**Sprint-1: Multi-Source Sentiment-Driven Stock Prediction and Trading Strategy**

**Introduction**

In this sprint, the goal was to integrate sentiment analysis from multiple sources, including financial news, social media, and financial reports, into a stock market prediction model. The objective was to enhance trading decisions and improve investment returns by leveraging sentiment-driven insights. The system was designed to preprocess and analyze sentiment data, integrate it into predictive models, and validate trading strategies through backtesting and forward testing.

[US291: As a trader, I want to integrate sentiment analysis from diverse sources, including financial news, social media, and financial reports, into my stock market prediction model so that I can enhance my investment returns, make better-informed trading](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291)

**Conditions of Satisfaction**

**Data Coverage & Quality:** Data pipelines must collect timely, relevant, and clean data from multiple sources.

**Sentiment Analysis Efficacy:** Models should accurately capture sentiment signals, including sarcasm and irony.

**Predictive Model Integration**: Sentiment features must show measurable improvements in stock predictions.

**Trading Strategy & Validation:** Sentiment-driven trading signals should be backtested and forward-tested.

**Performance Metrics:** Achieve target risk-adjusted returns and maintain controlled drawdowns.

**Documentation & Explainability:** Clear documentation of the entire process and model decisions.

**Definition of Done**

1. Data pipelines successfully collect, clean, and store sentiment data from multiple sources.

2. Sentiment analysis models achieve predefined accuracy benchmarks.

3. Predictive models incorporating sentiment data outperform baseline models.

4. Trading strategies demonstrate positive returns in backtesting, out-of-sample testing, and forward testing.

5. Risk management metrics confirm stable performance.

6. Complete documentation of models, trading strategies, and evaluation methods.

**Tasks**

[MSS 1.1: Data Collection and Processing (20 ph)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/292)

Subtask 1: Develop pipelines for news articles (4 ph)

Subtask 2: Collect social media data (4 ph)

Subtask 3: Gather financial reports (4 ph)

Subtask 4: Clean and preprocess data (4 ph)

Subtask 5: Conduct exploratory data analysis (4 ph)

**Tasks I Worked On**

Data Collection Agent Implementation

MSS 1.1: Data Collection and Processing (20 ph) 292

Developed automated pipelines to collect and preprocess sentiment data from news, social media, and financial reports.

Implemented preprocessing steps to handle missing data, noise, and multilingual content.

Conducted exploratory data analysis to ensure data quality and relevance.

**Sprint-2: Sentiment Analysis Model Development and Feature Engineering for Stock Prediction**

**Introduction**

In this sprint, the goal was to develop a comprehensive sentiment analysis model to process and interpret sentiment data, followed by engineering features that would be integrated into stock market prediction models. The sentiment models aimed to analyze and quantify the sentiment extracted from various sources, while the feature engineering process focused on creating meaningful features to enhance predictive accuracy. Additionally, model stacking techniques were explored to further optimize the performance of the stock prediction model.

[US291: As a trader, I want to integrate sentiment analysis from diverse sources, including financial news, social media, and financial reports, into my stock market prediction model so that I can enhance my investment returns, make better-informed trading](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291)

**Conditions of Satisfaction:**

**1. Sentiment Analysis Model Development:**

- Lexicon-based, machine learning, and deep learning models should effectively process and analyze sentiment data.

- Each model type (lexicon-based, machine learning, deep learning) must meet specific accuracy benchmarks.

**2. Feature Extraction and Engineering:**

- Extracted sentiment features must be relevant and representative of market movements.

- Feature engineering techniques should demonstrate a clear improvement in model performance when integrated.

**3. Model Stacking:**

- Implement model stacking to combine different models and improve prediction accuracy.

- Stacked models should outperform individual models in predictive accuracy.

**4. Addressing Data Challenges:**

- Address challenges such as handling missing data, noise, and multilingual content in sentiment analysis models.

**5. Integration with Predictive Models:**

- Sentiment features must be successfully integrated into stock prediction models.

- Models incorporating sentiment features must outperform baseline predictive models.

**6. Validation:**

- Validate the models using appropriate evaluation techniques such as backtesting and forward testing to ensure reliable predictions and actionable trading signals.

**Definition of Done:**

1. Lexicon-based models, machine learning models, and deep learning models are trained and validated against predefined accuracy benchmarks.

2. Sentiment features are extracted, engineered, and integrated into the stock prediction models.

3. Model stacking is successfully implemented, and stacked models outperform individual models.

4. Addressed challenges related to missing data, noise, and multilingual sentiment data.

5. Predictive models incorporating sentiment data demonstrate improved stock market forecasting and actionable trading signals.

6. Complete documentation of models, feature engineering process, and integration into the prediction pipeline.

**Tasks:**

**[MSS 1.2: Sentiment Analysis Model Development (25 ph) 371](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/371)**

Agent: Sentiment Analysis Agent

**- Subtask 1: Train Lexicon-based Models (5 ph)**

- Develop and implement lexicon-based sentiment models to analyze sentiment in financial news, social media, and reports.

- Train the lexicon models to score sentiment (positive/negative/neutral) based on predefined lexicons.

- Evaluate model accuracy based on its performance on a labeled sentiment dataset.

**- Subtask 2: Train Machine Learning Models (5 ph)**

- Train machine learning models (e.g., logistic regression, SVM) using features derived from sentiment-labeled data.

- Fine-tune the hyperparameters of the machine learning models to improve accuracy.

- Validate the machine learning models using cross-validation techniques.

**- Subtask 3: Train Deep Learning Models (10 ph)**

- Implement and train deep learning models (e.g., LSTM, BERT) for sentiment analysis to capture more complex patterns and relationships.

- Preprocess the data for deep learning (e.g., tokenization, embedding layers).

- Train the models on large datasets and evaluate their performance against baseline models.

**- Subtask 4: Address Data Challenges (5 ph)**

- Handle missing data, noisy data, and multilingual content by applying appropriate preprocessing techniques.

- Use data augmentation techniques to improve the robustness of the models.

- Develop a strategy to handle imbalanced sentiment data if needed (e.g., oversampling/undersampling).

**[MSS 1.3: Feature Engineering and Model Integration (20 ph) 372](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372)**

Agent: Feature Engineering Agent

**- Subtask 1: Extract Sentiment Features (5 ph)**

- Extract sentiment-related features (e.g., sentiment polarity, intensity) from the output of sentiment analysis models.

- Identify key sentiment indicators that can be directly tied to stock market trends.

- Preprocess and structure the features in a format suitable for predictive models.

**- Subtask 2: Integrate Features into Prediction Models (5 ph)**

- Integrate the sentiment features with other technical and fundamental features used in stock prediction models.

- Develop a seamless pipeline to feed sentiment data into existing prediction models (e.g., regression, random forests).

- Evaluate the improvement in model performance with the integration of sentiment features.

**- Subtask 3: Explore Feature Engineering Techniques (5 ph)**

- Research and explore different feature engineering techniques (e.g., sentiment aggregation, temporal analysis) that can better capture the nuances in sentiment data.

- Test various combinations of sentiment features to optimize model performance.

- Identify feature importance and discard irrelevant or redundant features.

**- Subtask 4: Implement Model Stacking (5 ph)**

- Implement a model stacking strategy by combining different sentiment analysis models (lexicon-based, machine learning, deep learning) to improve prediction accuracy.

- Use ensemble methods such as stacking, boosting, or bagging to combine predictions from different models.

- Evaluate the performance of the stacked model in comparison to individual models.

**Tasks I Worked On**

**Sentiment Analysis Agent Implementation:**

**[- MSS 1.2: Sentiment Analysis Model Development (25 ph)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/371)**

- Developed and trained lexicon-based, machine learning, and deep learning models for sentiment analysis, addressing challenges like multilingual content and noisy data.

- Implemented and evaluated deep learning models such as LSTM and BERT for sentiment extraction.

- Focused on refining sentiment extraction processes to improve prediction performance.

**Feature Engineering Agent Implementation:**

**[- MSS 1.3: Feature Engineering and Model Integration (20 ph)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372)**

- Worked on extracting sentiment features (e.g., polarity, intensity) from sentiment analysis outputs and integrated them into predictive models.

- Explored feature engineering techniques to refine the sentiment features and enhance model accuracy.

- Collaborated on implementing model stacking to combine the strengths of individual models, improving overall prediction results.

**Sprint-3: Feature Engineering, Trading Strategy Development, and Backtesting**

**Introduction**

In this sprint, the focus was on refining the integration of sentiment-based features into predictive models while developing and backtesting trading strategies. The goal was to enhance stock market forecasting by leveraging sentiment-driven insights and optimizing trading strategies through rigorous backtesting. The feature engineering tasks aimed to extract, process, and integrate sentiment data effectively, while the trading strategy tasks focused on converting these insights into actionable signals and validating them through historical simulations.

[US291: As a trader, I want to integrate sentiment analysis from diverse sources, including financial news, social media, and financial reports, into my stock market prediction model so that I can enhance my investment returns, make better-informed trading](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291)

**Conditions of Satisfaction:**

1. Feature Engineering and Integration:

- Extract sentiment-based features that capture relevant market movements.

- Ensure proper integration of these features into stock prediction models.

2. Sentiment-Driven Trading Signals:

- Convert sentiment analysis outputs into actionable trading signals.

- Ensure trading signals align with market behavior and do not introduce noise.

3. Backtesting Framework Development:

- Implement a robust backtesting framework to evaluate trading strategies.

- The framework must support multiple strategies and allow performance comparisons.

4. Strategy Performance Validation:

- Conduct backtesting on historical data and assess risk-adjusted returns.

- Trading strategies incorporating sentiment must outperform baseline strategies.

5. Optimization and Refinement:

- Tune strategy parameters to enhance profitability and reduce drawdowns.

- Incorporate safeguards against overfitting by testing on out-of-sample data.

6. Documentation and Reporting:

- Fully document the strategy development, backtesting results, and key learnings.

- Provide a summary of the best-performing strategies and their market applicability.

**Definition of Done:**

1. Sentiment features are successfully extracted, engineered, and integrated into predictive models.

2. Trading signals derived from sentiment analysis are implemented and validated.

3. A fully functional backtesting framework is developed and supports multiple strategies.

4. Trading strategies incorporating sentiment demonstrate improved performance over baselines.

5. Strategy parameters are optimized to balance profitability and risk.

6. Complete documentation of the workflow, strategy performance, and key insights is provided.

**Tasks**:

[MSS 1.3: Feature Engineering and Model Integration (20 ph) 372](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372)

Agent: Feature Engineering Agent

Subtask 1: Extract Sentiment Features (5 ph)

- Extract sentiment-related features (e.g., polarity, intensity) from sentiment analysis outputs.

- Ensure features effectively capture market sentiment and contribute to predictive power.

- Structure sentiment features for seamless integration into predictive models.

Subtask 2: Integrate Features into Prediction Models (5 ph)

- Combine sentiment-based features with existing technical and fundamental indicators.

- Implement feature scaling, transformation, and selection techniques to optimize model performance.

- Assess the impact of sentiment features on stock market prediction accuracy.

Subtask 3: Explore Feature Engineering Techniques (5 ph)

- Investigate advanced feature engineering techniques (e.g., sentiment aggregation, event-based signals).

- Experiment with different transformations to identify the most impactful features.

- Optimize feature selection by removing redundant or noisy features.

Subtask 4: Implement Model Stacking (5 ph)

- Develop ensemble techniques to combine different sentiment models for improved accuracy.

- Test various stacking architectures (e.g., meta-modeling, weighted averaging).

- Evaluate the performance gains from stacked models compared to individual models.

[MSS 1.4: Trading Strategy Development and Backtesting (30 ph) 373](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/373)

Agent: Trading Strategy Agent

Subtask 1: Translate Sentiment Insights into Trading Signals (5 ph)

- Define rule-based and data-driven trading strategies incorporating sentiment analysis.

- Develop buy/sell/hold decision logic based on sentiment scores and market conditions.

- Implement filters to avoid false signals and improve strategy robustness.

Subtask 2: Develop Backtesting Frameworks (5 ph)

- Create a backtesting framework to evaluate trading strategies against historical data.

- Ensure support for multiple timeframes, asset classes, and execution styles.

- Incorporate transaction costs, slippage, and risk constraints in backtesting models.

Subtask 3: Backtest Trading Rules (10 ph)

- Conduct rigorous backtesting using historical sentiment and market data.

- Evaluate strategies based on key performance metrics (e.g., Sharpe ratio, drawdown, win rate).

- Compare sentiment-driven strategies against traditional strategies and baseline models.

- Identify and document the most effective sentiment-based trading approaches.

**Tasks I Worked On:**

Feature Engineering Agent Implementation:

**[MSS 1.3: Feature Engineering and Model Integration (20 ph)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372)**

- Extracted and processed sentiment features to enhance stock prediction models.

- Integrated sentiment data with existing indicators and validated improvements in prediction accuracy.

- Explored feature engineering techniques, including sentiment aggregation and event-driven transformations.

- Developed ensemble methods for model stacking, leading to more robust sentiment-based predictions.

Trading Strategy Agent Implementation:

**[MSS 1.4: Trading Strategy Development and Backtesting (30 ph)](**MSS 1.4: Trading Strategy Development and Backtesting (30 ph)** " \l "373)**

- Designed sentiment-based trading signals and ensured their effectiveness in decision-making.

- Implemented a backtesting framework capable of evaluating multiple sentiment-driven strategies.

- Conducted extensive backtesting and performance analysis to refine trading rules.

- Optimized trading strategies based on performance metrics and risk considerations.

This sprint successfully advanced the integration of sentiment analysis into stock market prediction, culminating in the development of backtested trading strategies. The enhancements in feature engineering and model stacking contributed to improved predictive accuracy, while rigorous backtesting ensured that sentiment-driven trading strategies were validated for practical use.

Sure! Here's a Sprint-4 write-up in the same structured and professional format as your Sprint-3 summary. This one covers evaluation, refinement, and documentation/reporting phases of your sentiment-driven stock market prediction project.

**Sprint-4: Strategy Evaluation, Forward Testing, and Final Documentation**

Introduction

In Sprint-4, the primary focus was on the rigorous evaluation of sentiment-driven trading strategies through out-of-sample testing, forward testing, and comprehensive performance analysis. The objective was to validate the reliability and robustness of the models developed in previous sprints under real-world conditions. This phase also emphasized effective documentation and reporting, aimed at providing transparency, reproducibility, and actionable insights to stakeholders. The tasks in this sprint ensured the strategies were not only theoretically sound but also practically viable in dynamic market environments.

[US291: As a trader, I want to integrate sentiment analysis from diverse sources, including financial news, social media, and financial reports, into my stock market prediction model so that I can enhance my investment returns, make better-informed trading](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) .

Conditions of Satisfaction:

1. Out-of-Sample Validation:

- Apply trading strategies to out-of-sample data to check generalizability.

- Confirm strategies are not overfitted and perform consistently on unseen data.

2. Forward Testing:

- Simulate or apply strategies in a near real-time environment.

- Assess if strategies align with past performance in evolving market conditions.

3. Performance Metrics Analysis:

- Conduct in-depth analysis using risk-adjusted performance indicators.

- Provide a comparative evaluation across strategies.

4. Complete Documentation:

- Record methodology, modeling decisions, and experimental results.

- Prepare visual presentations and written reports for stakeholders.

Definition of Done:

1. Out-of-sample testing is completed with analysis demonstrating model generalization.

2. Forward testing is implemented in a paper-trading or simulated environment.

3. Strategy performance metrics are fully analyzed and benchmarked.

4. All methodologies, processes, and results are documented in detail.

5. Presentations and final reports are prepared and shared with stakeholders.

Tasks:

[MSS 1.5: Evaluation and Refinement (15 ph) 374](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/374)

Agent: Evaluation Agent

- Subtask 1: Conduct Out-of-Sample Testing (5 ph)

- Select data segments not previously used in training or validation.

- Apply sentiment-driven strategies to this unseen data.

- Evaluate predictive accuracy and profitability to ensure robustness.

- Subtask 2: Analyze Performance Metrics (5 ph)

- Use key performance indicators (e.g., Sharpe ratio, Sortino ratio, CAGR, drawdown).

- Benchmark against baseline models and traditional strategies.

- Identify strategies with consistent outperformance across scenarios.

- Subtask 3: Perform Forward Testing in a Simulated or Live Environment (5 ph)

- Deploy trading strategies in a simulated environment (e.g., paper trading).

- Monitor real-time predictions and trading decisions.

- Validate consistency with historical and out-of-sample results.

[MSS 1.6: Documentation and Reporting (10 ph) 375](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/375)

Agent: Documentation Agent

- Subtask 1: Document Project Methodology and Results (5 ph)

- Provide detailed documentation of sentiment feature generation, modeling, and evaluation.

- Include technical workflows, rationale for decisions, and experimental design.

- Subtask 2: Prepare Presentations and Reports (5 ph)

- Summarize strategy results, insights, and business implications.

- Develop stakeholder-facing slide decks and executive summaries.

- Highlight top-performing strategies and suggested next steps.

**Tasks I Worked On:**

Evaluation Agent Implementation:

MSS 1.5: Evaluation and Refinement (15 ph)

- Executed out-of-sample testing to verify model generalization across new market data.

- Performed comprehensive metric-based analysis, highlighting strategies with optimal risk-reward profiles.

- Set up a simulated trading environment to perform forward testing under near-live conditions.

- Assessed consistency between forward test outcomes and historical backtesting results, confirming strategy reliability.

Documentation Agent Implementation:

MSS 1.6: Documentation and Reporting (10 ph)

- Compiled extensive documentation covering end-to-end methodology from feature engineering to final evaluation.

- Created polished presentations and detailed reports summarizing findings, strategy efficacy, and market readiness.

- Shared deliverables with key stakeholders for review and feedback, supporting informed decision-making.

**Conclusion**

Sprint-4 marked the final validation and delivery phase of the sentiment-driven trading project. Through out-of-sample and forward testing, we ensured the strategies were robust, generalizable, and capable of real-world deployment. Thorough documentation and reporting solidified the knowledge base, offering stakeholders clear visibility into the project’s outcomes and strategic value. This sprint not only confirmed the viability of sentiment-informed trading strategies but also prepared the groundwork for future live trading or product integration.

**Summary Table of Work**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User Story ID** | **User Story** | **Story Points** | **Task ID** | **Task** | **Task Hours** | **Status** | **Actual Hours** |
| [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | As a trader, I want to integrate sentiment analysis from diverse sources into my stock market prediction model. | 3 | [292](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/292) | MSS 1.1: Data Collection and Processing | 20 | Completed | 20 |
| [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | As a trader, I want to integrate sentiment analysis from diverse sources into my stock market prediction model. | 3 | [371](**MSS 1.2: Sentiment Analysis Model Development (25 ph)** " \l "371) | MSS 1.2: Sentiment Analysis Model Development (25 ph) | 25 | Completed | 25 |
| [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | As a trader, I want to integrate sentiment analysis from diverse sources into my stock market prediction model. | 3 | [372](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372) | MSS 1.3: Feature Engineering and Model Integration (20 ph) | 20 | Completed | 19 |
| [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | As a trader, I want to integrate sentiment analysis from diverse sources into my stock market prediction model. | 3 | [373](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/373) | MSS 1.4: Trading Strategy Development and Backtesting | 22 | Completed | 20 |

## **Summary Table of Commits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Commit Number | Commit Description (exactly as in github) | User Story | Task |
| 2/6/2025 | [d098a8b9fd840470223bf270e685c212543e9e48](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/462/commits/d098a8b9fd840470223bf270e685c212543e9e48) | Update app.py | [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | [292](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/292) |
| 2/26/2025 | [05468e5b12672956f910299b98c1a1db6b5fe47a](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/commit/05468e5b12672956f910299b98c1a1db6b5fe47a) | MSS 1.2: Sentiment Analysis Model Development (25 ph) | [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | [371](**MSS 1.2: Sentiment Analysis Model Development (25 ph)** " \l "371) |
| 3/8/2025 | [337947e5cf1863a21b33fbc6a0ae84073cb74e36](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/commit/337947e5cf1863a21b33fbc6a0ae84073cb74e36) | MSS 1.3: Feature Engineering and Model Integration | [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | [372](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/372) |
| 3/25/2025 | [c048d7132bda060de9d9dbe43d24342b6885805d](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/528/commits/c048d7132bda060de9d9dbe43d24342b6885805d) | MSS 1.4: Trading Strategy Development and Backtesting | [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | [373](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/373) |
| 4/22/2025 | [856016759f26a492f4c61f91381be38aab59e549](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/580/commits/856016759f26a492f4c61f91381be38aab59e549) | [MSS 1.5: Evaluation and Refinement (15 ph)\*\* #374](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/374) | [291](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/291) | 374 |