**PROJECT TITLE:** StockOracle: AI-Powered Stock Prediction & Forecasting System

**PROJECT DURATION:** January 2025 – March 2025  
  
  
  
**SUBMITTED BY:** Aritro Dutta

Lopamudra Mukherjee

Neeladri Bandopadhyay

Harsh Kumar Singh

**SUBMITTED TO:** Dr. Ambika Prasad Mishra

**TABLE OF CONTENTS**

1. Abstract
2. Introduction
3. Objectives
4. System Overview
5. System Architecture and Methodology

5.1) Data Collection and Preprocessing

5.2) Technical Indicator Calculation

5.3) Textual Data Generation and Sentiment Extraction

5.4) Topic Modelling with BERTopic

5.5) Forecasting and Sentiment Integration

1. Implementation Details

6.1) Tools and Libraries

6.2) Code Organization and Workflow

6.3) Data Storage and File Management

1. Experimental Setup and Evaluation

7.1) Data Scope and Acquisition

7.2) Performance Metrics and Validation

7.3) Testing Strategies and Backtesting

1. Results and Discussion

8.1) Data Processing and Quality

8.2) Indicator Reliability and Market Correlation

8.3) Topic Modelling Insights

8.4) Forecasting Accuracy and Decision Signals

8.5) Overall Impact and Use Case Scenarios

1. Challenges and Limitations
2. Future Enhancements
3. Conclusion
4. References

**ABSTRACT**

The project delves in the development of an AI-powered system designed for comprehensive stock market analysis by integrating quantitative and qualitative techniques. It collects five years of stock data for 10 companies, computes technical indicators like SMA and RSI, and enriches the dataset with textual summaries and sentiment scores. Leveraging BERTopic for topic modelling and Prophet for time-series forecasting, it uncovers underlying market trends and provides actionable investment signals. This multi-layered approach supports informed decision-making by combining rigorous data analysis with advanced natural language processing to capture market sentiment and emerging trends. Overall, the system empowers strategic investing.

**INTRODUCTION**

In today’s rapidly evolving financial markets, the ability to analyse vast amounts of data quickly and accurately is paramount. Traditional methods often rely solely on historical numerical data, but market dynamics also hinge on qualitative aspects such as investor sentiment and emerging trends.

StockOracle addresses these challenges by integrating multiple analytical layers:

* **Quantitative Analysis:** Utilizing historical stock data, computing technical indicators (e.g., SMA, RSI), and forecasting future trends with statistical models.
* **Qualitative Analysis:** Generating textual data from market movements, extracting sentiment, and employing topic modelling (BERTopic) to uncover underlying themes in financial texts.
* **Integrated Decision Making:** Combining these approaches to deliver clear BUY, SELL, or HOLD signals based on both data-driven and sentiment-based insights.

This multifaceted approach ensures a comprehensive evaluation of market conditions, setting the stage for more accurate and informed investment strategies.

**OBJECTIVES**

The project was designed with the following key objectives:

* **Data Acquisition:** Collect extensive historical stock data covering a 5-year period for 10 companies using the yfinance API.
* **Technical Analysis:** Calculate essential technical indicators including the 50-day and 200-day Simple Moving Averages (SMA\_50, SMA\_200) and the Relative Strength Index (RSI) using the pandas\_ta library.
* **Textual Data and Sentiment Analysis:** Generate descriptive textual data and corresponding sentiment scores from the processed numerical stock data, thereby providing a qualitative layer of analysis.
* **Topic Modelling:** Apply the BERTopic model to the refined textual dataset to identify latent topics and trends within the financial discourse.
* **Forecasting:** Implement time-series forecasting using Prophet to predict future stock price movements, integrating both historical trends and sentiment indicators.
* **Integration and Decision Support:** Combine all the analytical outputs into a cohesive system that provides actionable recommendations for market actions (BUY, SELL, HOLD).
* **Documentation and Reproducibility:** Ensure that every stage of the project is thoroughly documented and reproducible for further research and development.

**PROJECT OVERVIEW**

StockOracle is structured as a comprehensive, multi-stage analytical pipeline, comprising:

* **Stage 1 – Data Collection and Preprocessing:**
  + Raw stock data is acquired using yfinance and undergoes rigorous cleaning and normalization.
  + The resulting dataset, stored as stock\_data\_5\_years.csv, forms the basis for further analysis.
* **Stage 2 – Technical Analysis:**
  + Calculation of technical indicators such as SMA and RSI provides insight into market momentum and trend behaviour.
* **Stage 3 – Textual Data Generation and Sentiment Analysis:**
  + The system enriches the numerical dataset by generating textual narratives that describe market conditions and extracting sentiment scores.
  + The outcome is a refined dataset (refined\_textual\_data.csv) that blends quantitative and qualitative metrics.
* **Stage 4 – Topic Modelling:**
  + BERTopic is applied to the textual dataset to uncover latent topics and themes, offering deeper insights into the factors influencing market trends.
  + The topic distributions are visualized and stored in Topic\_Visualization.html.
* **Stage 5 – Forecasting and Decision Making:**
  + Time-series forecasting models predict future stock trends based on historical data and integrated sentiment analysis.
  + The final output includes actionable signals (BUY, SELL, HOLD) that guide investment decisions.

**SYSTEM ARCHITECTURE & METHODOLOGY**

**5.1) Data Collection and Preprocessing**

* **Data Source:**
  + Stock data for 10 companies is fetched using the yfinance API.
  + The time period spans from January 17, 2020 to January 16, 2025, providing a substantial dataset for robust analysis.
* **Data Cleaning:**
  + Raw data is checked for missing values, duplicates, and anomalies.
  + Data normalization techniques are applied to ensure consistency across different data points.
* **Storage:** The cleaned data is stored in stock\_data\_5\_years.csv and forms the primary dataset for subsequent processing.
* **Automation:** Scripts are written in Python to automate the data collection and cleaning process, ensuring reproducibility.

**5.2) Technical Indicator Calculation**

* **Indicators Computed:**
  + **SMA\_50 and SMA\_200:** Calculated using moving average formulas over 50-day and 200-day windows to capture short-term and long-term trends.
  + **RSI:** Computed to assess momentum and potential overbought or oversold conditions.
* **Libraries Used:** The pandas\_ta library is employed to efficiently compute these technical indicators.
* **Validation:** The computed indicators are validated against standard benchmarks to ensure accuracy.

**5.3) Textual Data Generation and Sentiment Extraction**

* **Text Generation:** Transform numerical stock data into descriptive textual summaries that highlight key market movements and trends.
* **Sentiment Analysis:**
  + Natural Language Processing (NLP) techniques are applied to extract sentiment scores from the generated texts.
  + These sentiment scores are crucial for providing a qualitative overlay to the quantitative analysis.
* **Output Dataset:** The enriched dataset is saved as refined\_textual\_data.csv, which serves as the input for topic modelling.

**5.4) Topic Modelling with BERTopic**

* **Model Overview:** BERTopic leverages transformer-based embeddings to capture semantic information from the textual data.
* **Process:**
  + The refined textual dataset is fed into the BERTopic model.
  + The model identifies clusters of related words and phrases, representing underlying topics in the financial domain.
* **Visualization:** Topic distributions and key representative words are visualized and outputted as an interactive HTML file (Topic\_Visualization.html).
* **Interpretation:** These topics provide insight into prevailing market sentiments and emerging themes, such as regulatory changes, economic shifts, or investor sentiment trends.

**5.5) Forecasting and Sentiment Integration**

* **Time-Series Forecasting:** Prophet is utilized for time-series forecasting, which models historical trends and predicts future stock price movements.
* **Integration with Sentiment:**
  + The forecasting model incorporates sentiment data to refine predictions.
  + This integration ensures that both quantitative trends and qualitative signals influence the final output.
* **Decision Framework:**
  + Forecasted trends are translated into actionable recommendations:
    - BUY: When the forecast indicates a significant upward trend.
    - SELL: When the forecast indicates a downward trend beyond a set threshold.
    - HOLD: When changes are minimal or uncertain.
* **Implementation:** The final integrated analysis is implemented in the script Sentiment\_Analysis\_and\_Forecasting.py.

**IMPLEMENTATION DETAILS**

**6.1) Tools and Libraries**

* **python:** The primary language used for all scripting and development.
* **yfinance:** For fetching historical stock data from online sources.
* **pandas and pandas\_ta:** Used for data manipulation and the computation of technical indicators.
* **bertopic:** Applied for advanced topic modelling on the textual data.
* **prophet:** Utilized for robust time-series forecasting.
* **jupyter notebooks:** Multiple notebooks (Stock\_Prediction.ipynb, Textual\_Data.ipynb, Topic\_Representation.ipynb) are used for exploratory analysis and prototyping.
* **Additional Libraries:** Libraries such as numPy, matplotlib, and seaborn may be used for data visualization and statistical analysis.

**6.2) Code Organization and Workflow**

* **Modular Design:**
  + The project is divided into clearly defined modules:
    - Data collection and preprocessing.
    - Technical indicator calculation.
    - Textual data generation and sentiment analysis.
    - Topic modelling.
    - Forecasting and final decision integration.
* **File Organization:**
  + **Notebooks:**
    - Stock\_Prediction.ipynb: For data collection and indicator computation.
    - Textual\_Data.ipynb: For generating and refining textual data.
    - Topic\_Representation.ipynb: For applying BERTopic and visualizing topics.
  + **Scripts:** Sentiment\_Analysis\_and\_Forecasting.py: For integrating forecasting and sentiment analysis.
  + **Data Files:**
    - stock\_data\_5\_years.csv: The primary dataset with technical indicators.
    - refined\_textual\_data.csv: The enriched textual and sentiment dataset.
  + **Visualizations:** Topic\_Visualization.html: Interactive output of the topic modelling process.

**6.3) Data Storage and File Management**

* **Data Versioning:** Each data processing step outputs a versioned file, ensuring reproducibility and traceability.
* **File Organization:** A structured directory layout separates raw data, processed data, notebooks, scripts, and visual outputs.
* **Backup and Documentation:** Regular backups and version control (via Git) are maintained to track changes across the project lifecycle.

**EXPERIMENTAL SETUP & EVALUATION**

**7.1) Data Scope and Acquisition**

* **Data Range:** The dataset encompasses 5 years of stock data, ensuring sufficient historical context.
* **Company Selection:** Data is collected for 10 diverse companies to represent different market segments and risk profiles.
* **Data Quality Assurance:** Automated scripts ensure that data is consistently cleaned and validated upon collection.

**7.2) Performance Metrics and Validation**

* **Technical Indicators Validation:** SMA and RSI computations are cross-verified against industry benchmarks and manual calculations.
* **Sentiment Analysis Evaluation:** Sentiment scores are compared with external sentiment analysis tools and manually validated.
* **Forecasting Accuracy:** Forecasting models are evaluated using error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).
* **Backtesting:** Historical data is used to simulate predictions, allowing the model’s performance to be validated against known outcomes.

**7.3) Testing Strategies and Backtesting**

* **Unit Testing:** Individual functions (e.g., data cleaning, indicator computation) are rigorously unit tested.
* **Integration Testing:** The end-to-end pipeline is tested to ensure that all modules work seamlessly together.
* **Backtesting Framework:** Historical predictions are compared to actual market movements to assess the model’s predictive power over different time horizons.

**RESULTS & DISCUSSION**

**8.1) Data Processing and Quality**

* **Data Consistency:** The collected data is free from major anomalies and is well-aligned across different companies.
* **Successful Preprocessing:** Missing values are effectively handled, ensuring a robust dataset for downstream analysis.

**8.2) Indicator Reliability and Market Correlation**

* **Technical Indicators:** Computed SMA and RSI values reflect market trends accurately and correlate with known financial patterns.
* **Validation:** The indicators were validated using both statistical measures and expert review.

**8.3) Topic Modelling Insights**

* **Latent Themes:** BERTopic successfully extracted coherent topics, such as market sentiment shifts, regulatory news, and economic indicators.
* **Visualization:** The interactive topic visualization allows users to drill down into specific themes, enhancing interpretability.
* **Interpretability:** The topics provide an additional layer of understanding that complements the numerical data.

**8.4) Forecasting Accuracy and Decision Signals**

* **Predictive Performance:** The forecasting module using Prophet achieved competitive accuracy, with error metrics indicating reliable performance.
* **Actionable Signals:** The integration of sentiment analysis helped refine the predictive signals, resulting in clear BUY, SELL, and HOLD recommendations.
* **Overall Impact:** The combined approach of quantitative and qualitative analysis results in a comprehensive market outlook.

**8.5) Overall Impact and Use Case Scenarios**

* **Investment Decision Support:** StockOracle provides a dual-layered analysis that supports informed investment decisions.
* **Research and Development:** The framework serves as a robust foundation for further research in financial analytics and AI-driven forecasting.
* **Business Applications:** The system can be integrated into larger financial platforms for real-time market analysis and decision support.

**CHALLENGES & LIMITATIONS**

1. **Data Quality Variability:** Inconsistent data quality across sources required intensive cleaning and normalization.
2. **Integration Complexity:** Merging quantitative models with NLP-based sentiment and topic analysis posed challenges in ensuring data alignment and consistency.
3. **Model Limitations:** Forecasting models are inherently sensitive to market volatility and unexpected economic events.
4. **Scalability Issues:** Handling larger datasets and additional financial instruments may require optimization of the current pipeline.
5. **Resource Constraints:** High computational costs for running deep learning-based topic models and extensive backtesting were noted.

**FUTURE ENHANCEMENTS**

1. **Real-Time Data Integration:** Implement live data feeds to enable dynamic, real-time analysis and forecasting.
2. **Advanced Modelling Techniques:** Incorporate deep learning models (e.g., LSTM, Transformers) for enhanced forecasting and sentiment analysis.
3. **Broader Data Sources:** Integrate financial news APIs and social media sentiment to enrich textual analysis.
4. **Interactive Dashboard Development:** Develop a comprehensive, user-friendly dashboard using Dash/Plotly for real-time visualization and interactive exploration of results.
5. **Automated Reporting:** Implement a reporting module to automatically generate insights and performance summaries for stakeholders.
6. **Scalability Improvements:** Optimize code and infrastructure to handle larger datasets and additional features, ensuring robustness and faster processing.

**CONCLUSION**

The project represents a significant step forward in financial analytics by merging technical analysis with advanced NLP and forecasting techniques. This project demonstrates that integrating quantitative data with qualitative insights leads to a more comprehensive understanding of market trends. The results indicate that the system is capable of delivering reliable predictions and actionable signals, thereby empowering investors to make better-informed decisions. Although challenges remain, particularly in scalability and integration, the current framework provides a robust foundation for future enhancements and further research in the field of AI-driven financial analysis.

**REFERENCES**

1. yfinance: <https://pypi.org/project/yfinance/>
2. pandas\_ta: <https://pypi.org/project/pandas-ta/>
3. bertopic: <https://pypi.org/project/bertopic/>
4. prophet: <https://pypi.org/project/prophet/>
5. Additional Resources:

* Research papers and articles on time-series forecasting, sentiment analysis, and financial market analytics.
* Documentation for Python libraries and frameworks used in the project.