Group Reflection Report

Project Overview

Project Name: Post Earnings Announcement Drift Trading Algorithm

Group Number: 10

Group Members: Archit Sharma, Bhuvesh Chopra, Braedon Kwan

GitHub Repository: https://github.com/retro2408/PEAD-Trading-Dashboard.git

Achievements Summary

We successfully achieved full end-to-end functionality of our trading algorithm, designed around the Post Earnings Announcement Drift (PEAD) anomaly. The system was rigorously tested on a decade of historical data, automatically executing regression modeling and backtesting routines with each run. These results dynamically populate our trading dashboard, providing users with up-to-date performance metrics as new earnings are released. This workflow ensures a seamless transition from data ingestion to decision-making and visualization. Furthermore, all modules—from data fetching to UI rendering—were integrated through well-defined interfaces, ensuring cohesive performance validated through unit and integration testing (as outlined in our V&V and Design documents).

Our regression models consistently achieved strong predictive performance, averaging an adjusted R^2 of approximately 85%. This was accomplished through meticulous feature engineering, including Variance Inflation Factor (VIF) filtering to eliminate multicollinearity. The resulting earnings forecasts closely aligned with actual values, as demonstrated across tickers such as MSFT, GOOGL, and GS. Backtesting further validated the algorithm's viability: an initial paper investment of $50,000 yielded a $300 gain over 10 years—achieved through only four trades annually and a short 2-hour holding period per trade. Together, these results highlight the system's practical effectiveness and its potential for real-world application under conservative risk exposure and minimal intervention.

Requirements Completion Summary

All P0 and P1 requirements, as defined in our pivoted direction post-January faculty meeting, were successfully implemented and in many areas exceeded their original specifications. Our

core pipeline—from data extraction and preprocessing to trade signal generation—was fully realized. The algorithm accurately parses historical earnings data through the Alpha Vantage API to extract key financial metrics (e.g., Reported EPS, Revenue, Accounts Receivable), and compares these against Expected EPS data sourced from Yahoo Finance. This EPS comparison forms the foundation of our signal generation logic, which was thoroughly tested and validated before being deployed into the backtesting module. As outlined in Section 5.3 of our documentation, rigorous performance evaluation over a 10-year period confirmed the strategy's stability and profitability.

In terms of P1 functionality, we delivered a fully functional and dynamic web-based dashboard that visualizes all trading signals and financial metrics with clarity. The homepage offers an overview of key stock-level information, including Sharpe ratio comparisons and search capabilities, while the detailed stock pages present equity curves, backtest PnL, trade history, and other performance visualizations. We implemented regression-based whisper number prediction using multivariate linear models trained on historical deltas between different EPS types. Further, the algorithm hardcodes trade thresholds and stop-loss rules to maintain capital protection. Although we did not implement the P2 features involving advanced ensemble models like XGBoost or Random Forests, the core system operates robustly with clearly defined margins and well-performing regression logic.

Lessons Learned Including What Worked and Did Not

Overall, our team is very satisfied with the outcome of this project and the direction it ultimately took—even though it differed from our original design. Following our January meeting with the professor, we made a strategic pivot by removing the live price extraction and order placement modules. This decision was based on privacy constraints, the lack of viable APIs for trading without user authentication, and the high cost of broker-integrated news APIs. Instead, we focused our efforts on creating a powerful and verifiable model for the Post Earnings Announcement Drift (PEAD) anomaly, enabling us to validate the strategy across a diverse basket of stocks using historical data.

This experience gave us a practical understanding of handling large financial datasets, building reliable backtesting systems, and working with regression models under real-world constraints. We gained hands-on skills in data cleaning, managing API limitations, and constructing multivariate models with meaningful predictors. Presenting our results through a dynamic dashboard taught us how to use modern data visualization tools effectively. Importantly, we also discovered that while the strategy performs well on some stocks, it underperforms on others—highlighting the need for selective application and more contextual feature inputs. We now see how the project lays a strong foundation for future exploration into other market anomalies, and we're excited by the prospect of expanding the framework to incorporate macroeconomic indicators such as currency shifts and inflation rates.

What worked particularly well was the use of historical earnings data to train regression models and generate meaningful trading signals, and how cleanly these results integrated into a presentable dashboard experience. We also appreciated the collaborative and exploratory nature of the capstone course, which allowed us to iterate on our ideas, adapt when necessary, and ultimately deliver a robust and informative system.