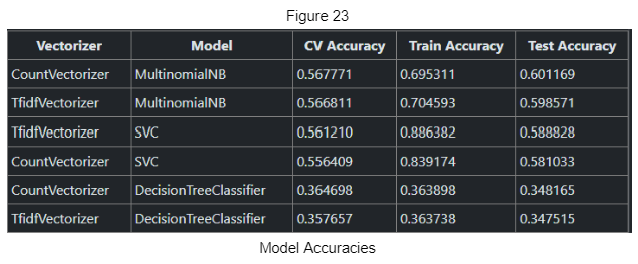
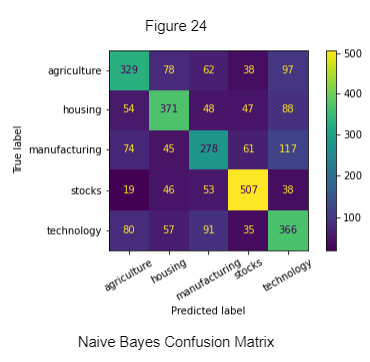


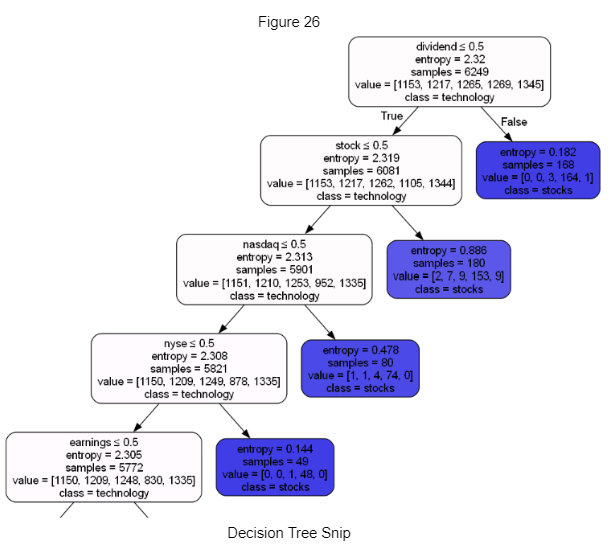
Figure 24 below shows the confusion matrix for the Naive Bayes model run with the data vectorized by count. The topic the model was able to predict the best was stocks, which had the highest precision of 74% and the highest recall of 76%. A high precision means the model returns more relevant results and less irrelevant ones for stocks. A high recall means the model returns most of the relevant results whether or not irrelevant ones are also returned. The worst precision was technology and the worst recall was stocks.



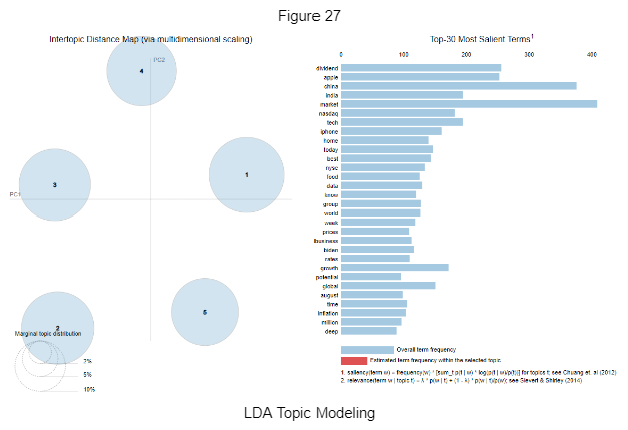
The Naive Bayes model also returned log probabilities of each word for each label. Figure 25 below shows the top 10 most important words in each topic for the prediction. From the word clouds above, it appears the algorithm is picking up on similar words like food and climate for agriculture and home and market for housing. Manufacturing and technology have more overlap. They even share ‘apple’ and ‘iphone’ in their top 10 words.



Even though the decision tree performed the worst at 35% accuracy, it still provides an interesting visualization that shows how the algorithm made its decisions. Figure 26 shows a small portion of the decision tree that shows how it predicted the stocks topic based on the words dividend, stock, nasdaq, nyse, and earnings.



Lastly, an LDA analysis was able to cluster the data into 5 distinct groups. When plotted in two dimensions, Figure 27 shows that all the topics are clearly separated. The words in each of the topics seem to be a mix of topics. One topic is clearly about housing and one is clearly about agriculture. Stock, technology, and manufacturing seem to have similar mixes of words.



## Sentiment Classification

The results of the sentiment classification show the model that had the highest classification was the Support Vector Machine run on the unbalanced data vectorized with Tf-Idf with a test accuracy of 70%. To compare with the balanced dataset, the accuracy for the SVM on balanced data is 64%. Figure 28 below shows the accuracies for each of the models.

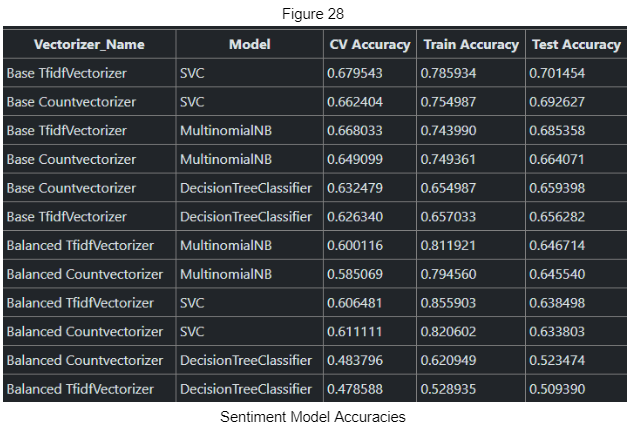
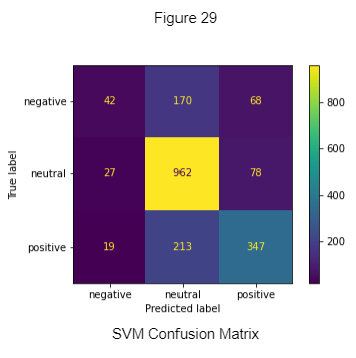
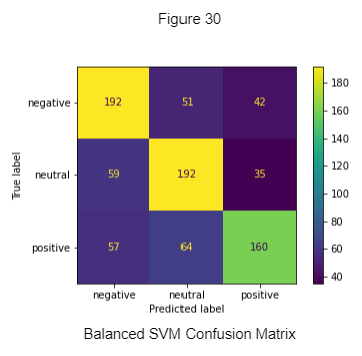


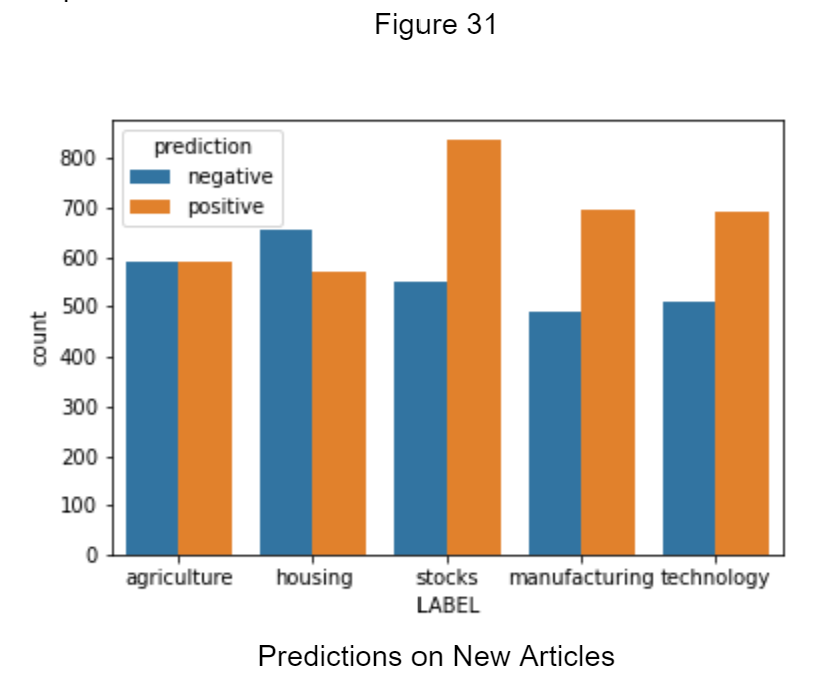
Figure 29 below shows the confusion matrix for the Support Vector Machine run on the unbalanced data vectorized with Tf-idf. Neutral had the highest precision of 72% and the highest recall of 90%. Negative had particularly low scores of a 48% precision and a 15% recall.



To show the difference of balancing the dataset, Figure 30 below shows the confusion matrix for the SVM model run on the balanced dataset vectorized with Tf-Idf. Neutral has a little less precision at 68% and recall at 57%, but negative now has a precision of 63% and a recall of 67%. Positive is not far from negative with 62% precision and 67% recall. Even though this model has a lower test accuracy than the model run on unbalanced data, this model will be better to use for sentiment analysis of news articles. Using the balanced dataset will allow the model to be reasonably accurate with each of the sentiments (Ohana & Tierney, p.4, 2009).



The result of the sentiment analysis was a model that can classify news articles as positive, negative or neutral. Figure 31 below shows the results of predicting the new articles from the topic analysis. The blue on the left represents negative predictions, and the orange on the right represents the positive predictions. This data could be used along with other financial models to create an accurate predictor.



# Conclusion

Traders on Wall Street are always working towards anticipating tomorrow’s changes in stock prices. One way to do this is to look at the financial numbers and make a forecast based on previous performance. The problem with this is that it doesn’t consider major events unfolding and changing the directory of the economy. To account for this investors look to the news to know what is going on and what people are talking about. Financial companies hire teams of analysts to set up computers to read the news. They combine this textual analysis with traditional financial analysis to make predictions about future prices.

Setting up a computer to read the news is a task that has different levels. First, the computer needs to access the news. Some places might have the news neatly arranged in categories, but others might not. It might be left up to the reader to understand what the category is. For a computer to do this, an analyst needs to set up a topic classifier. This tool predicts what topic a piece of text is about. With this tool, companies can add more data to the next steps in the process.

The next step after getting the news is to analyze it. One could take a deeper look into categories. Another analysis that can be done is a sentiment analysis. A sentiment analysis is about determining if the words in the text are more positive, more negative, or neutral. Now that the news is already organized into categories, one could run a sentiment analysis on a topic and see how positive or negative things are within that topic.

Once news data has been categorized into topics and the sentiment from each topic is retrieved, the next step would be to combine the results with other types of analysis like a financial analysis to make predictions about future stock prices. The goal of this analysis was to create a topic classifier and a sentiment classifier for the first two steps in the process.

The results of the topic classifier showed that with enough data, it was able to predict the correct topic 60% of the time. This might seem low, but for there being natural crossover in the topics, five possible outcomes, and having removed the topic words, these results are worth considering. What is even more exciting is that the model was not significantly biased towards any one topic. The results of the sentiment classifier showed that it was able to predict the correct sentiment 64% of the time. There was another model with a higher test accuracy but it couldn’t predict negative sentiment well, which makes it not so useful for stock analysis. These models can now be used in a larger system with a goal of predicting stock prices.

# Resources

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003, January). *Latent Dirichlet Allocation*. Journal of Machine Learning Research. https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=

Ohana, B., & Tierney, B. (2009). *Sentiment Classification of Reviews Using SentiWordNet*. Arrow@TU Dublin. https://arrow.tudublin.ie/

Weiss, S. M., Indurkhya, N., & Zhang, T. (2015). *Fundamentals of Predictive Text Mining* (2nd ed.). Springer London Heidelberg New York Dordrecht.