Post Earnings Announcement Drift (PEAD) Algorithm for Automated Trading

GROUP 10

Archit Sharma

Bhuvesh Chopra

Braedon Kwan

October 11th, 2024

1. Versions, Roles, and Contributions

1.1 Revision History

Version	Author(s)	Description	Date
Rev 0	All	Created first draft of document	11th October 2024

1.2 Project Personnel

Group Member	Role	Email
Archit Sharma	Developer, Machine Learning. Responsible for creating machine learning and AI Models to analyze earnings reports.	shara96@mcmaster.ca
Bhuvesh Chopra	Data Analyst. Responsible for connecting to the dataset, data preprocessing, cleaning and setting up the stream through API.	choprab@mcmaster.ca
Braedon Kwan	Full stack Developer Responsible for developing the web interface to integrate with the algorithm.	kwanb5@mcmaster.ca

1.3 Table of Contributions

Group Member	Contributions
Archit Sharma	Functional Requirements (P1, P2), Risks and Open Issues, Purpose
Bhuvesh Chopra	Functional Requirements (P0, P1), Data and Metrics, Project Constraints
Braedon Kwan	Functional Requirements(P3), Non-functional requirements, Stakeholders, Glossary

2. Glossary

Term	Definition
PEAD	Post Earnings Announcement Drift, a phenomenon where stock prices continue to drift after earnings surprises.
EPS	Earnings per share, a key metric for analyzing company performance.
NLP	Natural Language Processing is a subfield of Artificial Intelligence which utilizes rule-based or machine learning approaches to understand and interpret human language and will be used by us to interpret earnings reports.
API	Application Programming Interface, used for connecting to brokers and other data sources.
XG Boost	A supervised machine learning method for classification and regression.
Sharpe Ratio	Risk Adjusted Measure of Portfolio Returns.
XSS Attack	Hackers inject malicious code onto a legitimate website.
Momentum	Rate of acceleration of a stock's price or volume.
Whisper Numbers	The markets quantified expectations about the earnings per share.
Expected EPS	The Earnings Per Share provided by analysts from investment banks.
Sell Stock	A short seller borrows stock from a broker and sells that into the market. Later, they

	hope to buy back that stock at a cheaper price and return the borrowed stock in an effort to profit on the difference in prices.

3. Introduction

3.1 Purpose of the Project

This project aims to build an algorithm that exploits the **Post Earnings Announcement Drift (PEAD)** anomaly in financial markets. The system automates buy/sell/do nothing decisions based on discrepancies between expected and actual earnings data using key metrics such as Earnings Per Share (EPS), revenue, and gross profit margins. By doing so, it captures stock price drifts following earnings "surprises" and executes trades automatically.

Goals:

The goals of the project are primarily to develop an algorithm that will processes earnings reports in real time, that is whenever new earnings reports are published on broker news streams and use those earnings reports to trigger automated trades, using trade signals generated by the algorithm. Secondly, the project aims to refine and optimize trading decisions using machine learning models to predict upwards or downwards trends in stock metrics. Lastly, the project focuses on implementing and providing users such as student traders and analysts with a web-based interface to monitor and interact with the algorithm and use it to learn how earnings reports are analyzed to dictate better trade decisions.

3.2 Stakeholders

3.2.1 Direct Stakeholders

- **Group Members:** We are the primary, hands-on contributors to the project, tasked with designing, coding, testing, and deploying the algorithm and machine learning models. We will also be building a web-based interface for users. The successful deployment of the project will enhance our professional experience and potentially open career opportunities. We will benefit directly from gaining knowledge in algorithmic trading, machine learning, and web development.
- **Student Algorithmic Traders/Analysts:** The web implementation of the algorithm would allow other fellow students who aspire to learn about algorithmic trading and PEAD a chance to interact with a live algorithm, which allows them to understand the intricacies of the algorithm. Furthermore, it provides them with a chance to see how machine learning and NLP could be used to dictate better trade decisions.

3.2.2 Indirect Stakeholders

- **Academic Researchers:** Researchers interested in stock market anomalies or predictive modeling can use this project's output as a research tool or case study. Their stake is primarily academic, with a focus on discovering new insights into market behavior. They benefit by gaining access to a system that can provide real-time data and help identify market inefficiencies or anomalies. Though not directly involved, they would be affected if the algorithmic models fail to capture meaningful data or produce unreliable results, limiting the project's research potential.

3.3 Project Constraints

- **Real-Time Data Dependency:** The algorithm is heavily depended on real-time earnings data, which can be constrained by the availability and speed of data feeds (e.g., Yahoo Finance, Refinitiv, IB API news Stream). Data latency may limit the accuracy and timeliness of trade execution, particularly during high-traffic periods like earnings seasons.
- **Market Scenarios (Earnings Report Focus)**: The algorithm would be limited to stock metrics based on earnings reports. This means that other market factors such as macroeconomic data, geopolitical events, etc. are excluded from the trading decisions, narrowing the application of the model.
- **Machine Learning Constraints (For the Algorithm Itself):** The models must generalize well to real-time market conditions, yet the limited historical data (40 data points over 10 years) restricts the depth of learning for whisper number predictions.
- **Limited Stock Industry Selection**: For this project we would be focusing only on technology industry stocks, as they have high correlation, and all the members are most comfortable with them and follow them closely. Expanding the model to all market sectors may require significant additional data and refinement.
- **Regulatory and Compliance Constraints:** The system must adhere to financial regulations surrounding automated trading, such as limits on algorithmic trading activity and data privacy. These rules may limit the algorithm's functionality or necessitate certain safeguards.
- **Data Source Constraints:** Access to live streaming news data for live earnings reports requires premium API subscription from Interactive Brokers.
- **Broker Integration Constraints for other Users**: Although the project includes a web interface for interacting with the algorithm, users will need to manually integrate the system with their broker accounts. The algorithm will generate trade signals, but users must link their own broker accounts to the platform for execution. Due to privacy and security concerns, we cannot automate the creation, integration, or management of individual broker accounts or portfolios. This ensures that users maintain full control over their accounts while minimizing privacy risks and avoiding complex integration challenges.

4. Functional Requirements

4.1 P0 (Minimal Viable Product):

- The algorithm will connect to Interactive Brokers' news livestream API to retrieve live earnings reports.
- It will parse earnings reports to extract key financial metrics such as Reported EPS, Revenue, Profit, and Accounts Receivable.
- The algorithm will also retrieve Expected EPS from historical earnings reports, by accessing Interactive Brokers' news dataset.
- A comparison will be made between the Expected EPS and the Reported EPS.
- Based on this comparison, the algorithm will output a trade signal—buy, sell or do nothing.
- Before using the algorithm with real capital in live trading, rigorous back testing of the strategy will be performed to ensure accuracy and reliability. This is explained in more detail in the following section: 5.3 Performance Requirements.

4.2 P1 (High Priority):

- A fine-tuned algorithm should be able to use a regression model to predict the next whisper number by analyzing previous earnings reports, trends and surprises. The model should learn from generalizable patterns across similar stocks, making it more adaptable and capable of functioning with limited data. This is explained in detail in sections 4.2.1 Regression for Whisper Number Prediction and 4.2.2 Generalization of Whisper Numbers.
- A fine-tuned algorithm should be able to decide an optimal or at least an established stop loss level and an exit criterion for the strategy in order to minimize drawdown risks for the investor.
- The algorithm would hard code the required margins/thresholds between these three EPS's metrics which we are considering to generate a trading signal between buy ,sell or do nothing.

4.2.1 Regression for Whisper Number Prediction:

The machine learning model will be trained using deltas (margins) based on actual EPS, required EPS, and whisper EPS. The following techniques will be utilized:

- **Linear Regression** and **Decision Trees** for capturing relationships between previous reports and the whisper number.
- **Gradient Boosting (XGBoost)** will be employed to handle missing data and more complex non-linear relationships.

4.2.2 Generalization of Whisper Numbers:

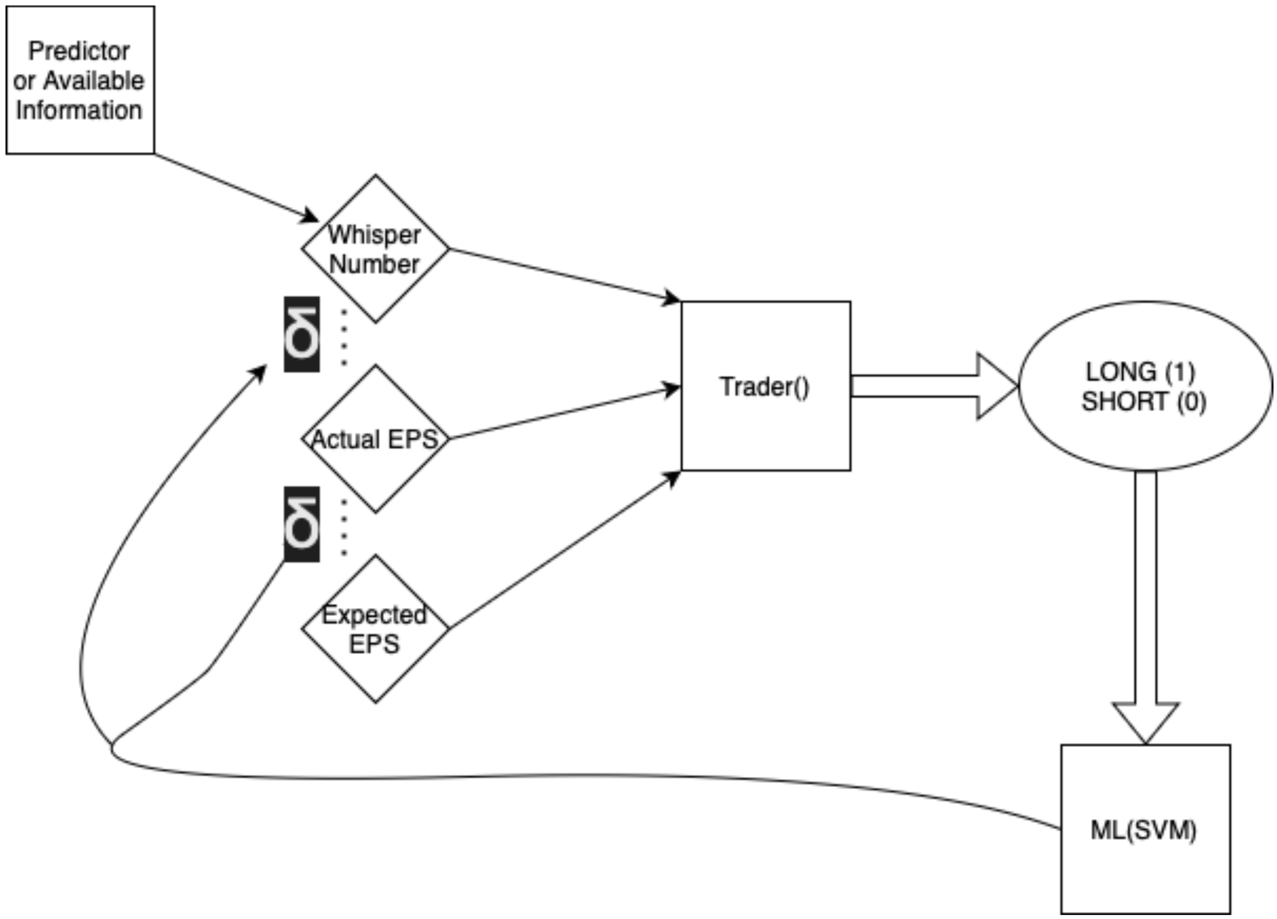
Since data is limited (40 points over 10 years), the model will generalize its learning using:

- **Transfer Learning**: Training the model on a broader dataset from similar companies and sectors, then fine-tuning it on the specific stock or company data.
- **Data Aggregation**: Incorporating sector-based earnings reports and general market sentiment to provide more robust predictions.

This approach ensures that the system can generalize across different stocks or sectors while using limited data effectively. The predicted whisper number will be used alongside actual and expected EPS margins to decide whether to **buy, sell or do nothing** based on market responses.

4.3 P2 (Medium Priority)

- The addition of a machine learning model to learn the margin and action determination would enhance the model as it would be able to generate trading signals by efficiently learning the optimal margin thresholds required. In addition, the reward function for the model would be the profit achieved by the trading entity during the back test. A more detailed explanation of the same is provided in section 4.3.1 Margin and Action Determination and through the figure provided below, which gives a visual representation for the same.



4.3.1 Margins and Action Determination

The **margins** between whisper, actual, and expected EPS will be crucial for determining market behavior. By training the model on these margins, the system will classify the appropriate **trading action**.

For **action determination** (buy, sell, do nothing), the following machine learning models will be used:

- **Decision Trees**: To classify the trading action based on the learned margin configurations.
- **Random Forests**: To improve classification accuracy by combining multiple decision trees.
- **XGBoost (Gradient Boosting)**: To handle non-linear relationships between EPS margins and trading actions, offering high performance in classification tasks.

Unlike the whisper number prediction, **action determination** is not generalizable across sectors or companies, and therefore **transfer learning** will not be applied for

this component. The **buy, sell, do nothing decisions** are highly specific to the company, stock, or sector, and require training on more targeted data. The model will rely on the direct relationship between the margins of the specific company's EPS numbers to determine the optimal action.

The models will be trained on historical data to learn which margin configurations typically lead to **buying, selling or doing nothing** decisions:

- **Buy**: When the actual EPS is significantly higher than the whisper number.
- **Sell**: When the actual EPS falls below the whisper number.
- **Do Nothing**: When the margins are small or negligible.

The classification output will guide real-time trading decisions based on the analysis of EPS deltas and margins.

4.4 P3 (Nice to Have)

- A web-based dashboard will receive and display trading signals from the algorithm, providing users with a clear and accessible overview of relevant stocks. For more detailed information refer to 4.4.1 Homepage Overview and 4.4.2 Detailed Stock Page.
- Technical indicators and candlestick patterns can be powerful tools to confirm trading signals. By combining these methods, traders can filter out false signals and increase the probability of successful trades. For more information refer to 4.4.3 Technical indicators and 4.4.4 Candle Stick Patterns.

4.4.1 Homepage Overview:

- Displays key information for each stock including stock name, symbol, current signal (buy/sell or do nothing based on latest eps release), last eps release date, next eps release date, and other key financial metrics.

4.4.2 Detailed Stock Page:

- Displays the financial metrics of the selected company in greater detail and has a price graph with the historical trading signals plotted in.

4.4.3 Technical Indicators:

- **Moving Averages:** smooths out price data to help identify trends.
- **Relative Strength Index:** a measure of momentum to identify overbought or oversold conditions.
- **Bollinger Bands:** uses volatility to determine price extremes.

4.4.4 Candle Stick Patterns:

- **Bullish/Bearish Engulfing**: these patterns indicate potential reversals.
- **Doji Candlestick:** indicates market indecision and can confirm potential reversals, particularly when seen at support or resistance levels.
- **Hammer and Hanging Man:** a hammer at a support level or a hanging man at a resistance level indicates a potential reversal.

5. Non-functional Requirements

5.1 Look and Feel Requirements

- **Homepage**: Clean, organized layout displaying stock name, symbol, buy/sell/do nothing signal, and next EPS release date.
- **Color Scheme**: Green for long (buy), red for short (sell), and neutral tones for other information.
- **Typography**: Use bold fonts for key data like stock symbols and signals, with smaller text for secondary details.
- **Consistency**: Uniform styling across all pages for charts, buttons, and data.
- **Responsive Design**: Ensure smooth functionality across desktops, tablets, and smartphones, with clear visuals on all screen sizes.

5.2 Usability and Humanity Requirements

- **Easy Navigation**: Simple homepage layout with clickable stocks for detailed information.
- **Signal Clarity**: Color-coded signals and clear labels to make buy/sell/do nothing signals easy to understand.
- **Intuitive Design**: Ensure the interface is straightforward and user-friendly, requiring minimal learning.

5.3 Performance Requirements

- **Back testing Performance**:
  The system should demonstrate statistically significant non-zero returns during back testing using a **T-distribution test** with a **confidence interval of 95%**. The p-value should indicate that returns are significantly different from zero, confirming that the model performs better than random chance.
- **Accuracy and Precision**:
  The model must maintain an accuracy of at least **75%** in predicting correct trades. The model's precision and recall must also be optimized to minimize false positives (incorrect buy/short signals) and maximize true positives (correct trading signals).
- **Risk-Adjusted Returns (Sharpe Ratio)**:
  The system should achieve a **Sharpe ratio** of at least **1.1** during back testing, ensuring that the model's returns are adjusted for the risk taken. This will measure the consistency of returns relative to the market's volatility.
- **Speed**: System must generate and execute trades within a reasonable timeframe after detecting earnings anomalies.

5.4 Security Requirements

- **API Rate Throttling**: Apply rate limiting to server to request to prevent DoS attacks.
- **User Authentication**: Use secure token-based authentication to protect against XSS attacks.

5.5 Legal Requirements

- **Compliance**: System must comply with regulatory requirements for automated trading platforms, particularly those involving earnings report data and trade execution. Regulations may vary depending on the region or market being traded, particularly for earnings-based trading.

5.6 Logging and Auditing Requirements:

The system must maintain detailed logs of all operations and transactions to ensure:

- **Traceability**: Every action taken by the system can be traced back for auditing purposes.
- **Error Handling**: All errors and exceptions will be logged for troubleshooting and improvement.
- **Performance Evaluation**: Logs will be analyzed to evaluate the model's performance over time and identify areas for enhancement.

5.6.1 Logging Components:

- **Trade Execution Logs**: Record details of every trade executed, including timestamps, trade type, stock ticker, quantity, price, and outcome.
- **Model Decision Logs**: Log the model's predictions and the corresponding actions taken, along with margins at the time of prediction.
- **Error and Exception Logs**: Capture errors and exceptions encountered, including timestamps, error messages, and relevant contextual information.

6. Data and Metrics

6.1 Data for Training/Building the Model

- **Features:**
  - **EPS Deltas**: The difference between the actual EPS, required EPS, and whisper EPS.
  - **Historical EPS Data**: Time series data on actual, expected, and whisper EPS over a span of 10 years for multiple companies in similar sectors.

    - **Market Trends and Sentiment**: Data on stock price movements, post-earnings announcements, and any relevant market sentiment indicators (e.g., from financial news sources).
    - **Sector Data**: Aggregated data from similar companies or industries to allow for transfer learning when data is scarce.
    - **EPS Data:** We would need to leverage regular expressions and pattern matching to parse the required information from a large unstructured dataset from the broker/Api to get the features we would need to pass over to the trading entity.
- **Label/Target:**
    - **Whisper Number Prediction**: Actual whisper EPS values used as ground truth for prediction.
    - **Action Determination**: Classification of whether to "buy", "sell", or "do nothing" based on the predicted margins between whisper, actual, and expected EPS.

6.2 Links to Dataset or Plan for Data Collection

- **Sources**:
    - **Financial Databases**: Use sources like Bloomberg, Reuters, or S&P for historical EPS and stock market data.
    - **Broker API Data**: Leverage broker APIs such as the Interactive Brokers API for live stock price data and post-earnings stock movements.
    - **News Feeds**: For market sentiment, use NLP to process earnings call transcripts and financial news.
- **Plan to Simulate Data**: If real data is insufficient, simulate stock price movements post-earnings announcements based on sector trends to generate synthetic data for training.

6.3 Performance Metrics

- **Accuracy (for Action Determination)**: Measures how many buy/sell/do nothing predictions the model gets correct.
- **Precision and Recall**: Used to evaluate trading action classifications (especially important in avoiding false positives in buy/sell decisions).
- **Area Under ROC Curve (for Whisper Number Prediction)**: To gauge the predictive power of the regression model when classifying whisper numbers in relation to actual EPS.
- **Back test Return Analysis**: Evaluate how much return the trading algorithm generates in backtesting.
- **T-Test for Non-Zero Returns**: Statistical test ensuring that back tested returns are significantly non-zero.

7. Risks and Open Issues

7.1 Risks

- **Data Latency**: Real-time market data may experience delays, affecting trade execution speed.
- **Model Overfitting**: Regression and machine learning models may perform well on historical data but fail to generalize on real-time data.
- **API Reliability**: If the broker's API experiences downtime, the system may fail to execute trades or gather real-time data, or if the news is released to selected sources before our broker, it might diminish our ability to generate returns.
- **Hold Period Determination** : This number currently is arbitrary, we cannot decide for an optimal hold period and even if we do it might be overfitted or specific to the training example, therefore this would potentially limit the returns, we would need to work on sound economic reasoning to come up with a good number here.

7.2 Open Issues

- **NLP Accuracy**: Parsing complex earnings reports in real-time may lead to incorrect signals if the NLP system misinterprets the data.
- **Complex Earnings Reports**: Certain non-standardized reports may not be fully processed by the system, leading to missing data or misinterpretation.