National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Aashritha Venkataraman
Student ID: x23267356

School of Computing
National College of Ireland

Supervisor: Arjun chikkankod

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aashritha Venkataraman
Student ID:	x23267356
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Arjun chikkankod
Submission Due Date:	11/08/2025
Project Title:	Configuration Manual
Word Count:	1500
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aashritha Venkataraman
Date:	14th September 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aashritha Venkataraman
x23267356

1 Project Overview

This project implements a comparative analysis of zero-shot, few-shot, and chain-of-thought sentiment analysis techniques for financial market prediction using the DJIA dataset.Ensemble methods yield an accuracy of 63.33% in the system, which is 8.33 percent higher than in baseline performance.

Key Components

- **Base Model:** XGBoost 22 engineered features
- **Sentiment Analysis:** Zero-shot, Few-shot and chain-of-thought prompting technique
- **Model Integration FinBERT(To keep it domain specific)**
- **Ensemble Methods:** Weighted averaging, voting and meta-classification

2 System Requirements

2.1 Hardware Requirements

- **CPU:** 8GB RAM, a minimum of 4 cores
- **GPUs NVIDIA GPU (Recommended for FinBERT)**
- **Storage:** 10GB free space
- **Network:** High-speed internet for API calls

2.2 Software Dependencies

2.2.1 Core Libraries

- pandas≥1.5.0
- numpy≥1.21.0
- scikit-learn≥1.1.0
- xgboost≥1.6.0

2.2.2 NLP Libraries

- transformers≥4.20.0
- torch≥1.12.0
- textblob≥0.17.1

2.3 Visualization

- matplotlib≥3.5.0
- seaborn≥0.11.0
- wordcloud≥1.9.0

3 Data configuration

3.1 Primary Dataset:

Data source: `https://www.kaggle.com/datasets/lykin22/stock-headlines?select=DJIA_table.csv`

- top25DJIANews.csv

3.1.1 Columns:

  - Date: Trading date in DD/MM/YYYY format
  - Market direction (0=Down, 1=Up) label
  - Top1-Top25: Financial news headlines

- GPT Scores: combinedHeadlines_GPTscores.csv

3.1.2 Columns:

  - - date: Trading date
  - - zero: Zero-shotsentiment score
  - - few-shot: Few-shot sentiment score
  - - cot: Chain-of-thought sentiment score

- Parameters of Data Preprocessing
  - TEMPORAL_WINDOW = 3
  - TRAIN_TEST_SPLIT = 0.8
  - HEADLINE_CHAR_LIMIT = 2000
  - MIN_HEADLINE_LENGTH = 10

4 Model Configuration

4.1 XGBoost Hyperparameters

Parameter : Value
n_estimators : 100
max_depth : 4
learning_rate : 0.1
subsample : 0.9
colsample_bytree : 0.9
min_child_weight : 5
reg_alpha : 0.1
reg_lambda : 1.0
random_state : 42
verbosity : 0

4.2 Configuration of feature engineering

Listing 1: Feature Engineering Parameters

```
FEATURE_CONFIG = {
    'strong_negative_terms': [
        'crash', 'plunge', 'collapse', 'meltdown', 'panic',
        'crisis', 'disaster', 'emergency', 'terror', 'attack'
    ],
    'strong_positive_terms': [
        'surge', 'soar', 'boom', 'rally', 'breakthrough',
        'record', 'historic', 'milestone', 'achievement'
    ],
    'fed_policy_terms': [
        'fed', 'federal', 'reserve', 'interest', 'rate',
        'monetary', 'policy', 'stimulus', 'qe', 'taper'
    ],
    'economic_indicators': [
        'gdp', 'inflation', 'unemployment', 'jobs',
        'employment', 'retail', 'sales', 'housing'
    ]
}
```

4.3 Prompt Engineering Setup

4.3.1 Zero-Shot Configuration

Listing 2: Zero-Shot Prompt Configuration

```
ZERO_SHOT_PROMPT = """
Analyze this financial news for sentiment with focus on market
   direction impact.
Provide a numerical score from -1 (very negative market impact)
   to +1 (very positive),
considering likely influences on stock-market direction and
   investor behavior.

Consider factors such as:
- Company performance indicators
- Economic policy signals
- Market volatility
- Investor confidence

News content: {aggregated_headlines}
Respond with numerical score only.
"""
```

4.3.2 Few-Shot Configuration

Listing 3: Few-Shot Examples Configuration

```
FEW_SHOT_EXAMPLES = [
    {
        "headline": "Apple reports quarterly earnings of $2.46
           per share, beating analyst estimates",
        "score": 0.7,
        "reasoning": "Strong positive: earnings beat, revenue
           growth signals healthy performance"
    },
    {
        "headline": "Federal Reserve announces potential interest
            rate cuts due to economic concerns",
        "score": -0.6,
        "reasoning": "Negative: economic weakness signals, policy
            uncertainty"
    }
]
```

4.3.3 Chain-of-Thought Configuration

Listing 4: Chain-of-Thought Structure

```
COT_STRUCTURE = {
    "step_1": "Financial Information Identification",
    "step_2": "Market Impact Assessment",
    "step_3": "Sentiment Direction Analysis",
    "step_4": "Quantitative Integration"
}
```

5 Pipeline Configuration

5.1 Manual Process of Generate of GPT Score:

This project used ChatGPT web application manually, not API calls.

Process Flow:

- Data Preparation: Took a rolling average of 3 days market value
- Manual Prompting: Input of Heading is fed into ChatGPT web interface in a structured manner to ensure uniformity
- Score Collection: the scores between the bounds of -1 and 1 are inserted manually in the form of numbers
- CSV: The creation of scores is made into xg boost 300rows that generate gpt.csv.
- Integration: GPT metrics used as a combination with baseline characteristics in training the model

5.2 Temporal Aggregation Settings

Listing 5: Temporal Aggregation Configuration

```
AGGREGATION_CONFIG = {
    'window_sizes': [1, 3, 5],  # Days
    'optimal_window': 3,        # Best performing
    'aggregation_method': 'chronological',
    'information_leakage_prevention': True
}
```

5.3 Performance Parameters

5.3.1 Expected Performance Metrics

Listing 6: Expected Model Performance

```
Baseline Model: 55.0% accuracy
Traditional Sentiment: 48.3% accuracy
Zero-shot: 58.3% accuracy
Few-shot: 60.0% accuracy
Chain-of-thought: 55.0% accuracy
Ensemble (Best): 63.33% accuracy
```

5.3.2 Optimization Targets

Listing 7: Performance Optimization Targets

```
PERFORMANCE_TARGETS = {
    'accuracy_threshold': 0.55,
    'improvement_target': 0.05,  # 5% improvement
    'processing_time_limit': 300,  # seconds per batch
    'memory_limit': '8GB'
}
```

6 Troubleshooting

6.1 Common Issues and Solutions

1. **Memory Errors**

   Problem: Out of memory during FinBERT processing

   Solution:

   ```
   # Reduce batch size
   FINBERT_BATCH_SIZE = 16
   # Enable gradient checkpointing
   torch.backends.cudnn.benchmark = False
   ```

2. **Data Format Issues**

   Problem: Date parsing errors

   Solution:

   ```
   # Ensure consistent date format
   df['Date'] = pd.to_datetime(df['Date'], dayfirst=True, errors
      ='coerce')
   ```

3. **Feature Scaling Problems**

   Problem: Poor model convergence

   Solution:

   ```
   # Use StandardScaler consistently
   scaler = StandardScaler()
   X_scaled = scaler.fit_transform(X)
   ```

6.2 Performance Optimization

6.2.1 For Large Datasets

```
# Chunked processing
CHUNK_SIZE = 1000
for chunk in pd.read_csv(file, chunksize=CHUNK_SIZE):
    process_chunk(chunk)
```

7 Conclusion

This is a configuration guide that gives a complete guideline on implementing the financial market prediction system. Its reported setup managed to combine manual sentiment analysis of ChatGPT with conventional feature engineering and XGBoost implementation, proving that formal prompt engineering can even improve the financial forecast performance by more than 8.33% compared to the baseline methods.

After reading this configuration guide, the researchers and practitioners will be able to reproduce the research results and use the identified proven framework to improve the analysis of financial sentiments. A structured analysis of hyperparameter tuning, feature deletion and feature selection, and ensemble learning offers a substantial basis not only in the context of academic research but also in the financial application field, connecting the theoretical presentations of the topic of prompt engineering and its concrete realization in the work of the predictive system in the financial market.