Sentiment Analysis of Financial News

A Data-Driven Study Using NLP, XGBoost, and SHAP for Return and Volatility understanding

Andreas Wendel

EC Utbildning

Examensarbete

2025–05

Abstract

This project looks into how financial news affects the market — not just in big headlines, but in everyday reporting. The goal was to see if we can predict whether a piece of news will lead to a positive, negative, or neutral return on a stock, using a combination of natural language processing, machine learning, and SHAP for explainability. News summaries and price data were collected through public APIs and connected by time. The text was cleaned, vectorized with TF-IDF, and fed into two models: Logistic Regression and XGBoost. To understand what the model learned, SHAP was used to build a sentiment lexicon, showing which words pushed predictions in different directions. Statistical testing showed that news doesn't consistently lead to positive or negative returns — but it does increase volatility. A deeper dive into the word "tariff" showed exactly that: strong impact on volatility, but unpredictable direction. Overall, the project shows that while news might not tell us where the market is going, it does show when it's about to move.

Abbreviations

NLP	Natural language processing
XGBoost	Extreme gradient boosting
CLT	Central Limit Theorem
TP	True positive
FP	False positive
TN	True negative
FN	False negative
TF-IDF	Term Frequency–Inverse Document Frequency
SHAP	SHapley Additive exPlanations (already present, but clarify in body too)
XGBC	XGBoost Classifier
LR	Logistic Regression
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
API	Application Programming Interface
GPU	Graphics Processing Unit
QQ-plot	Quantile-Quantile Plot

Table of Contents

1 Introduction

The Financial markets are increasingly influenced by the rapid spread of news through digital media. This transformation has elevated the importance of understanding how financial news impact the market behaviours. In this study, we investigate the usage of natural language processing (NLP) and financial modelling, with a focus on sentiment analysis.

Recent years have witnessed several major economic events that highlight the power of news to shape market movements:

- Covid-19 (2020-2022)
  The unpredictable and major spread of the pandemic caused the 2020 stock market crash which triggering the start of a global recession, followed by unprecedented government and central bank interventions. These actions cause much uncertainty in the market creating extreme volatility in the market.
  (Source: Wikipedia)

- Global Trade Tensions (2021-2025)
  The global trade tensions have increased immensely and in later years has been pushing nations into actions such as tariffs and trade restrictions such as in 2025 when the US imposed tariff on trade imports which also caused a crash, or restriction on chips sales to the Chinese to slow their advancement in AI development.
  (Source: Wikipedia, Miller&Chevalier)

- Geopolitical Conflicts
  Russia's invasion of Ukraine in 2022 and ongoing conflicts in the middle east and elsewhere contributed to the price shocks which were very noticeable in the energy sector. Russia were also under sanctions which removed them from the global market. These geopolitical conflicts also lead actions such as the destruction of the Nord Stream pipeline which was natural gas pipeline running from Russia to Germany. Disrupting the energy market. As reliance of Russian energy withered the energy price rose which was very noticeable under the winter of 2022 in Sweden.
  (Sources: ECIU, Vattenfall)

This study will take a deeper dive into both major and everyday news to analyse how words might have had an impact on a smaller scale

While the economic impact of such major headlines is often clear, countless smaller financial news stories are published every day by outlets such as Forbes, Bloomberg, and The Verge. The question remains: do these less prominent articles also influence financial markets—and if so, how?

This study explores not only high-impact news events but also the broader body of financial journalism to analyze whether the sentiment and specific language in these texts can provide signals for market behavior on both small and large scales.

1.1 Research Questions

Throught this investigation there are several key questions:

1. How can unstructured text data be processed effectively for machine learning models
2. How can we evaluate whether a news article has a positive or negative market impact
3. Which specific words or phrases most strongly influence market returns and why
4. Is there a measurable correlation between the volume of news and market volatility
5. What are the limitations of news-based sentiment forecasting financial performance
6. How can we interpret and trust the model's predictions using explainable ai techniques such as SHAP
7. Can this framework be used for practical applications such as risk assessment or decision-making in financial institutions?

2 Theory

The goal is to create a model that can predict if the following articles will have a positive, negative or neutral return on the underlying. That defines this as a multi-classification problem which implies that the model will give us the Probability for each label.

We may than use the model to analyse how words impact the decision the model makes

2.1 Statistical inference

Statistical inference is the process of drawing conclusions about a population based on data collected from a sample. It allows researchers to make probabilistic statements about population parameters (such as the mean or variance) using observed data and formal statistical models.

(Source: Wikipedia, Statistisk dataanalys)

2.1.1 Central Limit Theorem (CLT)

- **Definition & Purpose**:
  The central Limit Theorem (CLT) is one of the fundamental parts in statistics, it states that the sampling distribution of the sample mean will in approximation be normal distributed as the size of the sample becomes larger, regardless of the shape distribution of the population. This theorem lays hands for many broad statistical methods such as the t-test, by justifying the use of normality-based inference even when the underlying data is not normally distributed.
- **Mathematical Expression**:
  Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables with mean μ and finite variance σ^2 The Central Limit Theorem states: $\frac{\overline{X_n}-\mu}{\sigma/\sqrt{n}} \rightarrow d\mathcal{N}(0,1)$

  Where:
  - $(\overline{X_n})$: is the sample mean
  - (μ): is the population mean
  - (σ): is the population standard deviation
  - (n): is the sample size
  - $(\rightarrow d\mathcal{N}(0,1))$: convergences the distribution to the standard normal

  This shows that as $(n \rightarrow \infty)$, the distribution of the standardized mean approaches the standard normal distribution

  (Source: Wikipedia, Investopedia)

2.1.2 Welch's t-test

- **Definition & Purpose**:
  Welch's t-test is a statistical method used to determine whether there is a statistically significant difference between the means of two independent groups, without assuming equal variances. It is a refinement of the standard Student's t-test and is particularly useful in real-world scenarios where the assumption of equal variance may not hold.

- **Mathematical Formula**:
  The t-statistic is calculated as: $t = \frac{\overline{X_1}-\overline{X_2}}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}}$

Where:

- $(\overline{X_1}, \overline{X_1})$: are the sample means
- (s_1^2, s_2^2): are the sample variances
- (n_1, n_2): are the sample sizes

The degrees of freedom are approximated using the Welch–Satterthwaite equation:

$$df = \frac{\left(\frac{s_1^2}{n_1}\right)^2 + \left(\frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_2^2}{n_2}\right)^2}$$

This formula is used to test the null hypothesis that the two groups means are equal.

(Source: Investopedia)

2.1.3 Leven's Test

- **Definition & Purpose**:
  Levene's test is used to assess the equality of variances (homogeneity of variance) across different groups. It's particularly useful when the assumption of equal variances is crucial for subsequent analyses like ANOVA or t-tests.

- **When to Use**:
  This test is appropriate when you need to verify that multiple groups have equal variances, especially when the data may not be normally distributed.
  (Source: Statistics How To)
- **Mathematical Formula**: Levene's test involves computing the absolute deviations of each observation from its group mean and then performing an ANOVA on these deviations. The test statistic is: $W = \frac{(N-k)}{(k-1)} \cdot \frac{\sum_{i=1}^{k} N_i (Z_{i\cdot} - Z_{\cdot\cdot})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (Z_{ij} - Z_{i\cdot})^2}$

  Where:
  - $(Z_{ij} = |Y_{ij} - \overline{Y_i}|)$: Absolute deviation from group mean
  - (N): Total number of observations
  - (k): Number of groups
  - (N_i): Number of observations in group i
  - $(Z_{i\cdot})$: Mean of deviations in group i
  - $(Z_{\cdot\cdot})$: Overall mean of deviations

  The test evaluates whether the group variances are statistically significantly different.
  (Source: Wikipedia)

2.1.4 Correlation Analysis

Correlation analysis is a statistical technique used to determine the degree and direction of a linear relationship between two continuous variables. It tells us:

- **Strength**: How strong the association is
- **Direction**: Whether the variables increase together (positive) or one increases as the other decreases (negative)

2.1.4.1 *Pearson's Correlation Coefficient (r)*

Pearson's correlation coefficient, denoted as r, is the most common measure of linear correlation between two variables. Its value ranges from -1 to 1:

- r=1: Perfect positive linear relationship
- r=−1: Perfect negative linear relationship
- r=0: No linear relationship

2.2 Data Transformation

To prepare unstructured financial news for quantitative analysis, several preprocessing steps are required. These steps are essential to convert raw textual information into a structured and standardized form suitable for machine learning models. The following subsections describe the natural language processing (NLP) techniques, text-cleaning procedures, and financial return labeling methodology used in this study.

2.2.1 NLP

NLP techniques are applied to reduce linguistic variability and extract meaningful features from textual data. These steps help standardize and simplify the content while retaining its semantic integrity.

2.2.1.1 *Stopwords*

Stopwords are common words such as "the," "is," "at," and "which" that typically carry little Value in by themselves. By removing stopwords we may reduce noise in the data and minimizes the dimensionality of the feature space. In this study, the stopwords set provided by the nltk library was used, with additional custom stopwords added to better suit financial text.

2.2.1.2 *Lemmatization*

Lemmatization is the process of converting words to their base or dictionary form (lemma). For example, the words "running" and "ran" would be lemmatized to "run." This step ensures that different forms of a word are treated as a single token, thereby improving the model's ability to generalize. Lemmatization was performed using the nltk.stem library, which considers both context and part-of-speech tagging.

2.2.1.3 *Tokenazation (TfidVectorizer)*

Tokenization refers to the process of splitting text into individual words or tokens. This forms the basis for numerical feature extraction. The TfidfVectorizer from scikit-learn was employed to transform the tokenized documents into a numerical matrix of Term Frequency–Inverse Document Frequency (TF-IDF) scores. This method highlights terms that are important within a document but rare across the corpus, providing more meaningful input to the machine learning model.

2.2.2 Text Cleaning

Text cleaning is performed to eliminate elements that may distort analysis. This may include:

- Lowercasing all text to maintain uniformity.
- Removing punctuation, numbers, and special characters.
- Stripping HTML tags and extraneous whitespace.

- Filtering out non-alphabetic tokens.

These procedures were implemented using regular expressions and built-in Python string methods. The aim was to ensure the resulting text is as noise-free and consistent as possible before vectorization.

2.2.3 Return Labeling

To align financial news with market impact, each article was labeled based on the return of the associated financial instrument following the news publication. Returns were computed as the percentage change in price over different time horizons (e.g., 10 minutes, 30 minutes, 1 hour, 5 hours, 1 day).

The formula used is: $\mathrm{r_t} = \frac{P_{t+\Delta t} - P_t}{P_t}$

Where P_t is the price at the time of the news and Δt is the future time horizon.

Articles were categorized into three classes:

- **Positive**: Returns greater than a positive threshold (e.g, +0.2%)
- **Negative**: Returns less than a negative threshold (e.g, -0.2%)
- **Neutral**: Returns within the threshold range (e.g, -0.2% to + 0.2%)

This classification setup defines the problem as a multi-class classification task. Labeling is crucial to training the supervised model to associate language patterns with subsequent market behavior.

2.3 Modelling

2.3.1 Logistic regression

Definition & Purpose: Logistic Regression is a supervised learning algorithm commonly used for classification tasks. It estimates the probability that a given input belongs to a particular category by applying the logistic (sigmoid) function to a linear combination of input features.

Mathematical Formulation: $P(y = 1 \mid x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta n x_n)}}$

Here, $\beta_0, \beta_1, \ldots \beta_n$ are the model parameters estimated from the data using maximum likelihood estimation.

(Source: Scikit-learn)

2.3.2 XGboost Classifier

- **Definition & Purpose**:
  XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient-boosting decision trees. It is widely used for classification problems due to its high performance and robustness

2.3.2.1 Extreme gradient boosting

Algorithm Overview:
XGBoost builds an ensemble of decision trees sequentially. Each tree learns to correct the errors (residuals) made by the previous trees by fitting the gradients of the loss function. The final prediction is the weighted sum of all trees.

Objective Function:

The training objective includes both the loss function $l(y_i, \hat{y}_i)$ and a regularization term $\Omega(f_k)$ that penalizes model complexity:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda |w|^2$$

Where T is the number of leaves, w are the leaf weights, γ, λ control regularization

Key Features:

- Uses both first and second-order gradients for better accuracy.
- Includes regularization to avoid overfitting.
- Supports shrinkage (learning rate), column and row subsampling, and parallel computation.
- Offers GPU acceleration and custom loss functions.

"Extreme" Advantage:

The term "Extreme" refers to XGBoost's performance improvements over traditional gradient boosting, including advanced regularization, optimized tree pruning, and support for large-scale distributed training.

(Source: XGBoost Paper, Nvidia)

2.3.3 Evaluation

2.3.3.1 Confusion Matrix

- **Definition**:
  A confusion matrix is a table used to evaluate the performance of a classification model by displaying the true vs. predicted classifications.
- **Components**:
  - True Positives (TP)
  - True Negatives (TN)
  - False Positives (FP)
  - False Negatives (FN)

2.3.3.2 Classification Report

The classification report provides precision, recall, F1-score, and support for each class, offering a more detailed performance summary than accuracy alone.

2.3.3.2.1 Accuracy

- **Formula**: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- **Interpretation**: Measures the proportion of total correct predictions.

2.3.3.2.2 Recall

- **Formula**: $Recall = \frac{TP}{TP+FN}$
- **Interpretation**: Measures how well the model identifies all relevant instances.

2.3.3.2.3 Precision

- **Formula**: $Precision = \frac{TP}{TP+FP}$
- **Interpretation**: Indicates the accuracy of the positive predictions.

2.3.3.2.4 F1

- **Formula**: $\mathrm{F1} = \frac{Precision * Recall}{Precision + Recall}$
- **Interpretation**: Harmonic mean of precision and recall, useful when seeking a balance between the two.
  (Source: Wikipedia)

2.3.3.3 ROC Curves and AUC

- ROC Curve: A Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier by plotting the true positive rate (TPR) against the false positive rate (FPR).

- AUC (Area Under Curve): Represents the area under the ROC curve. A higher AUC indicates better overall model performance.
  (Source: Wikipedia)

2.3.4 Hyperparameter tuning

2.3.4.1 BayesianOptimizations

- **Definition**:
  Bayesian Optimization is a probabilistic model-based approach for finding the minimum or maximum of an objective function that is expensive to evaluate. It is particularly useful in hyperparameter tuning where model evaluations are time-consuming.
- **Process**:
  - Builds a surrogate function (e.g., Gaussian Process)
  - Uses an acquisition function (e.g., Expected Improvement) to decide where to sample next
- **Advantage**:
  - Bayesian Optimization is more sample-efficient, requiring fewer evaluations to find a near-optimal set of parameters.
  - It intelligently balances exploration (trying new regions) and exploitation (refining promising areas).

(Source: Practical Bayesian Optimization, note: read part 2 ” Bayesian Optimization with Gaussian Process Priors”)

2.4 SHAP (SHapley Additive exPlanations)

Definition & Purpose: SHAP (SHapley Additive exPlanations) is a model-agnostic explainability method based on cooperative game theory. It assigns each feature an importance value representing its contribution to the model prediction.

2.4.1 Importance and direction

- Importance is measured by the mean absolute SHAP values:

$$\text{Importance}_{\text{i}} = \sum_{\text{c=1}}^{\text{classes}} \left| \frac{1}{\text{n}} \sum_{\text{i=1}}^{\text{n}} \phi_{\text{i,f,c}} \right|$$

- Direction is measured by the mean SHAP values:

$$\text{Direction}_{\text{f,c}} = \frac{1}{\text{n}} \sum_{\text{i=1}}^{\text{n}} \phi_{\text{i,f,c}}$$

  - *j* is the feature index,
  - *c* is the class index,
  - *i* is the sample index,
  - where $\phi_{i,f,c}$ is the SHAP value of feature *j* for sample *i* and class c.

2.4.1.1 Log-odds

In classification, SHAP values are often expressed in terms of log-odds. The log-odds is defined as: $log - odds = log(\frac{P(y=1)}{1-P(y=1)})$

SHAP values shift the log-odds of the model prediction. A positive SHAP value increases the log-odds (and hence the probability), while a negative SHAP value decreases it.

The log-odds value can be converted to probability via the sigmoid function:

$$P(y = 1) = \frac{1}{1 + e^{-log-odds}}$$

(Source: Paper, Github, Wikipedia)

3 Methodology

3.1 Data Collection

Data has been Collected using various API's with an attempt to stay consistence when choosing what news articles to include.

A list of major stocks from different sectors was assembled

For storing the data a simple SQL database was setup to include table where:

- Stocks: Sectors and ticker name
- Prices: Historical prices from the given stock with a lookback period of 2 years starting from 2023-05-01 to 2025-05-01 and a granularity of minimum of 10min intervals
- News articles: Financial news released by different sources where they corresponded with the given stock

3.1.1 Source of articles

The articles were sourced by an API provided from "Alpha Vantage" which could be easily accessed with a simple request.get() followed by the url, header with a json type, Parameteres, and an apikey which is free.

Within their api they also included a parameter called "tickers" which would internally filter for articles with the given ticker, using that parameter you could source for articles corrolated to our chosen stocks.

The articles were from the same period as the prices starting from 2023-05-01 to 2025-05-01

In the request you get data such as the [Title, url, time_published, author, summary, source,…] all of the unfiltered data were stored in the Database with their corresponding stock. Note that in the request you do not get the whole articles text but just a summary of the article

(Source: Alpha Vantage)

3.1.2 Source of prices

The source of the prices were also sourced by an API which were provided from "Tiingo".

Tiingo is one of few free API's that provide granularity down to 10 min and charting history to 2 years. However they pose a request limit of 50 request every hour with 3 months of price data at 10 min granularity being 1 request, we hit the hourly limit after 3 stocks of price history.

A script was set up to request all the necessary stock data without hitting the request limit which total to 10 hours.

(Source: Tiingo)

3.2 Statistical testing

A set of statistical tests were performed to determine if there was a difference in market reaction whenever financial news were released. By comparing the % return after news were released versus randomised % returns we could determine if the thesis holds.

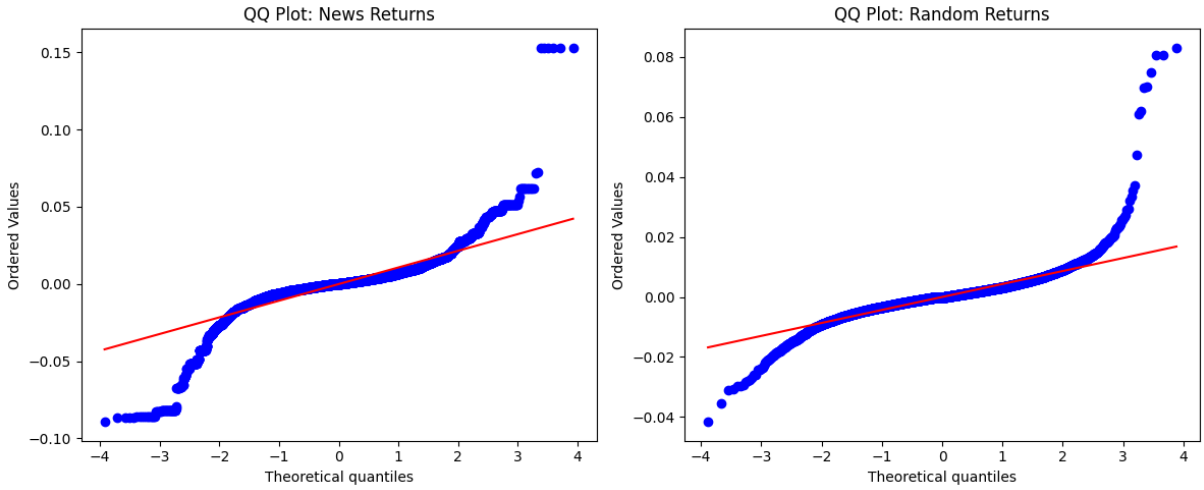


Figure 1

First, Most Statistical test have different assumptions and most of them assume that the distribution is normalised. Therefore, we plotted a simple QQ-plot on the 1-hour interval on the news returns as well as the random returns. (See figure 1)

After analysing the normal distribution and concluding that there are many outliers in the data

3.2.1 Mean returns and volatility

Continuing with the analysis we may also look at the differences between the news related returns and randomised returns. We conduct 2 tests

- **Volatility Test**:
  To test whether news-related returns exhibit greater variance than random returns, Levene's test was used. This test is suitable for non-normal distributions and checks if the variances of two or more groups are significantly different.
- **Mean Returns Test**:
  Mean returns will tell us if over the period would all financial news related returns be different from randomised returns. This would imply that the existence of news should increase the underlying's returns.
  This can be tested using a T-test which test if there is a significant difference in the mean between two distributions.

Depending on if the Levene's test is successful we can determine if there is a significant difference in variance, this also means that if the test is successful, we also need to adjust the T-test.

The T-test not only assume the distribution is normalised but also that the variance is equal.

When the distribution does not have equal variance, we can instead use the Welch's T-test which is specifically intended for situations where heteroscedasticity is a problem.

We noted that the Distributions are non-normal but because we have a sample size of *"news_returns size: 15891, random_returns size: 13455"* we can assume that the CTL applies and therefore we can use the t-test.

(See figure 2) you can see that almost all the mean % returns converge to 0.0, meaning that investing on news breakout is not significantly different from buying and selling at random. What we can notice is that the variance is significantly different meaning that news has an impact in a form of

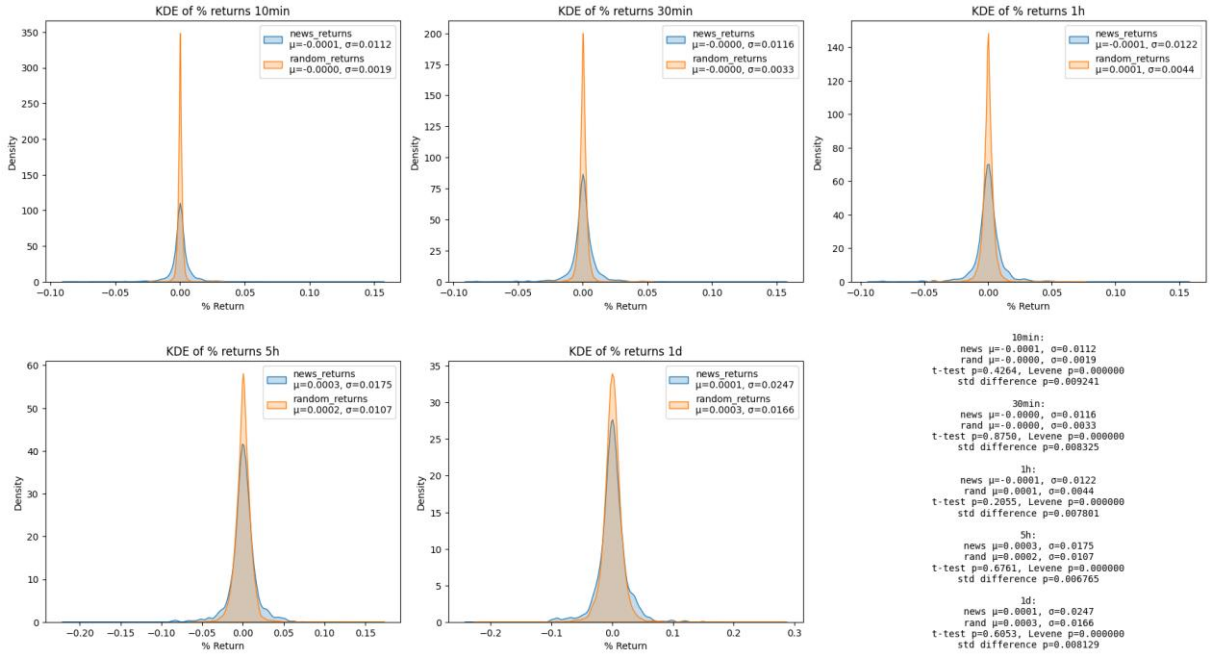


Figure 2

volatility meaning that there is a reaction to news releases. Note that with a high frequency close to the mean most returns random, or news related doesn't how a huge impact.

3.3 Preprocessing

Preprocessing is a critical step in any data-driven workflow. In this project, it serves as the foundation for extracting meaningful patterns and building robust models. Once the data was collected from the respective APIs, preprocessing involved combining the article and price datasets and applying various transformations to prepare the data for modelling.

3.3.1 Data transformations

Two key types of data transformations were applied:
feature engineering for the textual data and target labelling for the market returns.

- **Features:**
  The primary features are derived from the textual content of the financial news articles. To extract meaningful numerical representations from the text, natural language processing (NLP) techniques were employed. These include cleaning, tokenization, lemmatization, and vectorization using the TF-IDF (Term Frequency–Inverse Document Frequency) method. These steps transform raw text into structured numerical vectors suitable for machine learning algorithms.
- **Labels:**
  The target variable (label) was derived from price data by calculating the future return after a fixed time interval (e.g., one hour after publication). Based on a predefined threshold, the returns were then categorized into three classes:
  - Positive: Return above the threshold.
  - Negative: Return below the negative threshold.
  - Neutral: Return between the two thresholds.

This categorization allowed us to frame the problem as a multi-class classification task.

3.3.2 NLP

To transform the text into usable input for our models, several NLP preprocessing techniques were applied:

- **Text Cleaning:**
  Special characters, numbers, and capital letters were removed or normalized to reduce noise in the data.
- **Stopword Removal:**
  Common English and financial-specific stopwords were excluded to focus on more meaningful words.
- **Lemmatization:**
  Words were reduced to their base or dictionary form using a lemmatizer, which helps unify variations of the same word.
- **Vectorization:**
  The cleaned and tokenized text was vectorized using TF-IDF, converting it into a numerical format that captures both the frequency of a term and its importance across the entire corpus, the term will be a feature and have a numerical reference to it as an importance, the number of terms or features is chosen by the user. Here we can also chose if we would like single words or double words such as “corp” or “time buy”

3.4 Model building

This part involves training and evaluation of our machine learning model. The goal is to predict whether a news article will have a positive, negative or neutral market impact. By modelling we may also analyse how the model makes decision based on the given features

3.4.1 Logistic regression and XGBoost Classifier

Two different classification models were used “Logistic Regression (LR)” and “XGBoost Classifier (XGBC)”:

- **Logistic regression:**
  A simple yet effective linear model that estimates the probability of each class using a logistic (sigmoid) function. It's interpretable and often serves as a strong baseline for classification tasks

- **XGBoost Classifier:**
  A more powerful, tree-based ensemble method based on extreme gradient boosting. This model builds multiple decision trees iteratively, each one trying to correct the errors of the previous. It may also train on the GPU which will speed up the training process.

Both models were trained on the TF-IDF vectorized news text and the labeled sentiment classes.

3.4.2 Evaluation metrics

to evaluate model performance, several classification metrics were used:

- Accuracy: Proportion of total correct predictions.
- Precision: How many of the predicted positives were positive.

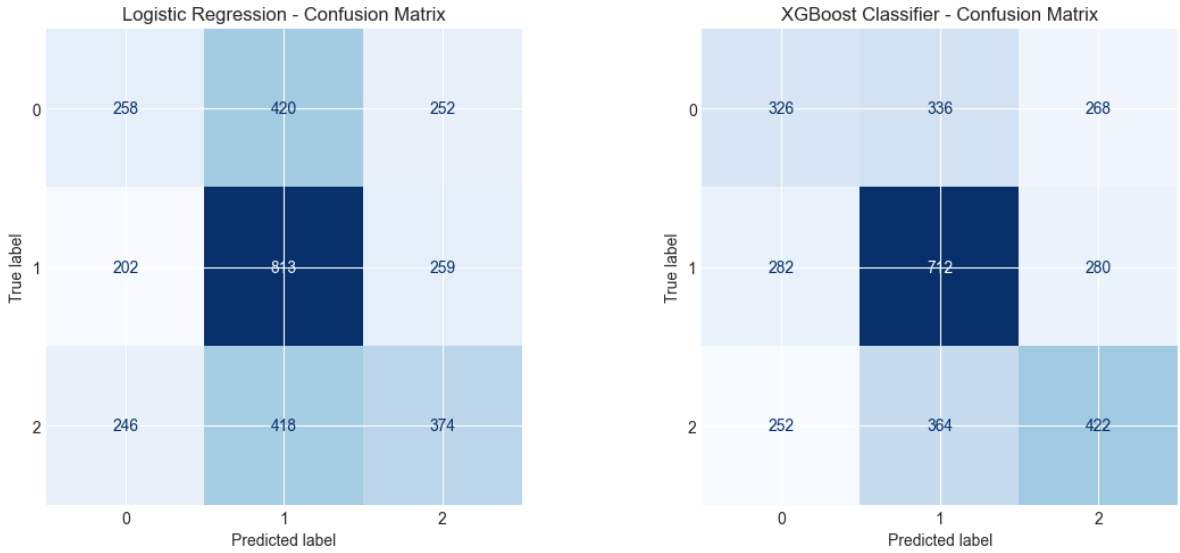


Figure 3

- Recall: How many of the actual positives were correctly identified.
- F1 Score: The harmonic means of precision and recall.
- ROC Curves & AUC: Used to evaluate how well the model separates each class. In the multiclass setting, One-vs-Rest ROC curves were computed.

Confusion matrices and classification reports were also plotted to visualize prediction patterns across the three classes (positive, neutral, negative). (See figure 3)

3.4.3 Optimising model

After testing the two models, you can carry on with a single model and attempt to improve its results by changing the model parameters. Bayesian Optimization was used to find the best combination of hyperparameters such as:

- Learning rate
- Max tree depth
- Subsample ratio
- Number of estimators

Bayesian Optimization is more sample-efficient than grid or random search, as it models the objective function and chooses the next set of parameters based on past results.

The objective of the optimization was to maximize the weighted F1 score, aiming for better balance between precision and recall across all classes

3.4.4 Testing model on custom text

A custom function was implemented to check whether the model could predict on new or fictional text. By inputting a custom text and then applying the process method on said text we can feed the features into our optimised model and get a predictive answer. Note that as the model has been trained on X features, new features or word will not be “seen” or impact the models decision hence new events which might have great impact on the market will not have the same effect on the model.

3.5 Sentiment analysis

This part of the project focuses on understanding why the model makes certain predictions by analyzing the influence of individual words. Instead of just knowing whether a prediction is positive or negative, we aim to interpret which features (words or phrases) are most responsible for those predictions.

3.5.1 SHAP

To achieve interpretability, we use SHAP (SHapley Additive exPlanations), a method based on game theory that assigns each feature a value representing its contribution to a prediction.

For each sample and class, SHAP returns how much each word pushed the model towards or away from a class. This makes it possible to trace back the impact of specific terms like "tariff" or "inflation."

3.5.1.1 Lexicon

Using the SHAP values, a sentiment lexicon was constructed. This lexicon lists the most influential words across all samples, showing:

- **Direction:**
  Whether a word tends to push predictions toward a specific sentiment (positive or negative).
- **Importance:**
  How strongly the word affects the model's decision, regardless of direction.

The SHAP values were averaged across all samples and classes, resulting in a table where each word had:

- Mean SHAP value per class
- Overall direction (based on the class it most pushes toward)
- Overall importance (absolute average impact)

This lexicon helps answer questions like: "Which words consistently lead to negative predictions?" or "What terms does the model associate with positive sentiment?"

(See figure 4) We can note that the highest importance that impact the model is the feature "Tariff" but looking a directional graph we see that the feature tariff is pushing the model to the neutral class. SHAP interprets this feature as a having an impact on the decision-making process in the model but

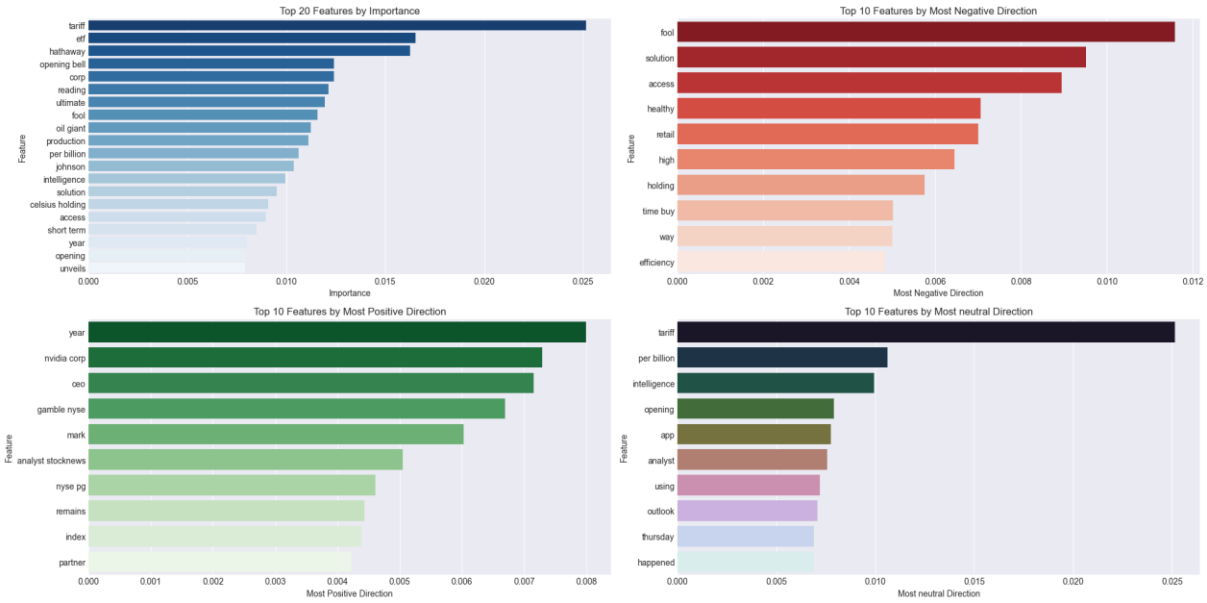


Figure 4

3.5.2 Feature deep-dive

After creating a custom lexicon of our features, we can investigate some of the word and se how they behave in the data.

3.5.2.1 "Tariff" occurrence and returns.

As an example, the word "tariff" was frequently flagged as highly influential. To understand its real-world effect:

- We filtered the dataset to only include articles where "tariff" was present.
- We plotted the number of these articles over time.
- We also measured the market volatility (standard deviation of returns) following those articles.
- Finally, a correlation test was run to see if the frequency of "tariff" mentions was statistically related to increased volatility or market moves.

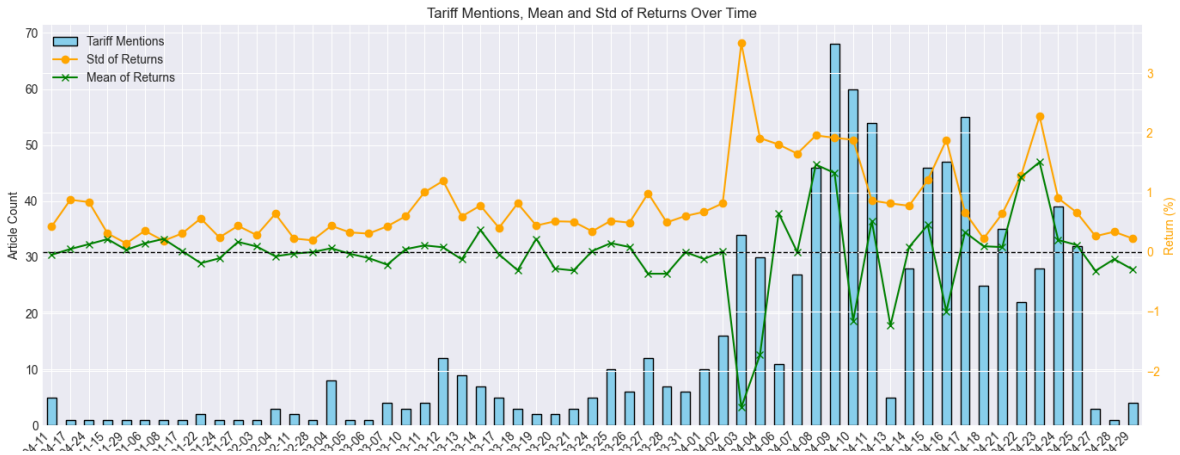


Figure 5

Figure 6

Looking at figure 5 we can notice that as the mentions of the word tariff increases the markets become more volatile but notice that the returns are very random meaning even if the word correlates to increased volatility there is no direct causation to if the price will return positive or negative hence increased push towards the neutral class (see figure 3).

Note that in figure 6 we plot the corrolation between the mentions of the feature “tariff” and the volatility and by the Pearson r metric there is a positive corrolation.

4 Resultat och Diskussion

4.1 Stock chosen

While building the database limits on what Companies should be included needed to be decided. First, the company needed to be public meaning they are traded in the stock market but they also needed to be large enough to be significant, smaller companies tend to be more volatile and move randomly based on traders.

(Source: Bankrate)

Another possibility would have been to use an ETF which most of the time follow the economic growth of the nation but just all companies would 1. not be moved by non-major news releases as "e.g news about apples new iPhone might not move the eft but it would impact the market on apples publicly traded shares"

There also needed to be enough news data for the underlying, small stocks tend not to receive enough reports.

Below (se matrix 1) are the randomly chosen stocks each from a given sector that are considered "Large-cap stocks" meaning that they have a market value of 10+ billion dollars.

Some difficulties arrived as we collected data it appeared that the API used "Alpha Vantage" could not for unknown reasons find news about 2 stocks "JPM and CMCSA" that lead to us removing their data from the training data. However, they still appear here as their returns were still used to calculate the randomised returns.

4.2 News data

The provided news from the "Alpha Vantage" API proved to be very good specially for being free, other API had paywalls or other restrictions. We also tested "Finnhub" but it proved to have some difficulties and being unreliable.

4.2.1 Text data

As the API only gave us the summary of the text "Ca.2000 characters at least" there might have been less data per article than originally was planned, some effort of data scraping the articles was test but proved to be quite inconsistent and time consuming therefore we decided to attempt to combine the title of the article and summary.

4.2.1.1 Text transformation

We decided to work with tokenization as we wanted to analyse the words rather than create a predictive model.

Other options could have been embedding such as with BERT, but it would prove difficult to do the feature analysis as we did which was our goal. (Source: Wikipedia)

Sector	Technology	Financials	Real Estate	Energy	E-commerce / retail	Foods and Consumables	Healthcare	Entertainment
Ticker	AAPL	BAC	PLD	XOM	AMZN	KO	JNJ	NFLX
Ticker	MSFT	GS	O	CVX	WMT	PG	PFE	DIS
Ticker	NVDA	JPM	SPG	SLB	TGT	PEP	PEP	CMCSA

Table 1

4.3 Modelling and evaluation

As we did extensive testing on the data and returns, we decided to move on with a dataset where the news had more impact, looking again at figure 2 we can see that the dataset with the lowest interval “10min” there appears to be the largest delta of variance as well as being significantly different from the random returns. Therefore, we continued to build our model on that dataset.

The chosen models are both well known, Logistic regression is very classic model and would fit this problem and XGBoost were primarily used for its performance. Originally Random Forest classifier was used but was replaced as it first slightly outperformed LR but had heavy computational cost. Therefore, changing to XGBC which essentially works but the main difference being that XGBoost uses a Boosting instead of bagging like Random Forest, but both build upon decision trees. XGboost also has better performance and GPU support making hyperparameter training faster.

The difference between the LR model and XGBC performance was very slim but none the less XGBC outperformed none the less. (see table 2)

The main evaluation you might look at is the Accuracy, but I would also take a look at the F1 score as it puts more emphasis on getting more of the TP, as there is a slight skew towards the neutral class (40% of the dataset) a model may be trained with some bias, this is also a reason to why the threshold for the return were set at 0.2%. An increase in the threshold would have created a larger unbalance.

The final model after hyperparameter tuning also increase some of our metrics where we mostly tried to improve the f1 score so that the model would attempt to be more generalised and not only predict good on the biased class “neutral”. In figure 7 we se a slight increase in overall stats such as Acc from “0.45 to 0.47”, and mainly in the F1 Score we improved from “ 0.447 to 0.464” as well as in AUC where we improved from “0.612 to 0.628”

LR-Model	Precision	Recall	F1-score	XGBC-model	Precision	Recall	F1-score	Support
Negative	0.37	0.28	0.32		0.38	0.35	0.36	930, 29%
Neutral	0.49	0.64	0.56		0.50	0.56	0.53	1275, 39%
Positive	0.42	0.36	0.39		0.44	0.41	0.42	1038, 32%
AVG Score	0.434	0.446	0.433		0.446	0.450	0.447	3242, 100%
Acc = 0.446	AUC = 0.607			Acc = 0.450	AUC = 0.612			TEST_DATA

Table 2

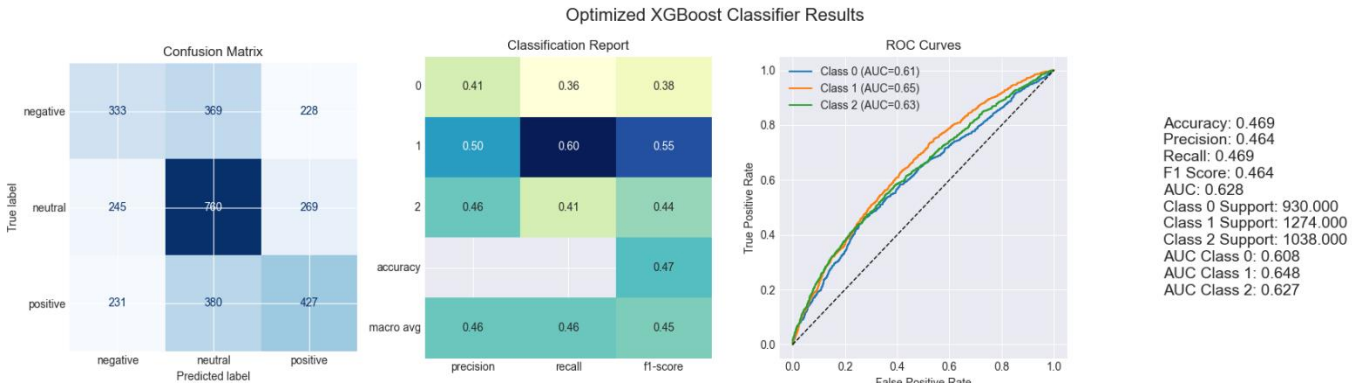


Figure 7

4.4 The Lexicon

After training the classification model using XGBoost, we used SHAP to understand how individual features influenced the model's predictions. The goal was to not only create a predictive model but also derive a market-specific sentiment lexicon, a collection of features that show us the associated impact on the model's decisions, both what feature pushes towards what class but also how much impact that feature has on the model, se example table 3.

Each word in the lexicon was evaluated using SHAP values, which quantify how much a feature contributes to pushing the model toward or away from a particular class. In this case, the classes were negative, neutral, and positive. By averaging the SHAP values across all predictions, we calculated two key metrics:

- **Direction:** The mean SHAP value for a word per class. A positive value suggests the word pushes the model toward that class, while a negative value pushes it away.
- **Importance:** The absolute mean SHAP value across all classes, indicating the overall influence a word has on the model's decisions, regardless of direction.

To better interpret the data, the lexicon also includes a column showing which class the word most strongly influences. For example, if a word like "tariff" had a strong positive SHAP value for the neutral class and smaller (or negative) values for others, it was labelled as most influential toward the neutral prediction.

An interesting outcome was that certain words, like "tariff", showed a strong influence on volatility but had inconsistent directional returns, suggesting a nuanced market reaction (se figure 5). These insights could be used in further research or even integrated into real-time sentiment screening systems.

feature	negative	neutral	positive	importance	most_influential_class	most_corroleted_class
tariff	-0.006095	0.025160	-0.000128	0.031383	neutral	neutral
etf	0.000609	-0.016533	0.001542	0.018685	neutral	positive
fool	0.011589	0.000000	-0.006897	0.018486	negative	negative
solution	0.009512	0.000008	-0.007629	0.017149	negative	negative

Table 3

4.5 Custom text prediction

By testing our own sentences, and let the model predict an outcome. We can quickly see the limitations of our model. The problem arises when new words or unheard-of topics would have appeared in the news.

Here are two examples (se figure 9), the first one has a clear message that the market is expected to move and we can even se what features contribute to the model. Notice how the feature “positive” has a negative correlation which is quite intresting. Compare this text to the other example where we talk about an extraterrestrial invasion which would most likely react badly in the markets but our model has never seen the word Alien or even invasion.

```
Custom Text: The stock market is expected to rise due to positive economic indicators.
Predicted Label: positive
Predicted Probabilities: {'negative': np.float32(0.16726872), 'neutral': np.float32(0.089476734), 'positive': np.float32(0.74325454)}
Features contributing to the prediction: ['due', 'economic', 'expected', 'indicator', 'positive', 'rise']
Feature Directions: {'due': 'positive', 'economic': 'neutral', 'expected': 'positive', 'indicator': 'negative', 'positive': 'negative', 'rise': 'neutral'}

Custom Text: Alien invasion is imminent and nations are preparing for the worst.
Predicted Label: neutral
Predicted Probabilities: {'negative': np.float32(0.18725808), 'neutral': np.float32(0.6196601), 'positive': np.float32(0.19308183)}
Features contributing to the prediction: ['imminent', 'nation', 'preparing', 'worst']
Feature Directions: {'imminent': None, 'nation': 'neutral', 'preparing': 'positive', 'worst': 'neutral'}
```

Figure 8

5 Slutsatser

5.1 How can unstructured text data be processed effectively for machine learning models

To turn chaotic blocks of finance text into something a model can chew on, we did a classic NLP pipeline: clean up the junk (punctuation, weird characters), remove the noise (stopwords), lemmatize words so “buying” and “bought” both become “buy,” and finally vectorize it using TF-IDF. This gave us a nice numerical format that models like Logistic Regression or XGBoost could understand without getting confused by grammar.

5.2 How can we evaluate whether a news article has a positive or negative market impact

We linked news articles to price movements of their corresponding stocks using returns over fixed intervals (like 10min, 1h). If the return was above 0.2%, it was “positive,” below -0.2% was “negative,” and anything in between was “neutral.” This gave us a way to train the model with real market reactions.

5.3 Which specific words or phrases most strongly influence market returns and why

This is where SHAP came in clutch. By measuring each word’s contribution to the prediction, we could build a sentiment lexicon. Words like “tariff” had high importance — they didn’t always mean the market went up or down, but they did consistently affect volatility. That’s the nuance: some words stir up the market, but don’t necessarily push it in a clear direction

5.4 Is there a measurable correlation between the volume of news and market volatility

Yes — at least for some terms. For instance, we saw that as mentions of “tariff” increased, so did volatility. Even if returns were random, the market clearly reacted. So, while you might not get rich trading every “tariff” mention, you’d probably see more turbulence. This may also be followed by the huge impact on the global trade war that is currently taking place, in a couple of years the word tariff might be as important or impact full as it is today

5.5 What are the limitations of news-based sentiment forecasting financial performance

A big one is that models only understand what they’ve seen before. Say something wild happens — aliens land on Wall Street — our model won’t know what to do with that. It also struggles with sarcasm, subtle context, and new slang. And since we only had article summaries (not full texts), some nuance was probably lost. This may be somewhat “fixed” by increasing data both on a

timescale but also on the text and context, we might never know what happens tomorrow or how the world react, and a model probably won't know either. I would say its more of a memory bank of previous patterns than a forecast as news and events are very random and volatile.

5.6 How can we interpret and trust the model's predictions using explainable ai techniques such as SHAP

SHAP was a game-changer. Instead of just getting a "positive" label, we could break down which words influenced that outcome. That helps trust the model — or at least debug it when it makes dumb calls. Plus, it gave us transparency that's useful if you're handing this off to risk managers or traders who want to know "why" as much as "what."

5.7 Can this framework be used for practical applications such as risk assessment or decision-making in financial institutions?

Yes, with more curated and broader data I believe it could, but with a warning label. It's not magic, but it's a powerful tool when combined with other inputs. It could support risk teams by highlighting potentially volatile news or help traders gauge sentiment trends across sectors. If you know the model's blind spots (like aliens), it's a strong add-on to a decision-making toolbox.

Appendix A

API params

- Alpha Vantage

  url = 'https://www.alphavantage.co/query'

  params = {

  'function': 'NEWS_SENTIMENT',

  'tickers': ticker,

  'apikey': apikey,

  'limit': limit,

  'sort': sort

  }

- Tiingo

  url = f"https://api.tiingo.com/iex/{ticker}/prices"

  headers = {

  'Content-Type': 'application/json'

  }

  params = {

  'startDate': current.isoformat(),

  'endDate': chunk_end.isoformat(),

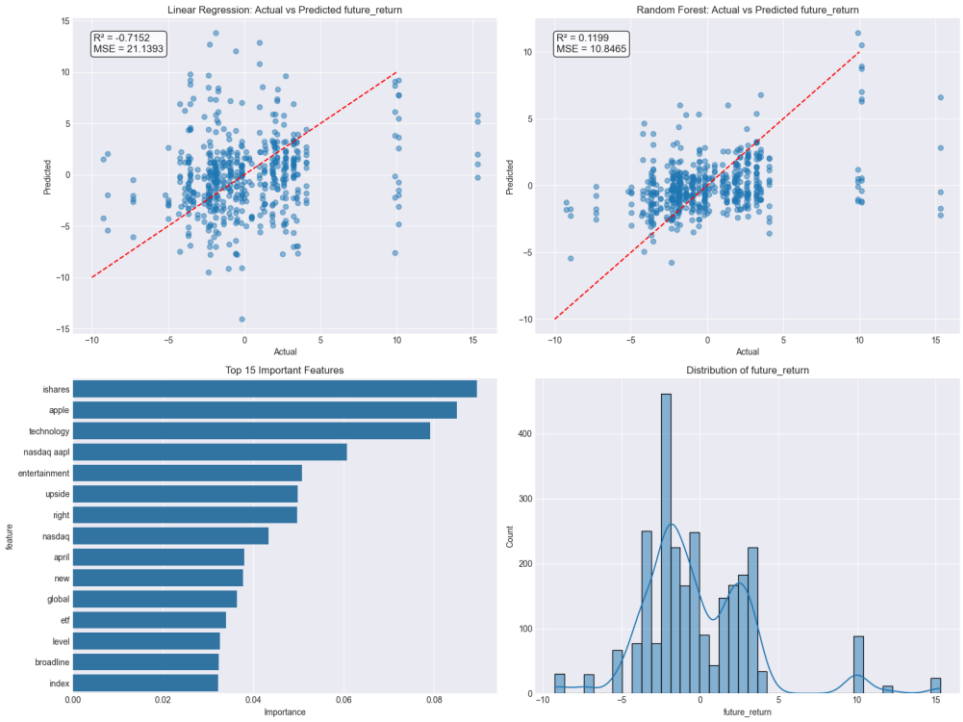
  'resampleFreq': '10min',

  'columns': 'open,high,low,close,volume',

  'token': api_token

  }

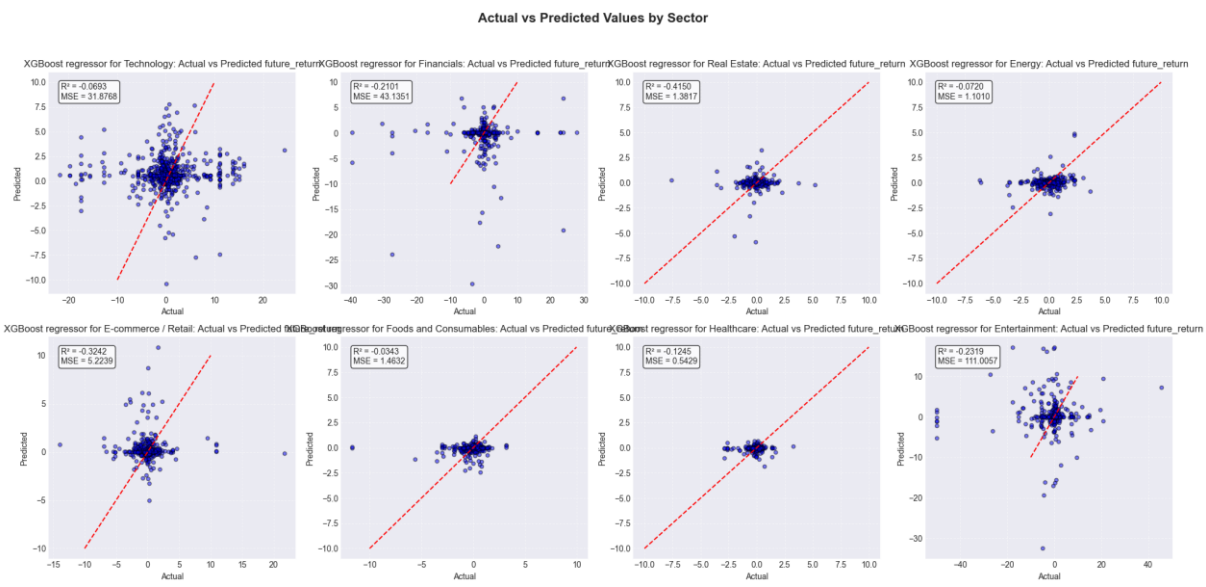
- Linear models result*



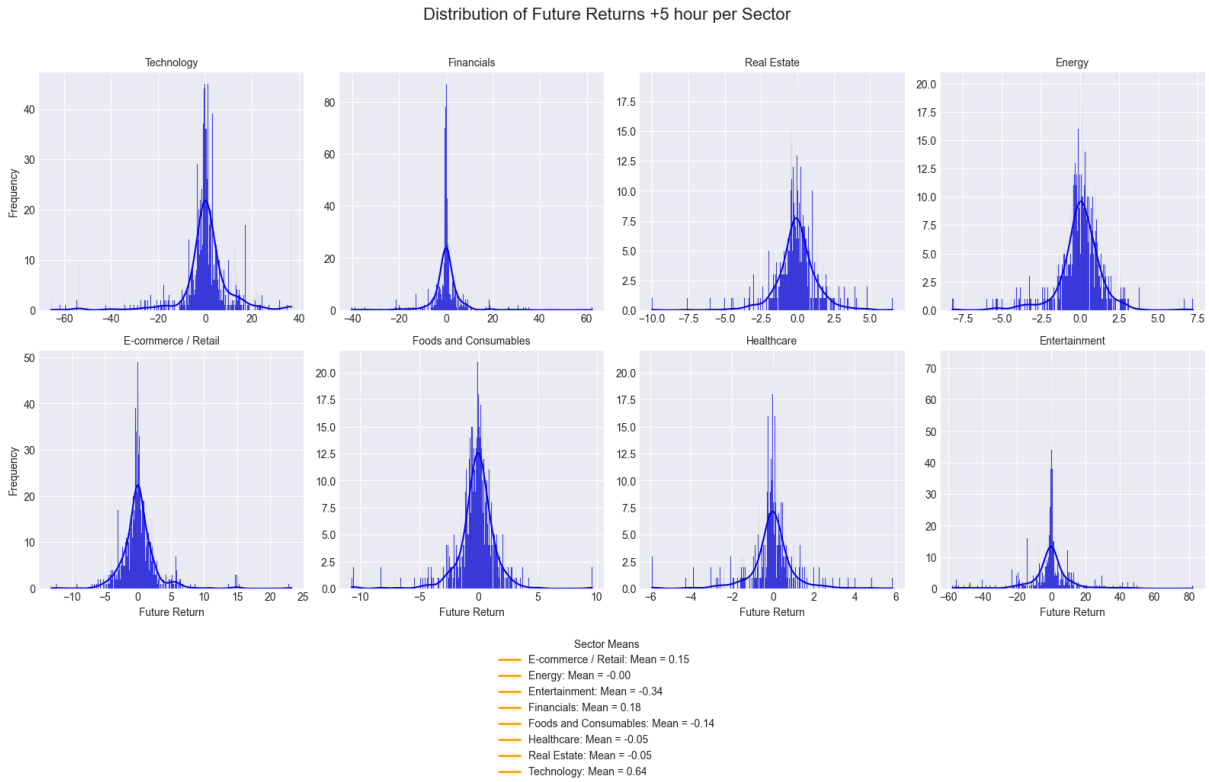
- R^2 score for the model given each sector:

- Sector based values based on numerical prediction not classes,

As this was discontinued, we did not attempt to do the same with classes “to few data points”



- Distrubution per sector, 5 hour interval



References

1. Wikipedia contributors. (n.d.). 2020_stock_market_crash. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/2020_stock_market_crash, Wikipedia

2. Wikipedia contributors. (n.d.). 2025 stock market crash. Wikipedia. Retrieved https://en.wikipedia.org/wiki/2025_stock_market_crash, Wikipedia

3. Miller & Chevalier. (2025). Trade Compliance Flash: Key Takeaways from New BIS Restrictions on AI Chips to China. Retrieved from https://www.millerchevalier.com/publication/trade-compliance-flash-key-takeaways-new-bis-restrictions-ai-chips-china, Miller&Chevalier

4. Energy and Climate Intelligence Unit (ECIU). (2024). Two years of Russia's war on Ukraine: the gas crisis, price rises and energy security. Retrieved from https://eciu.net/insights/2024/two-years-of-russias-war-on-ukraine-the-gas-crisis-price-rises-and-energy-security, ECIU

5. Vattenfall. (n.d.). Rörligt elpris – Prishistorik. Retrieved from https://www.vattenfall.se/elavtal/elpriser/rorligt-elpris/prishistorik/, Vattenfall

6. Wikipedia contributors. (n.d.). Statistical inference. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Statistical_inference, Wikipedia

7. Statistisk dataanalys. Svante Körner, Lars Wahlgren(2015). Statistisk dataanalys. Retrieved from https://www.studentlitteratur.se/kurslitteratur/matematik-och-statistik/statistik/statistisk-dataanalys/?srsltid=AfmBOoqdb4GStN6i45YiTZ6b4yAVMFM7_mMeRF6XkICIDjSRvSFrNCuL; Statistisk dataanalys

8. Wikipedia contributors. (n.d.). Central limit theorem. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Central_limit_theorem; Wikipedia

9. Investopedia. (n.d.). central limit theorem. Retrieved from https://www.investopedia.com/terms/c/central_limit_theorem.asp; Investopedia

10. Investopedia. (n.d.). T-Test. Retrieved from https://www.investopedia.com/terms/t/t-test.asp; Investopedia

11. Statistics How To. (n.d.). Levene Test for Equality of Variances. Retrieved from https://www.statisticshowto.com/levene-test/; Statistics How To

12. Wikipedia contributors. (n.d.). Levene's Test. Retrieved from https://en.wikipedia.org/wiki/Levene%27s_test; Wikipedia

13. Scikit-learn Developers. (n.d.). Logistic Regression. Retrieved from https://scikit-learn.org/0.16/modules/generated/sklearn.linear_model.LogisticRegression.html; Scikit-learn

14. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Retrieved from https://arxiv.org/abs/1603.02754; XGBoost Paper

15. Nvidia Authors. (n.d.). XGboost. Retrieved from https://www.nvidia.com/en-us/glossary/xgboost/; Nvidia

16. Wikipedia contributors. (n.d.). F1 score. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/F-score; Wikipedia

17. Wikipedia contributors. (n.d.). ROC curve. Retrieved from https://en.wikipedia.org/wiki/Receiver_operating_characteristic; Wikipedia

18. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. Retrieved from https://papers.nips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf ; Practical Bayesian Optimization : Bayesian Optimization with Gaussian Process Priors

19. Lundberg, S., & Lee, S.-I. (2017). SHAP (SHapley Additive exPlanations). GitHub. Retrieved from https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html, https://github.com/shap/shap ; paper, github

20. Wikipedia contributors. (n.d.). Logit. Retrieved from https://en.wikipedia.org/wiki/Logit; Wikipedia

21. Alpha Vantage. (n.d.). Stock Market API Documentation. Retrieved from https://www.alphavantage.co/documentation/; Alpha Vantage

22. Tiingo. (n.d.). API Documentation. Retrieved from https://www.tiingo.com/; Tiingo

23. Bankrate. (n.d.). large cap vs small cap-stocks. Retrieved from https://www.bankrate.com/investing/large-cap-vs-small-cap-stocks/; Bankrate