A Sentiment-Driven Financial Trading Pipeline

GROUP NO. 08
DIPEN PRAJAPATI
CHAITANYA DEOGAONKAR

INTRODUCTION

- Converts the tone and context of financial headlines/articles into quantitative sentiment scores
- Enables integration of unstructured text data into structured trading, risk, and research models
- Uses machine learning or deep learning models to interpret financial language accurately
- Supports both real-time and historical news processing
- Helps identify bullish or bearish signals before they are reflected in price movement

Complements traditional methods like:

- Technical indicators (e.g., RSI, MACD)
- Fundamental analysis (e.g., earnings, valuations)

Enhances decision-making by:

- Detecting market-moving sentiment early
- Filtering relevant news from noise
- Quantifying market psychology and narrative shifts

NATURAL LAUNGUAGE PROCESSING

What is NLP?

- Subfield of AI and Linguistics focused on human language understanding
- Enables machines to understand, interpret, and generate human language
- Converts unstructured text into structured data
- Extracts meaning, intent, and syntax from language
- Used in chatbots, voice assistants, language translation, and text classification
- Powers sentiment analysis, spam filters, NER, and speech recognition
- Helps machines handle context, ambiguity, and tone
- Automates insights from large-scale textual data

WHAT IS SENTIMENT ANALYSIS?

What is Sentiment Analysis?

- A Application of Natural Language Processing (NLP)
- Also known as opinion mining
- Automatically detects and quantifies attitude, emotion, or stance expressed in:
  - Text, speech, Other media
- Widely used in: Product reviews, Social media analysis, Financial news interpretation, Customer service feedback

Advanced Emotion Classes (optional in some models):

- Joy, Anger, Fear, Love, Surprise, etc.

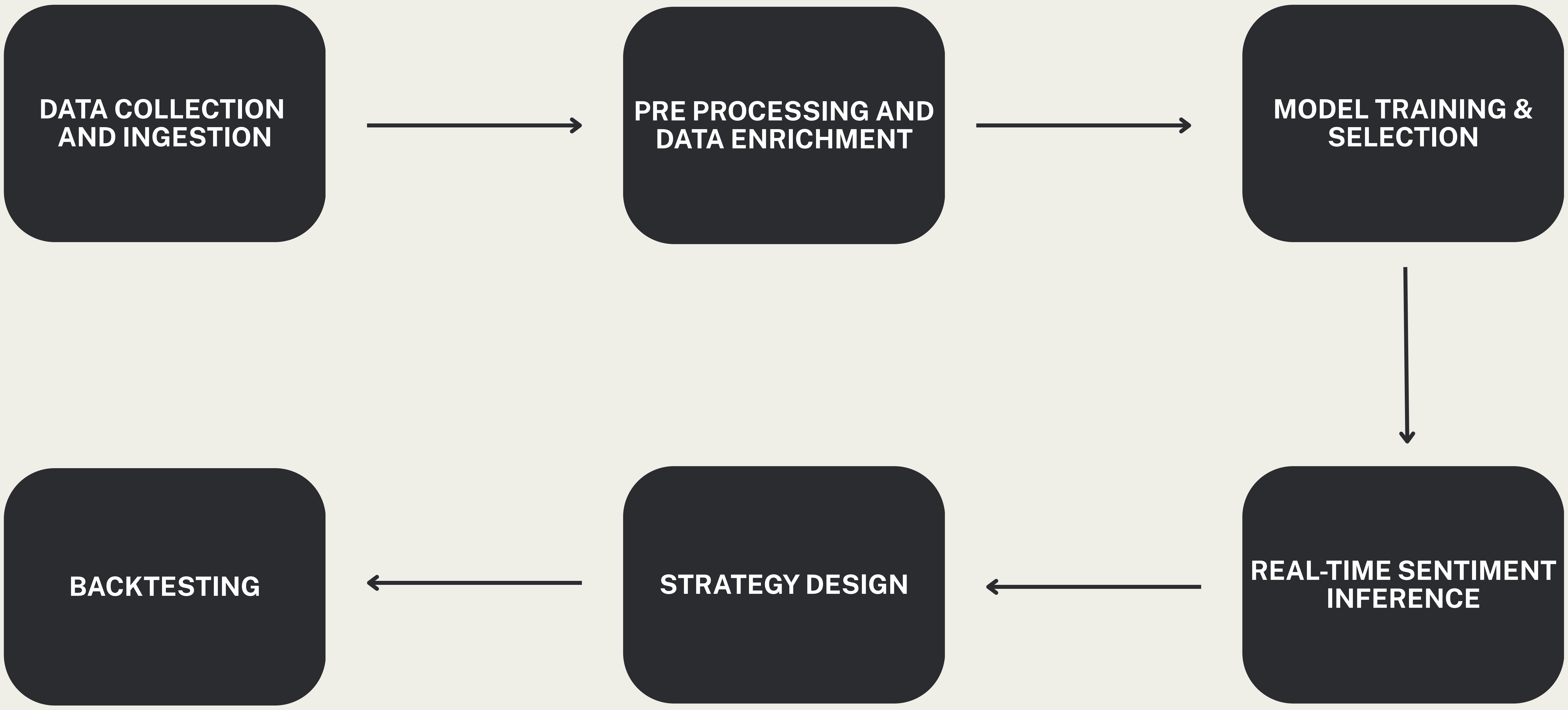
Sentiment Analysis in This Project

- Calculates sentiment of real-time financial news headlines
- Predicts trade decisions based on market tone and outlook
- Integrates sentiment with technical indicators to generate:
  - BUY, SELL, or HOLD signals

Types of Financial Sentiment Categories

- Positive - Indicates a bullish market outlook
- Negative - Suggests a bearish market trend
- Neutral - Signals market stability or indecisiveness

PROJECT FLOWCHART



DATA COLLECTION AND INGESTION

PRE PROCESSING AND DATA ENRICHMENT

MODEL TRAINING & SELECTION

REAL-TIME SENTIMENT INFERENCE

STRATEGY DESIGN

BACKTESTING

Evaluating Models on Financial News Sentiment Data

DATA COLLECTION AND INGESTION

The data was collected from various sources like:

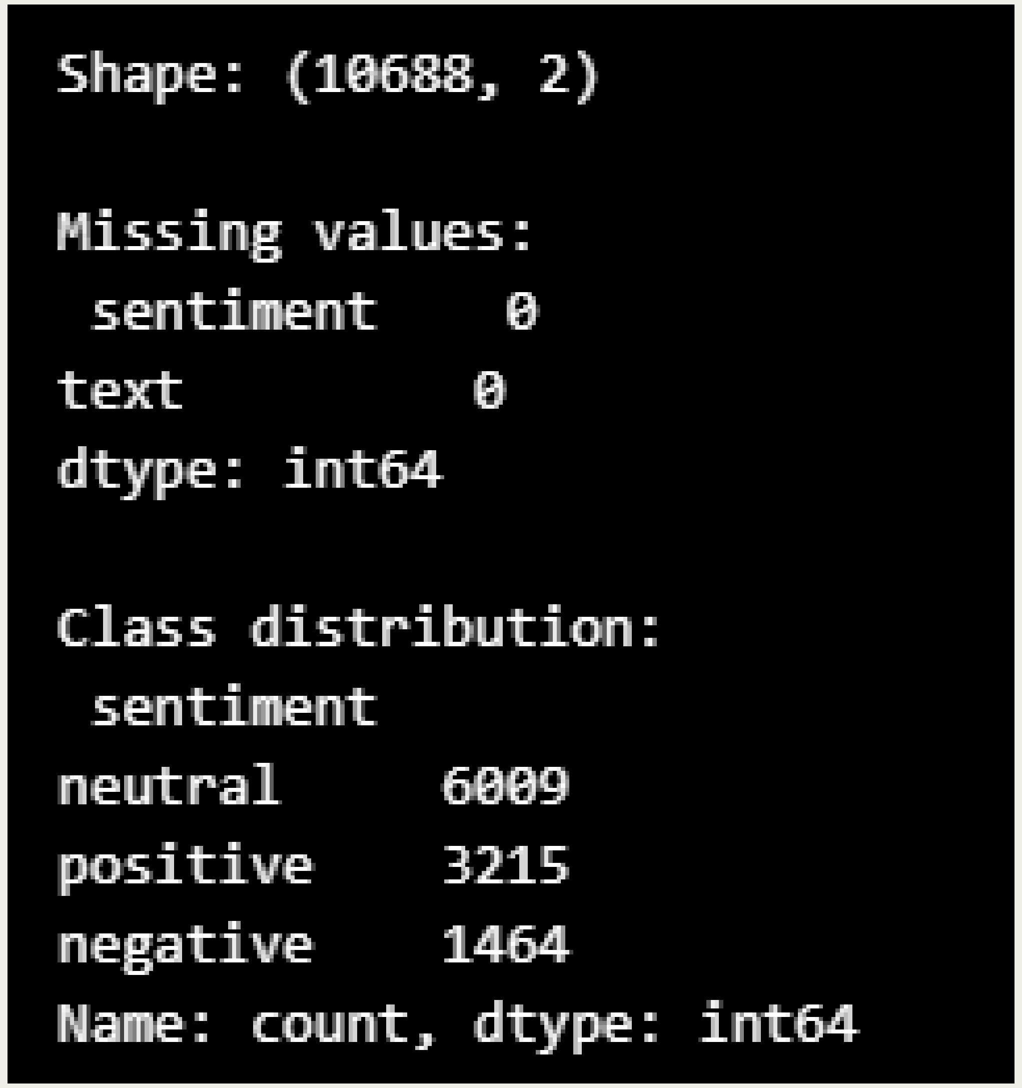
1. Kaggle - training and testing the models
2. Yahoo Finance - downstream tasks
3. News API - downstream tasks

PRE PROCESSING AND DATA ENRICHMENT

The collected data was explored, preprocessed and analyzed and converted to desired format.

The preprocessing tasks included:

1. Handled Missing Values – Removed incomplete rows for data integrity
2. Cleaned Text – Removed special characters and noise from headlines
3. Encoded Sentiments – Converted labels (Positive/Negative/Neutral) to numeric form
4. Tokenized Text – Prepared text inputs for transformer models
5. Applied TF-IDF – Extracted word importance for traditional ML models



```
Shape: (10688, 2)

Missing values:
 sentiment    0
text         0
dtype: int64

Class distribution:
 sentiment
neutral     6009
positive    3215
negative    1464
Name: count, dtype: int64
```

MODEL TRAINING & SELECTION

The models used for training the data and providing the final sentiment of the financial news were:

1. Logistic Regression
2. Naive Bayes
3. Support Vector Machine
4. Random Forest Classifier
5. FinBERT

1.LOGISTIC REGRESSION

- A supervised machine learning algorithm used for classification tasks
- Commonly used as a baseline model in Natural Language Processing
- Fits a linear combination of features and applies the logistic (sigmoid) function
- Produces a probability output used for binary or multiclass classification

Advantages:

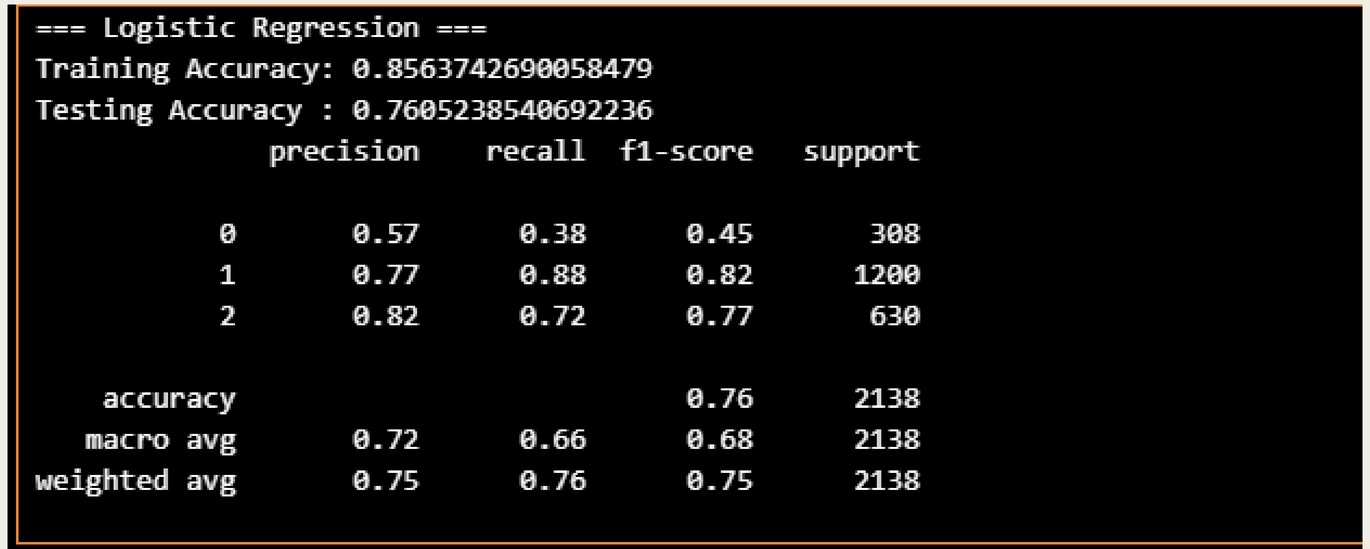
- Easy to interpret, fast to train, and computationally efficient
- Ideal for small datasets and real-time applications

Limitations

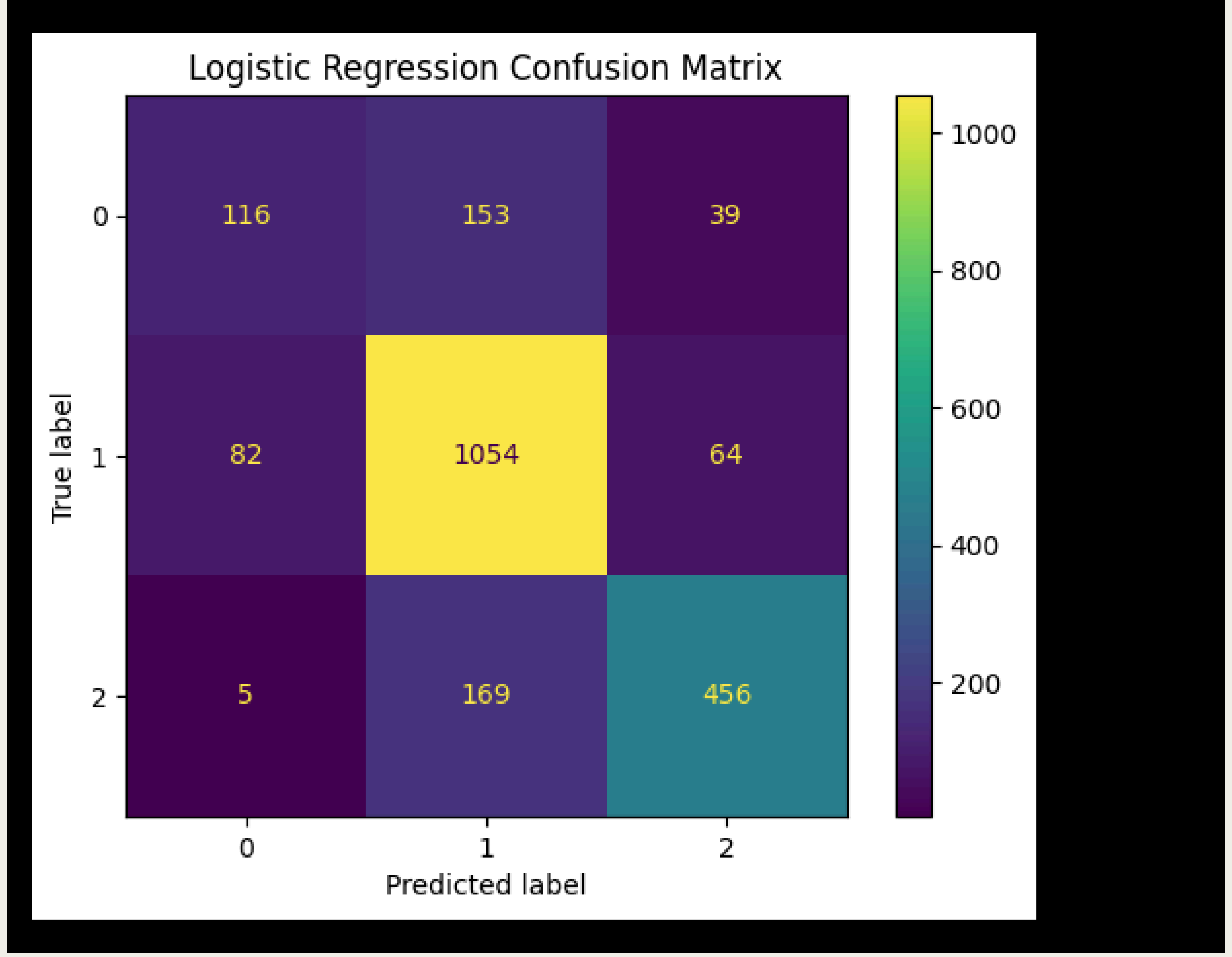
- Only captures linear decision boundaries
- Performance degrades with imbalanced class distributions
- Not well-suited for complex, non-linear sentiment patterns

$$p = \frac{1}{1+e^{-z}}$$

$$z = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$



```
=== Logistic Regression ===
Training Accuracy: 0.8563742690058479
Testing Accuracy : 0.7605238540692236
              precision    recall  f1-score   support

           0       0.57      0.38      0.45       308
           1       0.77      0.88      0.82      1200
           2       0.82      0.72      0.77       630

    accuracy                           0.76      2138
   macro avg       0.72      0.66      0.68      2138
weighted avg       0.75      0.76      0.75      2138
```



Logistic Regression Confusion Matrix

2.NAIVE BAYES

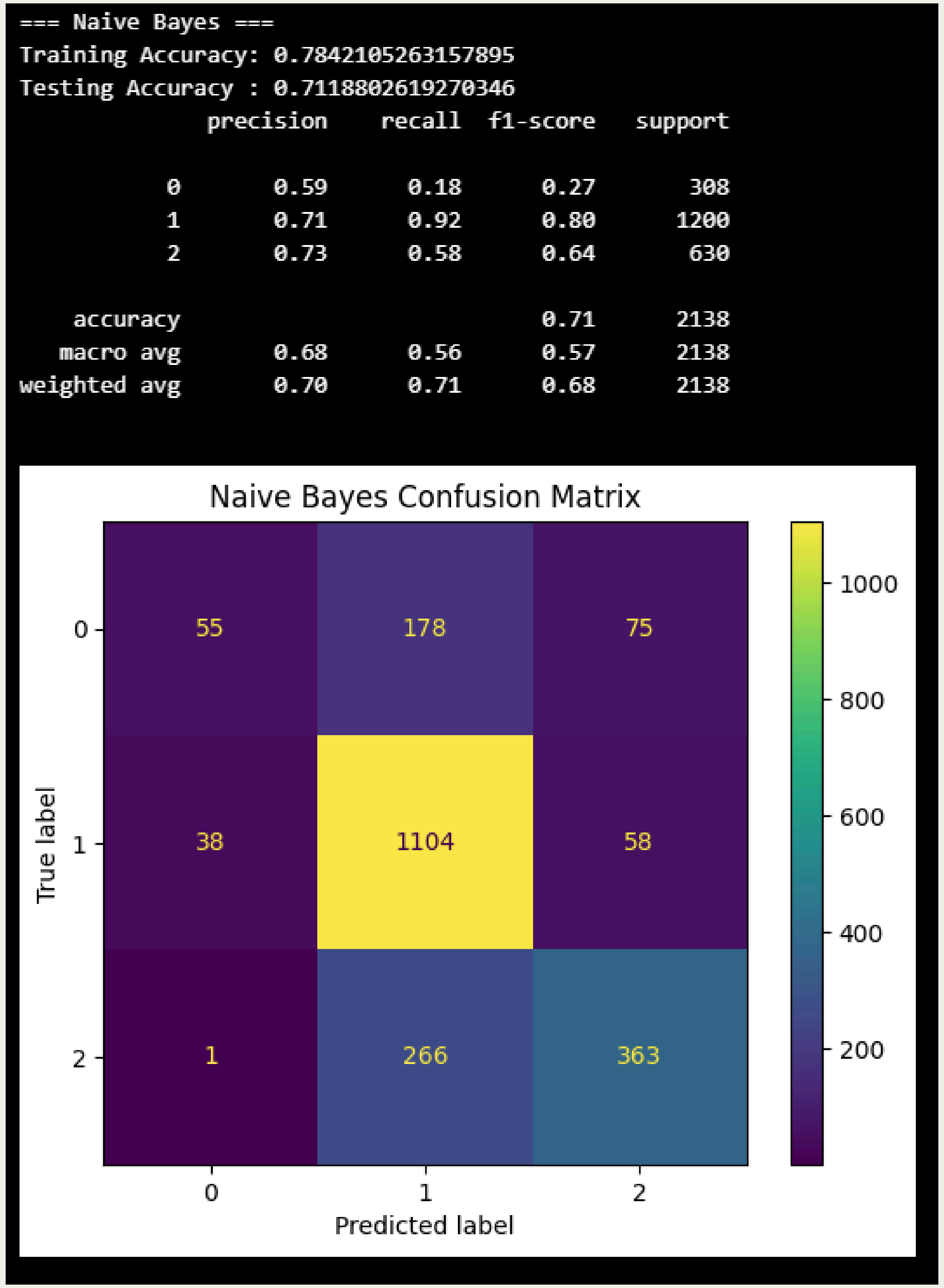
- A supervised probabilistic classification algorithm based on Bayes’ Theorem
- Assumes that all features are conditionally independent given the class label (naive assumption)
- Calculates the posterior probability of each class and selects the one with the highest value

Advantages:

- Often outperforms complex models on short text and noisy data
- Naturally probabilistic — outputs class confidence

Limitations:

- Not ideal for long-range dependencies or capturing context
- Performs poorly when the training data is insufficient or heavily unbalanced



```
=== Naive Bayes ===
Training Accuracy: 0.7842105263157895
Testing Accuracy : 0.7118802619270346
              precision    recall  f1-score   support

           0       0.59      0.18      0.27       308
           1       0.71      0.92      0.80      1200
           2       0.73      0.58      0.64       630

    accuracy                           0.71      2138
   macro avg       0.68      0.56      0.57      2138
weighted avg       0.70      0.71      0.68      2138
```

3.SUPPORT VECTOR MACHINE(SVM)

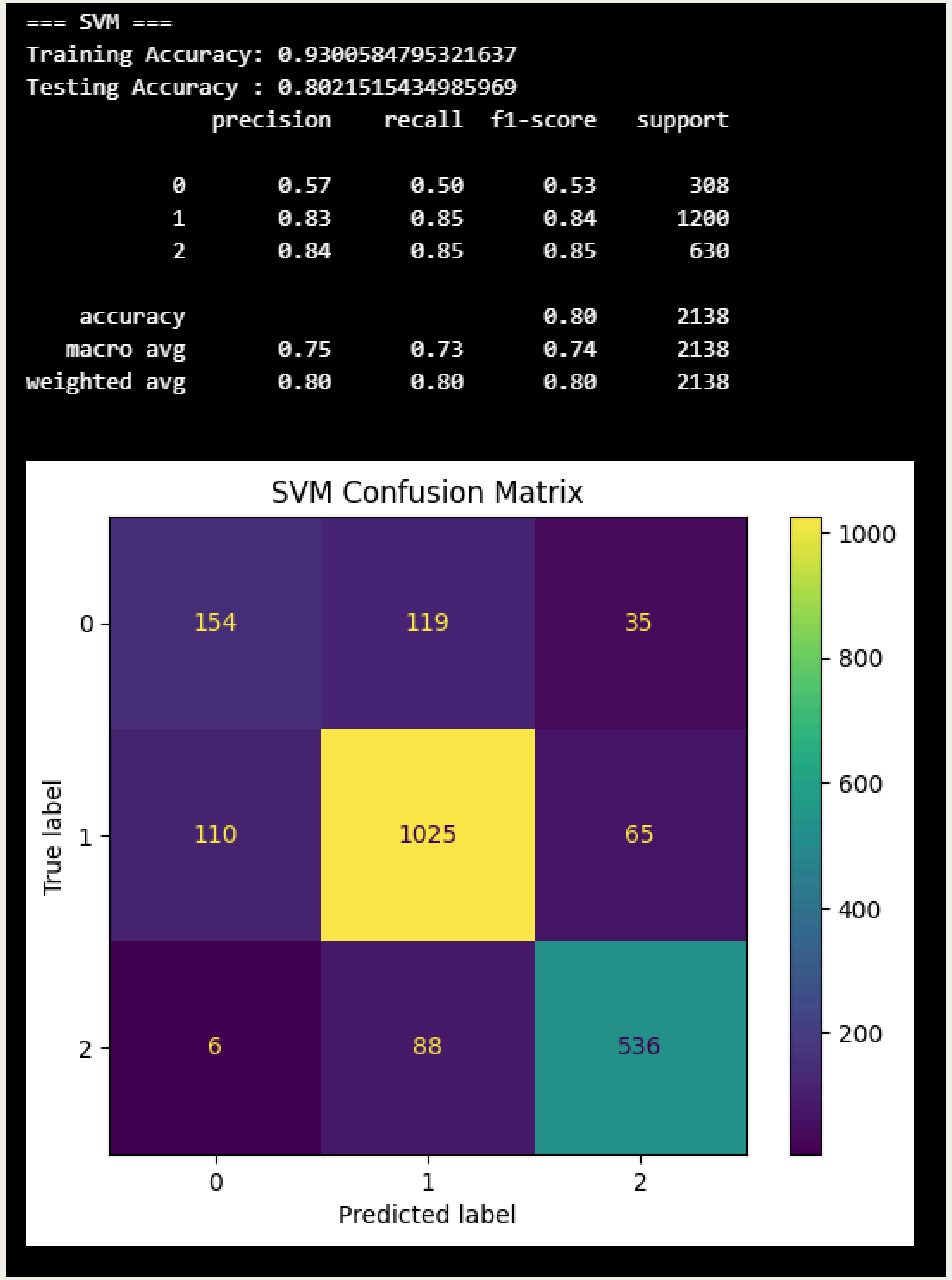
- A supervised machine learning algorithm used for classification and regression
- Finds the best hyperplane that separates data points of different classes with the maximum margin
- Effective in high-dimensional spaces, like those created by text vectorization

Advantages

- Works well for binary and multiclass classification
- Handles high-dimensional data (perfect for NLP)
- Can model non-linear relationships using kernels
- Robust to overfitting, especially with proper regularization

Limitations

- Not ideal for large-scale datasets (slower training time)
- Harder to interpret compared to logistic regression



```
=== SVM ===
Training Accuracy: 0.9300584795321637
Testing Accuracy : 0.8021515434985969
              precision    recall  f1-score   support

           0       0.57      0.50      0.53       308
           1       0.83      0.85      0.84      1200
           2       0.84      0.85      0.85       630

    accuracy                           0.80      2138
   macro avg       0.75      0.73      0.74      2138
weighted avg       0.80      0.80      0.80      2138
```

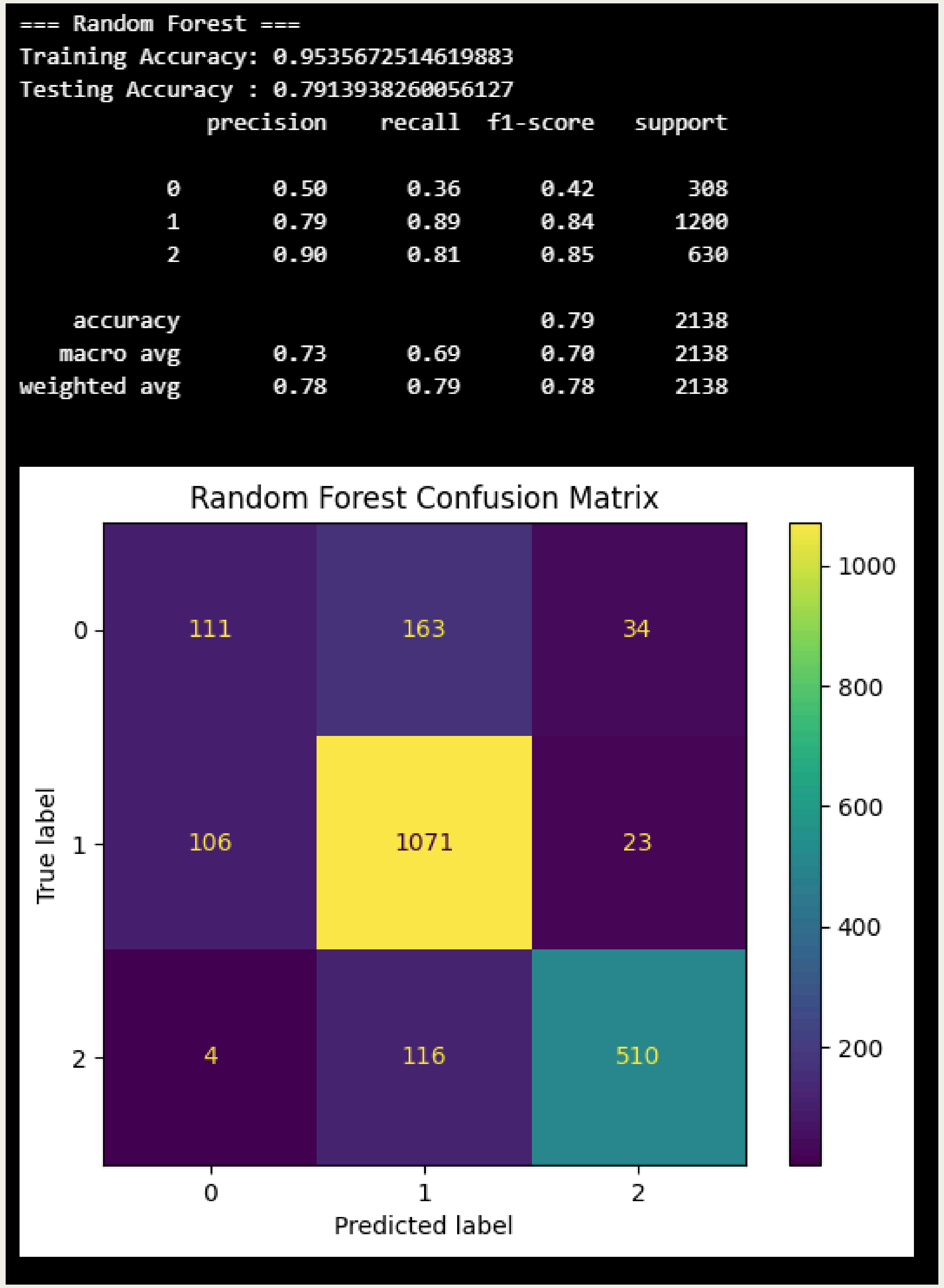
SVM Confusion Matrix

4.RANDOM FOREST CLASSIFIER

Random Forest is an ensemble of decision trees trained of different bootstrap samples of data and each tree votes for the its probabilistic decision and the majority of the probabilities or the probability average decides the final class.

Workflow

1. Create bootstrap sample
2. Grow an unpruned decision tree from the sample.
3. Store the tree.
4. Calculate the majority vote or the mean of class probabilities.
5. Take the average of the numeric outcomes(Regression).



```
=== Random Forest ===
Training Accuracy: 0.9535672514619883
Testing Accuracy : 0.7913938260056127
              precision    recall  f1-score   support

           0       0.50      0.36      0.42       308
           1       0.79      0.89      0.84      1200
           2       0.90      0.81      0.85       630

    accuracy                           0.79      2138
   macro avg       0.73      0.69      0.70      2138
weighted avg       0.78      0.79      0.78      2138
```

Random Forest Confusion Matrix

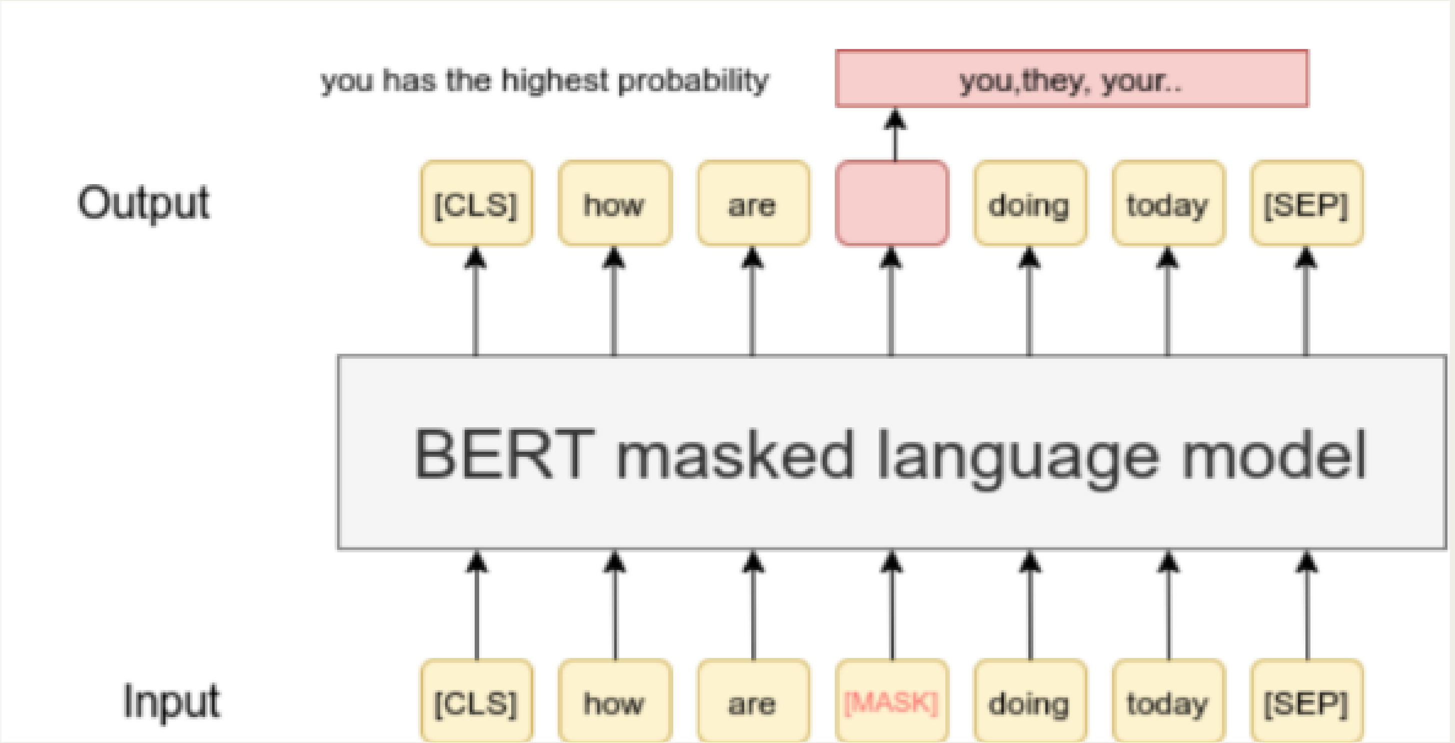
	0	1	2
0	111	163	34
1	106	1071	23
2	4	116	510

5. FINBERT

FinBERT is a domain specific version of BERT(Bidirectional Encoder from Transformers. To understand about FinBERT we first need to understand BERT model as FinBERT is derived from BERT.

What is BERT?

- BERT(Bidirectional Encoder from Transformers) was developed by Google in 2018 to learn the conceptional meanings from the text. BERT is a bidirectional model i.e it can read text from left-to-right and right-to-left.
- BERT is a MLM (Masked Language Model) i.e it randomly masks 15% of the tokens from the text and tries to predict these tokens.



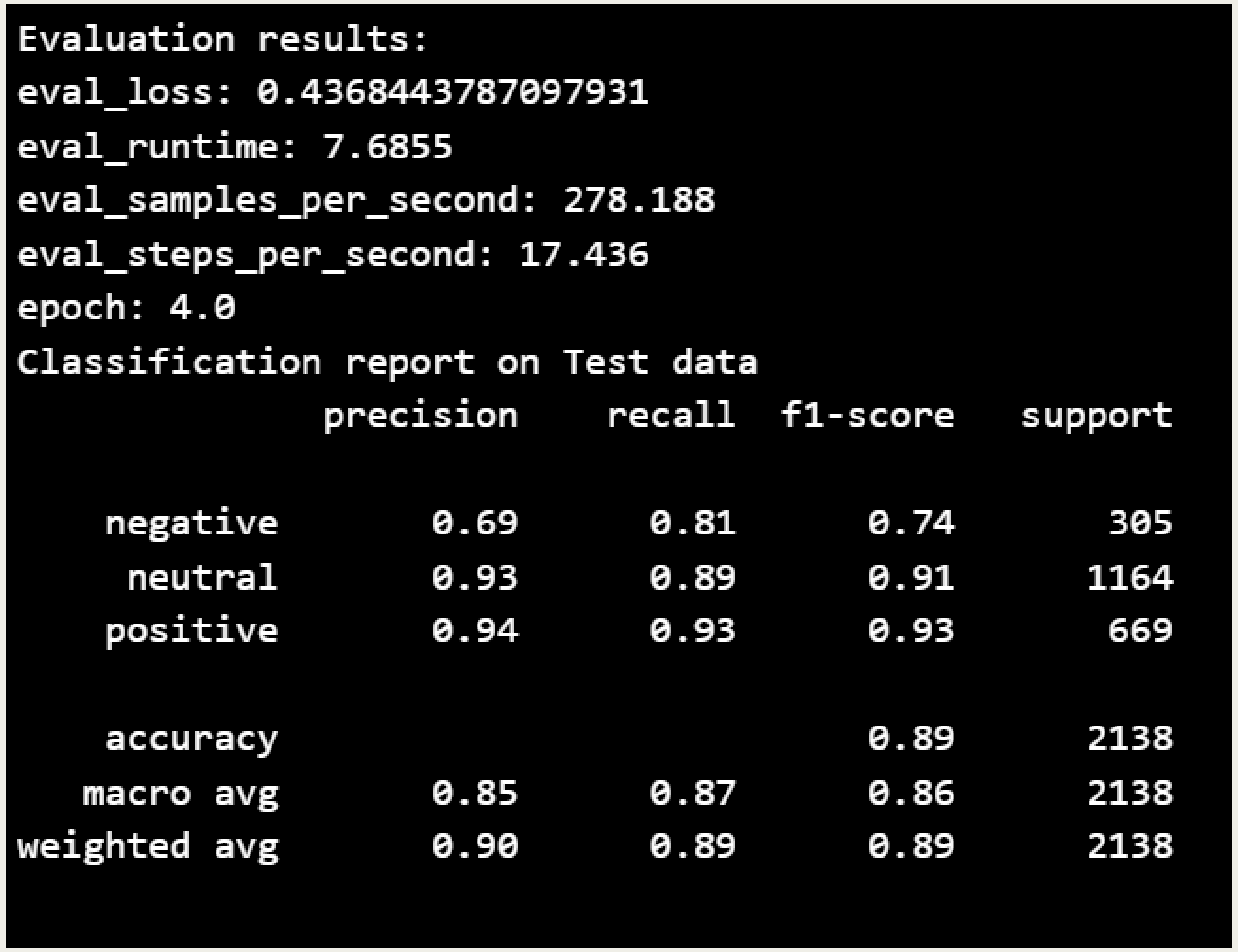
How does BERT calculate the probability of the winning word?

1. Mask a given token from the text.
2. All tokens pass through the transformer encoding layer. Self attention lets the mask position attend every other word and its final hidden vector h_{mask} encodes the bidirectional context from the text.
3. The raw score(logits) is calculated by: $z=W^{T}h_{mask}+b$
4. The softmax activation function then calculates the probabilities,

$$P(token=v|context)= exp(z_{v})/summation_{v'}\ exp(z_{v'})$$

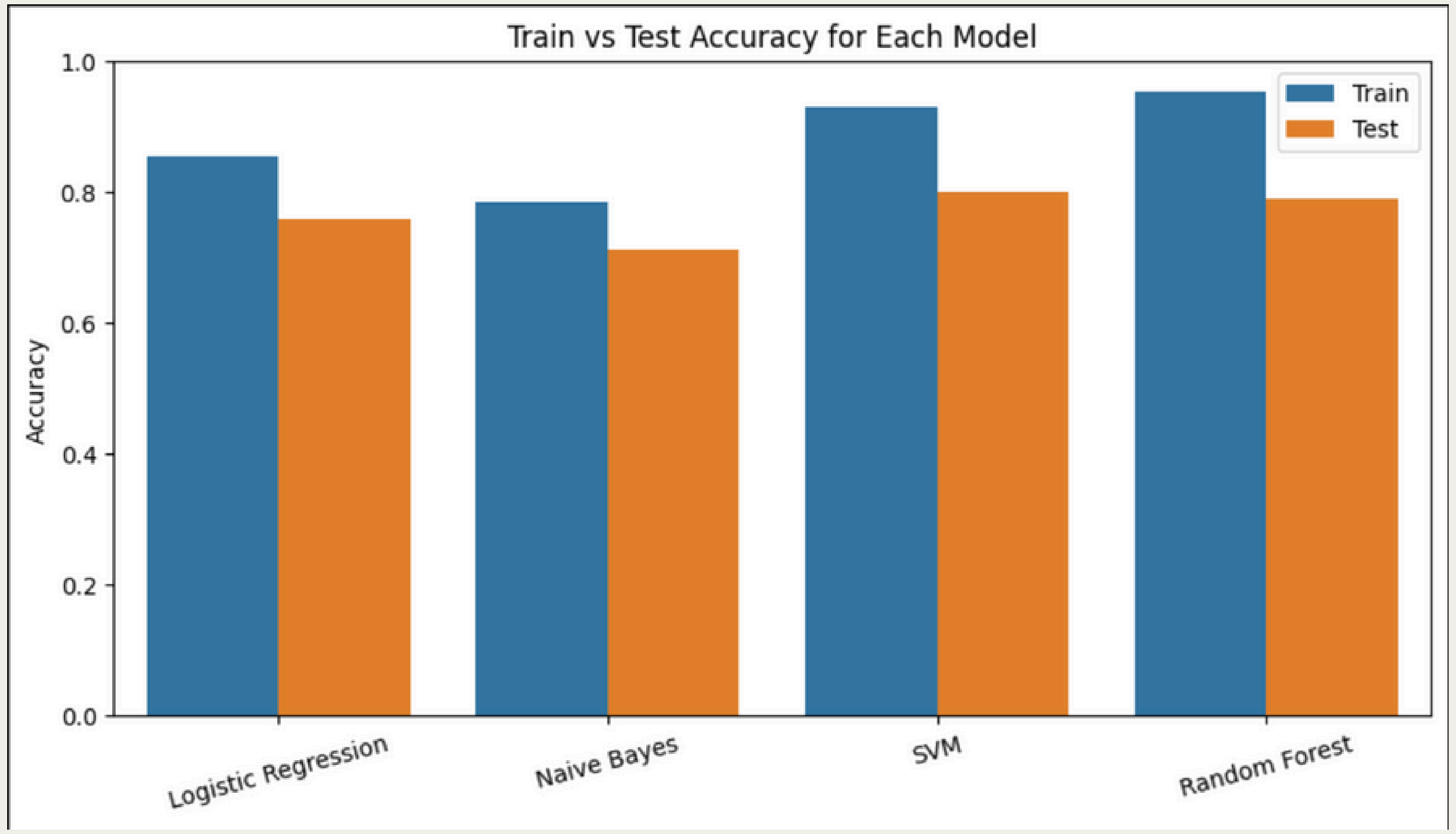
5. FINBERT

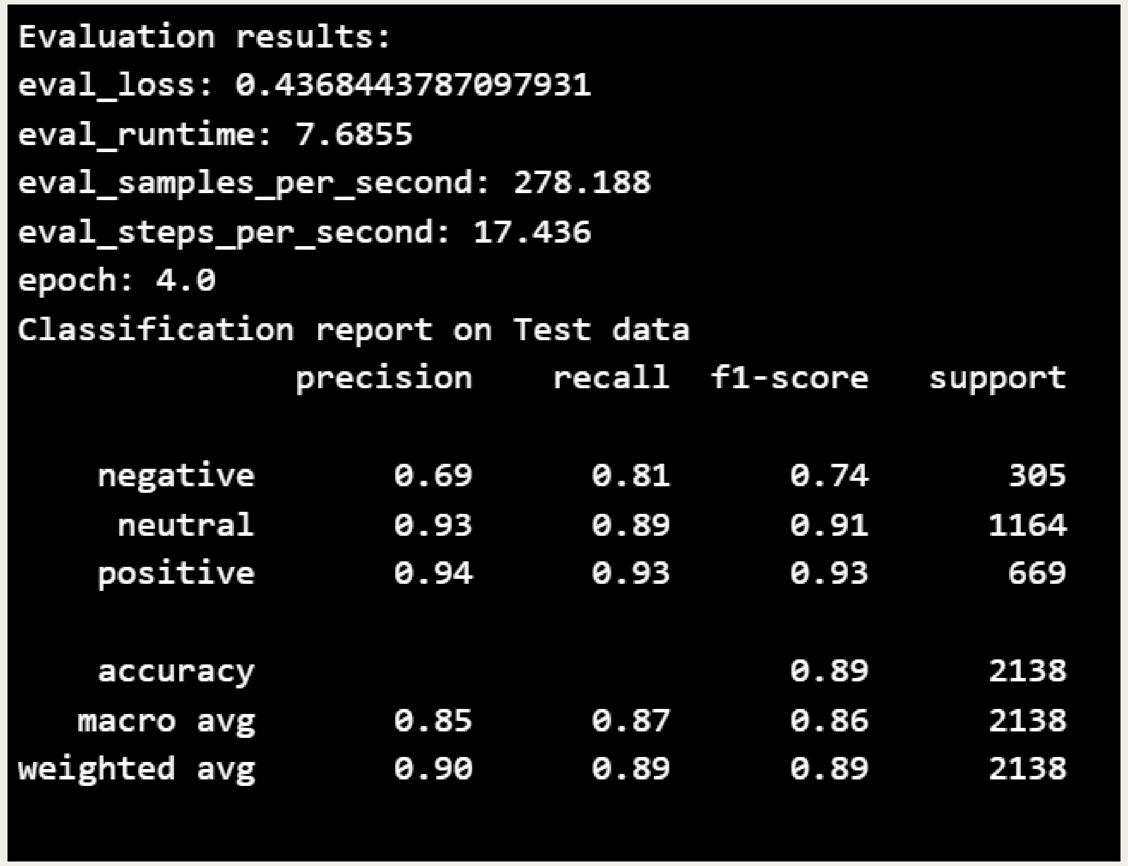
- FinBERT is a domain specific version of BERT(Bidirectional Encoder from Transformers. To understand about FinBERT we first need to understand BERT model as FinBERT is derived from BERT.
- FinBERT is specifically trained for predicting the sentiments of financial news.
- This domain specific training helps FinBERT to understand the financial jargons and provide with a more accurate and precise sentiments for the financial news.



```
Evaluation results:
eval_loss: 0.4368443787097931
eval_runtime: 7.6855
eval_samples_per_second: 278.188
eval_steps_per_second: 17.436
epoch: 4.0
Classification report on Test data
              precision    recall  f1-score   support

    negative       0.69      0.81      0.74       305
     neutral       0.93      0.89      0.91      1164
    positive       0.94      0.93      0.93       669

    accuracy                           0.89      2138
   macro avg       0.85      0.87      0.86      2138
weighted avg       0.90      0.89      0.89      2138
```

MODEL COMPARISON AND CONCLUSION:





```
Evaluation results:
eval_loss: 0.4368443787097931
eval_runtime: 7.6855
eval_samples_per_second: 278.188
eval_steps_per_second: 17.436
epoch: 4.0
Classification report on Test data
              precision    recall  f1-score   support

    negative       0.69      0.81      0.74       305
     neutral       0.93      0.89      0.91      1164
    positive       0.94      0.93      0.93       669

    accuracy                           0.89      2138
   macro avg       0.85      0.87      0.86      2138
weighted avg       0.90      0.89      0.89      2138
```

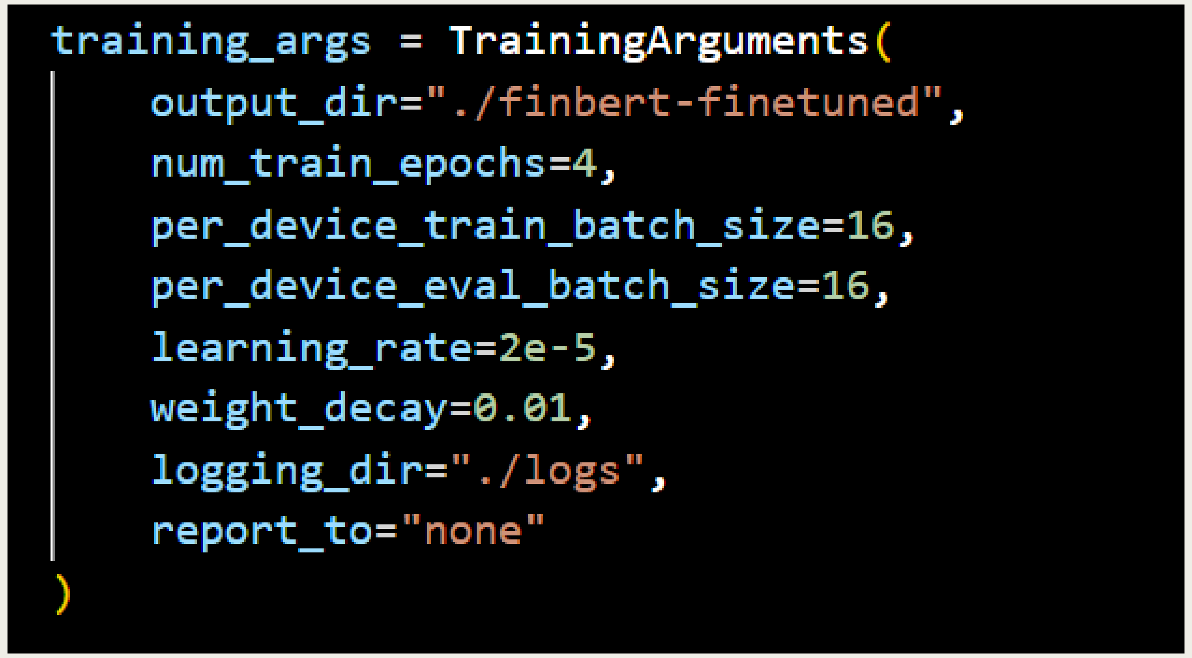
FinBERT Evaluation Results

From the above we can conclude that FinBERT is outperforming the other models and finetuning the FinBERT model can yield better performance and achive higher sentiment prediction accuracy. The finetuned FinBERT model is then used for providing real time financial sentiment strategy and backtesting.

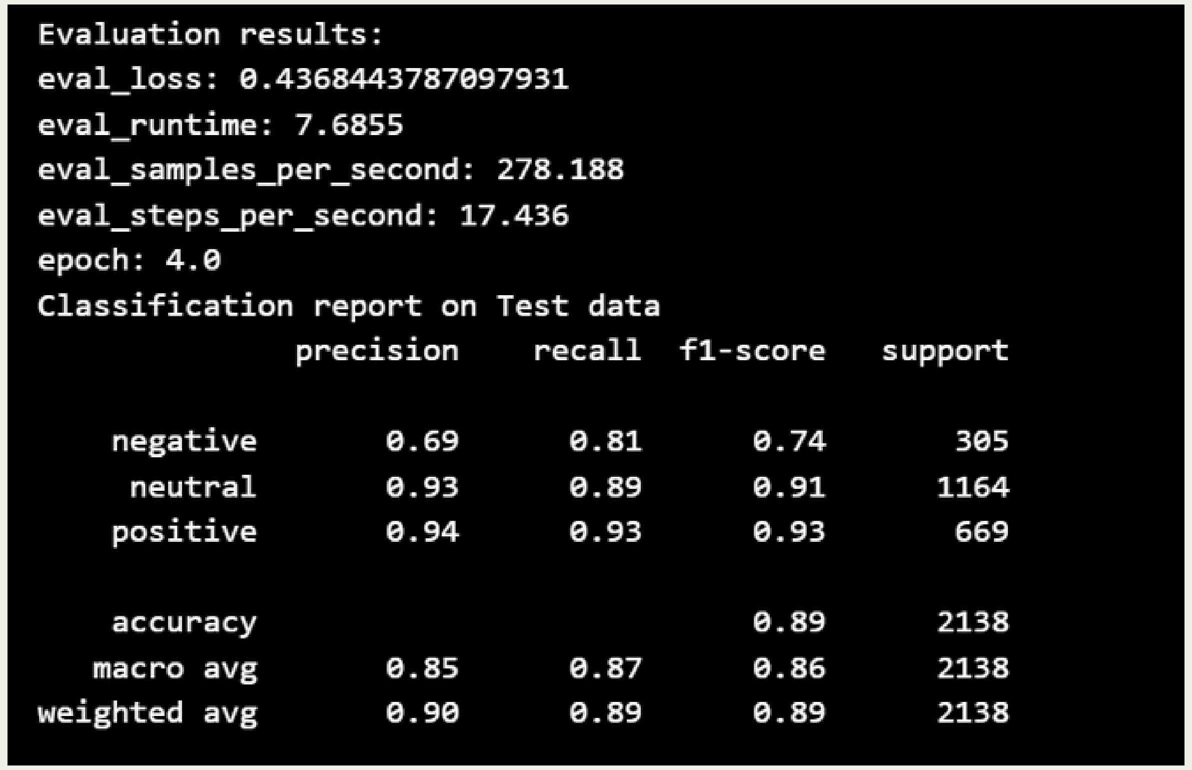
FINETUNED FINBERT

After initializing and finding the accuracy of FinBERT on the unseen test data, finetuning FinBERT was a essential part of the project. The parameters changes during finetuning of FinBERT were:

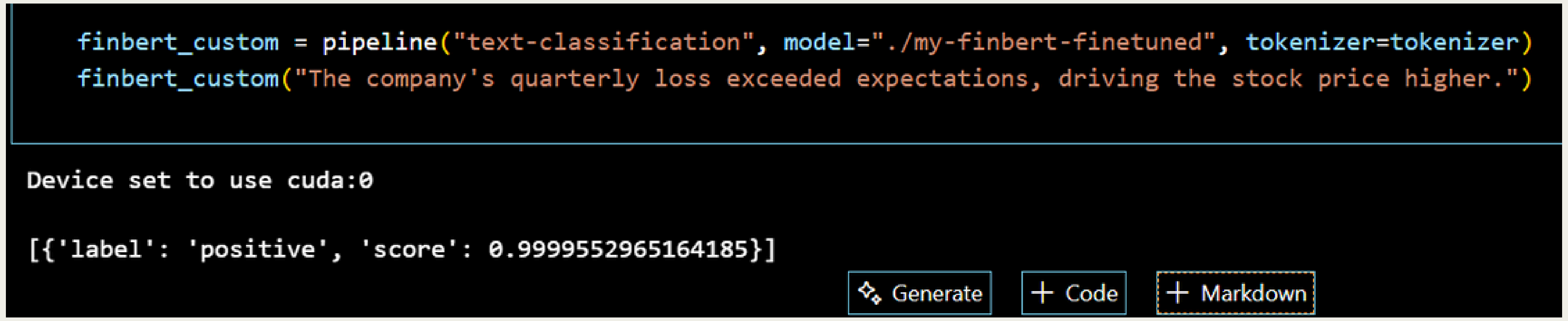
1. Epochs
2. Training batch size
3. Evaluation batch size
4. Learning Rate
5. Weight Decay



```
training_args = TrainingArguments(
    output_dir="./finbert-finetuned",
    num_train_epochs=4,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    learning_rate=2e-5,
    weight_decay=0.01,
    logging_dir="./logs",
    report_to="none"
)
```



```
Evaluation results:
eval_loss: 0.4368443787097931
eval_runtime: 7.6855
eval_samples_per_second: 278.188
eval_steps_per_second: 17.436
epoch: 4.0
Classification report on Test data
              precision    recall  f1-score   support

    negative       0.69      0.81      0.74       305
     neutral       0.93      0.89      0.91      1164
    positive       0.94      0.93      0.93       669

    accuracy                           0.89      2138
   macro avg       0.85      0.87      0.86      2138
weighted avg       0.90      0.89      0.89      2138
```



```
finbert_custom = pipeline("text-classification", model="./my-finbert-finetuned", tokenizer=tokenizer)
finbert_custom("The company's quarterly loss exceeded expectations, driving the stock price higher.")
```

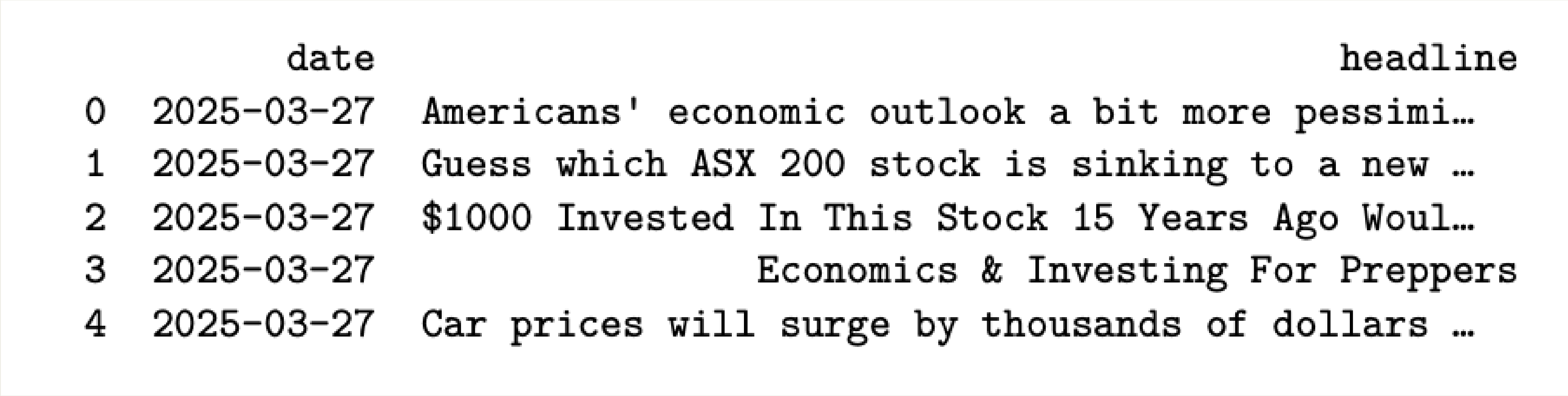
```
Device set to use cuda:0

[{'label': 'positive', 'score': 0.9999552965164185}]
```

Real-Time Market Simulation Using Sentiment-Driven Strategy

NEWS COLLECTION & SENTIMENT ANALYSIS

News fetching

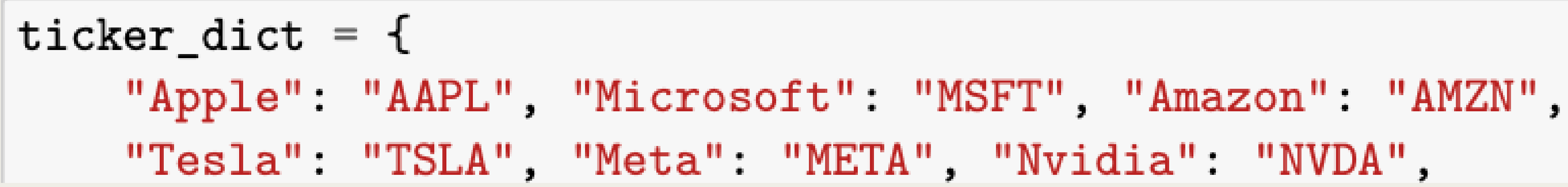
- Objective: Fetch real financial news for trading day - 1
- Source: NewsAPI
- Example of raw headlines



```
         date                                         headline
0  2025-03-27  Americans' economic outlook a bit more pessimi…
1  2025-03-27  Guess which ASX 200 stock is sinking to a new …
2  2025-03-27  $1000 Invested In This Stock 15 Years Ago Woul…
3  2025-03-27               Economics & Investing For Preppers
4  2025-03-27  Car prices will surge by thousands of dollars …
```

Ticker Mapping & Extraction

- Parse and identify tickers from headlines
- General news mapped to S&P 500 Index (SPY)



```
ticker_dict = {
    "Apple": "AAPL", "Microsoft": "MSFT", "Amazon": "AMZN",
    "Tesla": "TSLA", "Meta": "META", "Nvidia": "NVDA",
```

Sentiment Analysis

- Sentiment class (Positive, Neutral, Negative)
- Sentiment confidence score

TECHNICAL INDICATOR CALCULATION

RSI(Relative Strength Index)

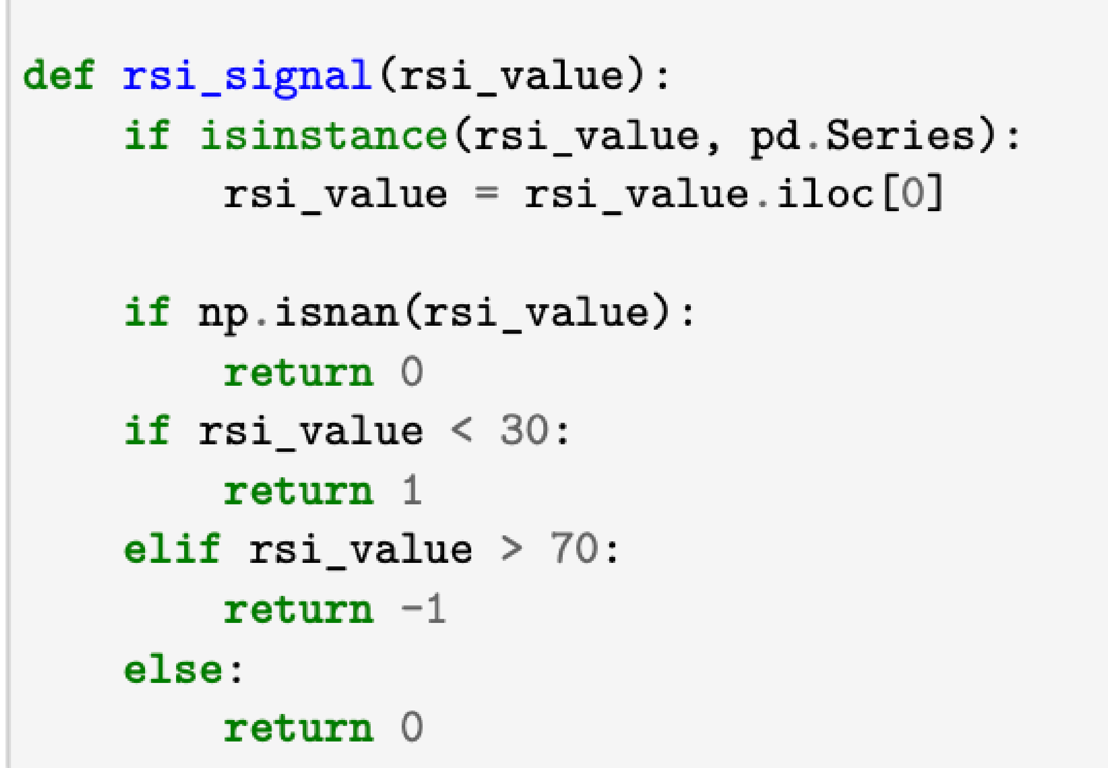
Formula:

RSI = 100 − [100 / (1 + RS)]

Where:

- RS = Average Gain / Average Loss
- Gain/Loss calculated over a specified period (commonly 14 days)

- RSI is a momentum oscillator that measures the speed and change of recent price movements.
- Helps identify overbought or oversold conditions in a stock.
- Values range from 0 to 100.



```python
def rsi_signal(rsi_value):
    if isinstance(rsi_value, pd.Series):
        rsi_value = rsi_value.iloc[0]

    if np.isnan(rsi_value):
        return 0
    if rsi_value < 30:
        return 1
    elif rsi_value > 70:
        return -1
    else:
        return 0
```

TECHNICAL INDICATOR CALCULATION

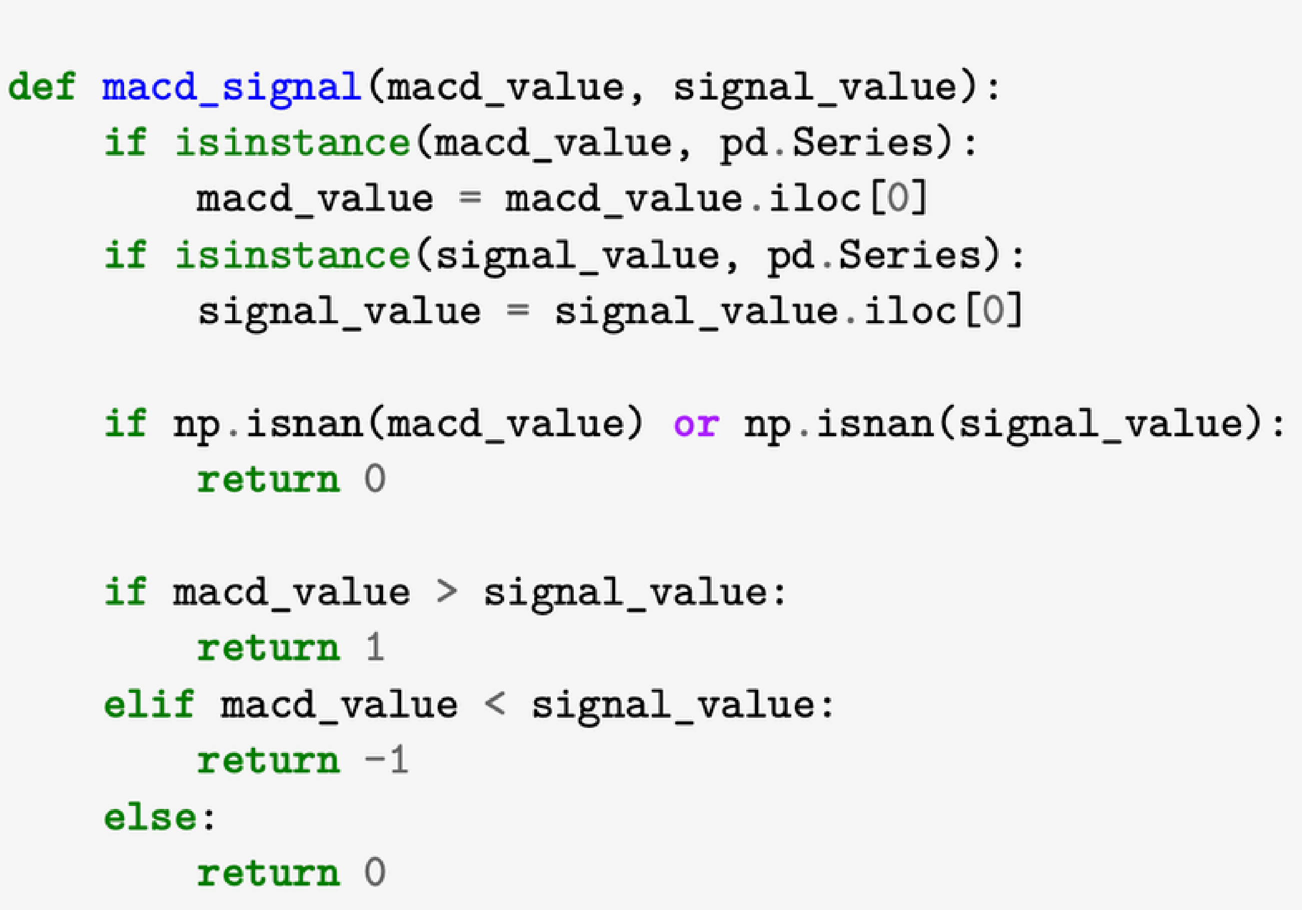
MACD(Moving Average Convergence Divergence)

Formula:

- MACD Line = EMA(12) − EMA(26)
- Signal Line = EMA(9) of MACD Line

- These lines are used to identify buy/sell crossover signals.

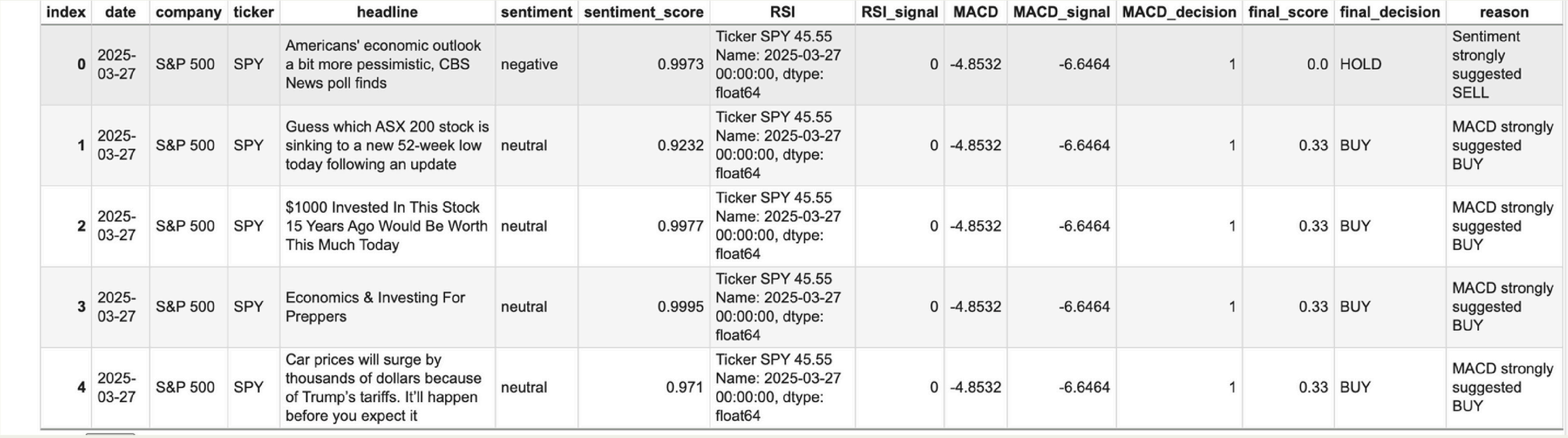
- MACD is a trend-following momentum indicator.
- Shows the relationship between two Exponential Moving Averages (EMAs) of closing prices.
- Used to spot changes in the strength, direction, momentum, and duration of a trend.



```python
def macd_signal(macd_value, signal_value):
    if isinstance(macd_value, pd.Series):
        macd_value = macd_value.iloc[0]
    if isinstance(signal_value, pd.Series):
        signal_value = signal_value.iloc[0]

    if np.isnan(macd_value) or np.isnan(signal_value):
        return 0

    if macd_value > signal_value:
        return 1
    elif macd_value < signal_value:
        return -1
    else:
        return 0
```

FINAL DATAFRAME

```python
rows.append({
    "date": date,
    "company": company,
    "ticker": ticker,
    "headline": headline,
    "sentiment": sentiment,
    "sentiment_score": round(sent_score, 4),
    "RSI": round(rsi_val, 2),
    "RSI_signal": rsi_action,
    "MACD": round(macd_val, 4),
    "MACD_signal": round(macd_sig, 4),
    "MACD_decision": macd_action,
    "final_score": round(final_score, 2),
    "final_decision": final_dec,
    "reason": reason
})
```



index	date	company	ticker	headline	sentiment	sentiment_score	RSI	RSI_signal	MACD	MACD_signal	MACD_decision	final_score	final_decision	reason
0	2025-03-27	S&P 500	SPY	Americans' economic outlook a bit more pessimistic, CBS News poll finds	negative	0.9973	Ticker SPY 45.55 Name: 2025-03-27 00:00:00, dtype: float64	0	-4.8532	-6.6464	1	0.0	HOLD	Sentiment strongly suggested SELL
1	2025-03-27	S&P 500	SPY	Guess which ASX 200 stock is sinking to a new 52-week low today following an update	neutral	0.9232	Ticker SPY 45.55 Name: 2025-03-27 00:00:00, dtype: float64	0	-4.8532	-6.6464	1	0.33	BUY	MACD strongly suggested BUY
2	2025-03-27	S&P 500	SPY	$1000 Invested In This Stock 15 Years Ago Would Be Worth This Much Today	neutral	0.9977	Ticker SPY 45.55 Name: 2025-03-27 00:00:00, dtype: float64	0	-4.8532	-6.6464	1	0.33	BUY	MACD strongly suggested BUY
3	2025-03-27	S&P 500	SPY	Economics & Investing For Preppers	neutral	0.9995	Ticker SPY 45.55 Name: 2025-03-27 00:00:00, dtype: float64	0	-4.8532	-6.6464	1	0.33	BUY	MACD strongly suggested BUY
4	2025-03-27	S&P 500	SPY	Car prices will surge by thousands of dollars because of Trump's tariffs. It'll happen before you expect it	neutral	0.971	Ticker SPY 45.55 Name: 2025-03-27 00:00:00, dtype: float64	0	-4.8532	-6.6464	1	0.33	BUY	MACD strongly suggested BUY

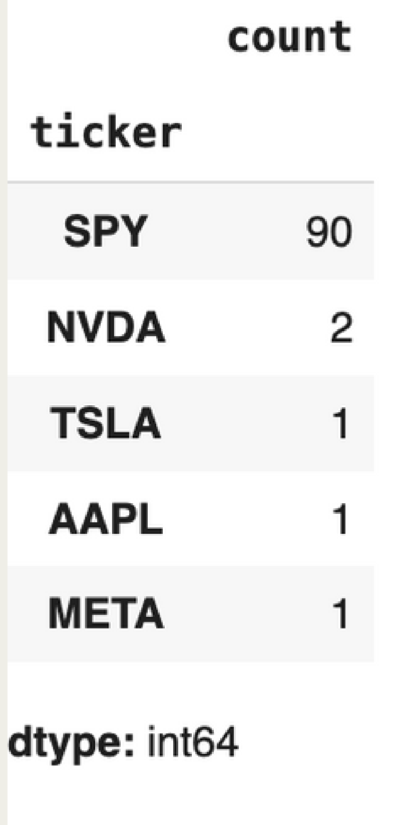
TICKER AGGREGATION AND EACH TICKER PLOT

Aggregated dataframe:

- Count multiple news items per ticker
- Average sentiment and final decisions
- Result: One signal per ticker per day

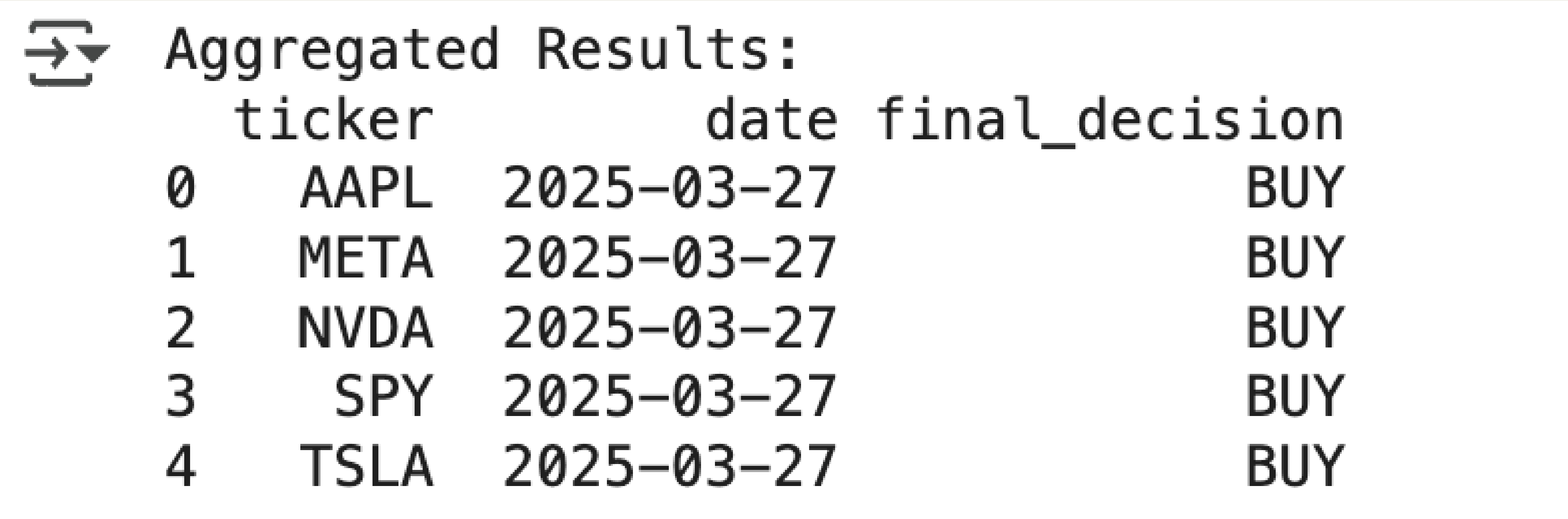
- If one has the highest count, use it
- If there's a tie, apply numeric mapping:
- BUY = +1, HOLD = 0, SELL = –1

- Take average:
- **> 0 - BUY**
- **< 0 - SELL**
- **= 0 - HOLD**



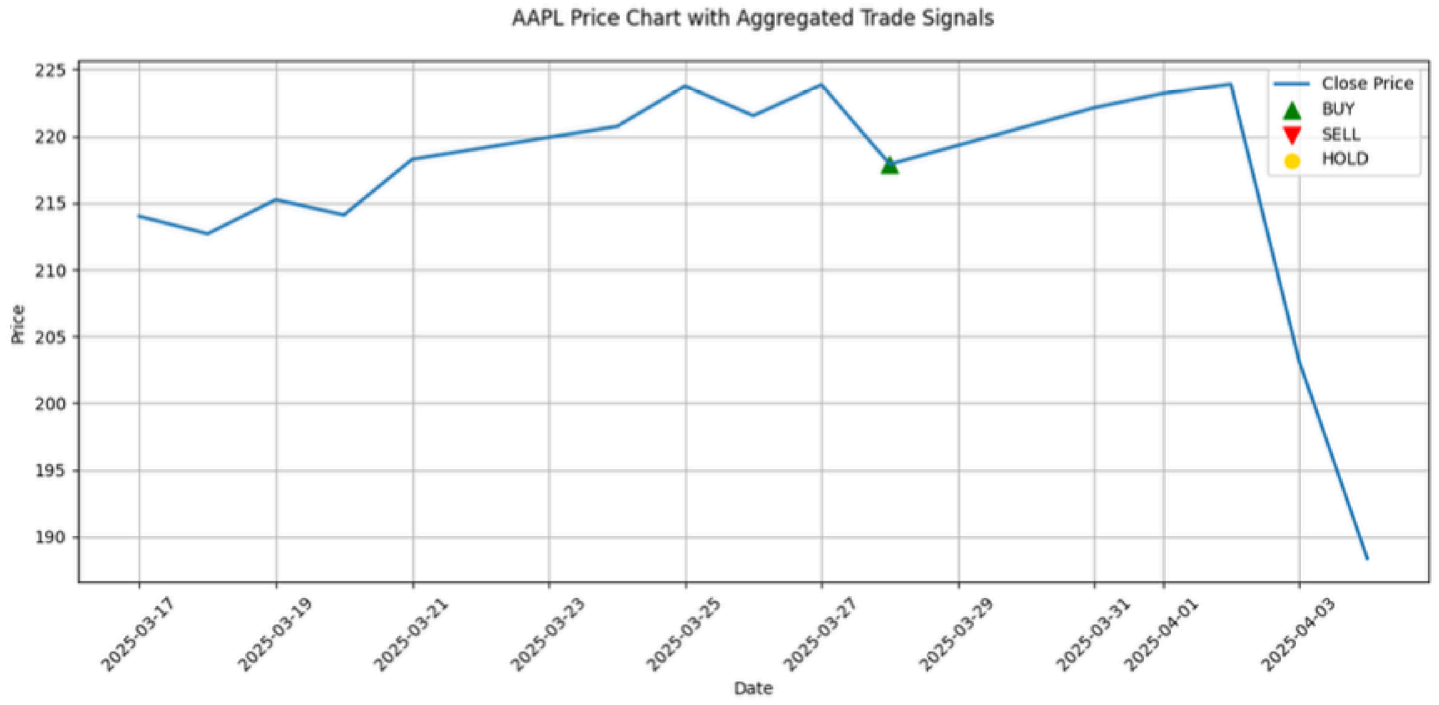
	count
ticker	
SPY	90
NVDA	2
TSLA	1
AAPL	1
META	1

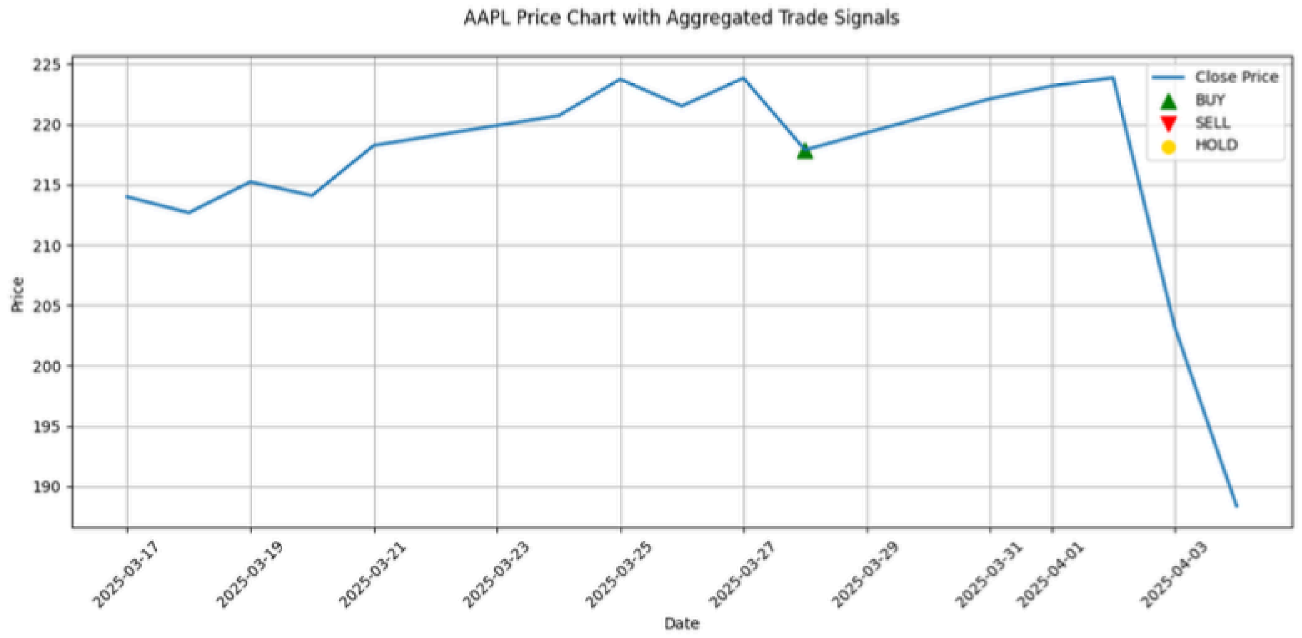
dtype: int64

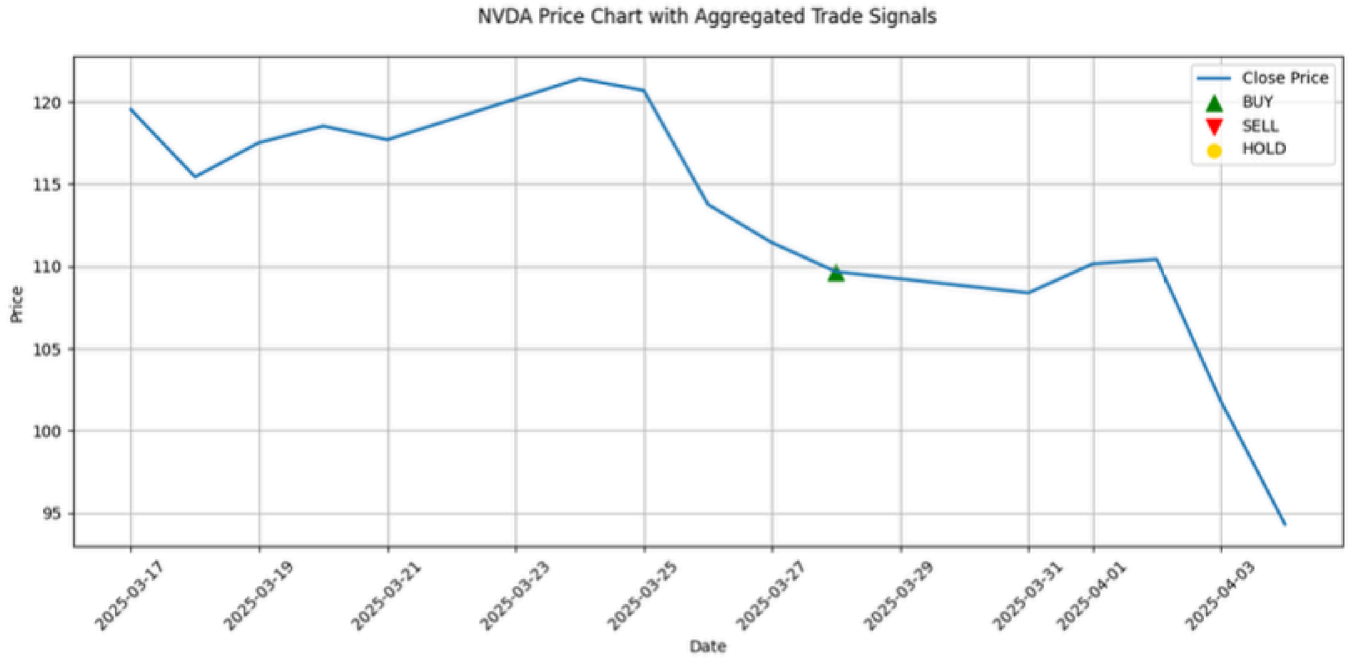


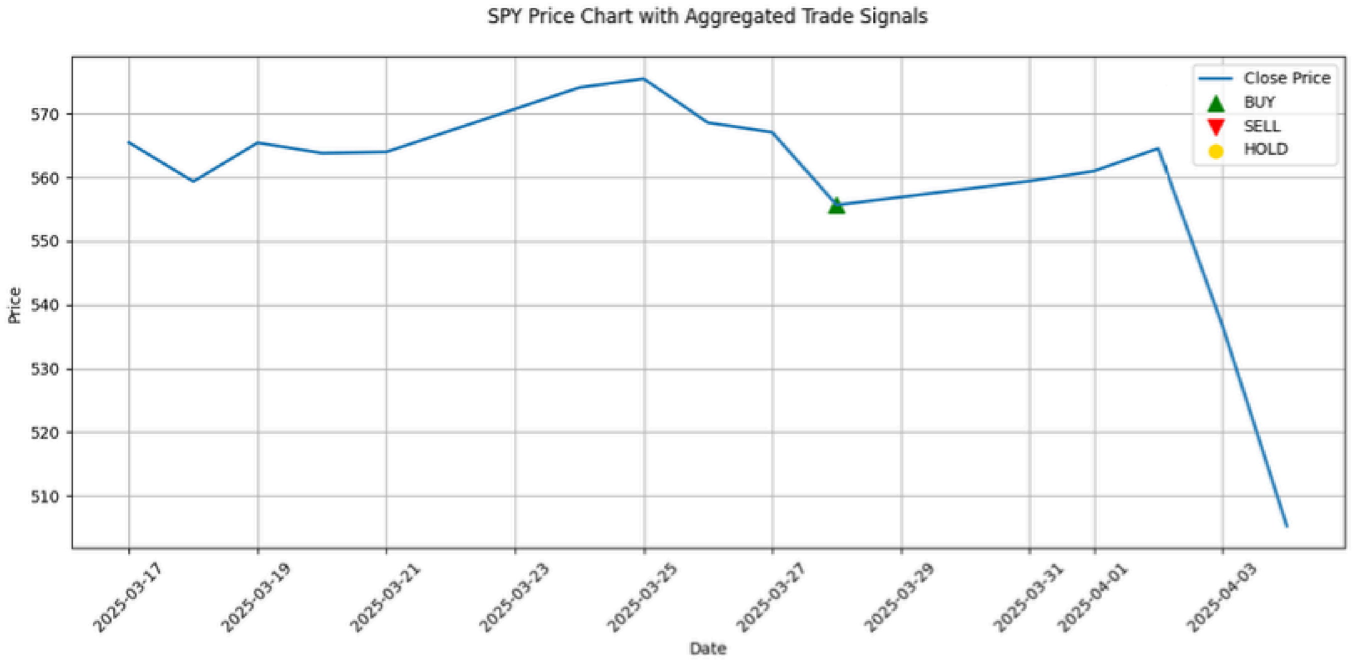
```
Aggregated Results:
  ticker        date final_decision
0   AAPL  2025-03-27            BUY
1   META  2025-03-27            BUY
2   NVDA  2025-03-27            BUY
3    SPY  2025-03-27            BUY
4   TSLA  2025-03-27            BUY
```

TICKER AGGREGATION AND EACH TICKER PLOT


AAPL Price Chart with Aggregated Trade Signals


AAPL Price Chart with Aggregated Trade Signals


NVDA Price Chart with Aggregated Trade Signals


SPY Price Chart with Aggregated Trade Signals

BACKTESTING

Portfolio Simulation:

Initial Setup

- Starting capital: $100,000
- Portfolio tracked with: cash, stock holdings and trade log

Portfolio Management

- Profit Tracking:
    - Realized Profit – from closed (sold) positions
    - Unrealized Profit – from currently held positions

Per-Ticker Tracking Includes(Log df):

- Ticker Name
- Buy price(opening price of bought day)
- Number of shares
- daily final decision
- closing price of that date
- daily profit
- overall profit on that stock

BACKTESTING

Trading Logic:

BUY Conditions

- Triggered only if final_decision = BUY
- Cash is equally divided across all BUY tickers of the day

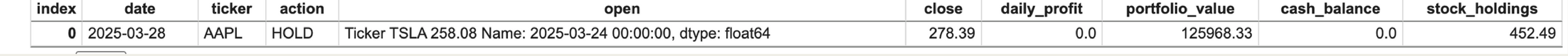
HOLD Conditions

- Continue holding till new signal comes for that ticker is HOLD
- No action is taken
- Position is re-evaluated daily

SELL Conditions

- Ticker is currently held AND New final_decision = SELL
- OR holding period reaches 5 trading days

All trades executed at next day's opening price

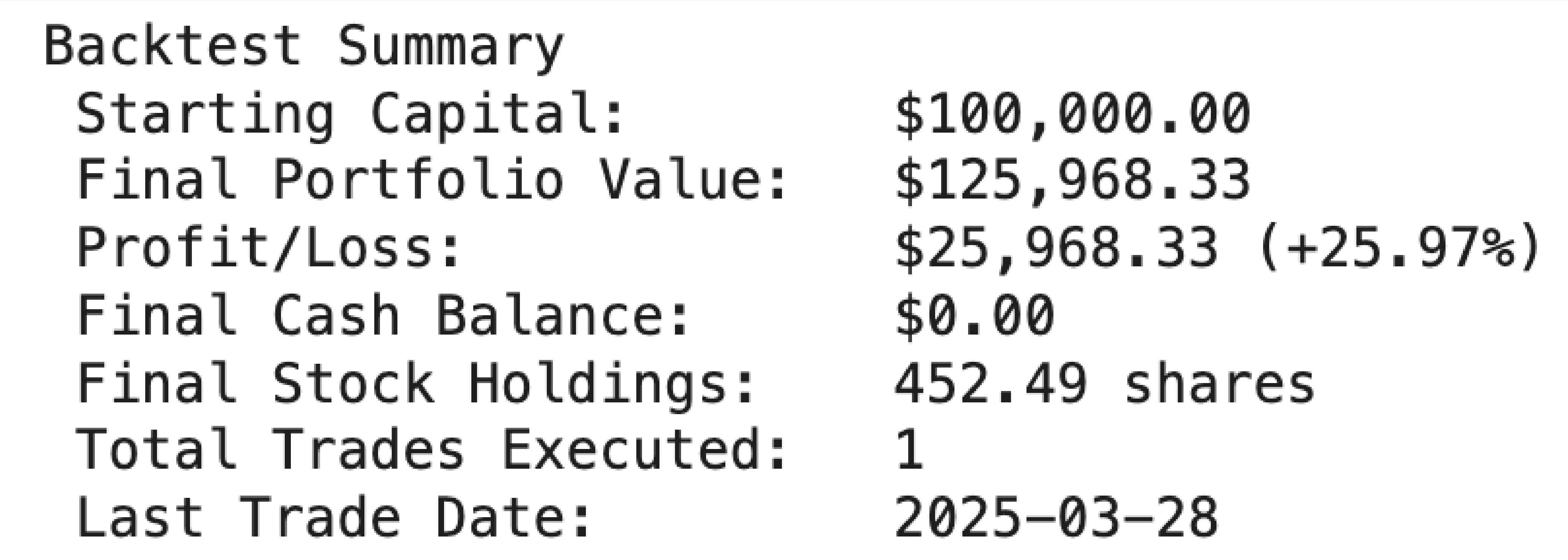


index	date	ticker	action	open	close	daily_profit	portfolio_value	cash_balance	stock_holdings
0	2025-03-28	AAPL	HOLD	Ticker TSLA 258.08 Name: 2025-03-24 00:00:00, dtype: float64	278.39	0.0	125968.33	0.0	452.49

BACKTESTING

Portfolio summary:

- Day starting Capital: $100,000.00
- Day Ending Portfolio Value: $125,968.33 (With Realised and Unrealised gain)
- Net Profit: $25,968.33 (+25.97%)
- Available Cash Balance: $0.00
- Total Number of Stock Holdings: 452.49 shares
- Daily Total Trades Executed: 1
- Last Trade Date: March 28, 2025



```
Backtest Summary
 Starting Capital:          $100,000.00
 Final Portfolio Value:     $125,968.33
 Profit/Loss:               $25,968.33 (+25.97%)
 Final Cash Balance:        $0.00
 Final Stock Holdings:      452.49 shares
 Total Trades Executed:     1
 Last Trade Date:           2025-03-28
```

Thank you!