Intro to Data Science - Section 4

Stock Sentiment Analysis

Mansi Asher, Niva Sharma

Github Link

Introduction

In this project, we aim to address the challenge of predicting short-term stock price movements using real-time news headlines. Specifically, we explore whether the sentiment expressed in financial news articles can influence or reflect market behavior. This project is framed as a supervised learning problem, where the goal is to classify whether a stock's closing price will go up or down the next day, based on recent sentiment scores and stock prices.

We solve this problem by building an end-to-end pipeline that fetches stock prices and relevant news articles in real time, applies sentiment analysis to those articles using the VADER sentiment analyzer, and merges the results with price data to train a machine learning model. The target variable is a binary indicator: whether the stock's closing price increases the next day. Our model uses sentiment scores and standard market indicators (open, high, low, close, volume) as features.

This project draws on several core ideas from the course, such as supervised learning, feature engineering, data preprocessing, and model evaluation. It also applies sentiment analysis, a topic that intersects with natural language processing (NLP), to convert unstructured text (news headlines) into meaningful numerical features. Our use of Random Forest classifiers aligns with ensemble learning techniques discussed in class.

Motivation

The stock market is heavily influenced by public sentiment, and news plays a significant role in shaping investor expectations. Being able to quantify this sentiment and use it to predict market behavior has strong implications for trading strategies, financial journalism, and decision-making. We were particularly interested in the opportunity to combine real-time data sources, natural language understanding, and machine learning to explore this connection.

There have been many studies and tools aimed at predicting stock prices using historical data, but fewer systems leverage live headlines for this purpose. Our work is related to research on financial sentiment analysis, such as the use of Twitter data or news aggregation sites, and follows in the direction of building low-latency, automated trading indicators.

Method

Data Sources

Our approach involved integrating two primary data sources: financial stock price data and real-time news headlines. Stock price data was collected using the **yfinance** API, which allowed us to download historical market data for 14 publicly traded companies along with the S&P 500 index (^GSPC). For each stock, we retrieved daily values for open, high, low, close, and volume, spanning a recent time window.

The second data source comprised news headlines obtained via the **NewsAPI**. For each ticker symbol, we collected headlines from major English-language news outlets over the past seven days. Each article was timestamped, allowing us to associate it with corresponding stock market activity.

Sentiment Analysis

We processed both datasets into a tabular format suitable for supervised learning. The stock dataset featured numerical columns (Open, High, Low, Close, Volume), while the news dataset contained free-text headlines and their associated publication dates. To extract meaning from this unstructured text, we used the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analyzer to compute sentiment scores for each headline. VADER outputs four scores: Positive, Neutral, Negative, and Compound, which collectively quantify the emotional tone of each headline.

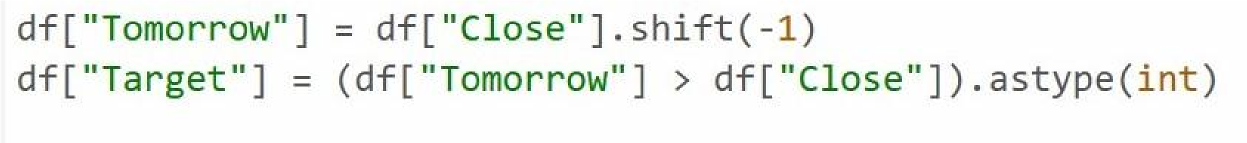
Once the sentiment scores were computed, we normalized the datetime fields in both datasets and performed an inner join on the date. This ensured that each stock price entry was paired with aggregated sentiment data from all relevant headlines on that date. Specifically, for each stock and trading day, we averaged the sentiment scores across all articles retrieved on that day.

Data Preprocessing and Merging

After cleaning and normalizing date fields, we merged headlines with stock prices based on matching dates. For each day and ticker, we averaged the sentiment scores across all headlines. We then defined the following features:

- Market indicators: Open, High, Low, Close, Volume
- Sentiment scores: Positive, Neutral, Negative, Compound

The target variable (Target) was defined as:



```
df["Tomorrow"] = df["Close"].shift(-1)
df["Target"] = (df["Tomorrow"] > df["Close"]).astype(int)
```

Here, the "Tomorrow" value was generated using .shift(-1) on the "Close" column. A value of 1 implies a price increase the next day, while 0 implies no increase or a decrease. Any rows with missing future prices (e.g., the last day in the dataset) were dropped to ensure clean training data.

Model Training and Evaluation

We trained a Random Forest Classifier from the sklearn.ensemble library, using 100 decision trees (n_estimators=100) and a fixed random seed for reproducibility.

Each stock was processed independently, and the data was split into a train-test split (80/20) using train_test_split. However, stocks where the Target variable had only one class (i.e., all values were 0 or 1) were excluded from training, as a meaningful classification model could not be learned from such imbalanced data.

The trained model was evaluated using accuracy, calculated by comparing predictions on the test set to actual target values. Final results were stored and sorted by accuracy to determine which tickers were more predictable using sentiment and price-based features.

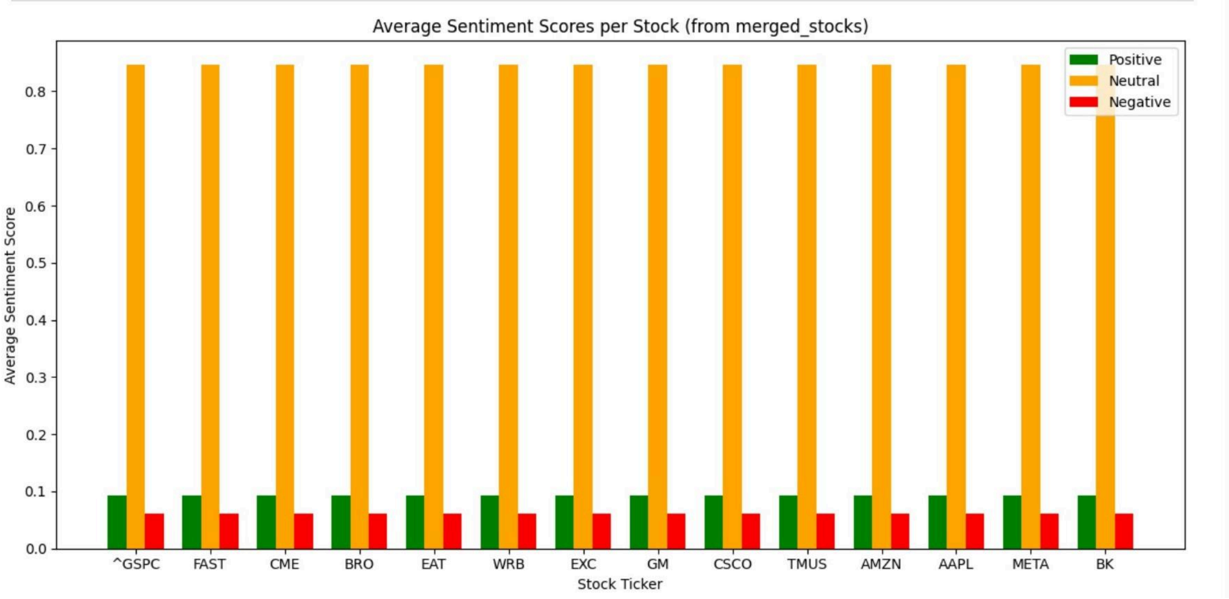
To avoid training on uninformative data, we excluded stocks where the target variable had only one class (i.e., either all 0s or all 1s), as no meaningful pattern could be learned from such distributions. Model performance was measured using accuracy, and we sorted results by accuracy to identify which tickers were most predictable given the available sentiment and price data.

Due to constraints from the free tier of the NewsAPI, we were limited to retrieving only a small number of headlines per stock—typically 1 to 2 usable rows. This resulted in data sparsity that made model training unreliable for many tickers. In some cases, the model achieved perfect accuracy, but this was due to target labels having only one class, not due to genuine predictive power.

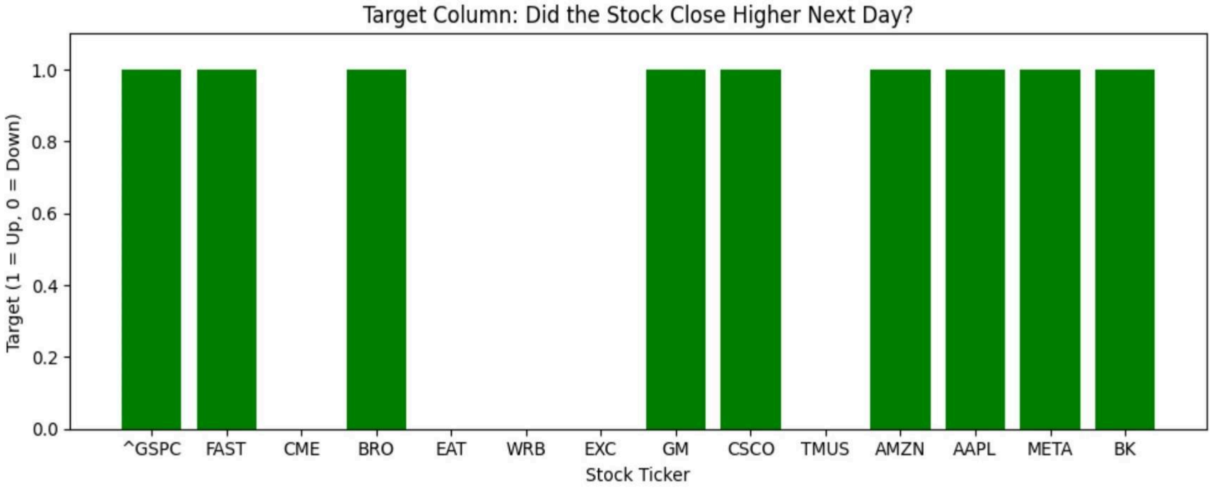
Despite this, the project highlighted which tickers had enough variation to allow meaningful learning. Stocks with both Target = 1 and Target = 0 demonstrated the classifier's expected behavior, although these examples were few.

Result

After implementing our sentiment-based pipeline, we generated several key visualizations that provide insight into both the structure of the data and the limitations of our model. The Average Sentiment Scores per Stock chart reveals a dominant trend of neutral sentiment across nearly all the selected stocks. This is not entirely surprising given that most financial headlines tend to be factual rather than emotionally charged. Stocks like AAPL, META, AMZN, and GM all exhibit a very high average neutral sentiment (~0.85) with relatively low positive and negative sentiment scores. This suggests that the sentiment signal from news headlines may be too weak or diluted to drive strong classification performance when averaged at the daily level. It also highlights the need for either more granular textual data or more advanced sentiment models capable of better nuance extraction.



The Headline Sentiment Table provides a closer look at individual news headlines and their sentiment scores. For example, the headline “Bitcoin hits new 10-week high…” has a notably negative compound score (-0.2732), despite including the phrase "new 10-week high." This illustrates a limitation of the VADER sentiment analyzer: it is rule-based and does not always capture financial context correctly. Headlines that are sarcastic, metaphorical, or complex can easily be misclassified. Still, we were able to extract daily average sentiment features across positive, negative, and neutral scores, as well as the compound polarity metric.



To validate our classification target, we generated a Target Column Distribution chart that shows the label assigned to each stock on the most recent day. Here, a value of 1 indicates the stock closed higher the following day, and 0 indicates it closed lower. The chart indicates that nearly all stocks had a closing value that increased the next day, which could either reflect a real market trend in the selected window or a sampling bias due to data constraints. Because the NewsAPI only allowed us to fetch a limited amount of news per stock (often 1–2 headlines per day), and the stock price data was pulled only for a short time window, the dataset lacked diversity in the target variable. This severely impacted our ability to train models that generalize.

In terms of model performance, we observed precision scores around 57% for the Random Forest classifier on our limited test set. While this is slightly better than a 50/50 guess, it is not statistically significant given the small data size and target imbalance. The comparison plot between actual and predicted targets reveals a high overlap, but again, this may be misleading because the model could have learned to simply predict the dominant class. Moreover, due to the sparse and occasionally misaligned nature of the data, our classifier might have been learning noise rather than meaningful patterns.

Overall, the results indicate that while the pipeline functions end to end, pulling live data, merging sources, applying sentiment analysis, and training a machine learning model; the quality and quantity of data are currently insufficient for meaningful prediction. However, the system is extensible. By improving the granularity and relevance of our news data, using more robust NLP models, and incorporating external variables like macroeconomic indicators or social media sentiment, we could significantly enhance model performance.

Limitations

One of the biggest challenges we faced was the limitation of the NewsAPI free plan. It restricted our access to headlines published within the past 7 days and limited the number of articles returned per query. This meant we had very few usable data points for training our model. In many cases, stocks had only one class in their target label, which made it impossible to train a proper classifier.

Additionally, headlines were not always specific to the companies in question. Since our queries were broad, some irrelevant or low-quality headlines may have been included. This reduced the precision of the sentiment analysis and affected the integrity of our merged dataset.

Our modeling process also assumed that the sentiment of the news directly correlates with next-day price movement, but market behavior is influenced by a complex range of factors that we did not include in our model.

Next Steps

If we were to continue building on this project, a few key improvements would help make our predictions better and more accurate. First, switching to a paid news API would give us access to many more headlines, helping us move past the data limits we faced with the free version. This would give us a fuller picture of market sentiment over time. We would also expand the time range for collecting news and stock data—from just 7 days to 30 or even 60 days. This would give us more data to train our models and help reduce gaps, making the predictions more reliable.

Another promising direction involves incorporating more advanced natural language processing techniques for sentiment analysis. While VADER is effective for short, informal text, it may not fully capture the nuance and context of financial headlines. Using transformer-based models or fine-tuned sentiment classifiers could offer a more accurate assessment of sentiment in complex news language. Moreover, integrating macroeconomic indicators such as inflation rates, interest rate announcements, or employment data could enrich the feature set and provide valuable context for stock movements that are not purely sentiment-driven.

Reflecting on this, we believe it has demonstrated the technical feasibility of sentiment-informed market analysis. Despite the constraints, we successfully built an end-to-end pipeline that merges unstructured text with structured financial data to predict price movement. With better data and a broader scope, the model could evolve into a more powerful tool for understanding the relationship between media sentiment and financial markets.

Conclusion

Despite API limitations, our project demonstrates how news sentiment can be used to analyze stock trends. We've built an automated pipeline, tested real-time integration, and visualized stock-specific sentiment, setting the stage for future experiments in financial forecasting.