Approach Used

1. **Data Augmentation:**
    - **Synonym Replacement:** Introduced variation by replacing words with their synonyms from **WordNet**.
    - **Balancing Data:** Used augmentation to ensure an equal number of **positive**, **negative**, and **neutral** labels.
2. **Text Preprocessing:**
    - Converted text to **lowercase**.
    - Removed **special characters and extra spaces**.
    - Encoded sentiment labels using LabelEncoder.
3. **Feature Engineering:**
    - Applied **TF-IDF vectorization** (max_features=10,000) to convert text into numerical features.
4. **Model Training & Evaluation:**
    - Split the dataset into **80% training and 20% testing**.
    - Trained a **Logistic Regression model** for classification.
    - Evaluated using **accuracy, precision, recall, and F1-score**.

Challenges Faced

- **Data Imbalance:** The dataset had an uneven distribution of labels, addressed through **data augmentation**.
- **Synonym Replacement Limitations:** Some synonyms from WordNet **did not fit the financial context**, potentially affecting accuracy.
- **Feature Representation:** TF-IDF is effective but does not capture deep semantic relationships.

Model Performance & Improvements

1. **Performance Metrics:**
    - **Accuracy: 86.46%**
    - **Class-wise F1-Scores:**
        - **Neutral:** 93%
        - **Positive:** 84%
        - **Negative:** 82%
2. **Observations:**
    - The model performs **well for neutral sentiments**, but **positive and negative sentiment classification could be improved**.
    - Precision and recall for **negative sentiment** (82%) indicate room for enhancement.
3. **Possible Improvements:**
    - **Use Word Embeddings:** Implement **Word2Vec** or **FinBERT** for better financial context understanding.

- **Fine-tune Logistic Regression:** Adjust **regularization (C parameter)** for better performance.
- **Experiment with Deep Learning Models:** LSTMs or **transformers like FinBERT** may improve sentiment classification.