StockSense-RAG (News Impact on Stocks)

Sachin Kumar

***Abstract*— StockSense-RAG is a proof-of-concept system that demonstrates how combining semantic retrieval of financial news with quantitative price modeling can enhance stock equity analysis. The system collects news related to specific tickers, preprocesses and embeds the textual content into a vector database (ChromaDB), and enables natural language queries through a RAG pipeline. Concurrently, historical price data is retrieved (e.g., via yfinance, newsapi), features are engineered, and a baseline machine learning model (Random Forest classifier) generates short-term directional price predictions. The output — for a given user query — includes a narrative explanation based on relevant news, a predicted price movement, and supporting articles. This integrated methodology aims to offer investors more informed, context-aware insights than what could be achieved with purely quantitative or purely sentiment-based approaches.**

I. INTRODUCTION

Financial markets are characterized by volatility driven not only by historical prices but also by real-world events and news sentiment. Traditional forecasting tools such as ARIMA, regression models, and technical-indicator–based ML models fail to accurately incorporate unstructured textual information. This limitation becomes evident during news-driven market movements where numerical patterns alone offer insufficient predictive power.

Recent advances in Natural Language Processing (NLP), semantic search, and retrieval-enhanced transformers have introduced new opportunities for combining structured and unstructured financial data. Retrieval-Augmented Generation (RAG) allows large language models to ground responses in factual retrieved documents, improving interpretability and reducing hallucination. When combined with a machine-learning classifier trained on engineered financial features, a dual-stage system can both predict and explain stock movement.

StockSense-RAG is designed to address this need by providing a modular pipeline that automatically collects stock data, ingests relevant news, embeds and indexes textual content, and offers real-time RAG-based reasoning paired with ML-driven price forecasting.

II. RELATED WORK

Recent research has explored combining sentiment with numerical indicators for market prediction. Transformer-based models such as BERT, FinBERT, and RoBERTa have been used to classify news sentiment and correlate it with price swings. Vector-database-based retrieval (e.g., ChromaDB, Pinecone) has also gained importance for contextual question answering.

Hybrid ML architectures have also shown improvements in stock prediction accuracy by integrating textual features, though their interpretability remains limited. Retrieval-Augmented Generation is relatively new in financial analytics, offering interpretability advantages by grounding responses in actual market documents.

StockSense-RAG builds upon these findings by unifying:

- Semantic retrieval of financial news,
- LLM-driven contextual reasoning, and
- Supervised learning for quantitative prediction, in a single cohesive architecture.

III. SYSTEM ARCHITECTURE

The system architecture consists of six primary modules:

A. Data Ingestion Layer

- Retrieves historical OHLCV (Open, High, Low, Close, Volume) data using APIs such as yfinance, newsapi.
- Collects news articles using NewsAPI or locally stored datasets.
- Stores fetched data in project directories for reproducibility.

B. Preprocessing Module

- Cleans and tokenizes text (stopword removal, lowercasing, punctuation handling).
- Handles missing data in stock price series.
- Prepares structured and unstructured datasets for downstream tasks.

C. Embedding & Vector Database

- Converts news text into high-dimensional embeddings using SentenceTransformers.
- Indexes embeddings in ChromaDB for semantic similarity search.
- Enables top-k retrieval for user-specific queries.

D. Machine Learning Prediction Module

- Performs feature engineering on historical stock data: Daily returns, moving averages, Price momentum, Volatility indicators
- Trains a Random Forest classifier to predict next-day price direction.

E. RAG Query Engine

- Accepts natural language questions from the user.
- Converts them into embeddings.
- Retrieves relevant news from ChromaDB.
- Sends query + retrieved evidence to an LLM to generate a contextual explanation

F. Output Layer

- Explanation based on recent news
- Predicted price movement
- Supporting articles and metadata

IV. METHODOLOGY

A. Quantitative Data Modeling

OHLCV data is converted into features using time-series engineering techniques. A supervised learning problem is formulated to classify next-day direction.

B. Textual Modeling with Embeddings

News is embedded using transformer-based models to capture semantic meaning. These embeddings allow similarity-based ranking of documents.

C. Random Forest Classifier

Random Forest is chosen because:

- It handles nonlinearity.
- Works well on tabular data.
- Reduces overfitting through ensemble averaging.

D. Retrieval-Augmented Generation Integration

RAG retrieves top-k relevant documents and injects them into the LLM prompt, grounding the final answer in factual evidence.

V. RESULTS

Experiments show:

- The Random Forest classifier achieves reasonable performance on short-term directional tasks.
- The RAG system successfully retrieves relevant news based on semantic similarity.
- LLM-generated responses are significantly more interpretable than standard ML outputs.
- Integrating textual evidence improves the credibility and explainability of predictions.

While numerical prediction accuracy varies by ticker, the overall user experience is improved due to explanation-rich outputs.

VI. DISCUSSION

Strengths of the system include:

- Interpretability: Predictions are backed by real news events.
- Modularity: Components (ML, embeddings, LLM) can be upgraded independently.
- Practicality: Provides actionable insights in a human-friendly format.

Limitations include:

- Dependence on news availability.
- No deep sentiment scoring or multi-day forecasting yet.
- Accuracy depends on the choice of embedding and ML models.

VII. CONCLUSIONS

This project demonstrates the effectiveness of combining machine-learning–based forecasting with retrieval-based natural language reasoning. StockSense-RAG enhances traditional stock prediction with contextual explanations grounded in real news. Future improvements may include transformer-based time-series prediction, sentiment quantification, and reinforcement learning for multi-step forecasting.

APPENDIX

A. Repository Structure

```
StockSense-RAG/
 data/
 vectordb/
 models/
 src/
    ingestion/
    embeddings/
    rag/
    ml/
 api/
 README.md
```

B. Dependencies

- Python 3.10+
- ChromaDB
- SentenceTransformers
- FastAPI
- Scikit-learn
- NewsAPI
- OpenAI API

ACKNOWLEDGMENT

I would like to show my sincere gratitude to my Prof. Shafkat Islam for his insightful suggestions and continuous support throughout this project. This work would not be possible without his guidance and motivation.

REFERENCES

[1] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," NAACL, 2019.
[2] L. Breiman, "Random Forests," Machine Learning Journal, 2001.
[3] ChromaDB Documentation.
[4] NewsAPI Documentation.
[5] OpenAI GPT Developer Guide.