Sentiment Analysis of Company Related Articles to Predict Stock Price Direction

Raul Martinez (raulmart@umich.edu), Vincent Lee (leevw@umich.edu), Xincheng(Steven) Song (steveso@umich.edu)

Introduction

In the past decade, the financial landscape has undergone significant transformation. Wall Street, formerly a loud and chaotic environment with traders shouting orders across the floor and manually executing trades, has largely been replaced by algorithmic trading. This shift has been driven by the increasing success of algorithmic approaches in outperforming traditional fundamental analysis. Our group was motivated to explore algorithmic trading because it is a broad and quickly evolving research area with plenty of room to improve and grow.

In this project, we leveraged unsupervised topic models and sentiment scorers to analyze and quantify sentiment. We then utilized these sentiment scores as a feature in our supervised classification models to predict future stock price movements. Our ultimate goal was to create an algorithm that can effectively predict profitable trades and “beat the market”. While we will go into more detail later, our final model was able to predict the direction of a stock’s movement over 50% of the time.

Related work

1. Elon Musk Twitter Predictor: https://github.com/WithDD97/Twitter-sentiment-analysis-Elon-Musk/blob/main/Machine%20learning.ipynb

This project scrapes Tweets from Elon Musk, and uses sentiment analysis on them to predict stock price movements for TSLA. We are using news articles, which is different from this project that exclusively uses Twitter data

2. Stock Price Prediction using Sentiment Analysis and Deep Learning for Indian Markets: https://arxiv.org/pdf/2204.05783.pdf

This project uses sentiment analysis in conjunction with deep learning and other regression techniques. This project is different from ours because it used headlines only while we used the whole article, and it focuses on Indian stocks.

3. A sentiment analysis approach to the prediction of market volatility: https://www.frontiersin.org/articles/10.3389/frai.2022.836809/full

This project uses VADER sentiment analyzer and LDA topic modeling to predict daily changes in the FTSE100 returns and volatility. Our project used a longer time horizon than one day and focused on returns of specific stocks only.

Unsupervised Learning

Data Source

The python package benzinga is an implementation of Benzinga News API and was used to get the news articles. This package contains the ability to retrieve information about news articles over specific dates, pertaining to specific stock tickers. The team used this package to get news articles from 2019-01-04 to 2023-01-04 for the stock ticker AAPL. The API returned lists of dictionaries which contained important variables needed for sentiment analysis including the creation date of the article, the title of the article, the ticker for the stock and a url link for the entire article. However, the API did not provide the text of the actual article, to retrieve the text for the articles and persist them into the database

a web scraper was built, which would loop through each of the url's and scrape the article from the benzinga website, once the article was scraped it would update the row for the article scraped. This produced 5375 records for AAPL.

Feature engineering

Since the benzinga API contained a rate limit, the team utilized a PostGres database to persist data. This allowed for the team to query the data once from the API and not need to worry about the rate limit. The API returned the url for the individual news articles, but did not provide the actual article itself. To retrieve the articles and persist them into the database a web scraper was built, which would loop through each of the url's in the database and scrape the article from the benzinga website, once the article was scraped it would update the row for the article scraped. All results were stored in a database table called stock_news.

The final features we used for our unsupervised **topic model** were:

Feature	Description	Source
Stock Ticker	The unique stock ticker the article was related to	benzinga
Article Title	The title of the article	benzinga
Article url	The url link to the article	benzinga
Article Contents	Text from the article, which was the main focus of the sentiment analysis	benzinga
Article Date	The date the article was posted	benzinga

The final features we used for our unsupervised **sentiment scoring** model were:

Feature	Description	Source
Topics Probability Distribution	The probability of the belong to each topic	BERTopic Model
Topic Representative Documents	The documents represent each topic	BERTopic Model
Article Date	The date the article was posted	BERTopic Model

Methods description

The general process we were trying to achieve with the unsupervised model was raw text -> preprocessing -> sentiment scoring.

Here are the following preprocessing techniques that we tried but **did not** use:

Whole Article: FinBERT had a character limit, so we used a sliding window to analyze large texts in chunks (articles over the character limit were not split into chunks). We thought this would leave the most context, especially because FinBERT has its own preprocessing when analyzing sentiment. However, we think that topic modeling before sentiment analysis was ultimately better because it removed irrelevant details and noise. We tried this approach with both FinBERT and FinBERT-Tone sentiment scoring.

LDA: When we were manually checking the topics from our LDA topic modeler, we noticed that the topics were not very related to the article, which resulted in very neutral sentiment scores that did not help us much with prediction. We will discuss the decision to use topic modeling with our final model.

sBERT summary: Using sBERT for summarization removed too many pieces of context without identifying underlying themes and subtopics, so we think that using summarization as a preprocessing technique led to misinterpretations of sentiment. We tried this approach with both FinBERT and FinBERT-Tone sentiment scoring.

Here are the following sentiment scoring techniques that we tried but **did not** use:

AFINN: This sentiment analyzer represents text as a bag of words to analyze sentiment, which excludes crucial context clues. It is also not tailored to financial text, which is an important factor for us.

finVADER: This sentiment analyzer is tailored to financial text. However, it uses a sentiment lexicon to generate scores, which is a rigid approach that can fail to capture nuance such as phrasing and word order.

FinBERT-tone: This sentiment analyzer is very similar to our final model, but it led to overly confident sentiment scores that skewed towards extremes. It is possible that because FinBERT-tone was trained on a significantly larger dataset than our final model, FinBERT, it could be susceptible to overfitting.

Here is the final process for our unsupervised model that we **did** use:

Preprocessing: We decided to do topic modeling on our raw text before inputting the topics into our sentiment analyzer. This is because topic modeling can help reveal subtle and nuanced sentiment that might not be apparent in raw text. It also reduces noise and irrelevant details by focusing on specific topics, which leads to more accurate sentiment analysis. Adding a topic modeling step does add complexity and can remove useful context, but the team thought the tradeoff did not outweigh the benefits.

Our topic modeler that yielded the best results in manual evaluation and testing with our supervised model was BERTopic with tf-idf vectorizer. We were concerned about using a different embedding for topic modeling and sentiment scoring, but we noticed that using the FinBERT embedding for BERTopic yielded unrelated and neutral topics, which was the same problem we experienced with LDA. After BERTopic does vectorization, it reduces the dimensionality with uniform manifold approximation and projection (umap), clusters the data with hierarchical density based spatial clustering (hdbscan), then represents the topics with a modified version of tf-idf.

The accompanying visualization delineates the document-topic relationship as established through BerTopic's topic modeling framework. In this representation, individual documents are denoted by dots, with the color coding of each dot corresponding to its categorized topic. According to the legend, 19 topics are identified, excluding outliers represented by topic -1, thus not featured in the chart. Each topic's nomenclature reflects its defining keywords, providing an intuitive grasp of its thematic essence. For instance, Topic 3, labeled as _teslas_ev_electric, specifically pertains to discussions around Tesla and electric vehicles, illustrating the method's effectiveness in distilling and categorizing thematic content.

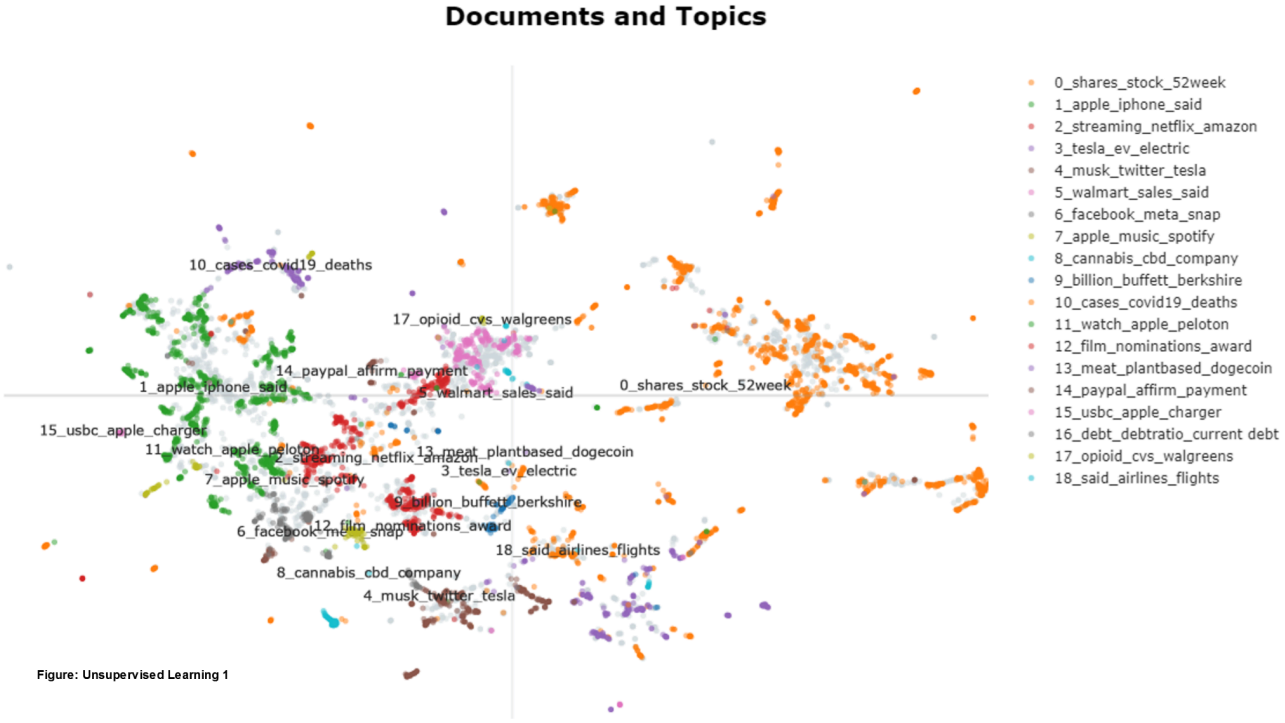

Figure: Unsupervised Learning 1

Sentiment Scoring: FinBERT's sentiment analyzer is tuned on large financial datasets. It relies on deep learning architectures that can understand complex sentence structures and context that a lexicon based approach cannot. FinBERT is also frequently updated by learning and training on new text, which allows it to maintain accuracy as financial linguistics change over time.

In our approach to calculating sentiment scores for articles, we integrated the capabilities of BERTopic and FinBERT. BERTopic served as our primary topic modeling tool, identifying 20 distinct topics within our dataset. This segmentation, as illustrated in Unsupervised Learning Figure 1, excludes outliers (represented by topic -1 and thus not shown in the chart), with each article linked to a specific topic through visual markers. The output from BERTopic yielded two pivotal sets of data: Topic Information and Topic Probability Distributions. Topic Information provides insights into the three most representative documents for each topic, highlighting articles with the strongest thematic ties. Applying FinBERT to these documents enabled us to assign a sentiment score to each topic. Topic Probability Distribution, on the other hand, assigns each article a set of probabilities indicating its alignment with the identified topics. These probabilities, when applied to the topic-specific sentiment scores, facilitate a nuanced calculation of sentiment for each article. Our comprehensive analysis culminated in the derivation of a composite sentiment score for the dataset, employing a weighted formula:

Composite Score = (1 × positive_score) + (0 × neutral_score) + (−1 × negative_score)

This composite score underpins the visualization presented in Unsupervised Learning Figure 2, where the x-axis denotes the composite sentiment score. Each dot, colored according to the article's sentiment, allows for an immediate visual assessment. Additionally, the titles of articles representing the extremities of sentiment—most negative, neutral, and positive—were examined to validate the coherence of our results.

The following visualization shows a distribution of sentiment scores from a sample of articles. A sample was used because showing the entire dataset would result in a very messy and busy visualization.

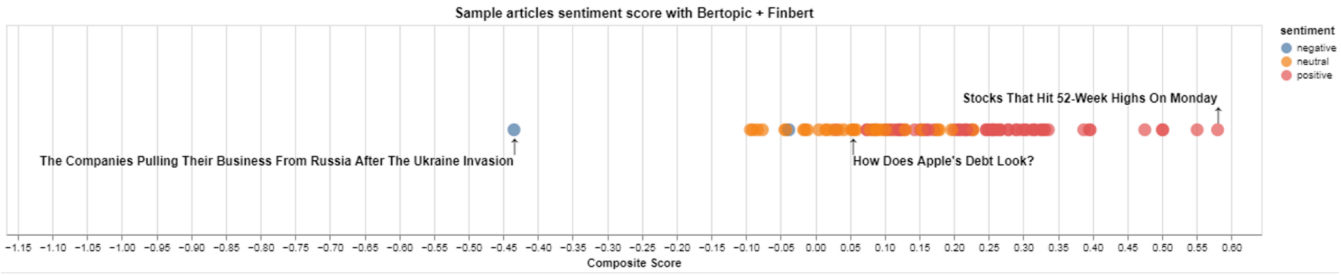

Figure: Unsupervised Learning 2

Hyper-Parameter Sensitivity Analysis

To understand the sensitivity of our sentiment scores to the number of topics (num_topics), we conducted a sensitivity analysis comparing 5-40 topics in increments of 5. As we can see in the graph below, over 15 topics, our distribution of positive, negative, and neutral scores do not change much. In the end, we decided to use 20 as the number of topics.

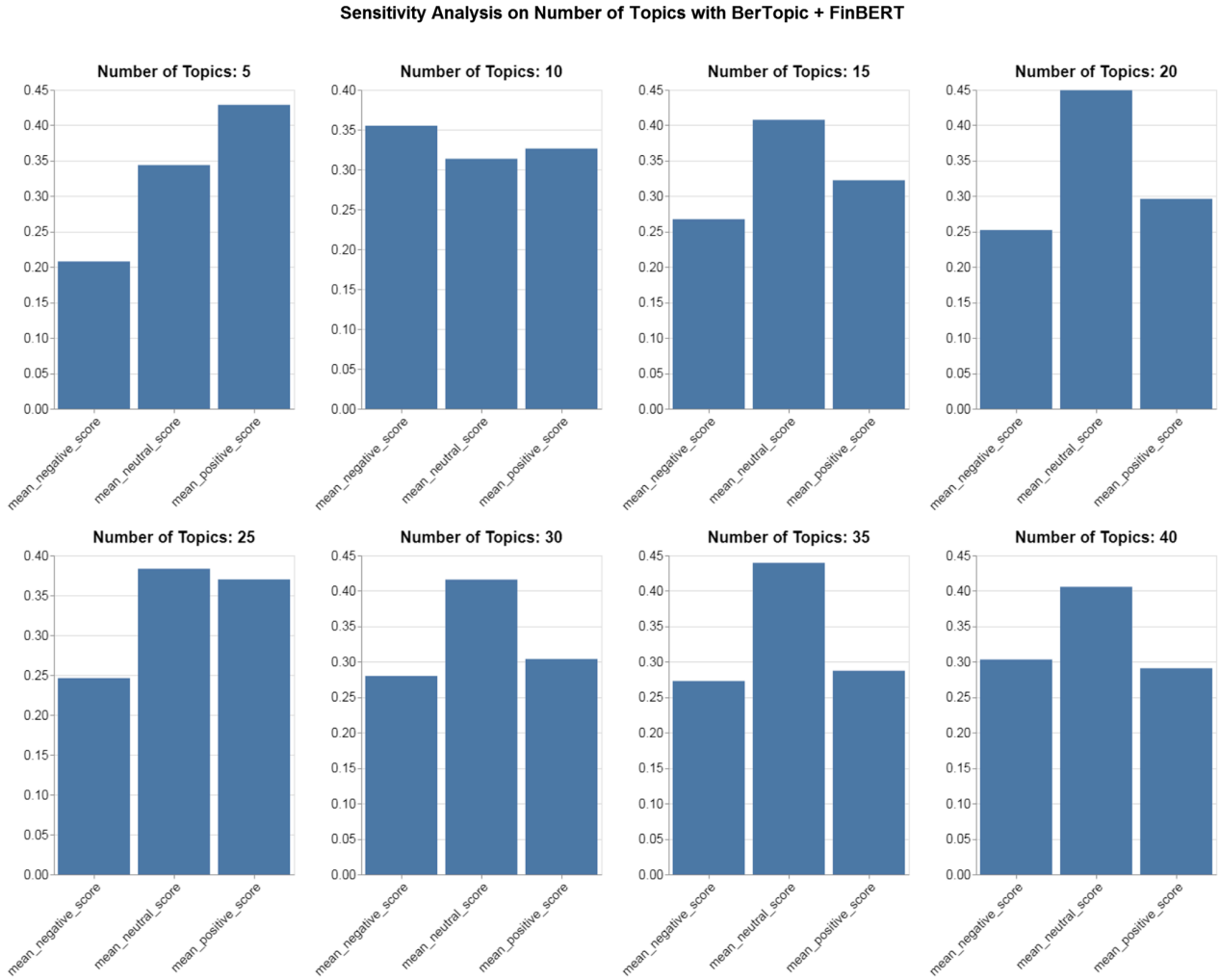

Figure: Unsupervised Learning 3

Unsupervised Evaluation

Evaluation is one of the major challenges with unsupervised learning. Even further, evaluation of a sentiment scorer is also difficult because sentiment is subjective. Different models and humans can read the same text, and come to different conclusions on the sentiment. We manually evaluated our sentiment scores by reading through the articles with the most positive, negative, and neutral sentiment scores, then adding green highlights for positive words/phases and red highlights for negative words/phrases. We expect the most positively scored articles to have a lot of green, the most negatively scored articles to have a lot of red, and the most neutrally scored articles to have very little highlighting. The articles are very large, so we added an example of this in Appendix A.

In the Supervised Evaluation section, we compared how the scores from our different sentiment scorers performed in our supervised model, which is another method we used to evaluate our unsupervised models.

Supervised Learning

Data Source

The python package yfinance was used to obtain daily stock price information. The team used this package to get stock prices from 2019-01-04 to 2023-01-04. The package returns records in a pandas DataFrame object and contains important predictive features such as open price, stock high, stock low. In addition, it contained the variable the team sought to predict: the closing price. This stock price data was combined with the matching sentiment data from the sentiment analysis scores from the Unsupervised Learning portion of the project.

Feature engineering

To prepare the data to be used by the supervised machine learning methods, the following steps were taken:

1. Retrieve Stock Price data: using the yfinance package, the daily stock prices were retrieved into a DataFrame.
2. Retrieve Stock Sentiment data: stock sentiments were retrieved from the database as a DataFrame. This DataFrame was grouped by date and aggregated the mean sentiment scores for positive, negative, and neutral sentiments. This was done to deal with days having multiple articles.
3. Merge DataFrames: the stock price DataFrame and the stock sentiment DataFrame were merged based on the date. The DataFrame is sorted on date.
4. Shift sentiment values: to avoid data leakage the stock sentiment scores for an article written on a particular day were shifted forward one day. This accounts for any article that may have been written after the close of the market.
5. Shift stock data features: The daily stock price information contains the open, high, low, and close. To be able to use these features without having data leakage, each of these values was shifted to the following day and renamed as a new column.
6. Add binary feature column for closing price: to determine whether or not the stock price increased for the day a binary feature column was added for 1 if the stock closed at a higher price than it opened and a 0 if it did not.

The final features for the DataFrame were:

Feature	Description	Source
open (numeric)	Opening price of the stock for the day.	yfinance
prev_high (numeric)	Highest price of the stock for the previous day.	Engineered from yfinance.
prev_low (numeric)	Lowest price of the stock for the previous day.	Engineered from yfinance.
prev_close (numeric)	Closing price of the stock for the previous day.	Engineered from yfinance.
positive (numeric)	Average positive sentiment for all articles written for the previous day.	Engineered from unsupervised learning output.
negative (numeric)	Average negative sentiment for all articles written for the previous day.	Unsupervised learning output.
neutral (numeric)	Average neutral sentiment for all articles written for the previous day.	Unsupervised learning output.
closed_higher (numeric)	Binary feature. This was the target label. 1 if the price closed higher, 0 if it did not.	Engineered from yfinance.

Methods description

The supervised learning workflow for all the models followed the same steps. The first processing step was to scale the data to prevent any feature from overpowering others; all predictive features were scaled using the MinMax scaler from sk-learn to be within a range of -1 to 1. To generate cross-validation splits, the team utilized TimeSeriesSplit, since the stock price data is a time-series, the training data must occur before the testing data, TimeSeriesSplit is built specifically for this use case. The data was split into 5 cross-validation splits each with a training size of 126 days, testing size of 31 days with 5 days of lag-time in between the training and testing data, this allowed for training sizes of 1/2 trading year with a testing window of just over a month of trading. With the data split, the team used GridSearchCV to test different hyperparameters for each model. This allowed the team to determine which hyperparameter combinations were most impactful by performing an exhaustive search of the different combinations in one function call, rather than manually testing each combination.

Four supervised machine learning (and one dummy classifier) algorithms were used to predict the target:

- Dummy Classification - Dummy model. This model was chosen as a baseline for all the actual supervised learning models as it always chose the majority class seen during fit.

- Logistic Regression - Linear model. This model was chosen as a base classifier.
- Nearest Neighbors Classification - Neighbors model. This model was chosen with the underlying assumption that data points with similar values (neighbors) would produce the same binary outcome.
- Random Forest Classifier - tree-based model in which the trees can be trained in parallel using bagging. Chosen as they are resistant to overfitting, with the final output being chosen from the combination of all the decision trees.
- Gradient Boosting Classifier - tree-based model in which the trees can be trained using boosting. Chosen because estimators have the ability to learn from the previous estimator's mistakes using gradient descent optimization.

Supervised Evaluation

In the unsupervised learning portion of the project the team evaluated several different methods of generating sentiment scores for articles, out of these methods 5 were evaluated for predictive performance. Below is the top performance for each sentiment algorithm:

Sentiment Generator	**Highest Mean Test Accuracy**	**Standard Deviation**
FinBERT sentiment scores on article summaries	0.58065	0.08893
FinBERT-tone sentiment scores on article summaries	0.55484	0.06255
FinBERT sentiment scores on entire article	0.56129	0.06321
FinBERT-tone sentiment scores on entire article	0.56774	0.07796
BERTopic with FinBERT sentiment scores	0.62581	0.12675

Out of all the approaches, the BERTopic algorithm with FinBERT was the most performant across the different supervised learning models in terms of accuracy. For this reason, it was chosen for further supervised learning evaluation.

Being able to predict whether or not a stock price will rise can provide useful information to the end user for both the positive and negative class. In addition, when looking at the spread of the target label, the dataset was very balanced. Additionally, there are equal penalties for incorrectly predicting either direction (false positive is as bad as false negative). For these reasons, the team chose to focus on accuracy as the main metric of evaluation over recall and precision.

Below are the results for the best model from each family with 5-fold cross validation:

Model	**Mean Test Accuracy**	**Standard Deviation**
Dummy Classifier	0.445161	0.080062
Logistic Regression	0.50323	0.07524

Nearest Neighbors Classification	0.54839	0.05398
Random Forest Classifier	0.58065	0.04562
Gradient Boosting Classifier	0.62581	0.12675

All the models were able to beat the Dummy Classifier. Out of all the models, the best performing model was Gradient Boosting. While overall not a great classifier of the data it contained a mean test accuracy approximately 24% higher than the worst performing model. However, it also contained the largest standard deviation of all the models.

Feature Importance Analysis

To interpret how the best model was utilizing the input features. The team used the *feature_importances_* property of the Gradient Boosting Classifier to see how the model weighted each input feature. Overall, most of the features were weighted rather equally with no one feature taking over 20% importance in the final output. However, three out of the four most important features *positive, open, negative, and neutral* were generated from the sentiment analysis.

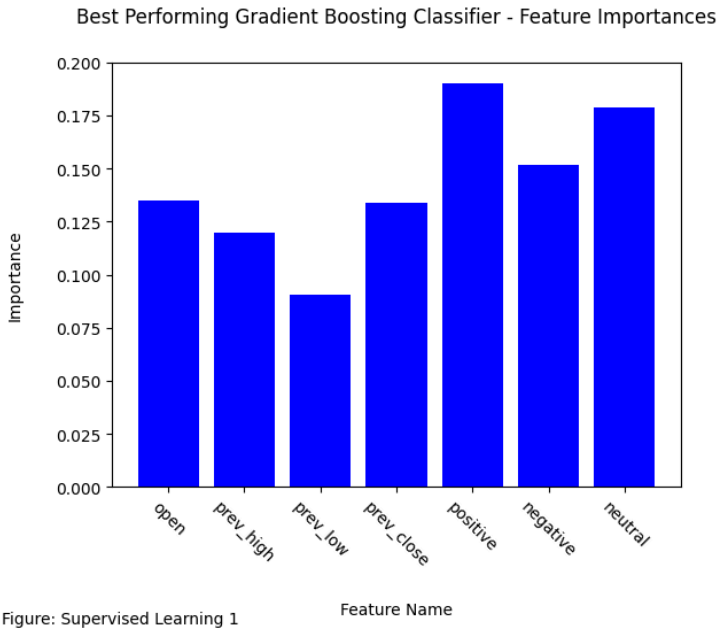

Figure: Supervised Learning 1

Feature Ablation

To understand if better accuracy could be obtained using a subset of the features, feature ablation was conducted on the best performing model. Below are the results for all the feature ablation test from each family with 5-fold cross validation:

Ablation Test	**Features**	**Mean Test Accuracy**	**Standard Deviation**
No Ablation	all	0.62581	0.12675

Only Sentiment Data	positive, negative, neutral	0.47097	0.05624
Only Stock Data	open, prev_high, prev_low, prev_close	0.54839	0.10201
Polarizing Score and Stock Price	open, positive, negative	0.57419	0.10872
Removing Least Important Feature	all except prev_low	0.58065	0.07356

Overall, the best performing model included all the available features, this could be a result of no one feature overpowering the other features.

Hyper-Parameter Sensitivity Analysis

To understand how different hyper-parameter choices affected the results of the Gradient Boosted Classification model, the team used the results from GridSearchCV to examine the 576 combinations of hyper-parameters used. This uncovered that most of the hyper-parameter combinations produced mean test scores that hovered in the 0.50 to 0.54 accuracy range. However, the model was sensitive to hyper-parameters tuning with the lowest mean test accuracy being 34% lower than the highest mean test accuracy. The best performing model used *learning_rate*=0.2, *max_depth=5, max_leaf_nodes=15, n_estimators=5.*

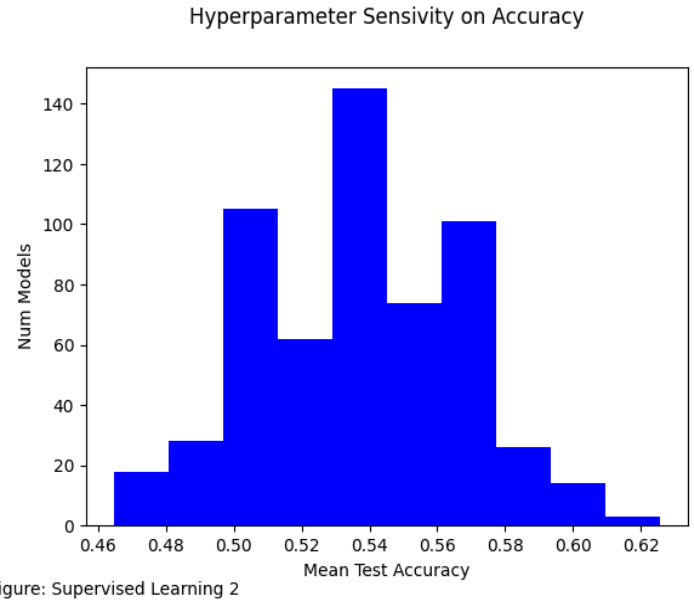

Figure: Supervised Learning 2

Training Size Sensitivity Analysis

An initial assumption the team had was that public sentiment for a stock would affect the stock direction differently at different periods of time. With this being said, it was assumed that stock sentiment predictors would not be as valuable in predicting stock price direction over the course of time. To analyze this, the

training and testing window sizes were doubled (252/60) and tripled (378/93) for the best performing Gradient Boosted Classifier and mean test accuracy fell to 0.52258 and 0.49677.

In contrast, the team also explored predicting stock price directions over smaller windows of time. Specifically, the team investigated training and testing windows of 21/5 to see if roughly a month's time-frame would provide a good basis for predicting stock direction with sentiments. Since, the training and testing size was drastically reduced, the number of cross-validations splits was increased to 20. A GridSearch was performed for different hyper-parameters settings, with the best performing models achieving an mean test accuracy of 0.6 over the 20 splits. Overall, the best model accuracy performance was found using training/testing size of 126/31 with five-fold cross-validation.

Trade Offs

From the results of the four models, the most evident tradeoff is speed vs accuracy. As models grew in complexity, so did their accuracy. The model which performed best was the slowest to train. It took 9 times longer on average to train the Gradient Boosted classifier than it did to train the Logistic Regression classifier. However, in the context of attempting to predict whether or not a stock price will rise. This is a trade-off most end-users would take, given that the Gradient Boosted Classifier used took under 8 seconds to grid search through all hyper-parameter combinations.

Failure analysis

To get a better understanding of where the model was failing, the team constructed confusion matrices over the 5 testing splits to better understand where errors were originating:

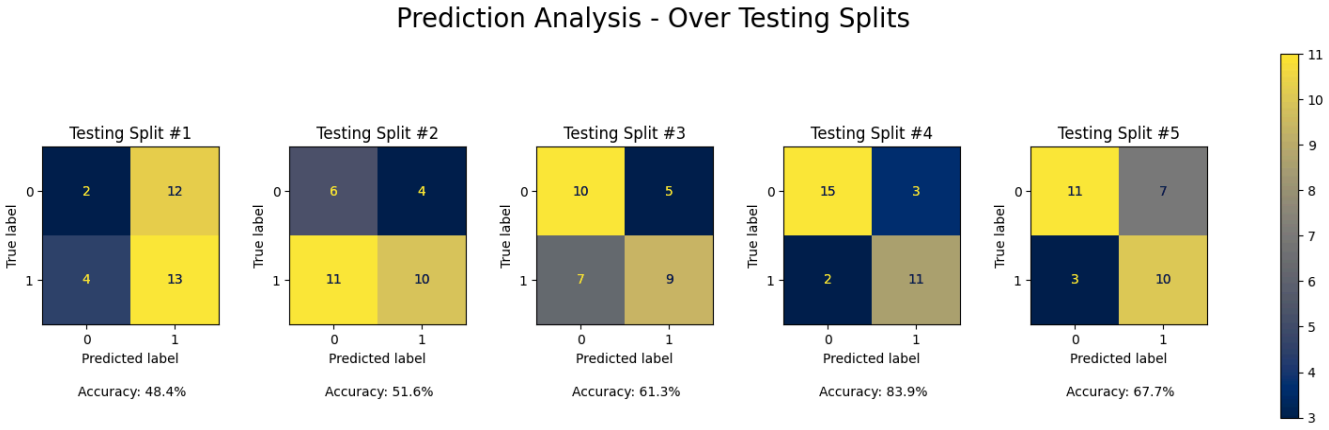

Figure: Supervised Learning 3

From this figure above, it is evident that the first two splits contained the worst performance. The first testing split suffers from a large amount of type 2 error (false-negatives) as it makes up a majority of the errors in the model. While the second split is just the opposite it suffers from a large amount of type 1 error (false-positives). The three remaining splits were more even in terms of type 1 and type 2 errors. To understand why these errors may be occuring the team looked into how the correlation between sentiments and the target label shifted between the train and test set for all five splits. Results are shown in the figure below:

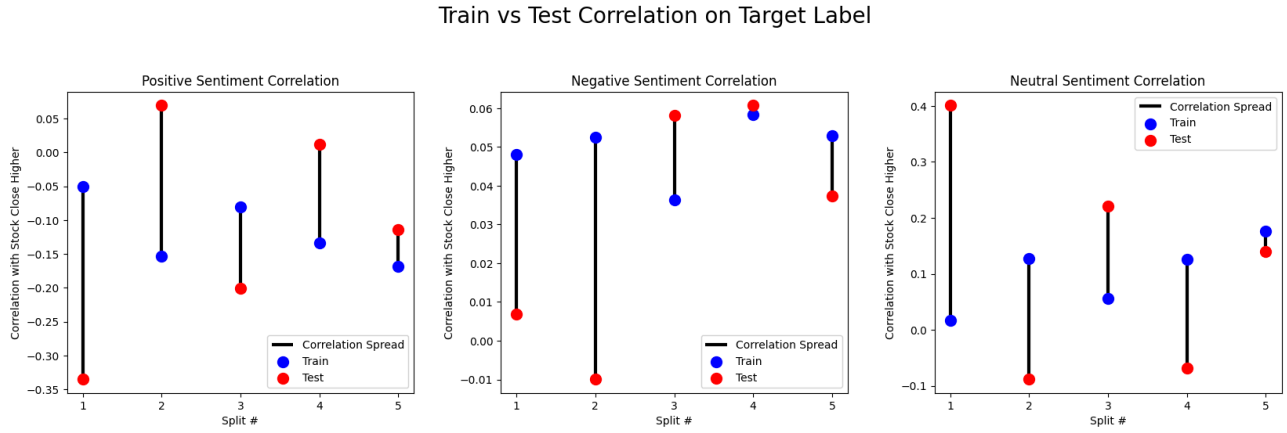

Figure: Supervised Learning 4

When looking at the difference in how the sentiments correlated between the target label. It is evident that the first two training splits had the largest amount of spread between how the sentiments correlated with the target label in the training set vs in the testing set. This could be a factor in the model outputting the incorrect classification.

The root cause of this change in correlation could happen for a number of different reasons. It could be that stock price direction is affected differently by article sentiments during different time windows. In addition, due to the limited number of articles in the dataset, it could be difficult for articles written to capture the actual public sentiment of the stock. Furthermore, there are certainly other confounding variables which will affect the direction of the stock price, which will not be captured in the sentiment analysis of online articles.

A future improvement that may assist in fixing these errors is to gather more data around the sentiment of the stock. In this project, only news articles written on the stock were considered. However, by adding in sentiment data over multiple different news sources such as social media platforms like X (formerly Twitter) and Reddit, could assist in capturing a better view of the sentiment of the stock.

Discussion

Unsupervised learning

Challenges

Evaluation of our sentiment analysis model was the largest challenge when building our unsupervised model. Sentiment is subjective, and there are so many different embeddings, preprocessing techniques, etc. that we spent a long time trying to get the “perfect” model. Of course, there is no perfect model, and we realized that real world models will have a lot more error than the models we use as examples in class.

Surprises

We were surprised how differently models with different embeddings scored the sentiment of our text. In extreme cases, one model would say a piece of text is neutral, another would say it is positive, and another would say it is negative. As mentioned in the challenges, there is not a perfect way to reconcile this.

Extending Solution

With additional time and resources it would be interesting to scrape more sentiment data from additional sources like social media websites. This might give a better representation of what public sentiment really is regarding a stock, or it might even show us that more data results in too much noise and a less accurate prediction. It would also be interesting to see if sentiment from different sources have different values. What if news stations were releasing positive articles at the same time as Twitter posts were all negative?

Supervised Learning

Challenges

Originally the team set out to predict what the closing price of the stock would be. The team tested a number of different parameter settings with different training and testing sizes. In a sense, it took an approach similar to what would be GridSearchCV squared where the team checked all combinations of both hyper-parameters, sentiment scores, and training and testing sizes. The results ended up being underwhelming with linear regression being able to fit the data best using the opening price as the feature of most importance. Due to this, the team decided to see if the dataset could be used for an alternative objective in classification. In addition, a standard training and testing window size was implemented to find the best performing model across the sentiment datasets.

Surprises

One interesting finding came from investigating the correlation of sentiment scores from BERTopic on the target label. When examining the correlations, the positive sentiment often had an inverse relationship to the target label. In addition, another interesting finding was while conducting sensitivity analysis on training size, a couple of models with smaller training/testing window sizes were able to achieve results just under the best performing model on the standard training and testing split sizes. Overall, the team went into the supervised evaluation with the idea that the best performance would be achieved from the sentiment scores generated by the FinBERT-Tone model, since this was trained on the most financial data of all the models used.

Extending Solution

With additional time, it would be interesting to be able to acquire additional data surrounding stock prices like the Earning Per Share. Due to API costs, this data was unable to be acquired since most of the providers of the data charged upwards of $100/mo for accessing the API. Also, with additional time it would be interesting to see how the different sentiment scores performed in the shorter time windows which were explored during training size sensitivity analysis.

Ethical Considerations

Unsupervised Learning

While not a major problem, one ethical consideration of our unsupervised learning model is that we are making assumptions about the sentiment of articles written by others. The subjectivity of sensitivity analysis and the imperfections of our model mean that we could be incorrectly assuming what the author intended. Again, the ethical consequences of this are very minor.

Supervised Learning

While the team does not feel like there are major ethical implications to providing this supervised learning model. One issue that does stand out is in the context of this model being used as a commercial product. For instance, if this model was used as part of a pitch by investment bankers to clients and was advertised as a model that can “beat the market.” While the model did slightly outperform random

guessing, communicating the results that the model achieved should be done with a thorough explanation of the performance the model achieved accompanied by the context in which it was tested.

GitHub Repository

https://github.com/StevenSongSTS/w24-milestone2-team18-leevw-steveso-raulmart

Statement of Work

Data Scraping: Raul Martinez
Unsupervised Learning: Raul Martinez, Steven Song, Vincent Lee
Unsupervised Learning Evaluation: Steven Song, Vincent Lee
Supervised Learning: Raul Martinez, Vincent Lee
Supervised Learning Evaluation: Raul Martinez
Project Management: Raul Martinez, Steven Song, Vincent Lee
Final Report: Raul Martinez, Steven Song, Vincent Lee

References

@GrabNGoInfo, Amy. “Topic Modeling with Deep Learning Using Python Bertopic.” Medium, GrabNGoInfo, 28 Mar. 2023, medium.com/grabngoinfo/topic-modeling-with-deep-learning-using-python-bertopic-cf91f5676504.

Grootendorst, Maarten P. “Topic Distributions.” BERTopic, maartengr.github.io/BERTopic/getting_started/distribution/distribution.html#example. Accessed 2 Mar. 2024.

Huang, Allen H., et al. “finbert: A large language model for extracting information from financial text*.” Contemporary Accounting Research, vol. 40, no. 2, 6 Jan. 2023, pp. 806–841, https://doi.org/10.1111/1911-3846.12832.

Saini, Anshul. “Gradient Boosting Algorithm: A Complete Guide for Beginners.” Analytics Vidhya, 10 Jan. 2024, www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/.

Sample Data Files

https://github.com/StevenSongSTS/w24-milestone2-team18-leevw-steveso-raulmart/tree/main/lib/sample_data_files

Appendix

Appendix A (Manual evaluation of sentiment scores from final model, BERTopic -> FinBERT):

Most positive: (positive: 0.5347991365835707, negative: 0.18029630884957215, neutral: 0.284906577734566): Looking at Q3, Apple AAPL earned $13.09 billion, a 1.85% increase from the preceding quarter. Apple also posted a total of $59.69 billion in sales, a 2.35% increase since Q2. Apple earned $12.85 billion and sales totaled $58.31 billion in Q2.Why ROCE Is SignificantReturn on Capital Employed is a measure of yearly pre-tax profit relative to capital employed in a business. Changes in earnings and sales indicate shifts in a company's ROCE. A higher ROCE is generally representative of successful growth in a company and is a sign of higher earnings per share for shareholders in the future. A low or negative ROCE suggests the opposite. In Q3, Apple posted an ROCE of 0.18%.Keep in mind, while ROCE is a good measure of a company's recent performance, it is not a highly reliable predictor of a company's earnings or sales in the near future.Return on Capital Employed is an important measurement of efficiency and a useful tool when comparing companies that operate in the same industry. A relatively high ROCE indicates a company may be generating profits that can be reinvested into more capital, leading to higher returns and growing EPS for shareholders.For Apple, the return on capital employed ratio shows the number of assets can actually help the company achieve higher returns, an important note investors will take into

account when gauging the payoff from long-term financing strategies.Q3 Earnings RecapApple reported Q3 earnings per share at $2.58/share, which beat analyst predictions of $2.04/share.

Most negative: (positive: 0.29240000000000005, negative: 0.6795, neutral: 0.028099999999999997)Companies Reporting Before The BellWalmart Inc. WMT is estimated to report quarterly earnings at $1.17 per share on revenue of $130.31 billion.Advance Auto Parts, Inc. AAP is expected to report quarterly earnings at $1.75 per share on revenue of $2.74 billion.The Home Depot, Inc. HD is projected to report quarterly earnings at $2.26 per share on revenue of $27.28 billion.Kohl's Corporation KSS is estimated to report quarterly loss at $1.75 per share on revenue of $2.16 billion.Dycom Industries, Inc. DY is expected to report quarterly loss at $0.04 per share on revenue of $746.64 million.SINA Corporation SINA is projected to report quarterly earnings at $0.07 per share on revenue of $385.62 million.Weibo Corporation WB is estimated to report quarterly earnings at $0.31 per share on revenue of $312.52 million.Eagle Materials Inc. EXP is projected to post quarterly earnings at $0.88 per share on revenue of $285.44 million. Companies Reporting After The BellUrban Outfitters, Inc. URBN is projected to post quarterly loss at $0.20 per share on revenue of $693.55 million.NetEase, Inc. NTES is estimated to post quarterly earnings at $4.01 per share on revenue of $2.21 billion.Taro Pharmaceutical Industries Ltd. TARO is expected to post quarterly earnings at $1.53 per share on revenue of $168.24 million.Red Rock Resorts, Inc. RRR is projected to post quarterly earnings at $0.18 per share on revenue of $429.30 million.Transcat, Inc. TRNS is estimated to post quarterly earnings at $0.28 per share on revenue of $45.33 million.Carriage Services, Inc. CSV is expected to post quarterly earnings at $0.33 per share on revenue of $71.52 million.Sociedad Quimica y Minera de Chile S.A. SQM is projected to post quarterly earnings at $0.25 per share on revenue of $487.43 million.

Most neutral: (positive: 0.1552, negative: 0.019733333333333335, neutral: 0.8250333333333333)GainersTaiwan Semiconductor, Inc. TSM stock rose 2.3% to $59.75 during Tuesday's pre-market session. The market value of their outstanding shares is at $256.0 billion. According to the most recent rating by China Renaissance, on December 17, the current rating is at Buy.Nokia, Inc. NOK stock rose 1.8% to $3.65. The market value of their outstanding shares is at $19.8 billion. The most recent rating by JP Morgan, on October 25, is at Neutral, with a price target of $4.20.Micron Technology, Inc. MU shares increased by 1.6% to $53.80. The market cap stands at $52.6 billion. According to the most recent rating by Wedbush, on December 17, te current rating is at Outperform.Uber Technologies, Inc. UBER shares rose 1.5% to $30.47. The market cap seems to be at $50.4 billion. The most recent rating by Citigroup, on December 10, is at Buy, with a price target of $46.00.ASML Holding, Inc. ASML shares surged 1.3% to $296.87. The market cap stands at $114.5 billion. The most recent rating by JP Morgan, on December 04, is at Overweight, with a price target of $310.00.Cheetah Mobile, Inc. CMCM stock surged 1.0% to $3.45. The market cap stands at $528.3 million. LosersApple, Inc. AAPL stock decreased by 0.2% to $279.28 during Tuesday's pre-market session. The market cap stands at $1.1 trillion. The most recent rating by Evercore ISI Group, on December 11, is at Outperform, with a price target of $305.00.