# Scienaptic AI Internship Project

# Project Title: Financial News Sentiment Analyzer

| | |
|---|---|
| **Intern Name:** | Aditya Namdeo |
| **Mentor # 1:** | Sayan Mahajan |
| **Mentor # 2:** | Victoria Alonso |
| **Vertical Head:** | Sayan Mahajan |

# Table of Contents:

**Introduction to problem & Objectives** **3**

- Background and Context 3
- Problem Statement 3
- Objectives 4
- Specific objectives 4
- Expected benefits 5

**Dataset** **5**

- Data gathering 5
- Data Description 5
- Data Preparation 6

**Implementation** **6**

- Assumptions 6
- Project Workflow and Methodology 6
- Models used 6
- Evaluation Metrics 14
- Implementation Challenges 15

**Results** **16**

- Exploratory Data Analysis(EDA) Results 16
- VADER Sentiment Analysis 18
- TEXTBLOB Sentiment Analysis 19
- Named Entity Recognition (NER) with spaCy 19
- N Gram Analysis 20
- TF-IDF Based N Gram Analysis 22
- Sentiment Analysis using FINBERT 23
- Sentence Embeddings and Clustering of Financial News 23
- Model Development and Evaluation Results 24
- Real Time Prediction Deployment Results 25

**Conclusion and Future Scope** **26**

- Conclusion 26
- Implications and Recommendations 26
- Future Scope 27

**Resources & Citation** **27**

- Software and Tools 27
- Research Papers and Publications 28
- Code Repositories 28
- Acknowledgements 28

# Introduction to Problem & Objectives

- ***Background and Context***

The financial markets are highly sensitive to global events, news, and public sentiment. In the modern financial ecosystem, unstructured data sources — particularly financial news — carry valuable, often real-time signals that can influence market behavior and investment decisions. With the rapid rise of Natural Language Processing (NLP) and transformer models, analyzing these large volumes of textual data has become not only feasible but strategically essential.

This project, **Financial News Sentiment Analyzer**, is the topic for my internship at **Scienaptic Systems Pvt Ltd.** The primary aim is to explore how nes sentiment and language patterns correlate with market movements, specifically the Dow Jones Industrial Average (DIJA) index from 2008 to 2016. By leveraging NLP tools like TextBlob, VADER, spaCy, and transformer-based models like FinBERT, the project provides a structured framework to convert the qualitative news data into quantitative insights.

In the context of a financial institution like **Scienaptic Systems Pvt Ltd,** such an analyzer can lead to more informed, data-driven trading strategies and a competitive advantage in identifying profitable opportunities with multiple practical applications like:

- Enhancing **risk management models** by incorporating real-time sentiment signals.
- Supporting **investment teams and portfolio managers** in identifying bullish or bearish market trends early.
- Strengthening **automated trading algorithms** by integrating sentiment-driven features alongside traditional quantitative data.
- Providing a **daily sentiment dashboard** for analysts to monitor market mood.

Overall, the project aligns with Scienaptic Systems' commitment to digital transformation and data-driven decision-making, demonstrating the potential of NLP to unlock actionable insights from financial news data.

- ***Problem Statement***

Financial markets react strongly to daily news, yet traditional models often overlook this unstructured text data. This project aims to automatically analyze financial news headlines to extract sentiment (positive, negative, or neutral) and study its relationship with DJIA index trends. By leveraging NLP techniques and transformer-based models like FinBERT, the goal is to help Scienaptic Systems Pvt Ltd integrate real-time sentiment signals into market analysis, supporting more data-driven and timely investment decisions.

# Internship

## 🔹 *Objectives*

- Build an NLP-based system to automatically classify financial news headlines into positive, negative, or neutral sentiment.
- Analyze and quantify the correlation between extracted news sentiment and stock market movements, specifically DJIA trends.
- Enable near real-time risk assessment and decision support for investment teams by turning unstructured news data into actionable insights.

## 🔹 *Specific Objectives*

To meet the overall goal of analyzing financial news sentiment and its relationship with market trends, the project was broken down into the following specific objectives:

### 1️⃣ Develop an NLP model to classify financial news sentiment

- Preprocessed and cleaned historical DJIA headline data (2008–2016).
- Applied sentiment analysis using rule-based models (**TextBlob**, **VADER**) and a domain-specific transformer model (**FinBERT**) to classify each day's combined headlines as positive, negative, or neutral.
- Explored **named entity recognition (NER)** using **spaCy** to extract key entities and topics from the news.

### 2️⃣ Correlate news sentiment with stock market fluctuations

- Aggregated daily sentiment results and analyzed correlations with DJIA movement labels (up/down).
- Created cross-tabulations and visualized sentiment trends vs market movement.
- Performed **n-gram analysis** to discover language patterns linked to market rises or declines.

### 3️⃣ Provide real-time risk assessments based on breaking financial news

- Engineered sentiment-based features (e.g., average daily sentiment, sentiment volatility, and positive/negative headline counts).
- Generated **sentence embeddings** using `sentence-transformers` and visualized them with **t-SNE** to reveal clusters of similar sentiment days.
- We developed an end-to-end NLP pipeline to predict DJIA movements from financial news by engineering sentiment and text features, training multiple ML models (Logistic Regression, Random Forest, SVM, XGBoost), and evaluating their performance.
- Finally, we built a real-time prediction tool using the best-performing model to generate market direction forecasts and risk scores from new headlines.

### 🔹 *Expected Benefits*

- **Improved decision-making:** Transforms unstructured financial news into sentiment insights, helping investors and analysts anticipate stock trends more accurately.
- **Automated risk assessment:** Flags high-risk or negative sentiment news in real time, supporting proactive market risk monitoring.
- **Efficient market monitoring:** Reduces manual effort required to track and analyze daily financial headlines, enabling faster, data-driven insights.
- **Alignment with industry needs:** Addresses the challenge of leveraging real-time news sentiment for market forecasting, contributing to more responsive and scalable risk management tools within the banking sector.

# Dataset

### 🔹 *Data Gathering*

The data for this project was sourced directly from **Kaggle**, specifically from the Financial News Dataset, which combines historical daily financial news headlines with corresponding Dow Jones Industrial Average (DJIA) market data. Covering the period from **2008-08-08 to 2016-07-01**, this dataset includes over 25 top news headlines per day alongside DJIA open, close, high, and low values. Since the dataset was already cleaned and compiled, there was no need for separate data scraping or external APIs. While this ensured high-quality historical coverage, one limitation is the absence of more recent or real-time data that could be used for live forecasting.

### 🔹 *Dataset Description*

The project uses the Financial News Dataset from Kaggle, which combines daily financial news headlines with Dow Jones Industrial Average (DJIA) stock data. Covering approximately **2008-08-08 to 2016-07-01**, the dataset includes around **1,987 daily records**. Each record consists of:

- A **Date** column
- A **Label** column indicating market movement (1 if DJIA closed higher or unchanged, 0 if it closed lower)
- **25 headline columns** (`Top1` to `Top25`) representing the top news headlines published each day
- Corresponding stock data columns: `Open`, `High`, `Low`, `Close`, `Volume`, and `Adj Volume.`

The label distribution is roughly balanced across the two classes (up vs down days). Overall, the dataset provides a rich mix of **textual features** (headlines) and **numerical market data**, making it suitable for sentiment analysis and market prediction tasks.

🔹 *Data Preparation*

To prepare the dataset for analysis, several preprocessing and cleaning steps were applied. First, the daily top 25 news headlines were **combined into a single text field** to create a consolidated view of each day's news. The combined text was then **cleaned using regular expressions** to remove unwanted characters such as `b'`, `b"`, and escape symbols. We also ensured **missing values** were handled appropriately by excluding incomplete rows and verified the dataset for duplicates. Further, **text normalization** steps like lowercasing and tokenization were used during later analyses (e.g., topic modeling and embedding generation).

# Implementation

🔹 *Assumptions*

- The **sentiment of daily financial news headlines** influences or reflects the daily DJIA market movement.
- Combining the day's top 25 headlines into a **single aggregated text** sufficiently captures the overall sentiment context for that day.
- The dataset from Kaggle is **accurate, clean, and correctly labeled**, requiring no additional manual labeling.
- Sentiment analysis tools like **TextBlob**, **VADER**, and the domain-specific **FinBERT** can effectively detect financial sentiment even in concise or nuanced headlines.
- External market factors beyond the scope of the news headlines (e.g., macroeconomic data releases) are not directly modeled in this analysis.

🔹 *Project Workflow & Methodology*

- **Data Collection:**
  Used two datasets from Kaggle:
  1️⃣`Combined_News_DJIA.csv` containing daily financial news headlines and DJIA movement labels (0/1).
  2️⃣`DJIA_stock_data.csv` containing historical stock prices and trading data.

- **Data Preprocessing:**
  Merged datasets on the date column. Cleaned text by removing null values, punctuation, and special characters. Formatted and combined the top 25 headlines into a single daily text input for analysis.

- **Exploratory Data Analysis (EDA):**
  Explored label distributions, DJIA price trends, and generated article-level statistics (e.g., average word count, frequency analysis). Visualized frequent words and created word clouds for different sentiment classes.

- **Sentiment Analysis:**
  Applied **VADER** and **TextBlob** to compute sentiment polarity scores for daily headlines. Visualized sentiment trends over time and analyzed correlation with DJIA market movements using heatmaps and rolling averages.

- **Named Entity Recognition (NER):**
  Used **spaCy** to extract key entities such as companies, people, and locations frequently mentioned in the headlines.

- **N-Gram Analysis:**
  Computed frequent bigrams and trigrams to identify common phrase patterns. Compared n-gram usage on market up days (label 1) vs. down days (label 0).

- **TF-IDF Analysis:**
  Applied TF-IDF vectorization on unigrams, bigrams, and trigrams to highlight significant terms and phrases across different sentiment or market movement labels.

- **Advanced Sentiment Techniques:**

  ✅ Used **FinBERT**, a financial-domain BERT model, to capture context-aware sentiment specific to finance.
  ✅ Generated sentence embeddings with transformer models and used clustering (e.g., t-SNE) to identify and visualize groups of similar news content.

- **Model Development and Evaluation**
  We developed an end-to-end NLP pipeline to predict DJIA movements from financial news by engineering sentiment and text features, training multiple ML models (Logistic Regression, Random Forest, SVM, XGBoost), and evaluating their performance.
- **Real-Time Prediction Deployment**
  Finally, we built a real-time prediction tool using the best-performing model to generate market direction forecasts and risk scores from new headlines.

🔹 ***Models Used***

To analyze the sentiment of financial news and explore its relationship with DJIA movements, the project combined traditional NLP methods with modern

transformer-based models. These models were selected for their ability to handle short headline texts and to extract both surface-level and context-aware sentiment information relevant to financial markets.

**Models and techniques applied:**

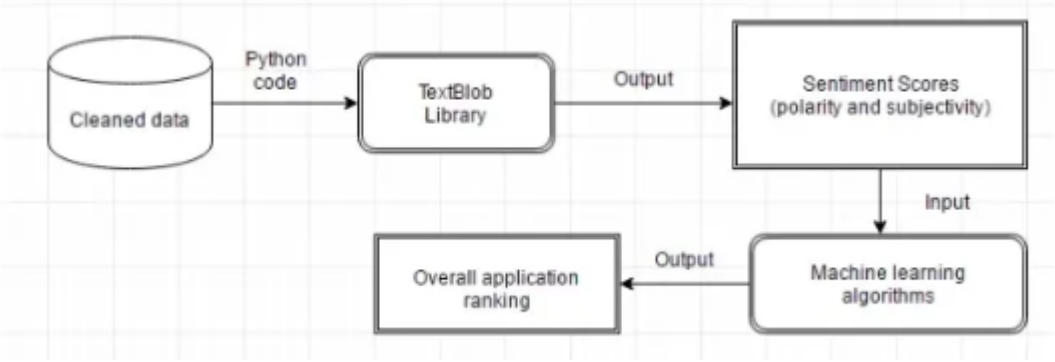
- **TextBlob & VADER:**
  Rule-based sentiment analysis tools used to generate quick polarity and subjectivity scores from aggregated daily headlines.
- **FinBERT:**
  A pre-trained transformer model specialized in financial text, capturing nuanced sentiment that generic models might miss.
- **TF-IDF vectorization:**
  Applied on unigrams, bigrams, and trigrams to identify key terms and phrase patterns across the dataset.
- **Sentence embeddings & clustering:**
  Generated using transformer models (`sentence-transformers`), then visualized using **t-SNE** to identify clusters of similar sentiment days.
- **Logistic Regression:**
  Uses a linear relationship between input features and the probability of DJIA going up or down, serving as a strong, interpretable baseline.
- **Random Forest:**
  Ensembles multiple decision trees to capture complex, non-linear patterns in sentiment and text data for market movement prediction.
- **SVM (Support Vector Machine):**
  Finds an optimal hyperplane to separate up vs. down market days, leveraging high-dimensional sentiment and text features.
- **XGBoost:**
  An advanced gradient boosting algorithm that iteratively improves predictions by focusing on misclassified samples, aiming for better accuracy on financial data.

These combined approaches provided a comprehensive view of daily news sentiment and its potential impact on market movements.

✅ **TextBlob**

TextBlob uses a **lexicon-based approach**, relying on a predefined dictionary of words associated with positive or negative sentiment. When processing a text, it tokenizes it into words, looks up each word's polarity score, and calculates an overall **average sentiment score**. The architecture is lightweight, purely

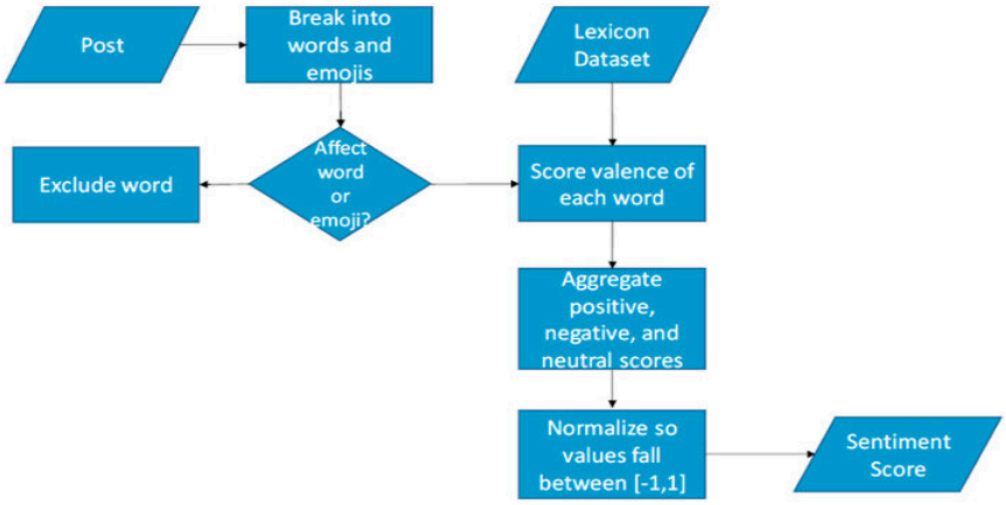
rule-based, making it suitable for quick, real-time applications without requiring large computational resources.




✅ **VADER (Valence Aware Dictionary and sEntiment Reasoner)**
VADER is also rule-based but specifically tuned for **short texts like headlines and social media**. It extends simple lexicons by including rules that account for:

- Capitalization (e.g., "GOOD" vs "good")
- Punctuation (e.g., “!” increases intensity)
- Degree modifiers (e.g., "very good" vs "good")
- Emoticons and slang

Internally, VADER combines these rules with its sentiment dictionary, producing compound scores that better reflect the emotional weight of short phrases.



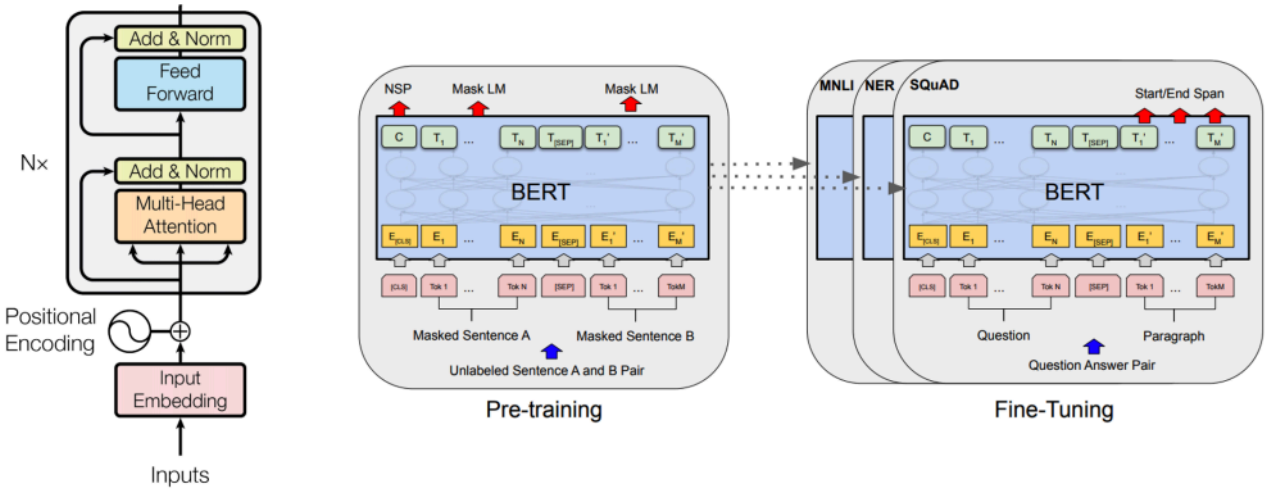

/ VADER Sentiment Scoring Process

✅ **FinBERT**

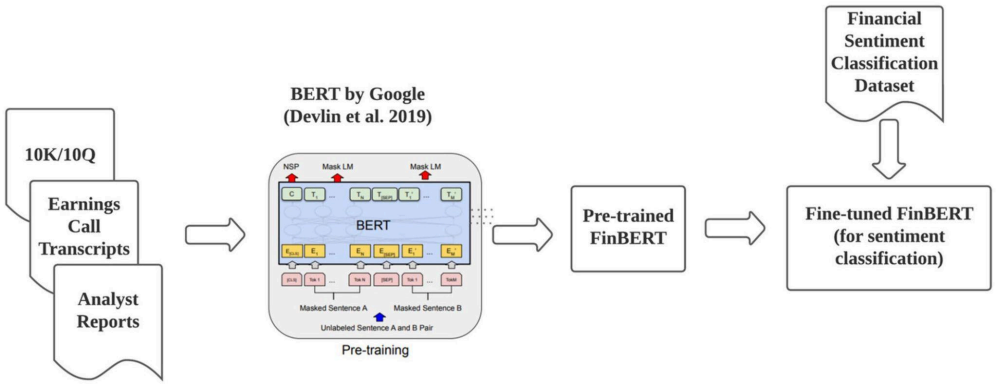
FinBERT is built on the **BERT (Bidirectional Encoder Representations from Transformers)** architecture, which includes:

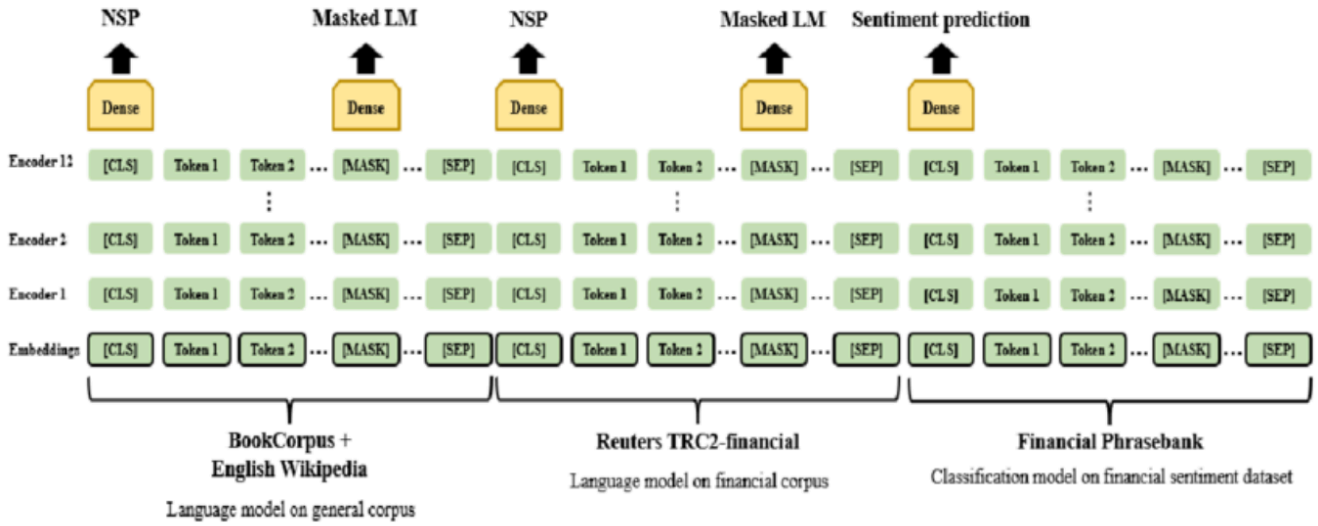
- 12 transformer encoder layers
- Multi-head self-attention mechanisms
- Feed-forward neural networks

  The transformer layers allow FinBERT to **capture bidirectional context**, understanding each word in relation to the words before and after it. FinBERT is further **fine-tuned on financial texts**, making it more sensitive to domain-specific phrases and tone. Its output classifies text into sentiment labels: positive, negative, or neutral.




A multi-layer bidirectional Transformer






FinBERT model architecture (Source: Adapted from [22, 24])

### ✅ TF-IDF Vectorization

TF-IDF is a **statistical text representation technique**. It transforms raw text into numeric vectors by:

- Counting term frequency (how often a word appears in a document)
- Scaling by inverse document frequency (how rare the word is across all documents)

The result highlights words that are frequent in a given day's headlines but rare in the entire corpus, identifying terms that could be market-moving. This method doesn't use deep learning; it's purely based on text statistics.

$$TF(t,d) = \frac{(Number\ of\ occurrences\ of\ term\ t\ in\ document\ d)}{(Total\ number\ of\ terms\ in\ the\ document\ d)}$$

$$IDF(t,D) = \log_e \frac{(Total\ number\ of\ documents\ in\ the\ corpus)}{(Number\ of\ documents\ with\ term\ t\ in\ them)}$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

✅ **Sentence Embeddings & t-SNE Clustering**

Sentence embeddings are produced by **transformer encoders** (similar to FinBERT/BERT) that transform a sentence into a fixed-size high-dimensional vector capturing its semantic meaning.

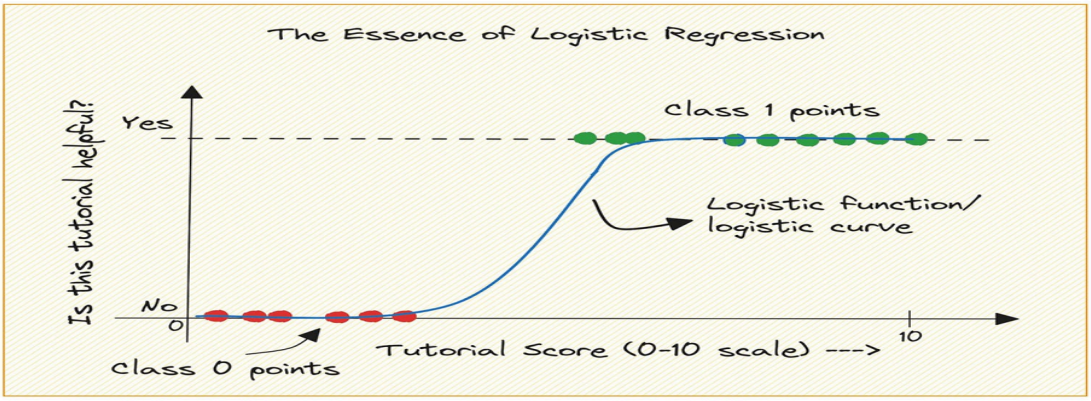
- Each dimension represents a latent feature learned by the model
- Embeddings from similar headlines cluster together in this vector space
  **t-SNE** (t-distributed Stochastic Neighbor Embedding) then reduces these high-dimensional embeddings to 2D or 3D, allowing us to visualize clusters of semantically similar news content.

✅ **Logistic Regression:**

Logistic Regression models the relationship between engineered features (like VADER, TextBlob, FinBERT sentiment scores, TF-IDF vectors, etc.) and the binary target (DJIA up or down).

It calculates the probability that the market will rise using a sigmoid activation over a linear combination of inputs.

Simple and highly interpretable, it serves as a solid baseline for financial news prediction by identifying overall trends and sentiment polarity.

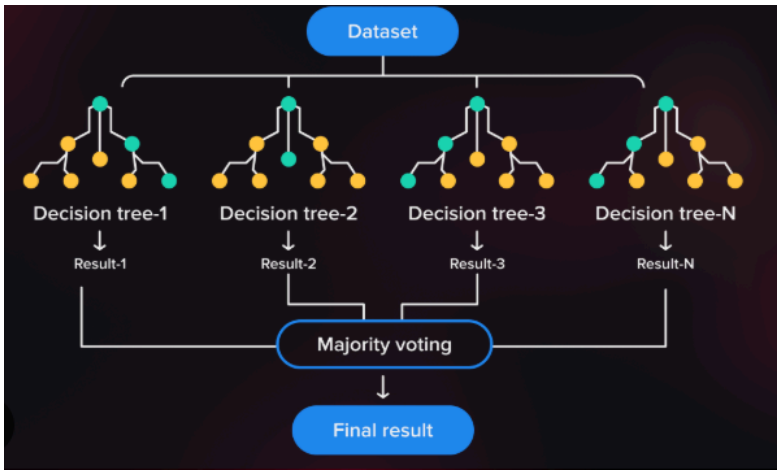



✅ **Random Forest:**

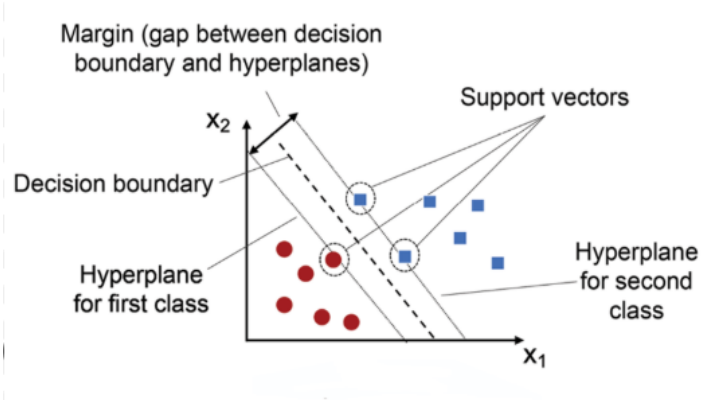
Random Forest creates an ensemble of decision trees, each trained on random subsets of data and features, to predict DJIA movements.

By averaging predictions across many trees, it reduces overfitting and captures complex non-linear relationships between text-based sentiment features and market direction.

This makes it robust to noise and suitable for diverse headline-driven fluctuations.

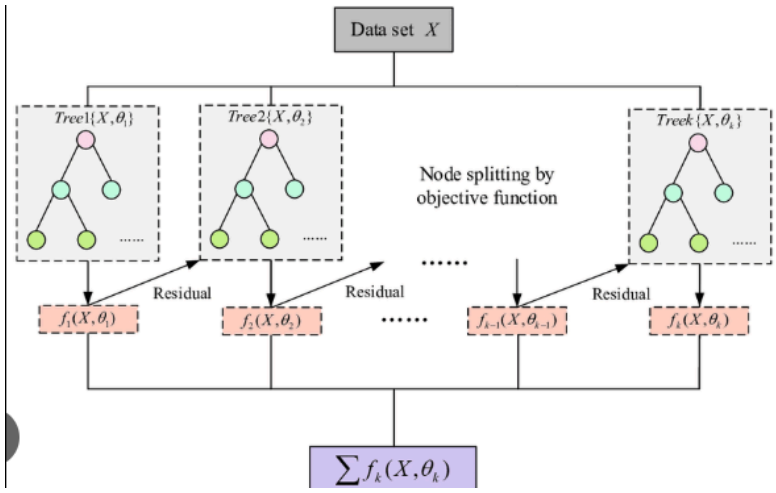



✅ **SVM (Support Vector Machine):**
SVM constructs an optimal hyperplane in high-dimensional feature space (built from sentiment and TF-IDF vectors) that best separates up vs. down days.
It relies on margin maximization, which makes it effective even with relatively few training examples.
By using kernels, SVM can also capture nonlinear boundaries driven by subtle shifts in financial news sentiment.




✅ **XGBoost:**
XGBoost applies gradient boosting on sequential decision trees, where each new tree corrects errors from the previous one.
It efficiently handles large feature spaces and sparsity from TF-IDF data, making it highly suited for text-driven predictions.
Its regularization and built-in handling of missing data improve robustness, aiming to

better capture headline sentiment effects on DJIA trends.




- ***Evaluation Metrics***

To assess the effectiveness of sentiment analysis and text modeling techniques, the project used standard NLP and data analysis metrics aligned with its objective: understanding how news sentiment correlates with DJIA movements rather than predicting exact prices.

**Key metrics and validation methods:**

- **Correlation Analysis:**
  Used Pearson correlation to examine the relationship between computed sentiment scores (from VADER, TextBlob, FinBERT) and actual DJIA movement labels (up or down).
- **Rolling Average & Trend Visualization:**
  Applied rolling averages on sentiment scores and plotted them against DJIA price trends to visually identify consistent patterns or divergences.
- **Word Frequency & TF-IDF Scores:**
  Evaluated most significant words and phrases contributing to sentiment shifts, helping interpret model outputs.
- **Topic Coherence (for LDA):**
  Measured topic coherence to validate the quality and interpretability of discovered topics.
- **t-SNE Clustering Visualization:**
  Used t-SNE plots to visually assess how well sentence embeddings grouped similar headlines, supporting qualitative evaluation.

- **Accuracy:** Measures the overall proportion of correct predictions among all predictions.
- **Precision:** Indicates how many predicted “up” days were actually correct, reflecting prediction reliability.
- **Recall:** Shows how well the model captures all actual “up” days, highlighting sensitivity.
- **F1 Score:** The harmonic mean of precision and recall, balancing both for imbalanced classes.
- **ROC AUC:** Summarizes the model’s ability to distinguish between up and down days across all thresholds.

### 🔹 *Implementation Challenges*

During the project, several practical and technical challenges emerged — from handling noisy textual data to managing the complexity of transformer-based models. These challenges were addressed through preprocessing, careful model choice, and iterative testing.

**Key challenges and mitigation strategies:**

- **Noisy and short headline texts:**
  Headlines lacked full context, making sentiment harder to interpret. This was mitigated by combining multiple daily headlines and aggregating sentiment scores.
- **Computational resources for transformers:**
  Running models like FinBERT and sentence embeddings was resource-intensive. We optimized by batching data and limiting embedding generation to key periods.
- **Imbalanced data distribution:**
  The number of up vs. down days was not perfectly balanced. Visualization and correlation analysis helped interpret trends rather than purely relying on balanced prediction.
- **Subjectivity in lexicon-based models:**
  Rule-based tools like VADER and TextBlob sometimes misclassify neutral finance-specific terms. Using FinBERT, trained on financial text, reduced this bias.

These trade-offs and solutions kept the analysis robust, while also highlighting areas for potential future improvements like advanced fine-tuning and real-time streaming.

# Results

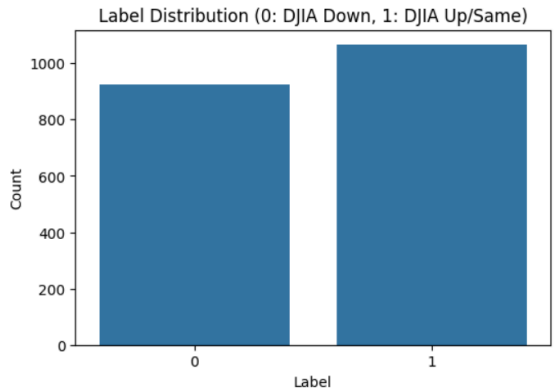
- ***Exploratory Data Analysis (EDA) Results:***

1) Basic dataset overview:

```
dtypes: float64(5), int64(2), object(27)
memory usage: 528.5+ KB
None
              Label          Open          High           Low         Close
count   1989.000000   1989.000000   1989.000000   1989.000000   1989.000000
mean       0.535445  13459.116048  13541.303173  13372.931728  13463.032255
std        0.498867   3143.281634   3136.271725   3150.420934   3144.006996
min        0.000000   6547.009766   6709.609863   6469.950195   6547.049805
25%        0.000000  10907.339844  11000.980469  10824.759766  10913.379883
50%        1.000000  13022.049805  13088.110352  12953.129883  13025.580078
75%        1.000000  16477.699219  16550.070312  16392.769531  16478.410156
max        1.000000  18315.060547  18351.359375  18272.560547  18312.390625

             Volume     Adj Close
count  1.989000e+03   1989.000000
mean   1.628110e+08  13463.032255
std    9.392343e+07   3144.006996
min    8.410000e+06   6547.049805
25%    1.000000e+08  10913.379883
50%    1.351700e+08  13025.580078
75%    1.926000e+08  16478.410156
max    6.749200e+08  18312.390625
```

We can see the overview from the above image.

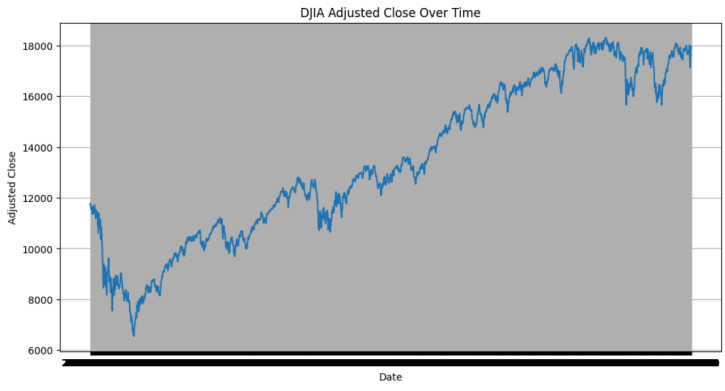
2)Distribution of DJIA labels showing proportion of up vs. down market days:




```
Label
1    53.544495
0    46.455505
```

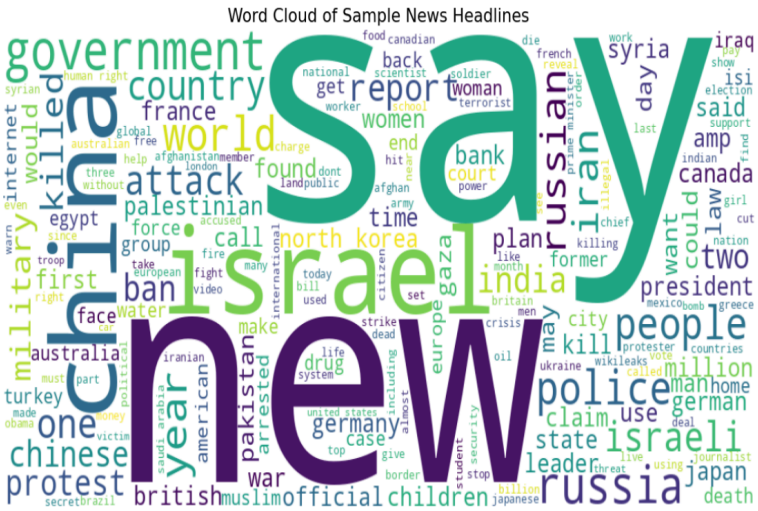
Visualizes the distribution of DJIA movement labels (0 for down, 1 for up/same) to check for class imbalance, and prints their percentages to understand the proportion of market rise vs. fall days in the dataset.

3)Plotting DJIA Trend Over Time:




Visualizes the historical trend of DJIA adjusted closing prices over the data's time span, helping identify overall Market movements, peaks and dips.

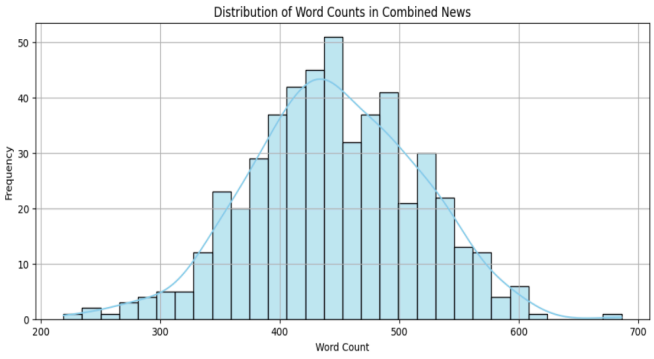
4)Word Cloud of Sample News and Word Cloud for Positive vs. Negative News

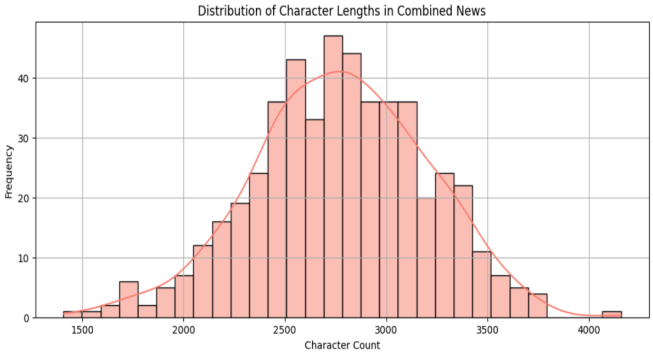




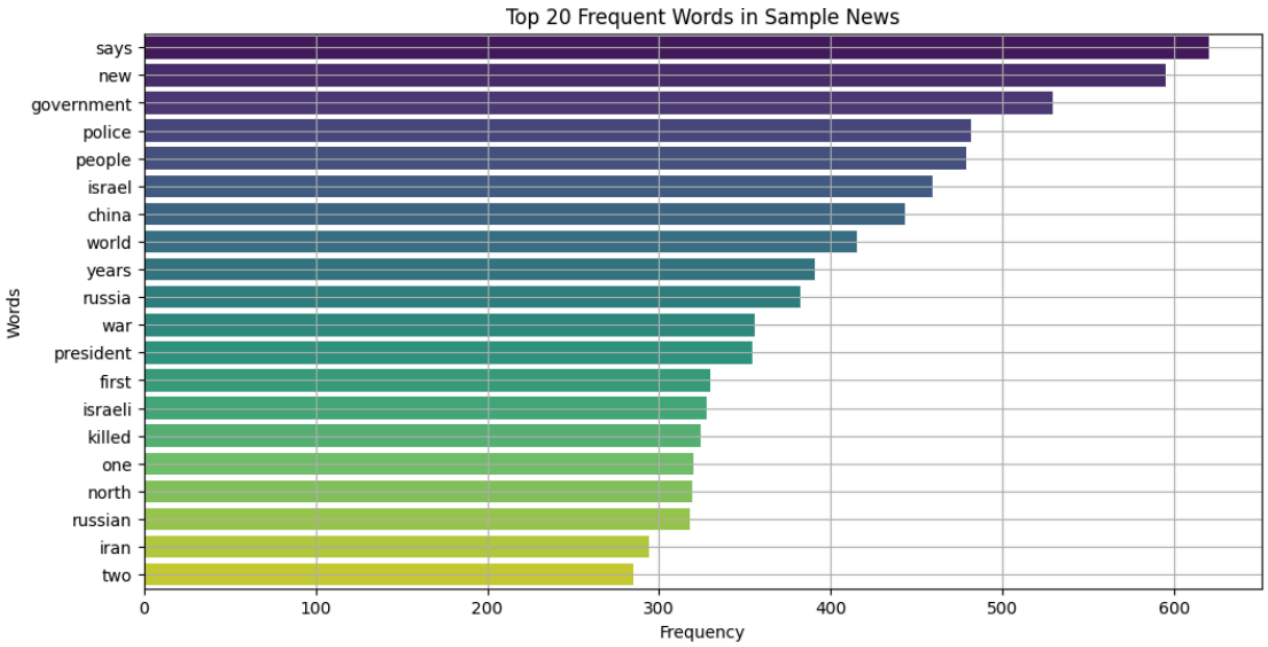
5) Distribution of Headline Lengths (Word Count & Length)

Calculates and plots the distributions of word counts and character lengths in the combined daily headlines to understand typical text length and identify outliers or very short/long days.




6) Word Frequency Bar Chart

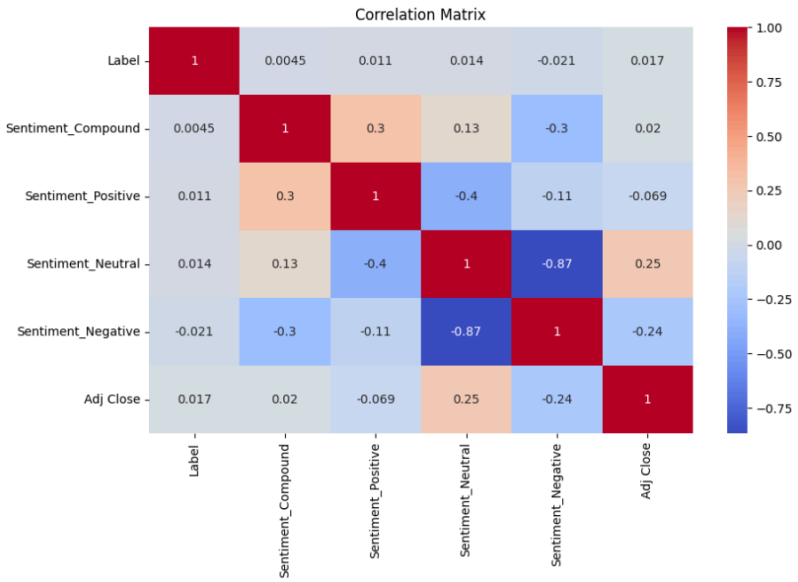


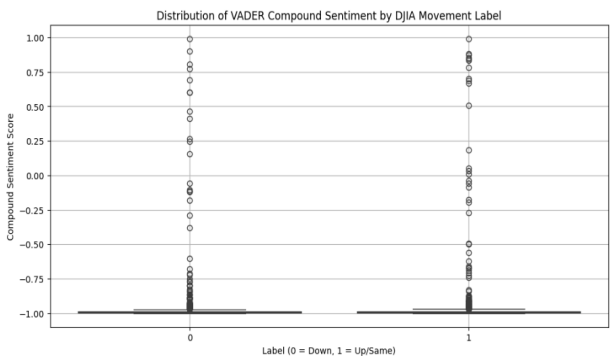

The above gives the top 20 frequent words in the sample news.

- ***VADER Sentiment Analysis:***

Here,we apply VADER to generate compound, positive, neutral, and negative sentiment scores for each day's combined news, then summarizes these scores with descriptive statistics to understand overall sentiment distribution.

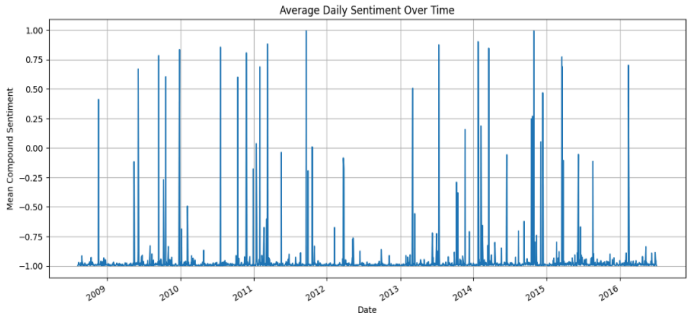
| | Sentiment_Compound | Sentiment_Positive | Sentiment_Neutral | Sentiment_Negative |
|---|---|---|---|---|
| **count** | 1989.000000 | 1989.000000 | 1989.000000 | 1989.000000 |
| **mean** | -0.957369 | 0.065675 | 0.772018 | 0.162315 |
| **std** | 0.199673 | 0.020968 | 0.041819 | 0.038575 |
| **min** | -0.999500 | 0.007000 | 0.588000 | 0.059000 |
| **25%** | -0.996400 | 0.051000 | 0.746000 | 0.135000 |
| **50%** | -0.993200 | 0.064000 | 0.773000 | 0.159000 |
| **75%** | -0.985500 | 0.079000 | 0.802000 | 0.188000 |
| **max** | 0.991700 | 0.153000 | 0.894000 | 0.316000 |

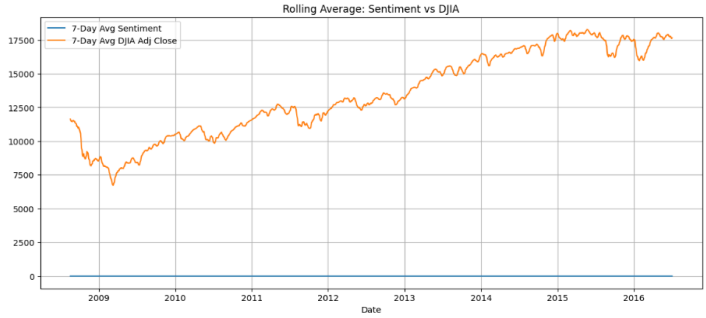


Correlation heatmap among VADER sentiments

Sentiment distribution by DJIA label



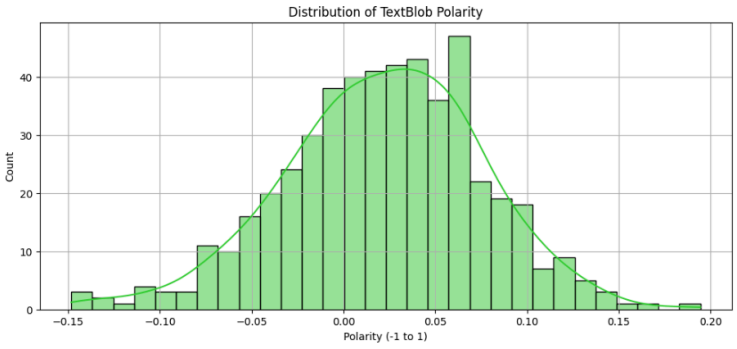



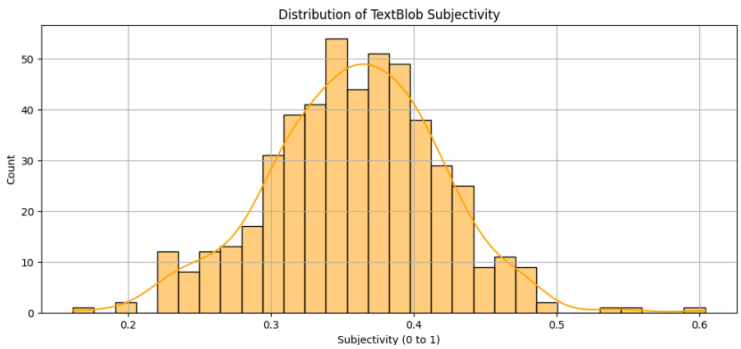

Average daily sentiments over time

Rolling averages of sentiments and DJIA

- ***TEXTBLOB Sentiment Analysis:***

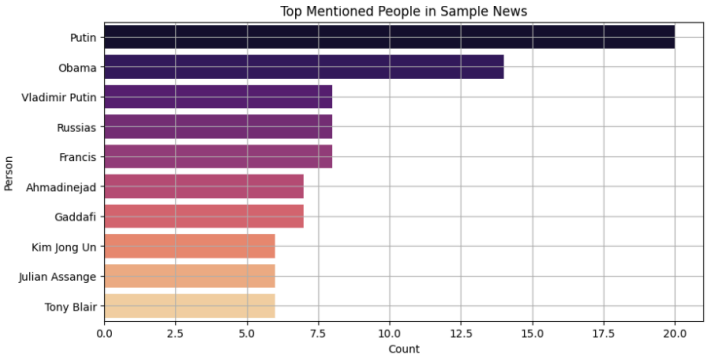
Calculates polarity (positive/negative tone) and subjectivity (opinion vs. fact) scores for each combined news entry using TextBlob, then visualizes their distributions to see overall sentiment trends and text subjectivity.

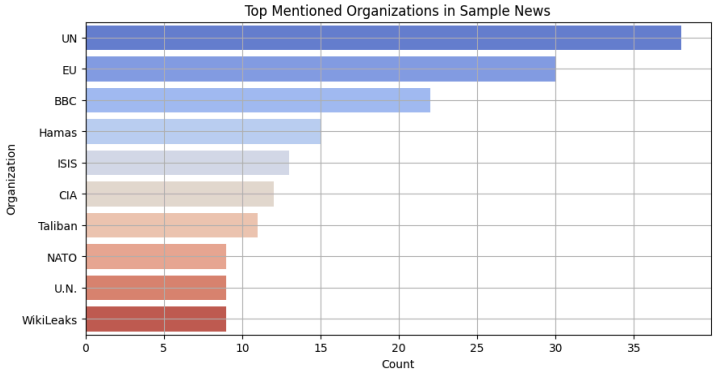



- ***Named Entity Recognition(NER) with spaCy:***

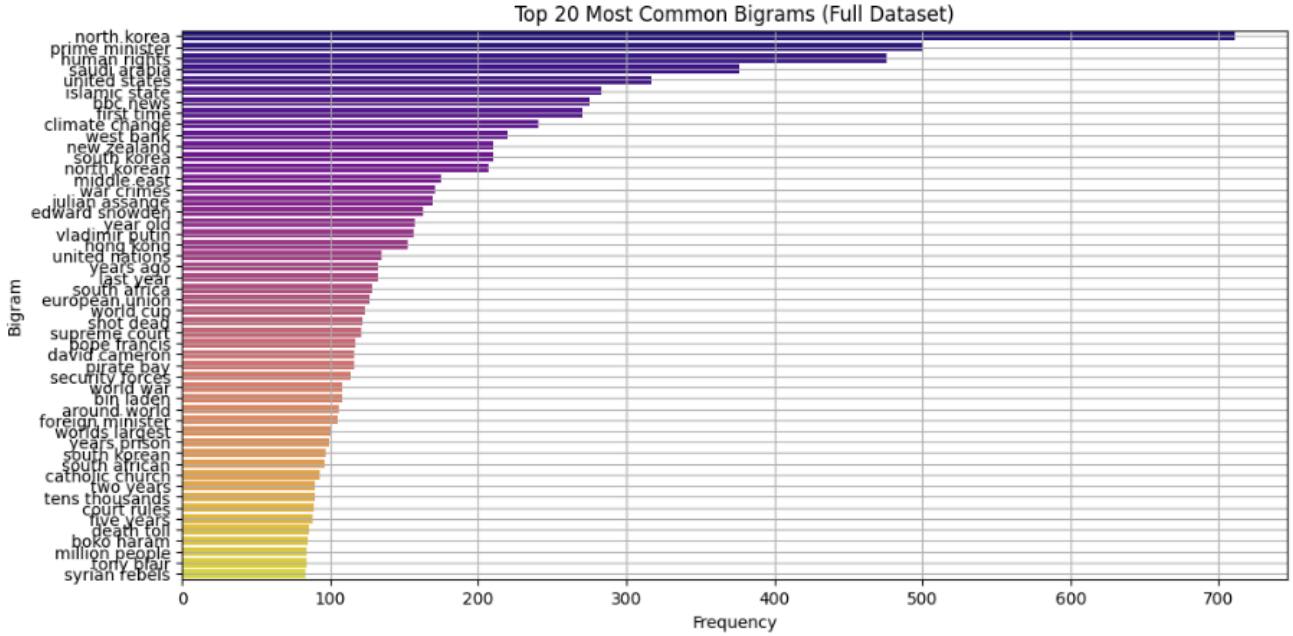
Uses spaCy to extract and count the most frequently mentioned organizations and people in a sample of headlines, then visualizes the top 10 entities with bar charts to highlight key players often discussed in financial news.



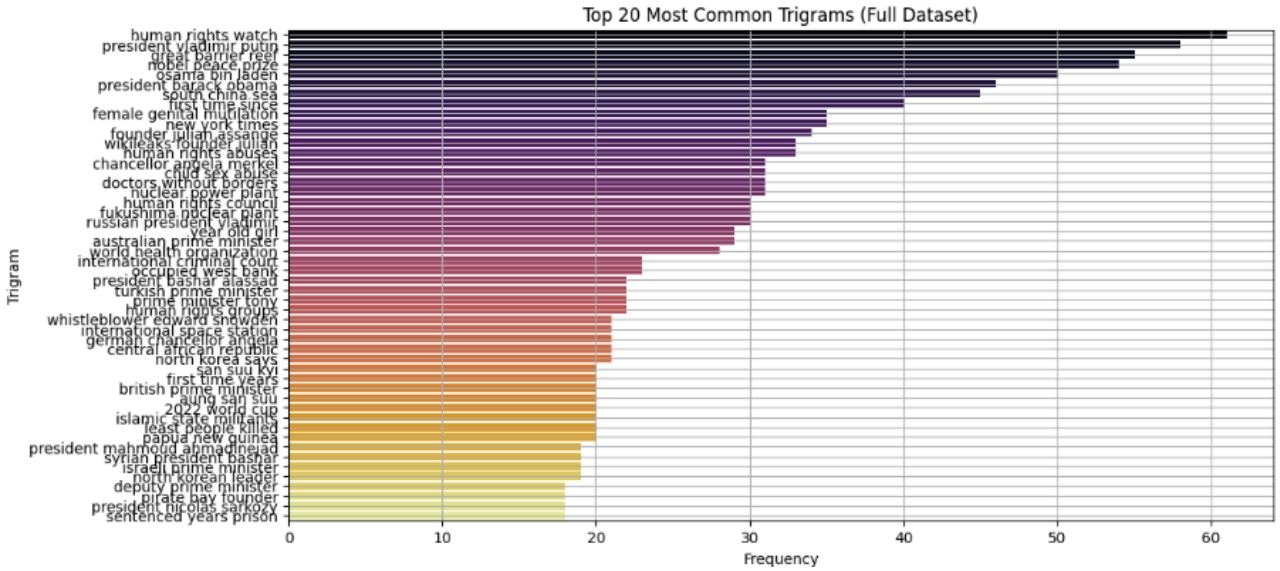




- ***N Gram Analysis :***

1) **Bigram Frequency Analysis:**
   Generates and counts the most frequent two-word combinations (bigrams) across the entire cleaned news dataset, then visualizes the top bigrams to reveal common phrase patterns in financial headlines.

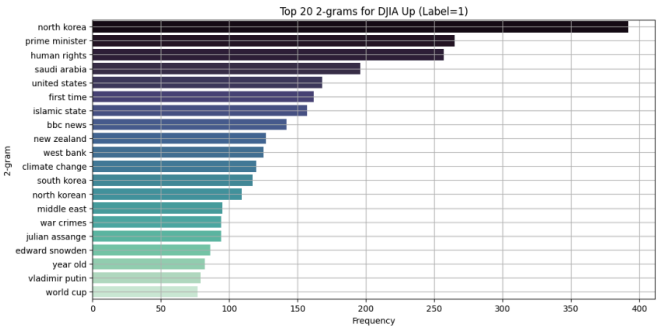


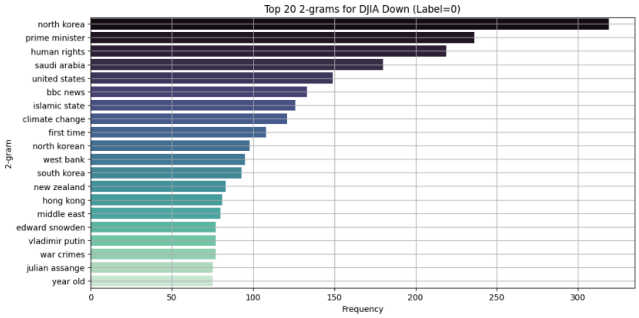

2) **Trigram Frequency Analysis:**
   Extracts and ranks the most frequent three-word combinations (trigrams) from the full cleaned news dataset, then plots the top trigrams to highlight recurring multi-word phrases in financial headlines.



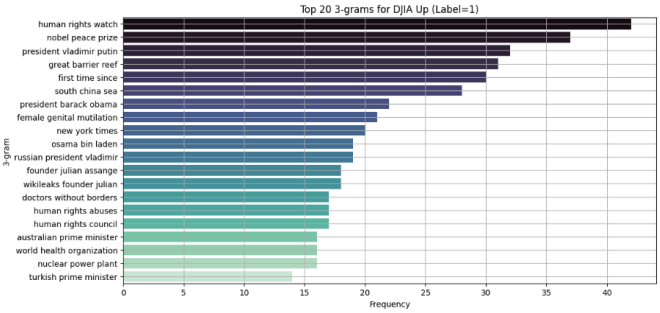

3) **Comparative N-gram Analysis for Market Movement:**

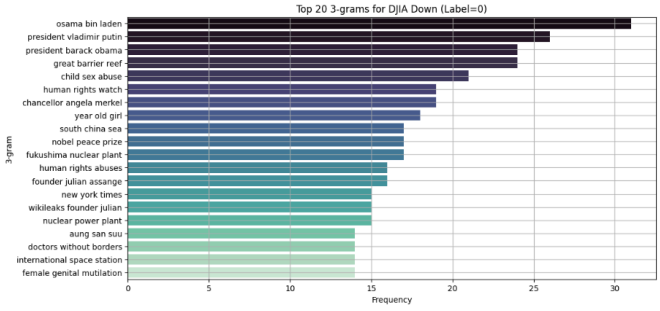
Cleans and splits the dataset into days when DJIA went up vs. down, then plots the most common **bigrams** separately for each label to reveal distinct language patterns linked to rising or falling markets.




Extracts and visualizes the top **trigrams** separately for days when DJIA increased vs. decreased, helping to identify unique three-word phrase patterns that may be associated with positive or negative market sentiment.

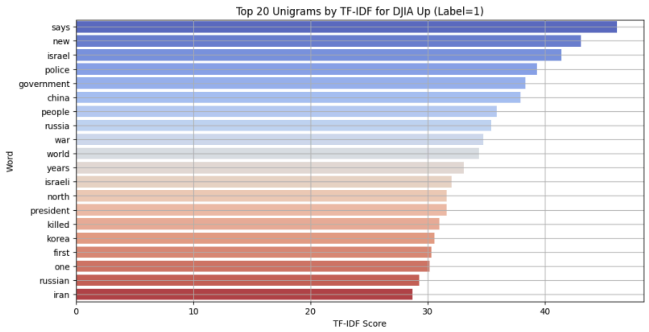


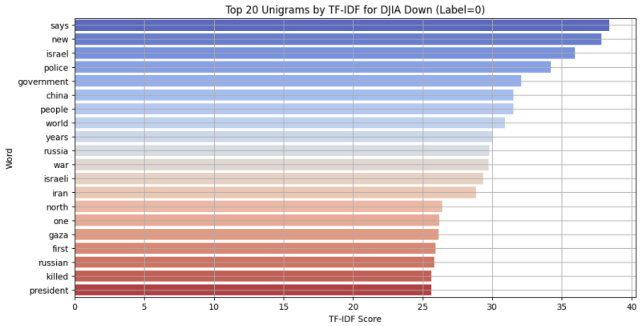



## 🔹 TF-IDF Based *N Gram Analysis :*

### 1)TF-IDF Analysis of Unigrams by Market Direction:

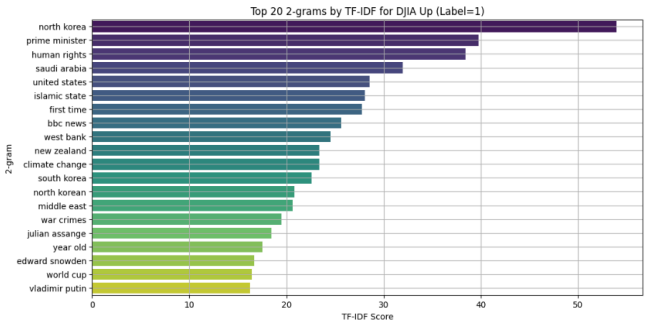
Calculates and plots the top unigrams with the highest TF-IDF scores separately for days when DJIA went up vs. down, highlighting the most distinctive and informative words characterizing positive and negative market movements.

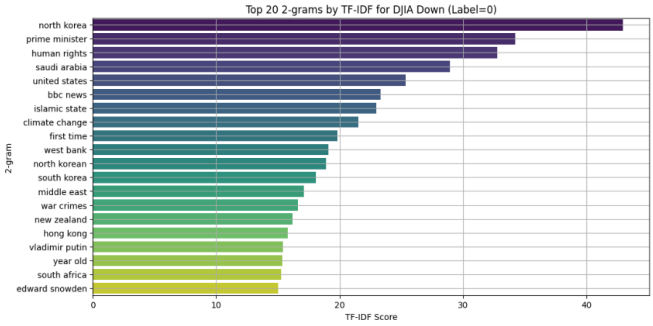



### 2)TF-IDF Analysis of Bigrams by Market Direction:

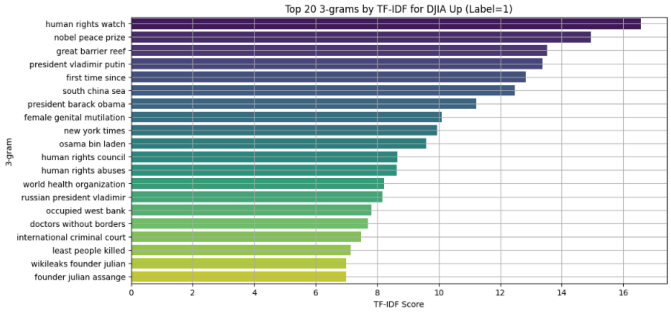
Computes and visualizes the most distinctive bigrams (two-word combinations) with the highest TF-IDF scores separately for DJIA up vs. down days, helping reveal key phrase patterns linked to each market condition.

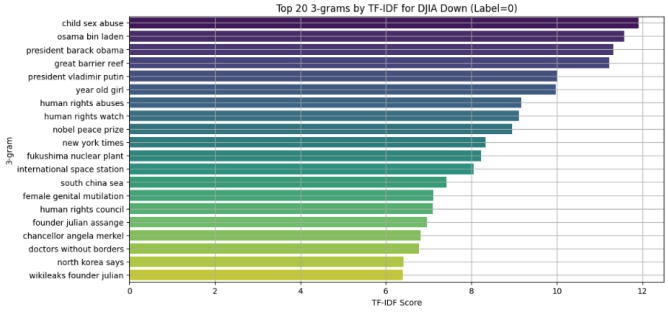



### 3) TF-IDF Analysis of Trigrams by Market Direction:

Identifies and visualizes the top trigrams (three-word combinations) with the highest TF-IDF scores for days when the DJIA increased vs. decreased, revealing richer multi-word patterns distinctive to each market trend.



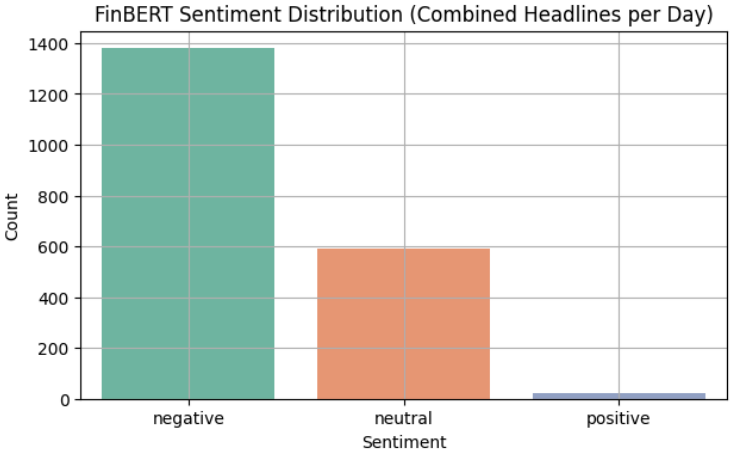



# Internship

## 🔹 *Sentiment Analysis using FINBERT :*

Loads the ProsusAI/finbert transformer model — specifically trained for financial text — and applies it to each day's combined news headlines. For each headline set, it predicts a sentiment label (positive, negative, or neutral) and a confidence score, then adds these to the dataframe. Finally, it visualizes the overall distribution of predicted sentiments to show whether financial news is generally seen as positive, negative, or neutral across the dataset.

This approach provides a domain-adapted view of sentiment, more aligned with how financial markets might interpret news compared to generic sentiment tools.

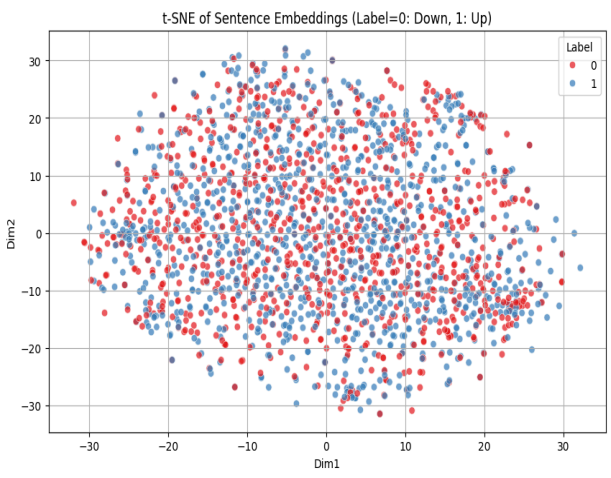


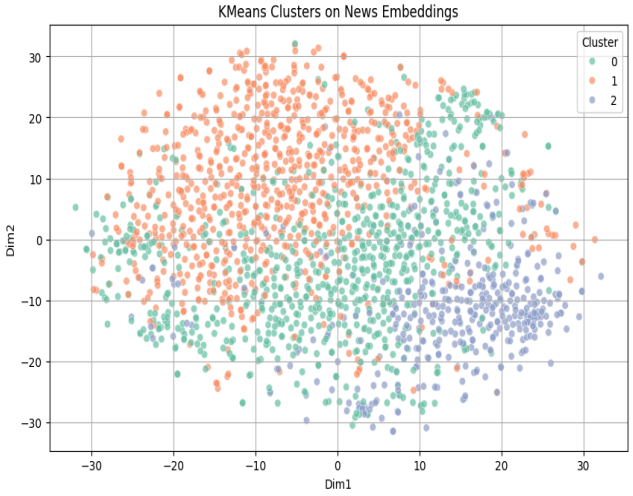

| DJIA Movement (Label) | 0 | 1 |
|---|---|---|
| FinBERT | | |
| negative | 639 | 740 |
| neutral | 277 | 312 |
| positive | 8 | 13 |

## 🔹 *Sentence Embeddings & Clustering of Financial News :*

Uses the pre-trained **all-MiniLM-L6-v2** model from SentenceTransformers to convert each day's combined news headlines into dense vector embeddings that capture semantic meaning.
Then, it reduces these embeddings to 2D with **t-SNE** for visualization, coloring by DJIA label (up or down) to explore if sentiment patterns align with market movement.
Finally, it applies **KMeans clustering** on these embeddings and plots the clusters, helping reveal hidden structures and groups of similar news days that might influence the market similarly.

This approach adds a semantic layer to the analysis, beyond simple word counts or sentiment, by grouping days based on deeper textual similarity.



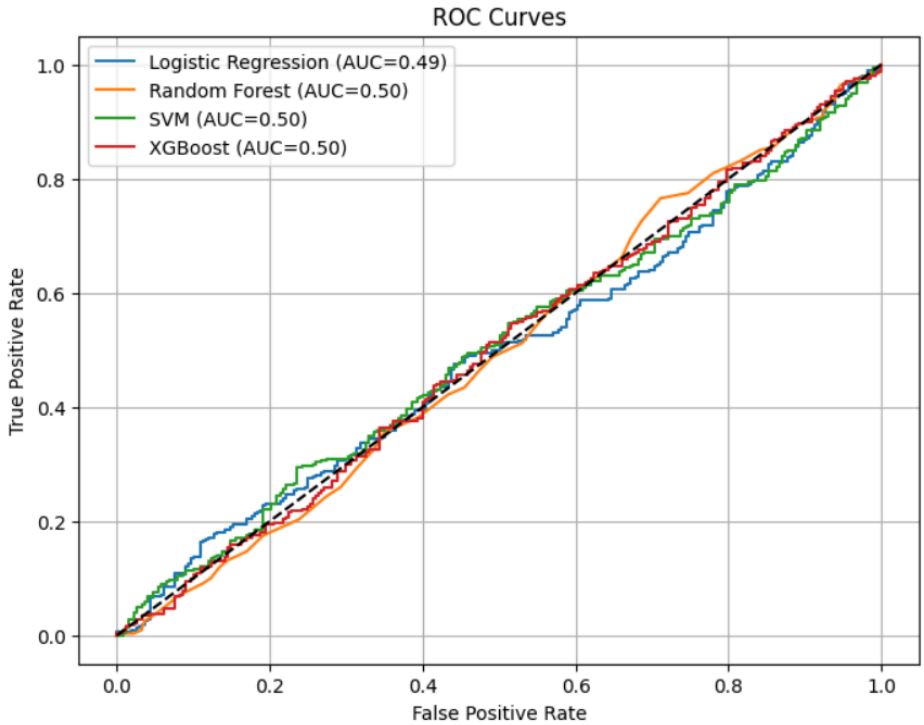
- ***Model Development and Evaluation Results:***

```
Model: Logistic Regression
Accuracy: 0.5309882747068677
Precision: 0.5352112676056338
Recall: 0.95
F1 Score: 0.6846846846846847
ROC AUC: 0.49247518050541517

Confusion Matrix:
 [[ 13 264]
 [ 16 304]]
```

```
Model: Random Forest
Accuracy: 0.5058626465661642
Precision: 0.5370919881305638
Recall: 0.565625
F1 Score: 0.5509893455098934
ROC AUC: 0.49688628158844766

Confusion Matrix:
 [[121 156]
 [139 181]]
```

```
Model: SVM
Accuracy: 0.5175879396984925
Precision: 0.5317460317460317
Recall: 0.8375
F1 Score: 0.6504854368932039
ROC AUC: 0.5017148014440433

Confusion Matrix:
 [[ 41 236]
 [ 52 268]]
```

```
Model: XGBoost
Accuracy: 0.5125628140703518
Precision: 0.5403899721448467
Recall: 0.60625
F1 Score: 0.5714285714285714
ROC AUC: 0.49931182310469313

Confusion Matrix:
 [[112 165]
 [126 194]]
```



- ***Comparison of Models:***

| Model | Accuracy | Precision | Recall | F1 Score | ROC AUC |
|---|---|---|---|---|---|
| Logistic Regression | **0.53** | 0.53 | **0.95** | **0.68** | 0.49 |
| SVM | 0.518 | 0.53 | 0.83 | 0.65 | **0.50** |
| XGBoost | 0.51 | **0.54** | 0.60 | 0.57 | 0.49 |
| Random Forest | 0.50 | 0.53 | 0.56 | 0.55 | 0.49 |

Observations:

- Logistic Regression has the best recall (0.95) and best F1 score (0.68).
- SVM is slightly behind but more balanced.
- XGBoost has higher precision but lower recall.
- Overall, Logistic Regression shows best balance for predicting “market up” days (high recall is useful for flagging possible market ups).

- ***Real-Time Prediction Deployment Results:***

We built a real time detection model which when given a financial news text returns the VADER score, TEXTBLOB polarity, FINBERT label, logistic regression model prediction and risk score.

```
new_headlines = [
    "Federal Reserve announces unexpected interest rate hike",
    "Tech stocks rally as major company reports record profits"
]

result_df = real_time_predict_lr(new_headlines)
print(result_df)
```

```
                                            Headline  VADER_Compound
0  Federal Reserve announces unexpected interest ...          0.4588
1  Tech stocks rally as major company reports rec...          0.4404

   TextBlob_Polarity FinBERT_Label Prediction  Risk_Score
0             0.1000      negative  Market Up       0.596
1             0.0625      negative  Market Up       0.629
```

# Conclusion and Future Scope

🔹 ***Conclusion:***

In this project, we successfully built an end-to-end **Financial News Sentiment Analyzer** tailored for stock market trend prediction—aligning closely with Scienaptic Systems' objective of enabling **near real-time risk assessment and decision support**

Starting with comprehensive data preprocessing and EDA, we explored sentiment trends using **VADER, TextBlob, and FinBERT**, performed **NER, N-gram and TF-IDF**, and extracted meaningful features to train multiple ML models (**Logistic Regression, Random Forest, SVM, XGBoost**) for predicting DJIA movements.

By integrating a real-time prediction pipeline, the system can now convert unstructured financial headlines into actionable insights—helping analysts anticipate market shifts and manage investment risk more effectively.

Overall, this project demonstrates the power of combining NLP and machine learning to transform large volumes of financial text into **timely, data-driven insights** for investment teams at Scienaptic System.

🔹 ***Implications and Recommendations:***

This project highlights how advanced **NLP techniques and sentiment analysis** can significantly improve financial market monitoring and decision-making, especially in fintech contexts like Scienaptic Systems.

By transforming large volumes of unstructured news data into quantifiable sentiment scores and predictive insights, investment teams gain a clearer, real-time view of market sentiment trends and associated risks.

✅ **Recommendations:**

- Integrate the developed sentiment pipeline into Scienaptic Systems' existing analytics platform for **daily market monitoring.**
- Use FinBERT and topic modeling to automatically **flag high-impact news** for investment teams, enabling proactive risk management.
- Extend the model to include **real-time news feeds or social media sentiment** for even faster detection of market-moving events.
- Regularly retrain and validate models with recent data to adapt to evolving market language and trends.

Together, these steps can help Scienaptic Systems move toward **automated, sentiment-driven forecasting**, making investment decisions both faster and more data-informed.

### 🔹 *Future Scope:*

While this project lays a strong foundation, there are several promising directions to deepen and expand its impact:

- **Real-Time Integration:** Connect the sentiment analysis pipeline to **live financial news feeds or social media APIs** to provide real-time alerts and risk signals to investment teams.
- **Model Enhancement:** Experiment with more advanced architectures, such as **LSTM, GRU, or transformer-based models (e.g., fine-tuned BERT variants)** to capture sequential context and subtle language nuances in news.
- **Granular Sentiment Scoring:** Move from daily-level sentiment to **headline- or sentence-level scoring**, enabling finer analysis of news impact.
- **Broader Data Sources:** Incorporate other relevant data like **macroeconomic indicators, analyst reports, and social sentiment** to build multi-modal predictive models.
- **Explainability:** Add explainable AI tools (e.g., SHAP, LIME) to better understand why the model predicts certain market movements, increasing trust for analysts and stakeholders.
- **Custom Risk Index:** Develop a specialized **risk index or dashboard** combining sentiment, market volatility, and historical patterns for decision support.

By exploring these avenues, future projects can build on this work to create even **smarter, adaptive, and explainable sentiment-driven forecasting systems** tailored for the dynamic world of financial markets.

# Resources & Citation

### 🔹 *Software and Tools*

The following software, frameworks, and libraries were used to implement, analyze, and visualize the project:

- **Python** (v3.11): Primary programming language for data preprocessing, analysis, and modeling.
- **Pandas** (v2.0+): Data manipulation and cleaning.
- **NumPy** (v1.24+): Numerical computing and array operations.
- **Scikit-learn** (v1.3+): Machine learning models, TF-IDF, n-gram analysis, evaluation metrics.
- **NLTK** (v3.8+): Text preprocessing and stopwords removal.
- **spaCy** (v3.6+): Named Entity Recognition (NER).
- **TextBlob** (v0.17+): Sentiment polarity and subjectivity analysis.

- **VADER Sentiment** (from `nltk.sentiment`): Financial sentiment scoring.
- **Transformers (Hugging Face)** (v4.40+): FinBERT model for domain-specific sentiment analysis and embeddings.
- **Sentence-Transformers** (v2.6+): Generating sentence embeddings for clustering and visualization.
- **XGBoost** (v2.0+): Gradient boosting machine learning model.
- **Matplotlib** (v3.7+) & **Seaborn** (v0.12+): Data visualization and plotting.

### 🔹 *Research Papers and Publications*

A detailed **literature review** of the research papers, articles, and publications that informed and shaped this project has been compiled separately.
This review discusses existing approaches, methodologies in financial sentiment analysis, use of transformer models like FinBERT, and prior work correlating news sentiment with stock market trends.

You can access the full document here:
👉 **[Link to Literature Review Document]**

### 🔹 *Code Repositories*

The complete code developed and implemented for this project has been **pushed to a dedicated GitHub repository** to ensure transparency, reproducibility, and ease of access for future reference or collaboration.
This repository contains data preprocessing scripts, exploratory data analysis notebooks, sentiment analysis pipelines, machine learning models, and visualization scripts.

You can access the full repository here:
👉 **[Link to GitHub Repository]**

### 🔹 Acknowledgments

I would like to extend my heartfelt gratitude to **Mr. Sayan Mahajan** , whose mentorship and guidance were invaluable throughout the course of this project.
Special thanks to my **colleagues and fellow interns** for their collaborative spirit, insightful discussions, and constant support, which significantly enriched the quality and depth of this work.

Their collective contributions played a crucial role in successfully executing this project during my internship at **Scienaptic Systems Pvt Ltd.**