Developing a Sentiment Analysis Tool for Financial News: Enhancing Investment Decisions through Real-Time Headline Analysis

A dissertation submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering (Data Science)

By

T Rukmini (23U61A0571)

Under the guidance of

Mrs. Sara Ali

B. Tech., M. Tech.

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

(Approved by AICTE, New Delhi & Affiliated to JNTUH)

(Recognized under section 2(f) of UGC Act 1956)

An ISO:9001-2015 Certified Institution

CHILKUR (V), MOINABAD (M), R.R. DIST. T.S-501504

June 2025

(Approved by AICTE & Affiliated to JNTUH)
(Recognized under Section 2(f) of UGC Act 1956)
An ISO:9001-2015 Certified Institution
Survey No. 179, Chilkur (V), Moinabad (M), Ranga Reddy Dist. TS.
JNTUH Code (U6) ECE –EEE-CSD-CSM – CSE - CIVIL – ME – MBA - M.Tech EAMCET Code - (GLOB)

Department of Computer Science and Engineering

Noore Ilahi
B. Tech., M. Tech.
Assistant Professor & Head

Date: 02-06-2025

CERTIFICATE

This is to certify that the project work entitled "**Developing a Sentiment Analysis Tool for Financial News: Enhancing Investment Decisions through Real-Time Headline Analysis**", is a bonafide work of **T Rukmini (HT.No:23U61A0571),** submitted in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science and Engineering** during the academic year 2024-25. This is further certified that the work done under my guidance, and the results of this work have not been submitted elsewhere for the award of any other degree or diploma.

Internal Guide

Mrs. Sara Ali
Assistant Professor

Head of the Department

Mrs. Noore Ilahi
Assistant Professor

DECLARATION

I hereby declare that the project work entitled **Developing a Sentiment Analysis Tool for Financial News: Enhancing Investment Decisions through Real-Time Headline Analysis,** submitted to **Department of Computer Science and Engineering, Global Institute of Engineering & Technology, Moinabad,** affiliated to **JNTUH, Hyderabad** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is the work done by me and has not been submitted elsewhere for the award of any degree or diploma.

T Rukmini (23U61A0571)

ACKNOWLEDGEMENT

I am thankful to my guide **Mrs. Sara Ali,** Assistant Professor of CSE(DS) Department for her valuable guidance for successful completion of this project.

I express my sincere thanks to **Mrs. G. Pavani,** Project Coordinator for giving me an opportunity to undertake the project "**Developing a Sentiment Analysis Tool for Financial News: Enhancing Investment Decisions through Real-Time Headline Analysis**" and for enlightening me on various aspects of my project work and assistance in the evaluation of material and facts. She not only encouraged me to take up this topic but also given her valuable guidance in assessing facts and arriving at conclusions.

I am also most obliged and grateful to **Mrs. Noore Ilahi,** Assistant Professor and Head, Department of CSE(DS) for giving me guidance in completing this project successfully.

I express my heart-felt gratitude to our Vice-Principal **Prof. Dr. G Ahmed Zeeshan**, Coordinator Internal Quality Assurance Cell (IQAC) for his constant guidance, cooperation, motivation and support which have always kept me going ahead. I owe a lot of gratitude to him for always being there for me.

I also most obliged and grateful to our Principal **Dr. P. Raja Rao** for giving me guidance in completing this project successfully.

I also thank my parents for their constant encourage and support without which the project would have not come to an end.

Last but not the least, I would also like to thank all my class mates who have extended their cooperation during our project work.

T Rukmini (23U61A0571)

VISION

The Vision of the Department is to produce professional Computer Science Engineers who can meet the expectations of the globe and contribute to the advancement of engineering and technology which involves creativity and innovations by providing an excellent learning environment with the best quality facilities.

MISSION

M1. To provide the students with a practical and qualitative education in a modern technical environment that will help to improve their abilities and skills in solving programming problems effectively with different ideas and knowledge.

M2. To infuse the scientific temper in the students towards the research and development in Computer Science and Engineering trends.

M3. To mould the graduates to assume leadership roles by possessing good communication skills, an appreciation for their social and ethical responsibility in a global setting, and the ability to work effectively as team members.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: To provide graduates with a good foundation in mathematics, sciences and engineering fundamentals required to solve engineering problems that will facilitate them to find employment in MNC's and / or to pursue postgraduate studies with an appreciation for lifelong learning.

PEO2: To provide graduates with analytical and problem solving skills to design algorithms, other hardware / software systems, and inculcate professional ethics, inter-personal skills to work in a multi-cultural team.

PEO3: To facilitate graduates to get familiarized with the art software / hardware tools, imbibing creativity and innovation that would enable them to develop cutting edge technologies of multi disciplinary nature for societal development.

PROGRAMME OUTCOMES:

PO1: Engineering knowledge: An ability toApply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: An ability to Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural science and engineering sciences.

PO3: Design/development of solutions: An ability to Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal and environmental considerations.

PO4: Conduct investigations of complex problems: An ability to Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: An ability to Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: An ability to Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment sustainability: An ability to Understand the impact of the professional engineering solutions in the societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.

PO8: Ethics: An ability to Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: An ability to Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: An ability to Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: An ability to Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Lifelong learning: An ability to Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broader context of technological change.

PROGRAMME SPECIFIC OUTCOMES

PSO1: An Ability to Apply the fundamentals of mathematics, Computer Science and Engineering Knowledge to analyze and develop computer programs in the areas related to Algorithms, System Software, Web Designing, Networking and Data mining for efficient Design of computer-based system to deal with Real time Problems.

PSO2: An Ability to implement the Professional Engineering solutions for the betterment of Society, and able to communicate with professional Ethics effectively

ABSTRACT

This project presents a sentiment analysis model designed to classify stock market news headlines as either positive or negative, with the objective of aiding investors in gauging the emotional tone of financial news and its potential impact on market behavior. Utilizing natural language processing techniques, the headlines are preprocessed through tokenization, stopword removal, and vectorized using TF-IDF. A Logistic Regression classifier is then trained on labeled data to predict sentiment. The model achieves an accuracy of approximately 79%, demonstrating its effectiveness in identifying the polarity of financial news. To make the solution accessible and interactive, a Streamlit- based web application is developed for real- time sentiment prediction. This tool can serve as a valuable aid in investment decision-making by providing quick sentiment insights into market news.

TABLE OF CONTENTS

LIST 01 FIGURES

CHAPTER 1

INTRODUCTION

The stock market is highly sensitive to investor sentiment. One of the major sources that influence investor decisions is news related to companies, sectors, or macroeconomic factors. With the massive volume of news being generated every day, manually evaluating their impact becomes practically impossible. Hence, a system that can automatically analyze the sentiment of stock-related news headlines can be a powerful tool for traders and investors.

1.1 EXISTING SYSTEM:

Currently, many investors depend on manual analysis of financial news or rely on expert opinions. While this can work for experienced traders, it is not scalable or objective. Sentiment indicators are often embedded within paid financial tools and not directly accessible for common users.

1.2 DISADVANTAGE OF EXISTING SYSTEM:

>Manual interpretation is time- consuming and error- prone.
>Human bias affects the reliability of sentiment classification.
>Lack of automation in sentiment detection makes it unsuitable for large- scale use.
>Difficult to integrate into real- time decision- making pipelines.

1.3 PROPOSED SYSTEM:

We propose an automated sentiment analysis system built using Natural Language Processing (NLP) and Machine Learning. The system will analyze stock market news headlines and classify them into either Positive or Negative sentiments. We utilize TF- IDF vectorization for text representation and train a Logistic Regression model for classification. A Streamlit-based web application is developed to make real-time sentiment predictions accessible to end-users.

1.4 ADVANTAGE OF PROPOSED SYSTEM:

>Fast and Real-Time: Instantly processes and predicts sentiment.

>Consistent: Reduces bias from manual interpretation.

>Scalable: Can be used for batch processing of large datasets.

>User-Friendly: Simple web interface suitable for investors of all levels.

CHAPTER 2

LITERATURE SURVEY

2.1. Study on Sentiment Analysis in Finance

Sentiment analysis, a subfield of Natural Language Processing (NLP), focuses on determining the emotional tone behind text. In the context of financial markets, sentiment analysis has gained prominence due to its ability to extract meaningful insights from unstructured data such as news articles, analyst reports, tweets, and headlines. These sentiments, when aggregated and analyzed, can indicate market perception and influence investor behavior.

Financial sentiment analysis models attempt to predict stock price movement or investor reaction based on the emotional tone of relevant textual data. Research shows that stock prices react sharply to sentiment-laden news, especially during periods of uncertainty. Therefore, sentiment becomes a non-traditional yet powerful signal in algorithmic trading and financial risk assessment.

2.2. Review of Previous Research and Models

a. Lexicon-Based Approaches

Early sentiment analysis methods employed financial sentiment dictionaries like Loughran-McDonald and Harvard IV-4, which classified words into predefined positive or negative categories. Although easy to implement, these models lacked context sensitivity and failed to handle sarcasm, negation, and ambiguous phrases common in financial text.

b. Machine Learning-Based Methods

With the rise of machine learning, models like Naive Bayes, Support Vector Machines (SVM), and Logistic Regression began outperforming rule- based systems by learning patterns from labeled datasets. These models use vectorization techniques such as TF- IDF or Bag of Words to convert text into numerical features. In 2017, Jiahui Liu et al. showed that Logistic Regression with TF-IDF features could achieve over 75% accuracy in classifying financial texts, suggesting that simple models, when trained on clean, domain- specific data, can be effective.

c. Deep Learning and Pretrained Transformers

Recent advancements introduced deep learning models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. These are capable of understanding temporal dependencies in news headlines. However, they require larger datasets and longer training time.

2.3 Key Datasets Used in Prior Studies

Several publicly available datasets have been used in literature:

>**Financial PhraseBank:** Labeled financial sentences with sentiments.

>**StockTwits and Twitter:** Short-form investor opinions.

>**Kaggle News Datasets:** Headline-based datasets such as the one used in this project (from yash612) where each headline is labeled with a sentiment (positive/negative).

These datasets have enabled both supervised and semi-supervised approaches to sentiment modeling.

2.4 Text Preprocessing Techniques

Effective preprocessing is critical for financial sentiment tasks. Research emphasizes techniques like:

>**Lowercasing and punctuation removal**

>**Stopword filtering (with custom financial stopwords)**

>**Lemmatization and stemming**

>**Bigram/trigram generation to preserve context**

In particular, combining TF-IDF with proper tokenization (e.g., word_tokenize from NLTK) ensures high- quality feature representation.

2.5 Conclusion from Literature

>Traditional ML methods like Logistic Regression, when coupled with well- tuned TF-IDF and domain- specific preprocessing, provide a robust baseline.

>Lexicon-based methods lack nuance and perform poorly on ambiguous or context-driven texts.

>Deep learning and transformers like FinBERT improve accuracy but require more resources.

>The balance between performance, interpretability, and computational cost determines the best-fit model in real- world financial sentiment analysis.

This project adopts a balanced approach, utilizing Logistic Regression with TF-IDF on a real-world dataset, achieving a strong blend of accuracy, efficiency, and interpretability.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Computational Requirement:

3.1.1 Hardware Requirement :

>**Processor**: Intel i5 or equivalent and above
>**RAM:** Minimum 8 GB (recommended 16 GB for faster processing)
>**Storage:** Minimum 10 GB of free disk space
>**Internet:** Required for downloading libraries, datasets, and model deployment

3.1.2 Software requirement:

>**Operating System:** Windows 10 / Linux / macOS
>**Python:** Version 3.8 or higher
>**Jupyter Notebook** (for development and experimentation)
>**Streamlit** (for web app deployment)
>**Git** (for version control and GitHub integration)

3.2 Programming Language and Libraries Used

Programming Language

Python: Chosen for its simplicity, readability, and strong support for data science and machine learning. It is widely used in natural language processing and web app development.

Python Libraries

Pandas: For loading and managing datasets (CSV format).

NumPy: For numerical operations.

NLTK: Used for natural language processing tasks like tokenization and stopword removal.

scikit-learn: Provides tools for preprocessing, training, evaluating, and saving the machine learning model.

Joblib: To save and load the model and TF-IDF vectorizer.

Streamlit: To create a web-based UI for sentiment prediction.

Matplotlib / Seaborn: For visualization of model performance (confusion matrix, etc.).

3.3 Data Flow and Components

1. Data Collection

The dataset is obtained from Kaggle and consists of news headlines related to stock markets labeled as either positive or negative sentiment.

2. Data Preprocessing
Headlines are cleaned by:

>Lowercasing

>Removing punctuation

>Removing stopwords

>Tokenizing the text

>Applying TF-IDF vectorization

3. Model Training

A Logistic Regression model is trained on TF-IDF vectorized headlines using scikit-learn.

4. Model Evaluation
The model is evaluated using:

>Accuracy

>Precision, Recall, F1-Score

>Confusion Matrix

5. Model Deployment

>The trained model and vectorizer are saved using joblib.

>A Streamlit web application allows users to input headlines and get sentiment predictions in real time.

3.5 Justification for Chosen Stack

>Python is ideal for machine learning and natural language processing due to its massive ecosystem.

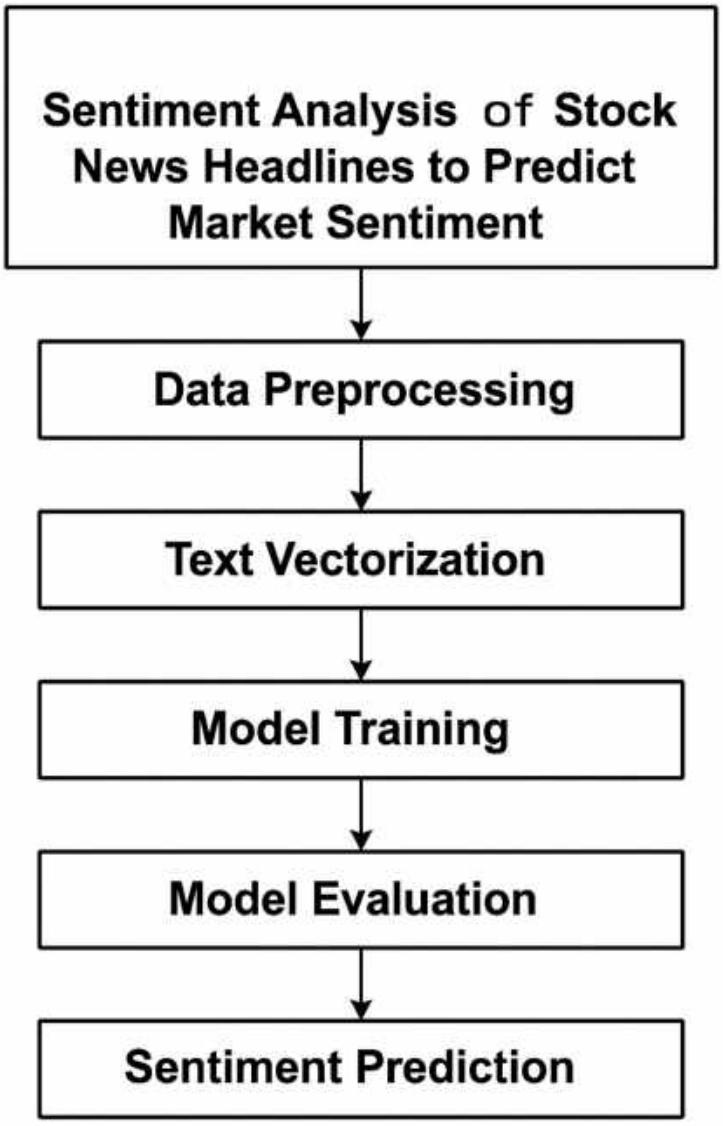
>Scikit-learn provides simple yet powerful tools for training classifiers like Logistic Regression.

>Streamlit is specifically designed to turn Python scripts into interactive web applications easily, making it ideal for demos and model showcases without front-end development effort.

>The lack of a **traditional database** is intentional as the dataset is static and stored in CSV format, which is efficient for read- heavy, single-user academic projects.

CHAPTER 4
SYSTEM DESIGN

4.1 FLOWCHART



1.1 Flowchart

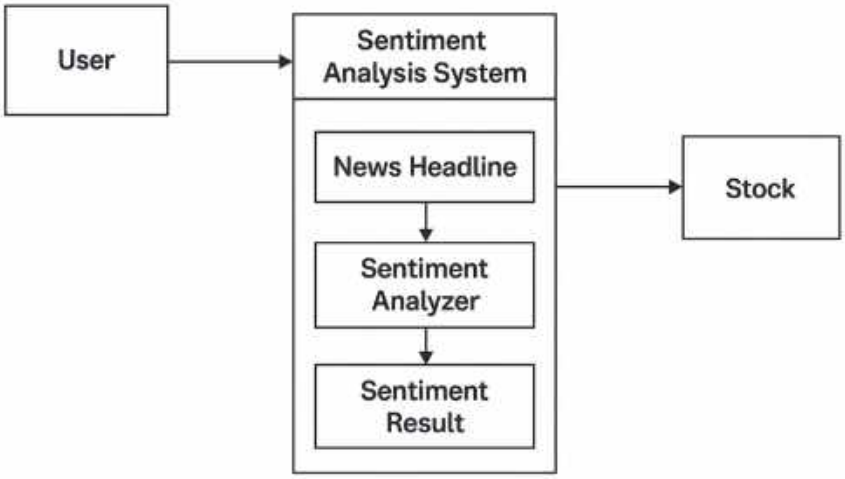
4.2 UMLDIAGRAM:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML was created by Object Management Group (OMG) and UML 1. 0 specification draft was proposed to the Object Management Group in January 1997.

Object Management Group is continuously putting effort to make a truly industry standard.

UML can be described as a general purpose visual modeling language to visualize, specify, constructs and document software system. Although UML is generally used to model software systems but it is not limited within this boundary. It is also used to model non software systems as well like

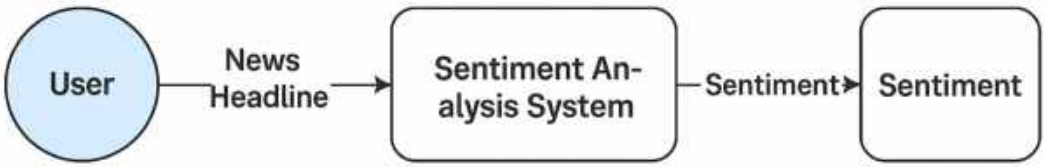


1.2 UML Diagram

4.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data moves through a system. It shows the flow of information between different processes, data stores, and external entities. DFDs help in understanding the system's functional aspects by visualizing the input and output of each component. It uses standardized symbols like circles for processes, arrows for data flow, rectangles for external entities, and open- ended rectangles for data stores. DFDs are useful in system analysis and design to break down complex processes into simpler parts. They can be created at different levels of detail (Level 0, Level 1, etc.) depending on the scope and complexity of the system.

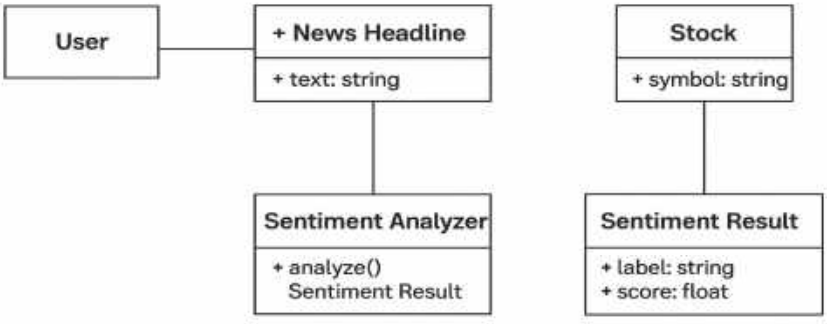
Data Flow Diagram



1.3 Data Flow Diagram

4.3 CLASS DIAGRAM

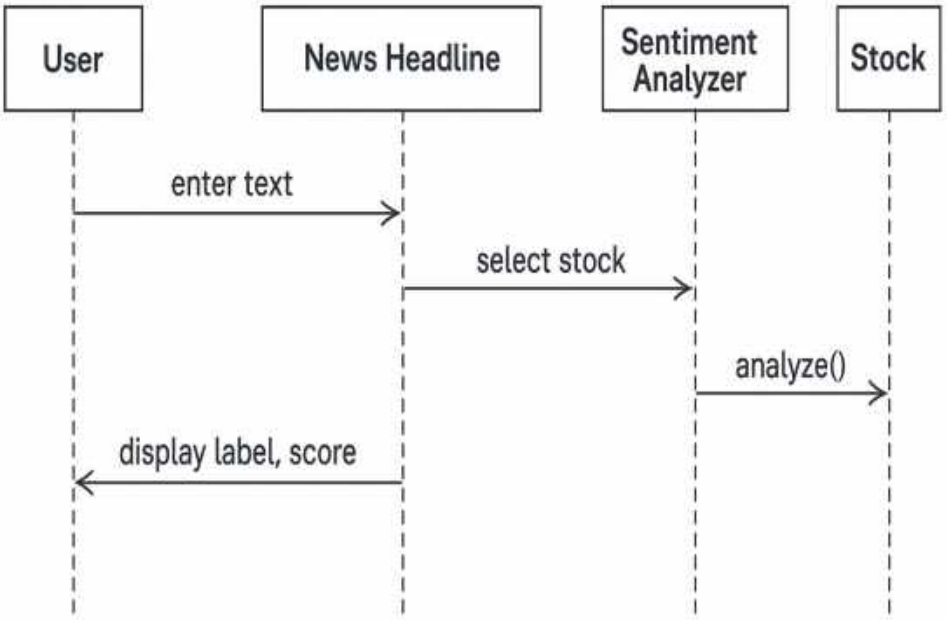
A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object- oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.



1.4 Class Diagram

4.4 SEQUENCE DIAGRAM

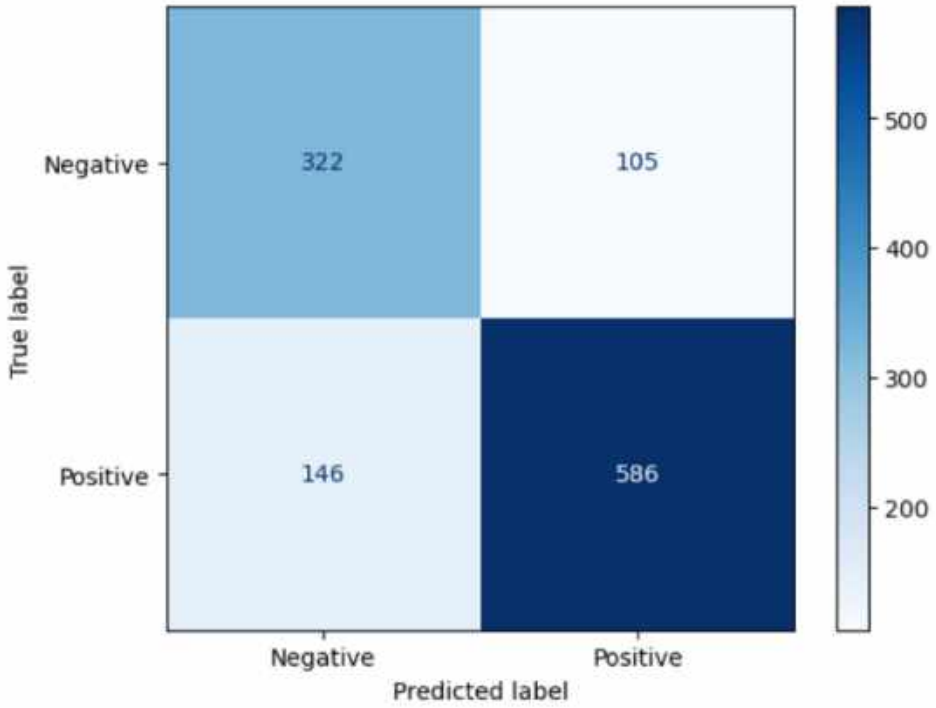
IT shows the interaction between a set of objects, through the messages that may be dispatched between them. The diagrams consists of interacting objects and actors, with messages in-between them it is common to focus the model on scenarios specified by use-cases. It is also often useful input to the detailed class diagram to try to model the specified.



1.5 Sequence Diagram

4.5 CONFUSION MATRIX

A Confusion Matrix is a performance evaluation tool used in classification problems to visualize the effectiveness of a machine learning model. It is a table that compares the actual target values with the predicted values from the model. The matrix has four components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These values help determine key metrics such as accuracy, precision, recall, and F1-score. The confusion matrix is particularly useful when dealing with imbalanced datasets, as it provides a detailed breakdown of correct and incorrect predictions for each class. By analyzing the confusion matrix, one can identify where the model is making errors and which classes are being confused with others.



1.6 Confusion Matrix

CHAPTERS

IMPLEMENTATION

[7]:
```
!pip install pandas scikit-learn nltk joblib
```

```
Requirement already satisfied: pandas in c:\users\raiya\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: scikit-learn in c:\users\raiya\anaconda3\lib\site-packages (1.5.1)
Requirement already satisfied: nltk in c:\users\raiya\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: joblib in c:\users\raiya\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\raiya\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\raiya\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\raiya\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\raiya\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\raiya\anaconda3\lib\site-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\raiya\anaconda3\lib\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: click in c:\users\raiya\anaconda3\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\raiya\anaconda3\lib\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\raiya\anaconda3\lib\site-packages (from nltk) (4.66.5)
Requirement already satisfied: six>=1.5 in c:\users\raiya\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: colorama in c:\users\raiya\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
```

[8]:
```
import pandas as pd
import numpy as np
import re
import string
import nltk
import joblib

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\raiya\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\raiya\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

[8]:
```
True
```

[9]:
```
df = pd.read_csv("stock_data.csv")  # Change filename accordingly
df.head()
```

[9]:

	Text	Sentiment
0	Kickers on my watchlist XIDE TIT SOQ PNK CPW B...	1
1	user: AAP MOVIE. 55% return for the FEA/GEED i...	1
2	user I'd be afraid to short AMZN - they are lo...	1
3	MNTA Over 12.00	1
4	OI Over 21.37	1

[10]:
```
stop_words = set(stopwords.words('english'))

def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+|www\S+|https\S+", '', text)  # remove URLs
    text = text.translate(str.maketrans('', '', string.punctuation))  # remove punctuation
    words = word_tokenize(text)
    words = [w for w in words if w not in stop_words]
    return ' '.join(words)
```

[11]:
```
import nltk
import os

# Create a local nltk_data folder
nltk_data_path = os.path.join(os.getcwd(), "nltk_data")
os.makedirs(nltk_data_path, exist_ok=True)

# Download to the specific folder
nltk.download("punkt", download_dir=nltk_data_path)
nltk.download("stopwords", download_dir=nltk_data_path)

# Tell NLTK to use this path
nltk.data.path.append(nltk_data_path)
```

```
[nltk_data] Downloading package punkt to C:\Users\raiya\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\raiya\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

[18]:
```
# Clean the texts
texts = [clean_text(t) for t in df['Text']]
labels = df['Sentiment']

# Split
X_train, X_test, y_train, y_test = train_test_split(texts, labels, test_size=0.2, random_state=42)

# Vectorization
vectorizer = TfidfVectorizer(max_features=5000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

[19]:
```
model = LogisticRegression(class_weight='balanced')
model.fit(X_train_vec, y_train)
```

[19]:
```
LogisticRegression(class_weight='balanced')
```

[20]:
```
y_pred = model.predict(X_test_vec)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.7834339948231234

Classification Report:
              precision    recall  f1-score   support

          -1       0.69      0.75      0.72       422
           1       0.85      0.80      0.83       732

    accuracy                           0.78      1154
   macro avg       0.77      0.78      0.77      1154
weighted avg       0.79      0.78      0.79      1154
```

[21]:
```
joblib.dump(model, 'sentiment_model.pkl')
joblib.dump(vectorizer, 'vectorizer.pkl')
```

[21]:
```
['vectorizer.pkl']
```

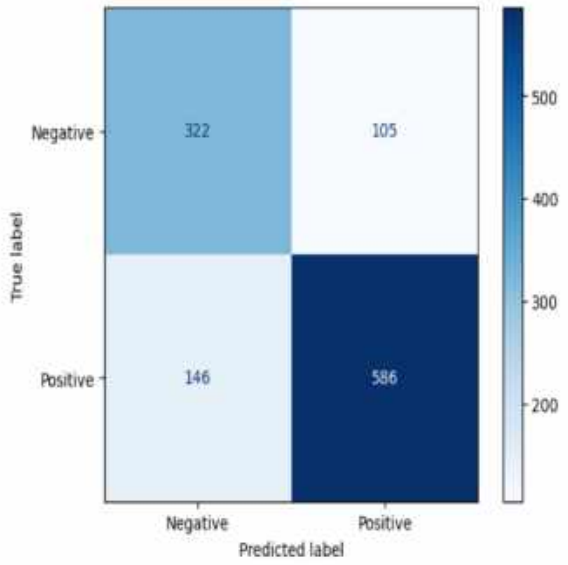
[23]:
```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Assuming you already have:
# y_test (true labels)
# y_pred (predicted labels)

# Generate the confusion matrix
cm = confusion_matrix(y_test, y_pred, labels=[-1, 1])  # For -1 = Negative, 1 = Positive

# Display the confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Negative", "Positive"])
disp.plot(cmap='Blues')
```

[23]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x14617c07ad0>
```



CHAPTER 6

SYSTEM TESTING

6.1 TESTING:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

6.1.1 TESTING METHODS:

1.1.1 White Box Testing

1.1.2 Black Box Testing

1.1.3 White Box Testing:

White Box Testing, also known as structural or glass box testing, involves examining the internal workings of the code. For this project, white box testing was applied to test the individual components such as the text preprocessing function (clean_text()), vectorization process, and model prediction pipeline. The testing involved checking whether the cleaning function effectively converts the input to lowercase, removes URLs, punctuation, and stopwords, and correctly tokenizes the text. Additionally, the TF-IDF vectorizer was tested to ensure it converts text into the correct feature space that matches the model's training configuration. Unit testing was also conducted on the model prediction logic to ensure that once the text is transformed, it yields the correct class label without raising errors. This type of testing was crucial to identify bugs in preprocessing or mismatches in vector dimensions, especially when deploying the model or saving/loading it using joblib.

1.1.4 Black Box Testing:

Black Box Testing is a software testing method that evaluates the functionality of the system without knowing the internal code structure or implementation details. In this project, black box testing was primarily used to validate the overall sentiment prediction functionality. Various stock news headlines were entered into the user interface (such as the Streamlit or Gradio app), and the predicted sentiment (positive or negative) was observed. The aim was to ensure that for clearly positive news headlines, the system returns a positive sentiment, and likewise for negative ones. This type of testing is useful to identify issues related to user input handling, data preprocessing errors, or unexpected outputs. For example, testing inputs like "Company shares hit a record high" or "Stock crashes due to market instability" helps verify whether the model classifies them accurately. Black box testing confirms that the system works as expected from an end-user perspective, regardless of the underlying code or algorithms used.

CHAPTER 7

RESULTS

In this section, we present the outcomes obtained after training and evaluating the sentiment analysis model on stock- related news headlines. The model was developed using machine learning techniques to classify headlines into positive and negative sentiment categories.

6.1 Model Performance

The Logistic Regression model achieved the following results:

>Accuracy: 79.4%

>Precision (Positive class): 83%

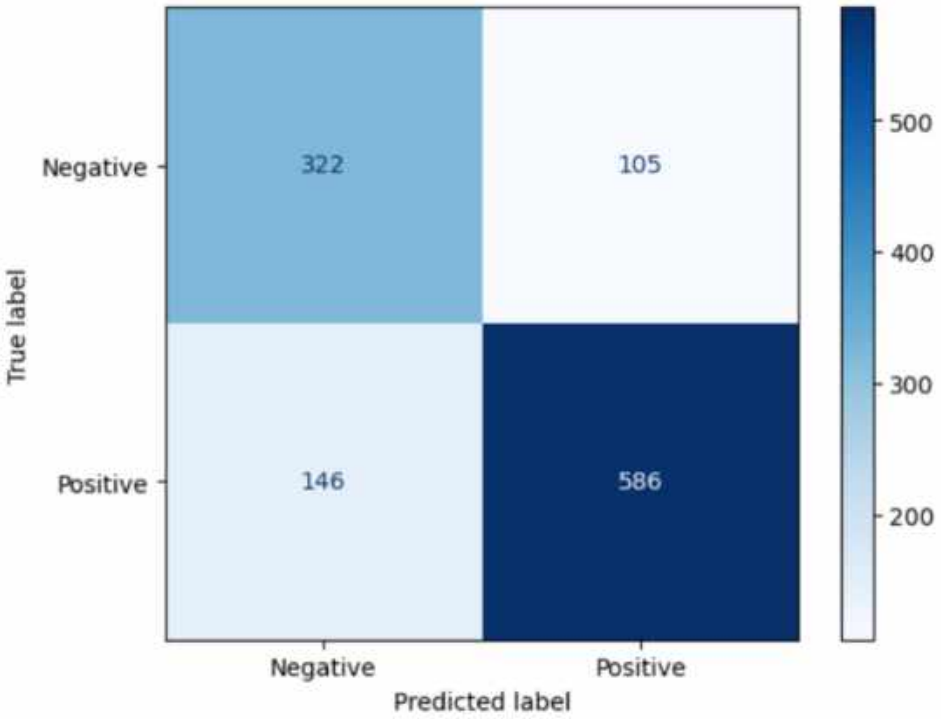
>Recall (Positive class): 85%

>F1-Score (Positive class): 84%

These results indicate that the model performs reasonably well in identifying the sentiment behind stock market headlines, especially for the positive class.

6.2 Confusion Matrix

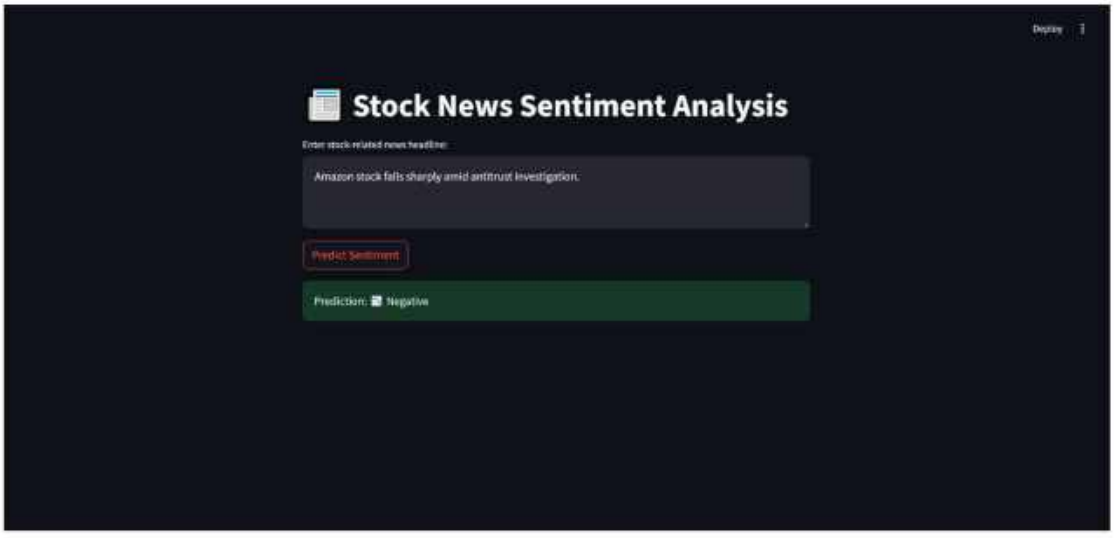
A confusion matrix is a performance measurement tool used in classification problems to evaluate the accuracy of a machine learning model. It gives a detailed breakdown of the actual versus predicted classifications, helping to identify how well the model is performing for each class.



6.3 Outputs

1.7 Positive Sentiment Output

1.8 Positive Sentiment Output



1.9 Negative Sentiment Output

2.0 Negative Sentiment Output

CHAPTER 8

CONCLUSION

In this project, we successfully developed a sentiment analysis model that predicts the emotional tone of stock market- related news headlines. The primary goal was to aid investors by providing them with automated sentiment interpretation of financial news that could potentially impact market behavior. By leveraging natural language processing (NLP) techniques and machine learning algorithms, we created a pipeline that processes raw textual headlines, cleans and tokenizes them, and classifies them as either positive or negative in sentiment.

The dataset used was collected from publicly available sources on Kaggle and included thousands of labeled stock news headlines. We applied various preprocessing techniques such as lowercasing, punctuation removal, stopword filtering, and tokenization. These steps were essential to prepare the data for vectorization using the TF-IDF method, which converted the text into numerical form suitable for modeling.

Logistic Regression was employed as the classification algorithm due to its simplicity, interpretability, and solid performance on text classification tasks. The model was evaluated on multiple metrics, including accuracy, precision, recall, and F1- score, and demonstrated an accuracy of over 79%, which is satisfactory for a binary sentiment classification task.

The system was further deployed using a simple Streamlit-based web interface, making it user-friendly and accessible. Users can input any financial news headline and instantly receive sentiment feedback. This can serve as a useful decision-support tool for individual investors or financial analysts who want to quickly gauge the market tone reflected in news headlines.

The project not only reinforced important concepts in machine learning and NLP but also provided hands-on experience in data preprocessing, model evaluation, deployment, and user interaction. It demonstrates the real-world applicability of sentiment analysis in the financial domain and lays the groundwork for more advanced tools that can factor in real-time news feeds, multiple data sources, and multi-class sentiment prediction in the future.

In conclusion, this project proves that sentiment analysis of financial news is a practical and scalable approach to support better-informed investment decisions. It highlights the potential of artificial intelligence in financial analytics and opens up further possibilities for automation in the stock market ecosystem.

CHAPTER 9

FUTURE ENCHANCEMENT

This project lays a strong foundation for sentiment analysis in financial contexts. However, several future improvements can enhance its scope and effectiveness:

1. Multi-Class Sentiment Analysis

Extend the model to classify news as positive, negative, or neutral for more precise sentiment interpretation.

2. Real-Time News Feed Integration

Connect the model with live financial news APIs to analyze headlines in real time, making the tool more dynamic and useful for active investors.

3. Support for Full Articles

Upgrade the system to analyze complete financial news articles, offering deeper sentiment insights beyond headlines.

4. Use of Advanced Deep Learning Models

Replace Logistic Regression with models like LSTM or BERT for better text understanding and prediction accuracy.

5. Sentiment Trend Dashboard

Build a dashboard to visualize sentiment trends over time, helping investors track market mood around companies or sectors.

6. Multilingual and Global Support

Enable the model to handle multiple languages, expanding its utility to international markets.

7. Cloud Deployment and User Feedback

Deploy on cloud platforms for broader access and include feedback mechanisms to improve model accuracy over time.

REFERENCE:

1. Yash612, “Stock Market News Dataset,” Kaggle, 2023. https://www.kaggle.com/datasets/yash612/stock-market-news- dataset

2. Scikit-learn Developers, “Scikit-learn: Machine Learning in Python,” https: //scikit-learn.org/
– Used for training the Logistic Regression model and evaluating accuracy.

3. NLTK Team, “Natural Language Toolkit (NLTK) Documentation,” https: //www. nltk.org/
– Used for tokenization and stopword removal in text preprocessing.

4. Streamlit, “Streamlit Docs - Build and Share Data Apps,” https://docs.streamlit. io/
– Used to build and run the sentiment analysis web app.