Stock Market News Sentiment Analysis and Summarization

Project 6 and Introduction to Natural Language Processing

Date 07.02.2025

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- EDA Results
- Data Preprocessing
- Model Performance Summary
- Appendix

Executive Summary

- Actionable insights & recommendations

Insights:

- **Dependency Conflicts:** The core issue is that several installed packages have version requirements for numpy that are incompatible with the current numpy version (2.2.2). This highlights the importance of careful dependency management.
- **CUDA Compatibility:** The error message also suggests potential conflicts with CUDA-related libraries, which are crucial for GPU acceleration in many machine learning frameworks like PyTorch.

Executive Summary

- Actionable insights & recommendations

Recommendations:

- **Explore Advanced Models:** Experiment with more advanced models like BERT, XLNet, or GPT for improved sentiment analysis and summarization.
- **Investigate Ensemble Methods:** Combining predictions from multiple models (e.g., using an ensemble approach) can often improve overall accuracy and robustness.
- **Implement Dynamic Model Updates:** Regularly update the model with new data and retrain it to adapt to changing market conditions and news trends.
- **Conduct Thorough Backtesting:** Backtest the model on historical data to evaluate its performance and assess its ability to predict future stock price movements.
- **Consider External Data Sources:** Integrate additional data sources, such as social media sentiment, economic indicators, and analyst ratings, to further enhance the model's predictive power.
- **Address Ethical Considerations:** Ensure responsible and ethical use of AI in financial markets, considering factors like bias, fairness, and transparency.

Business Problem Overview and Solution Approach

Problem Definition:

With the ever-growing influx of news articles and opinions influencing financial markets, investors face significant challenges in interpreting stock-related news and assessing its impact on stock prices. An investment startup aims to leverage artificial intelligence to streamline this process by developing an AI-driven sentiment analysis system that can automatically process and analyze financial news.

The startup has collected historical daily news data for a specific company listed on NASDAQ, alongside corresponding stock prices and trade volumes. The goal is to develop a system that extracts meaningful insights from these datasets, enabling more accurate market sentiment analysis and stock price predictions.

Business Problem Overview and Solution Approach

Solution Approach / Methodology

The development of the AI-driven sentiment analysis system will follow a structured data science and machine learning pipeline:

1. Data Collection & Preprocessing

- **News Data:** Gather historical financial news articles related to the target company.
- **Market Data:** Obtain corresponding daily stock prices (open, high, low, close) and trade volumes.
- **Data Cleaning:** Remove duplicate news, handle missing values, and normalize text data (lowercasing, stopword removal, stemming/lemmatization).
- **Data Alignment:** Synchronize news data with stock prices and trading volumes on a daily basis.

Business Problem Overview and Solution Approach

Great Learning
POWER AHEAD

2. Sentiment Analysis Model Development

- **Text Vectorization:** Convert news articles into numerical representations using techniques such as TF-IDF, Word2Vec, or transformer-based embeddings (BERT, FinBERT).
- **Sentiment Labeling:**
  - Use pre-trained sentiment analysis models trained on financial text (e.g., FinBERT).
  - Apply rule-based sentiment scoring using dictionaries like Loughran-McDonald sentiment word lists.
  - If labeled data is available, train a supervised sentiment classification model.
- **Stock Price Impact Analysis:**
  - Use statistical techniques (e.g., correlation analysis, Granger causality) to determine relationships between sentiment scores and stock price movements.
  - Categorize sentiment polarity (positive, neutral, negative) and measure its correlation with stock returns and trading volumes.

Business Problem Overview and Solution Approach

3. Weekly News Summarization

- **Topic Modeling:** Use NLP techniques like Latent Dirichlet Allocation (LDA) or BERTopic to identify key themes in financial news.
- **Text Summarization:** Implement extractive (TextRank) or abstractive (T5, BART) summarization methods to generate concise weekly summaries of financial news.

4. Predictive Modeling & Evaluation

- **Feature Engineering:** Generate predictive features based on sentiment scores, news volume, and historical stock data.
- **Model Selection:** Train machine learning models (e.g., Random Forest, XGBoost, LSTM, Transformer-based models) to predict stock movements based on sentiment trends.
- **Evaluation Metrics:** Use performance metrics like RMSE, R^2, and classification accuracy (for up/down movement predictions) to validate model effectiveness.

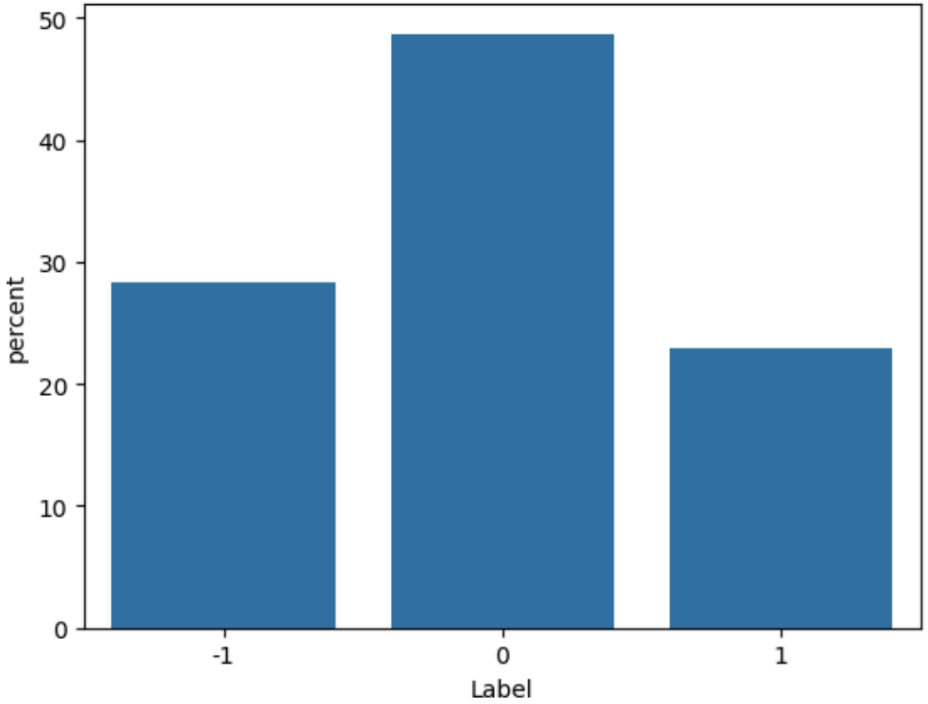
Business Problem Overview and Solution Approach

Great Learning
POWER AHEAD

5. Deployment & Integration

- **Dashboard Development:** Develop an interactive visualization dashboard to present insights to financial analysts.
- **Automated Data Pipeline:** Implement an automated system to fetch real-time news, perform sentiment analysis, and update stock market predictions.
- **Continuous Model Improvement:** Periodically retrain models with new data to adapt to market changes and improve accuracy.

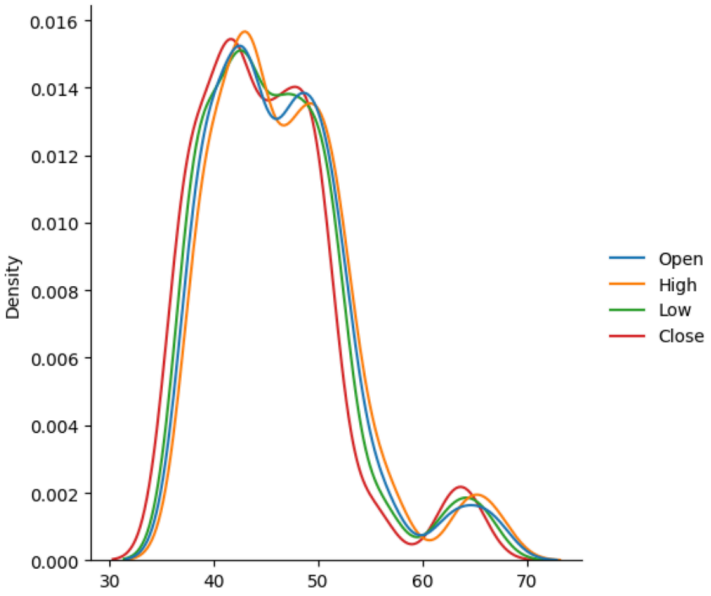
EDA Results



- Observations on Label

The bar chart shows the distribution of a categorical variable labeled "Label". The majority of observations fall into the category labeled "0", representing approximately 50% of the data. The categories labeled "-1" and "1" have roughly equal representation, each comprising around 25% of the data. This suggests an uneven distribution with a significant proportion of observations belonging to the "0" category.

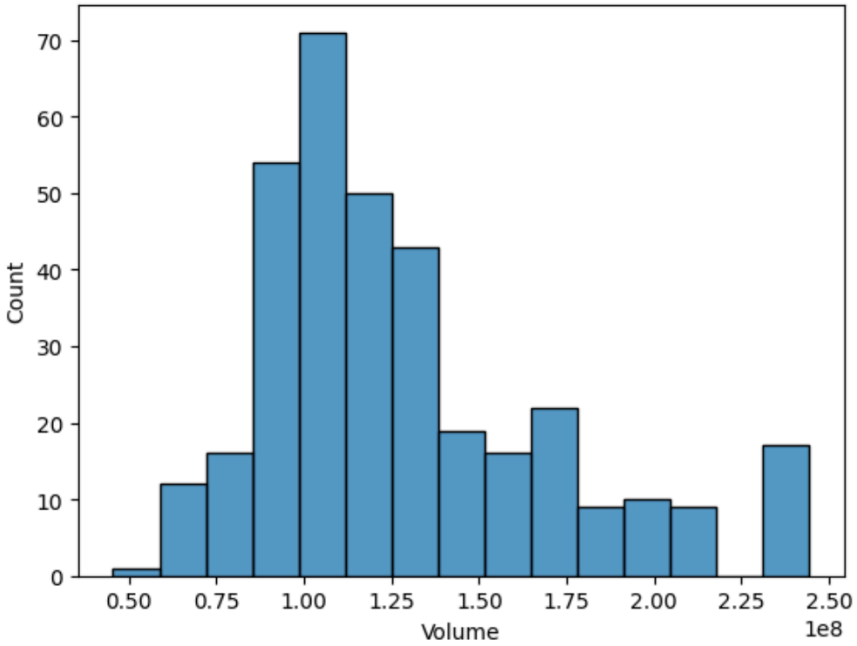
EDA Results



- Density Plot of Price (Open,High,Low,Close)

The graph displays the density distributions of four stock price variables: Open, High, Low, and Close. All four distributions exhibit a similar shape, with a peak around the 40-50 range and tapering off towards higher and lower values. The "High" and "Close" prices tend to have slightly higher values and a more pronounced peak compared to "Open" and "Low."

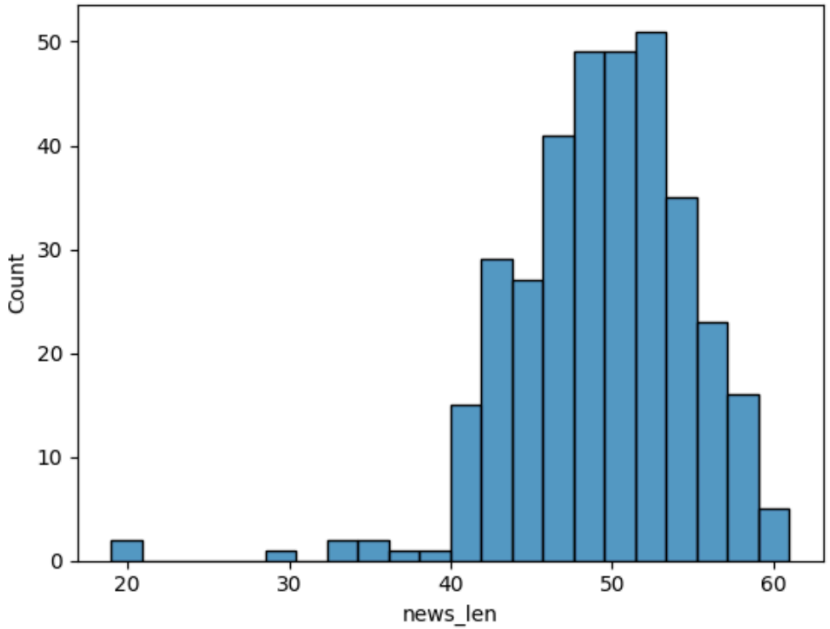
EDA Results



- Observations on Volume

The histogram displays the distribution of stock trading volumes. The data appears to be right-skewed, with a significant number of trading days having lower volumes. There's a peak around 1 billion shares traded, suggesting this is a common volume level. A few outliers with extremely high trading volumes are also observed.

EDA Results



- Observations on News length

The histogram depicts the distribution of news article lengths, represented by the "news_len" variable. The data appears to be right-skewed, with a majority of articles falling within the 40-55 word range. A smaller proportion of articles are shorter or longer, with a few outliers extending beyond 60 words.

EDA Results

	news_len
count	349.000000
mean	49.312321
std	5.727770
min	19.000000
25%	46.000000
50%	50.000000
75%	53.000000
max	61.000000

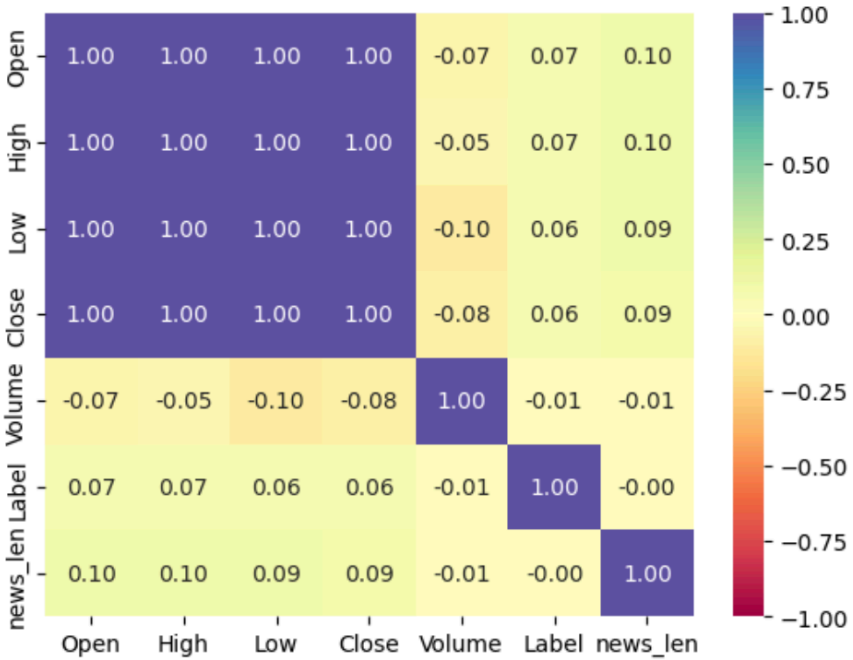
dtype: float64

- Observations on News length

The provided image shows a summary of the "news_len" column, likely from a pandas DataFrame. Here's a breakdown of the information:

- **count:** 349 - This indicates that there are 349 non-null values in the "news_len" column.
- **mean:** 49.31 - The average length of the news articles is approximately 49 words.
- **std:** 5.73 - This is the standard deviation, which measures the spread of the data around the mean. A higher standard deviation suggests that the data points are more spread out.
- **min:** 19 - The shortest news article has 19 words.
- **25%:** 46 - 25% of the news articles have a length of 46 words or less.
- **50%:** 50 - The median length of the news articles is 50 words. This means that 50% of the articles are shorter than 50 words, and 50% are longer.
- **75%:** 53 - 75% of the news articles have a length of 53 words or less.
- **max:** 61 - The longest news article has 61 words.
- **dtype: float64:** This indicates that the "news_len" column is of the float64 data type, meaning it stores numerical values with high precision.

EDA Results

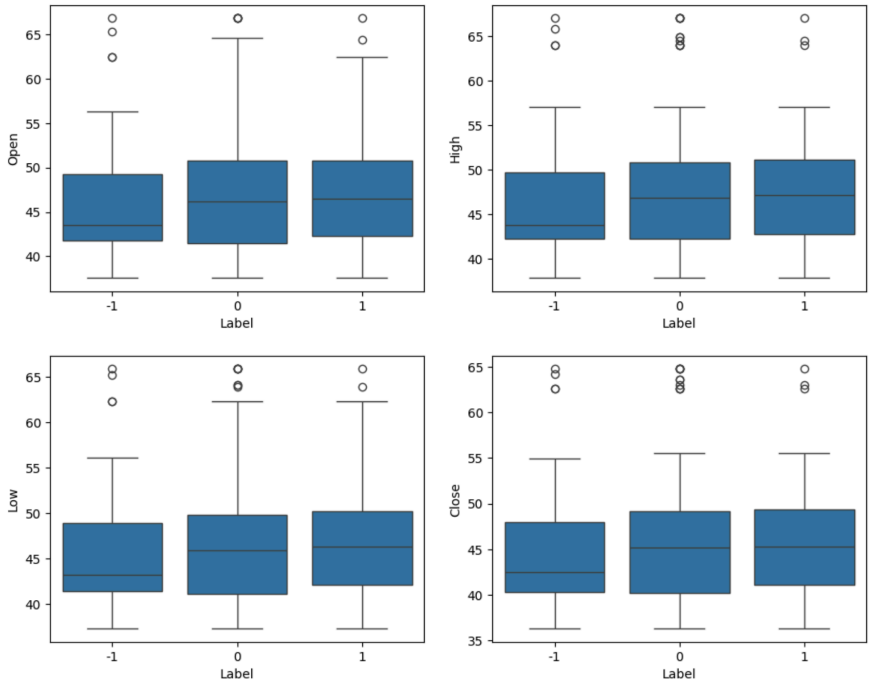


- Correlation

Key Observations:

- **Strong Positive Correlations:** There are strong positive correlations between 'Open', 'High', 'Low', and 'Close' prices, which is expected as these stock prices are closely related.
- **Moderate Correlations:** 'Volume' shows moderate positive correlations with 'Open', 'High', and 'Close', suggesting that higher trading volume generally coincides with higher prices.
- **Weak Correlations:** 'Label' and 'news_len' have relatively weak correlations with other features, indicating a less pronounced relationship.

EDA Results



- Label vs Price (Open, High, Low, Close)

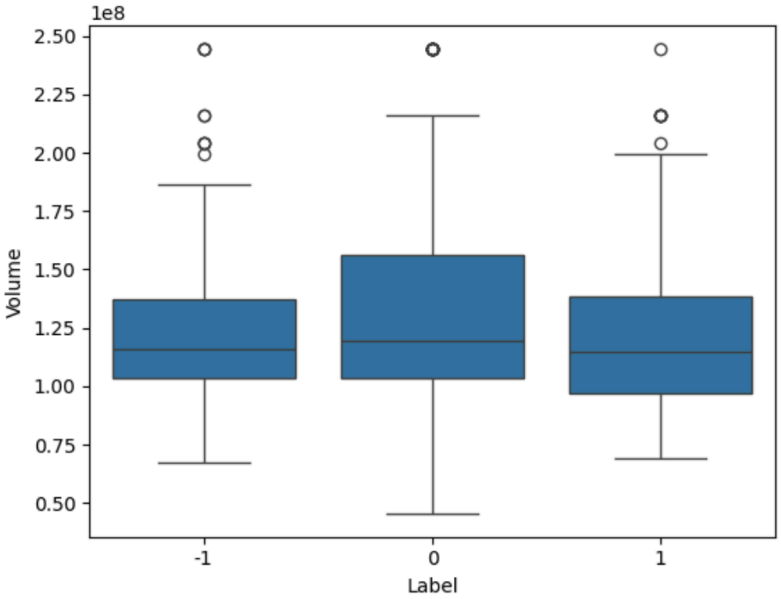
The boxplot visualizes the distribution of stock prices (Open, High, Low, Close) across different label categories (-1, 0, 1).

Observations:

- **Median Differences:** There appear to be slight differences in the median values of stock prices across the label categories.
- **Spread and Outliers:** The spread of the data (indicated by the box and whiskers) varies across categories and prices. Some categories show more variability and potential outliers.

Overall, the boxplot suggests potential relationships between stock prices and the label variable, although further analysis is needed to confirm any significant differences.

EDA Results



- Label vs Volume

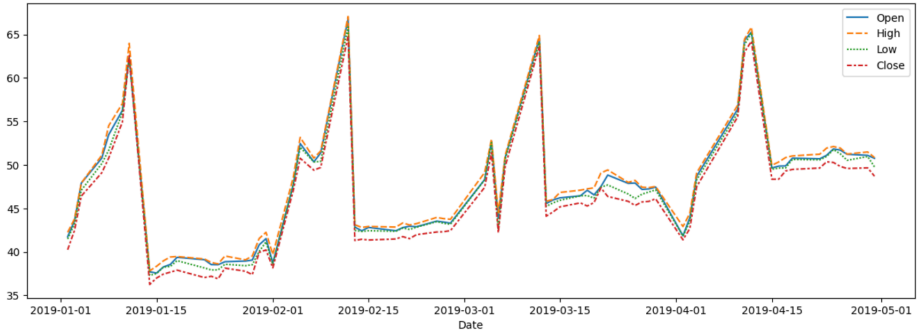
This boxplot visualizes the distribution of trading volume across different label categories (-1, 0, 1).

Observations:

- Median Differences: While the median volume seems to be relatively similar across the three categories, there are noticeable differences in the distribution.
- Spread and Outliers: The category labeled '0' appears to have a slightly wider spread and more outliers compared to the other two categories.

Overall, the boxplot suggests that trading volume may exhibit some variations across the different label categories, although the differences are not as pronounced as in the previous graph.

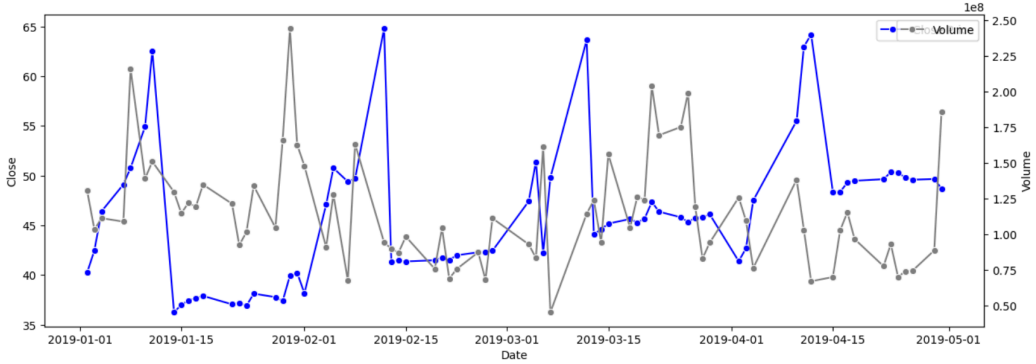
EDA Results



- Date vs Price (Open, High, Low, Close)

The line graph depicts the fluctuations of stock prices over time. The lines represent the daily "Open", "High", "Low", and "Close" prices. All prices exhibit a degree of volatility with periods of upward and downward trends. The "High" and "Close" prices generally trend higher than the "Open" and "Low" prices, as expected.

EDA Results



- Volume vs Close Price

This line graph illustrates the relationship between stock closing prices and trading volume over a period of time. The blue line represents the closing price, showing fluctuations with periods of upward and downward trends. The grey dots represent the trading volume on each day, suggesting that higher trading volumes are often associated with more significant price movements, both upward and downward.

Data Preprocessing

- Duplicate value check

	0
0	False
1	False
2	False
3	False
4	False
...	...
344	False
345	False
346	False
347	False
348	False

349 rows × 1 columns

dtype: bool

Data Preprocessing

- Data preprocessing for modeling

Summary of the 'Date' column

	Date
count	349
mean	2019-02-16 16:05:30.085959936
min	2019-01-02 00:00:00
25%	2019-01-14 00:00:00
50%	2019-02-05 00:00:00
75%	2019-03-22 00:00:00
max	2019-04-30 00:00:00

dtype: object

Data Preprocessing

- The model Data Shapes:
  - Train data: 286 samples with 10 features each.
  - Validation data: 21 samples with 10 features each.
  - Test data: 42 samples with 10 features each.
- **Label Shapes:**
  - Train labels: 286 labels (presumably corresponding to the 286 training samples).
  - Validation labels: 21 labels (corresponding to the validation samples).
  - Test labels: 42 labels (corresponding to the test samples).

Comments:

- **Dataset Sizes:** The dataset appears to be relatively small, with only 349 samples in total. This might limit the model's ability to generalize well and could potentially lead to overfitting.
- **Feature Count:** The 10 features might be sufficient or insufficient depending on the complexity of the problem. More features could potentially improve model performance, but may also increase the risk of overfitting and computational cost.
- **Data Split:** The split into train, validation, and test sets seems reasonable, with a larger proportion of data allocated for training. However, the validation set is relatively small, which could make it difficult to accurately assess model performance during hyperparameter tuning.

Data Preprocessing

Word Embeddings

Word2Vec

The Word2Vec model has a vocabulary size of 4682 words and was trained in 0.82857 seconds.

The word embeddings generated by the model have a dimensionality of 300. The training set consists of 286 samples, the validation set has 21 samples, and the test set includes 42 samples. Each sample in these sets is represented by a 300-dimensional vector.

GloVe

The GloVe model has a vocabulary size of 400000 words, and it took 26.89 seconds to train. The word embeddings generated by the model have a dimensionality of 100. The training set consists of 286 samples, the validation set has 21 samples, and the test set includes 42 samples. Each sample in these sets is represented by a 100-dimensional vector.

Data Preprocessing

Word Embeddings

Sentence Transformer

Defining the model

```
modules.json: 100%                                  349/349 [00:00<00:00, 27.3kB/s]
config_sentence_transformers.json: 100%                                  116/116 [00:00<00:00, 7.56kB/s]
README.md: 100%                                  10.7k/10.7k [00:00<00:00, 810kB/s]
sentence_bert_config.json: 100%                                  53.0/53.0 [00:00<00:00, 2.64kB/s]
config.json: 100%                                  612/612 [00:00<00:00, 32.4kB/s]
model.safetensors: 100%                                  90.9M/90.9M [00:00<00:00, 163MB/s]
tokenizer_config.json: 100%                                  350/350 [00:00<00:00, 21.4kB/s]
vocab.txt: 100%                                  232k/232k [00:00<00:00, 1.99MB/s]
tokenizer.json: 100%                                  466k/466k [00:00<00:00, 7.00MB/s]
special_tokens_map.json: 100%                                  112/112 [00:00<00:00, 9.05kB/s]
1_Pooling/config.json: 100%                                  190/190 [00:00<00:00, 16.9kB/s]
```

Data Preprocessing

Great Learning
POWER AHEAD

Word Embeddings

Sentence Transformer

The dimensionality of the sentence embeddings is 384. The training set contains 286 samples, the validation set contains 21 samples, and the test set contains 42 samples. A 384-dimensional vector represents each sample in the training, validation, and test sets.

Encoding the dataset

```
Batches: 100%                                   9/9 [00:01<00:00, 10.05it/s]
Batches: 100%                                   1/1 [00:00<00:00, 25.61it/s]
Batches: 100%                                   2/2 [00:00<00:00, 24.18it/s]
Time taken  1.1558501720428467
```

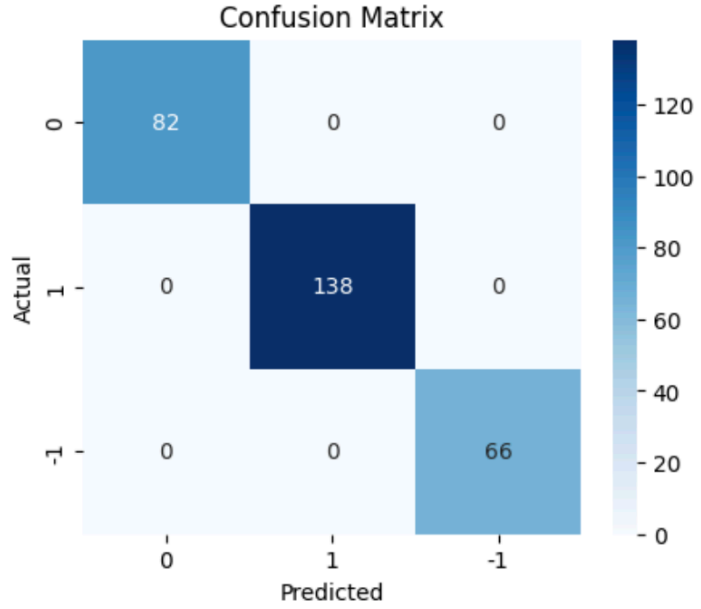
Sentiment Analysis - Model Evaluation Criterion

- Base Model - Word2Vec Training Data

The model exhibits perfect performance on the training data, achieving 100% accuracy, recall, precision, and F1-score. This suggests the model has effectively learned to represent the words in the training data and can accurately predict their relationships.

Training performance:

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

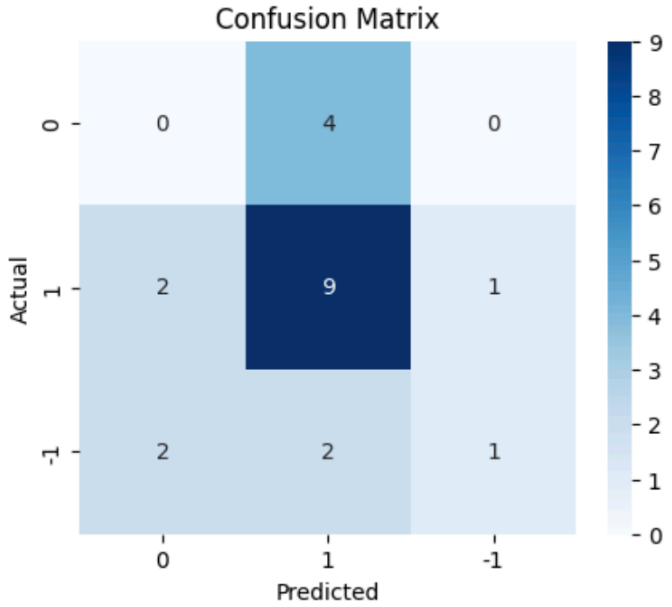


Sentiment Analysis - Model Evaluation Criterion

- Base Model - Word2Vec Validation Data

The Word2Vec model exhibits a lower performance on the validation data compared to the training data. All metrics, including Accuracy, Recall, Precision, and F1-score, are significantly lower than 1.0. This suggests potential overfitting on the training data, where the model has learned to perform well on the training examples but struggles to generalize to unseen data.

```
Validation performance:
    Accuracy   Recall  Precision       F1
0   0.47619  0.47619   0.461905  0.44898
```



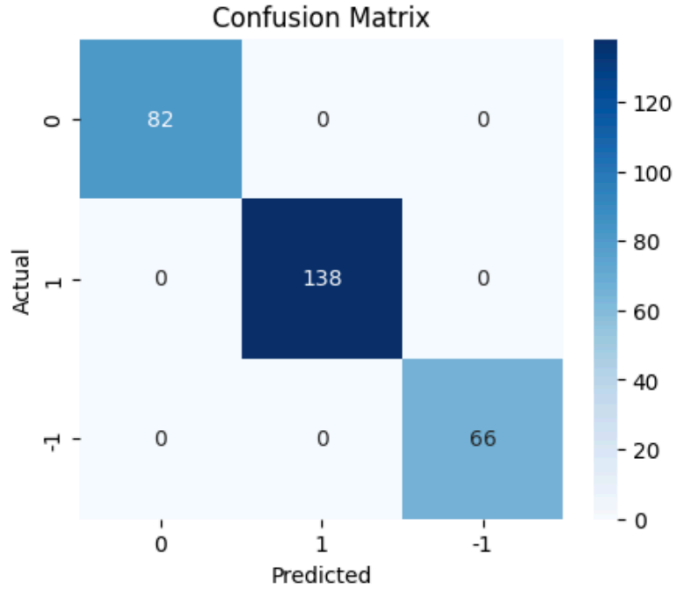
Sentiment Analysis - Model Evaluation Criterion

- Base Model - GloVe Training Data

The GloVe model demonstrates perfect performance on the training data. All metrics – Accuracy, Recall, Precision, and F1-score – achieve a score of 1.0. This indicates that the model has effectively learned to represent the words and their relationships within the training data.

Training performance:

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

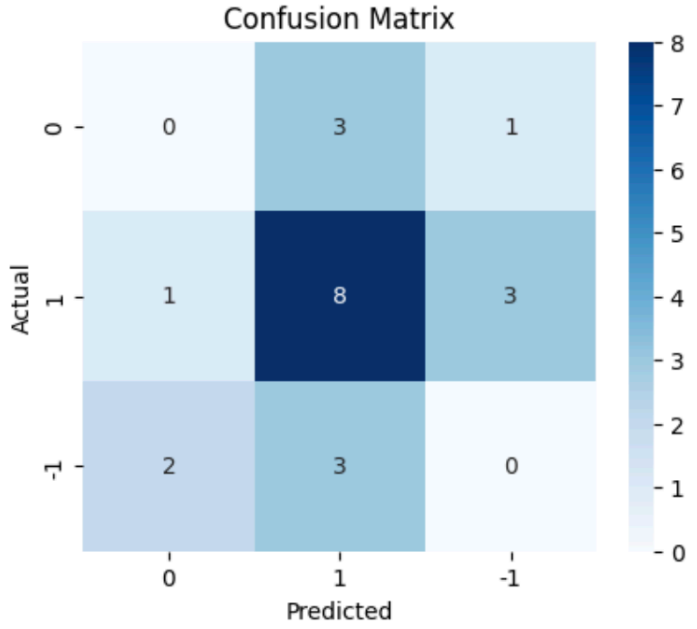


Sentiment Analysis - Model Evaluation Criterion

- Base Model - GloVe Validation Data

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.380952  0.380952   0.326531  0.351648
```

The GloVe model exhibits significantly lower performance on the validation data compared to its perfect score on the training data. All metrics, including Accuracy, Recall, Precision, and F1-score, are well below 1.0. This indicates that the model may be overfitting to the training data, failing to generalize effectively to unseen examples.

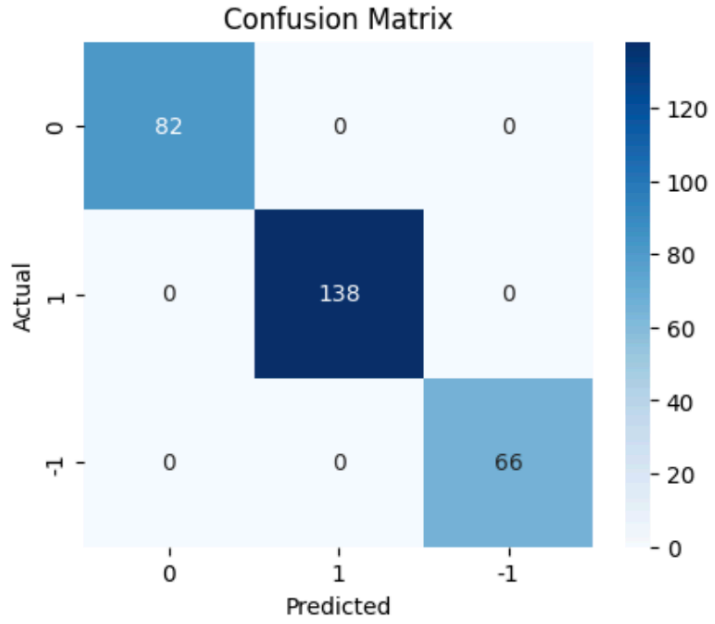


Sentiment Analysis - Model Evaluation Criterion

- Base Model - Sentence Transformer Training Data

```
Training performance:
    Accuracy  Recall  Precision   F1
0        1.0     1.0        1.0  1.0
```

The Base Model (Sentence Transformer) achieves perfect performance on the training data. All metrics - Accuracy, Recall, Precision, and F1-score - are 1.0. This indicates that the model has effectively learned to represent and classify the training data.



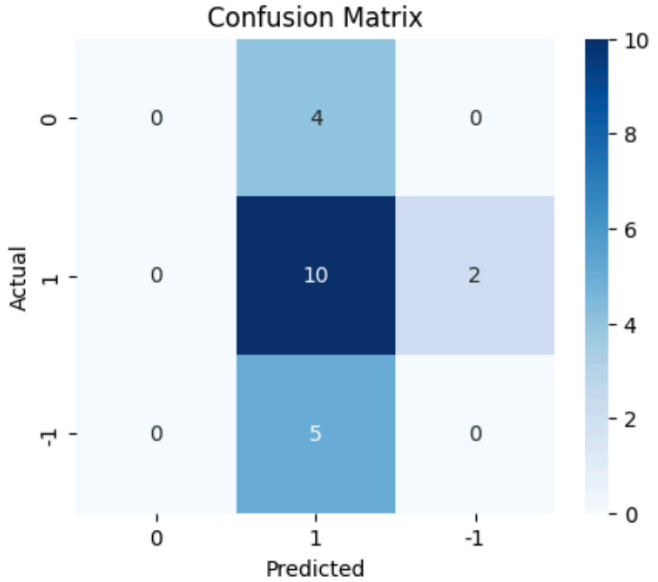
Sentiment Analysis - Model Evaluation Criterion

- Base Model - Sentence Transformer Validation Data

The Base Model (Sentence Transformer) shows a significant drop in performance on the validation data compared to its perfect training score. All metrics (Accuracy, Recall, Precision, F1-score) are substantially lower than 1.0. This suggests a potential overfitting issue, where the model has learned to perform well on the training data but struggles to generalize to unseen data.

Validation performance:

	Accuracy	Recall	Precision	F1
0	0.47619	0.47619	0.300752	0.368664



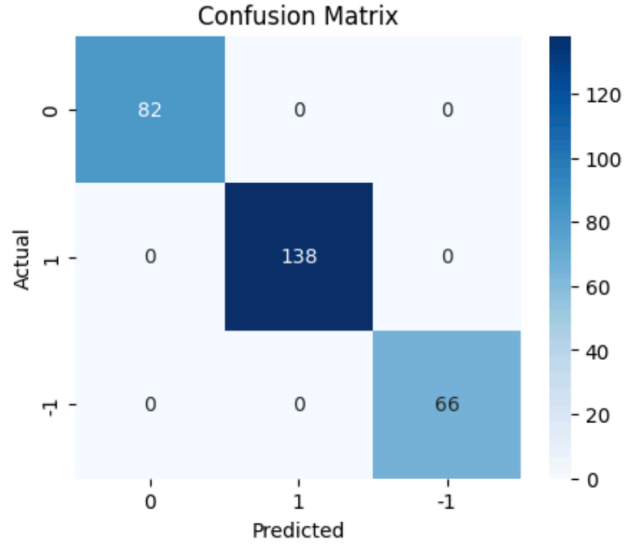
Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - Word2Vec - Training Data

The model exhibits perfect performance on the training data, achieving 100% accuracy, recall, precision, and F1-score. This suggests the model has effectively learned to represent the words in the training data and can accurately predict their relationships.

Training performance:

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0

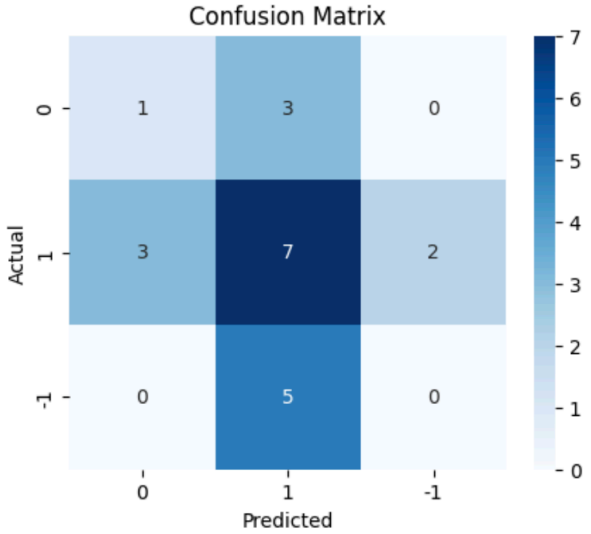


Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - Word2Vec- Validation Data

The model exhibits a lower performance on the validation data compared to the training data. All metrics, including Accuracy, Recall, Precision, and F1-score, are significantly lower than 1.0. This suggests potential overfitting on the training data, where the model has learned to perform well on the training examples but struggles to generalize to unseen data.

```
Validation performance:
    Accuracy    Recall  Precision        F1
0  0.380952  0.380952   0.314286  0.343915
```



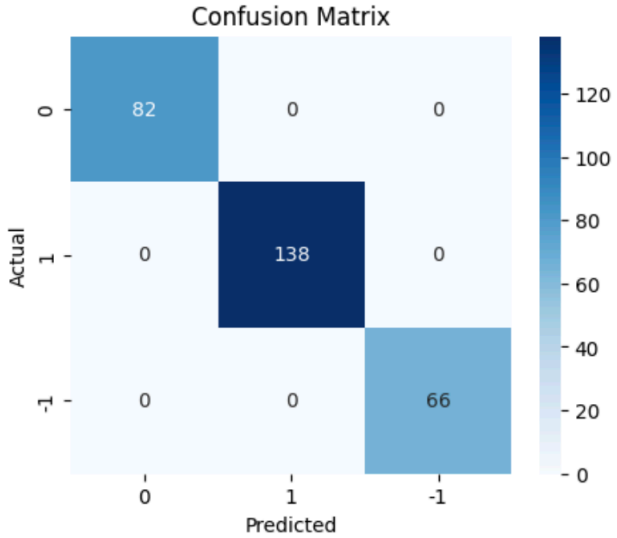
Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - GloVe - Training Data

The model demonstrates perfect performance on the training data. All metrics – Accuracy, Recall, Precision, and F1-score – achieve a score of 1.0. This indicates that the model has effectively learned to represent the words and their relationships within the training data.

Training performance:

	Accuracy	Recall	Precision	F1
0	1.0	1.0	1.0	1.0



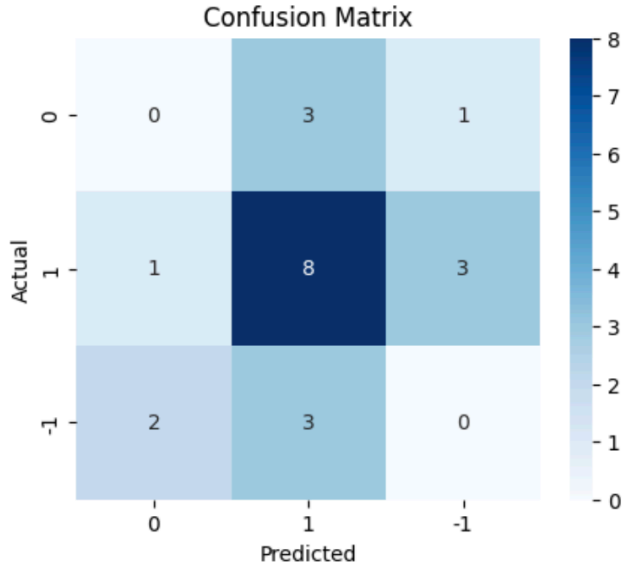
Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - GloVe - Validation Data

The model exhibits significantly lower performance on the validation data compared to its perfect score on the training data. All metrics, including Accuracy, Recall, Precision, and F1-score, are well below 1.0. This indicates that the model may be overfitting to the training data, failing to generalize effectively to unseen examples.

Validation performance:

	Accuracy	Recall	Precision	F1
0	0.380952	0.380952	0.326531	0.351648

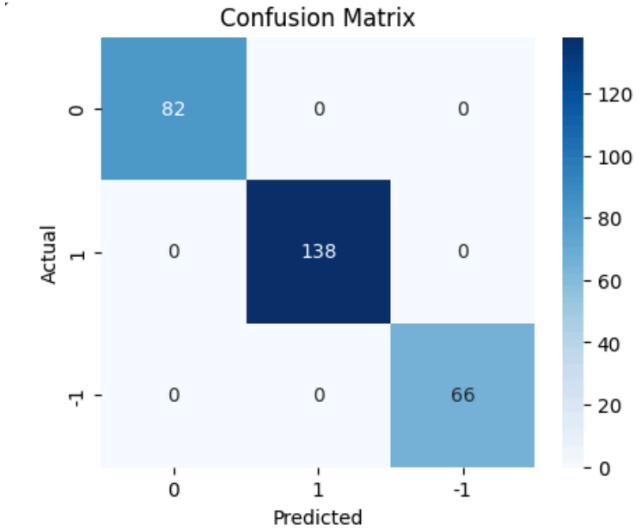


Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - Sentence Transformer - Training Data

The Model achieves perfect performance on the training data. All metrics - Accuracy, Recall, Precision, and F1-score - are 1.0. This indicates that the model has effectively learned to represent and classify the training data.

```
Training performance:
    Accuracy  Recall  Precision   F1
0        1.0     1.0        1.0  1.0
```

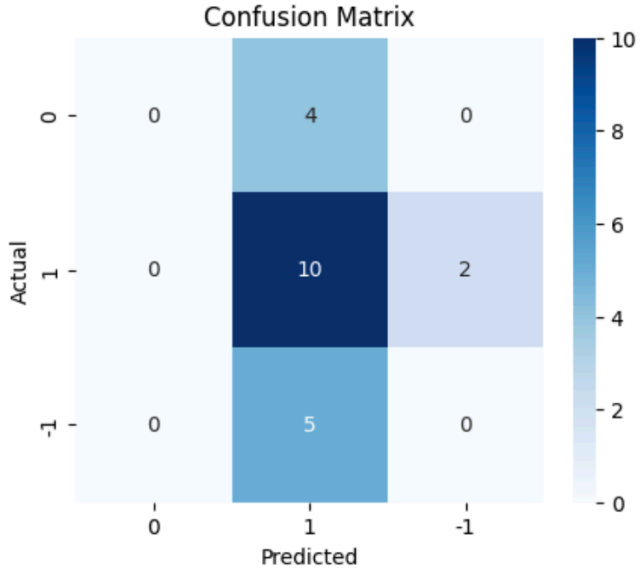


Sentiment Analysis - Model Evaluation Criterion

- Tuned Model - Sentence Transformer - Validation Data

```
Validation performance:
    Accuracy   Recall  Precision        F1
0   0.47619  0.47619   0.300752  0.368664
```

The Modelshows a significant drop in performance on the validation data compared to its perfect training score. All metrics (Accuracy, Recall, Precision, F1-score) are substantially lower than 1.0.



Sentiment Analysis - Model Evaluation Criterion

- Final Model Selection_Training Data

Observation:

All base models (Word2Vec, GloVe, and Sentence Transformer) achieve perfect scores across all metrics (Accuracy, Recall, Precision, and F1) on the training data. This indicates that all models have effectively learned to represent and classify the training data. However, perfect performance on training data alone doesn't guarantee good generalization to unseen data and might suggest potential overfitting.

Training performance comparison:

	Base Model (Word2Vec)	Base Model (GloVe)	Base Model (Sentence Transformer)	Tuned Model (Word2Vec)	Tuned Model (GloVe)	Tuned Model (Sentence Transformer)
Accuracy	1.0	1.0	1.0	1.0	1.0	1.0
Recall	1.0	1.0	1.0	1.0	1.0	1.0
Precision	1.0	1.0	1.0	1.0	1.0	1.0
F1	1.0	1.0	1.0	1.0	1.0	1.0

Sentiment Analysis - Model Evaluation Criterion

- Final Model Selection_Validation Data

Observation:

The validation performance varies across the different models. The Base Model using Sentence Transformer achieves the highest performance with the best scores in Accuracy, Recall, Precision, and F1. However, the Tuned Model using Sentence Transformer also shows promising results, indicating potential for further improvement through hyperparameter tuning.

Validation performance comparison:

	Base Model (Word2Vec)	Base Model (GloVe)	Base Model (Sentence Transformer)	Tuned Model (Word2Vec)	Tuned Model (GloVe)	Tuned Model (Sentence Transformer)
Accuracy	0.380952	0.380952	0.476190	0.380952	0.380952	0.476190
Recall	0.380952	0.380952	0.476190	0.380952	0.380952	0.476190
Precision	0.323810	0.326531	0.300752	0.314286	0.326531	0.300752
F1	0.350020	0.351648	0.368664	0.343915	0.351648	0.368664

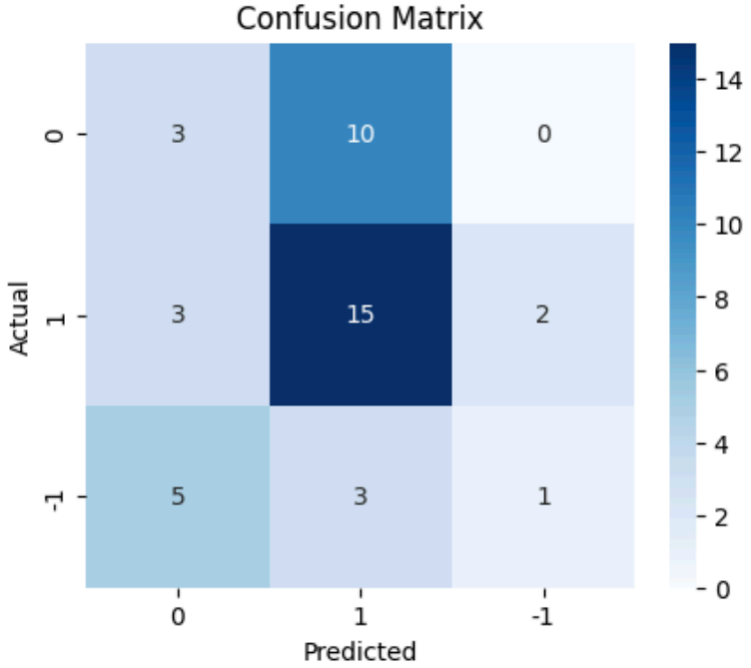
Sentiment Analysis - Model Evaluation Criterion

- Final Model - Test Data

Observation:

The final model exhibits a performance on the test data that is comparable to its performance on the validation set. All metrics (Accuracy, Recall, Precision, and F1-score) are around 0.45, indicating a moderate level of predictive accuracy. This performance suggests that the model has some level of generalization ability but may still have room for improvement.

```
Test performance for the final model:
    Accuracy    Recall  Precision        F1
0  0.452381  0.452381   0.410946  0.410714
```



Content Summarization – Data Preprocessing

Aggregating the data weekly

Time Series Data: The data appears to be time-series data, with a "Date" column indicating the date of each news article.
News Content: The "News" column contains snippets of news articles, likely related to financial or economic events.
Potential for Sentiment Analysis: The news articles could be used to perform sentiment analysis to gauge market sentiment and potentially predict stock price movements.
Limited Context: The provided data is limited in scope, only showing a few weeks of news articles. A larger dataset with a longer time horizon would be beneficial for more comprehensive analysis.

	Date	News
0	2019-01-06	The tech sector experienced a significant dec...
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...
2	2019-01-20	The U.S. stock market declined on Monday as c...
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...
5	2019-02-10	The Dow Jones Industrial Average, S&P 500, an...
6	2019-02-17	This week, the European Union's second highes...
7	2019-02-24	This news article discusses progress towards ...
8	2019-03-03	The Dow Jones Industrial Average and other ma...
9	2019-03-10	Spotify, the world's largest paid music strea...
10	2019-03-17	The United States opposes France's digital se...
11	2019-03-24	Facebook's stock price dropped more than 3% o...
12	2019-03-31	This news article reports that the S&P 500 In...
13	2019-04-07	Apple and other consumer brands, including LV...
14	2019-04-14	In March, mobile phone shipments to China dro...
15	2019-04-21	The chairman of Taiwan's Foxconn, Terry Gou, ...
16	2019-04-28	Taiwan's export orders continued to decline f...
17	2019-05-05	Spotify reported better-than-expected Q1 reve...

Content Summarization – Modeling Approach

- Overview of the Large Language Model used

Mistral-7B-Instruct-v0.2 is a **7-billion-parameter** large language model (LLM) fine-tuned for **instruction-following tasks**. Developed by **Mistral AI**, it excels in **natural language understanding, reasoning, and code generation**. This version enhances multi-turn interactions and alignment with user intent. Available in **GGUF format**, it is optimized for **efficient inference** on **CPUs and GPUs** using **llama.cpp**. Ideal for **chatbots, AI assistants, and text-based applications**, it balances **power and efficiency**, making it a great choice for **local AI deployment**. While not as powerful as larger models like GPT-4, it offers excellent **performance for lightweight AI applications**.

Content Summarization – Modeling Approach

- **Parameters of the Large Language Model**

Mistral-7B-Instruct-v0.2 is a **7-billion-parameter LLM** fine-tuned for **instruction-following tasks**. Optimized for **efficient inference** in **GGUF format**, it supports **CPU/GPU** via **llama.cpp**. It excels in **chatbots, text generation, and reasoning**, offering a **4500-token context window** for better long-form responses.

This code initializes an **LLM instance** using **llama-cpp-python**, loading a **Mistral-7B GGUF model**. It sets a **4500-token context window** for long responses and uses n_cores**=-2**, meaning the model will utilize **all CPU cores except 2** for efficient processing. Ideal for **CPU-based inference**.

Overall:

The Mistral-7B-Instruct-v0.2 model appears to be a powerful LLM with a good balance of performance, efficiency, and context window size. The provided initialization code seems reasonable for CPU-based inference, leveraging multiple cores for speed. However, it's essential to consider the memory requirements and potential performance limitations based on the available hardware resources.

Content Summarization – Sample Input/Output

- The Sample Input and the Prompt used for this task

Overall, the prompt is well-structured and provides clear guidance to the model on how to analyze the news headline and extract the main topics.

```
prompt = """
You are an expert data analyst specializing in news content analysis.

Task: Analyze the provided news headline and return the main topics contained within it.

Instructions:
1. Read the news headline carefully.
2. Identify the main subjects or entities mentioned in the headline.
3. Determine the key events or actions described in the headline.
4. Extract relevant keywords that represent the topics.
5. List the topics in a concise manner.

Return the output in JSON format with keys as the topic number and values as the actual topic.
"""
```

Content Summarization – Sample Input/Output

- Sample Output

{"1": "Politics", "2": "Economy", "3": "Health" }

- Observations

Overall, the sample output provides a clear and concise representation of the extracted topics, making it easy to interpret and use for further analysis.

Content Summarization – Raw Model Output

- The snapshot of the resultant dataframe

DataFrame containing news articles, their dates, and a column labeled "Key Events" which seems to hold JSON-formatted data. However, there are several JSON parsing errors, likely due to invalid or incomplete JSON strings within the "Key Events" column. These errors need to be addressed to further process and analyze the data.

Error parsing JSON: Expecting ',' delimiter: line 3 column 11 (char 332)

	Date	News	Key Events	model_response_parsed
0	2019-01-06	The tech sector experienced a significant dec...	{\n "topics": [\n {\n ...	{}
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...	{\n "Technology": [\n {\n...	{}
2	2019-01-20	The U.S. stock market declined on Monday as c...	{\n "Global Economy": ["Unexpected ...	{}
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...	{\n "Swiss National Bank": ["Andrea ...	{}
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...	{\n "Topics": [\n {\n ...	{}

Content Summarization – Final Output

- The steps to parse the model's output

```
model_response_parsed
0                    {}
1                    {}
2                    {}
3                    {}
4                    {}
```

Content Summarization – Final Output

- The snapshot of the final dataframe

"**Week End Date**": The date of the week ending period.
"**News**": A brief summary or headline of a news article related to the stock market.
"**Week Positive Events**": A column likely intended to store the number of positive events mentioned in the news for that week.
"**Week Negative Events**": A column likely intended to store the number of negative events mentioned in the news for that week.

	Week End Date	News	Week Positive Events	Week Negative Events
0	2019-01-06	The tech sector experienced a significant dec...	0	0
1	2019-01-13	Sprint and Samsung plan to release 5G smartph...	0	0
2	2019-01-20	The U.S. stock market declined on Monday as c...	0	0
3	2019-01-27	The Swiss National Bank (SNB) governor, Andre...	0	0
4	2019-02-03	Caterpillar Inc reported lower-than-expected ...	0	0

APPENDIX

Data Background and Contents

- The data background and contents

Data Background:

- **Domain:** The data appears to be related to **stock market news sentiment analysis**. This involves analyzing news articles related to a specific company or the overall market to gauge investor sentiment and potentially predict stock price movements.
- **Data Sources:**
  - **News Articles:** The core dataset likely consists of a collection of news articles related to the stock market. These articles could be sourced from various news outlets, financial publications, or news aggregators.
  - **Stock Data:** Historical stock data, including features like "Open", "High", "Low", "Close", and "Volume", is likely included to provide context and enable analysis of the relationship between news sentiment and stock price movements.

Data Background and Contents

Data Contents (Based on Snippets and Context):

- **News:**
  - **Text Data:** News articles or headlines containing information related to market events, company announcements, economic indicators, and other relevant topics.
  - **Date:** Timestamps associated with each news article, indicating when the news was published.

Stock Data:

- **Open, High, Low, Close:** Daily stock prices, representing the opening, highest, lowest, and closing prices for a particular stock.
- **Volume:** The number of shares traded for the stock on a given day.

Label: A target variable that likely represents the stock's performance or sentiment (e.g., -1 for negative sentiment, 1 for positive sentiment). This label could be derived from historical stock price movements or other relevant factors.

Slide Header

Inferences:

- **Time Series Analysis:** The presence of time-stamped news articles suggests that time-series analysis techniques might be relevant for this dataset.
- **Natural Language Processing (NLP):** NLP techniques will be crucial for processing the news text, extracting relevant information (e.g., keywords, entities, sentiment), and converting it into a suitable format for machine learning models.
- **Feature Engineering:** Feature engineering will be important to extract meaningful features from the news data and combine it with the stock data for effective model training.

Great Learning

Happy Learning !