Imperial: Coursework 2025

May 15, 2025

Abstract

This document outlines the coursework for the Machine Learning and Deep Learning in Finance course. The assignment is structured around an event-driven investment strategy, with a specific focus on earnings announcement dates. Students are expected to leverage advanced data science and machine learning techniques to forecast the abnormal returns of S&P 1500 constituents on the day of their earnings releases. This constitutes a complex and highly relevant problem in quantitative finance, regularly addressed by practitioners and researchers in the field.

Contents

1 Context

In practice, publicly listed companies report their earnings on a quarterly basis. These earnings announcements constitute key information events that can significantly impact stock prices—particularly on the trading day following the release. In this project, we restrict our focus to modeling and forecasting stock returns on the day after the earnings announcement.

Despite strict regulatory frameworks designed to prevent the leakage of material non-public information, investor expectations play a crucial role in shaping market reactions. Market participants—ranging from institutional investors to retail traders—frequently take speculative positions in anticipation of the announcements, using instruments such as short sales, call or put options, or direct long positions in the underlying equities.

To inform their expectations, investors may consider historical earnings release data, including the direction and size of past earnings surprises. While the relationship between past and future reactions is complex and not necessarily stable over time, such historical patterns are often used as heuristics or features in predictive models. However, anticipating the market's response remains a challenging task due to the influence of broader macroeconomic conditions, sector-specific factors, changes in investor sentiment and earlier reporting from firms in the same sectors or industries.

2 Available Datasets

To address the forecasting task, five datasets are provided:

- `X_train`: This dataset contains the training features. It includes a broad range of raw variables that require careful preprocessing and feature engineering to extract meaningful signals. The features span several domains, including price and volume-based indicators, fundamental data, short interest metrics, and analyst estimates as well as categorical sectors or industries GICS Classification. A column date is also present.
- `y_train`: This dataset contains the target variables corresponding to `X_train`. The primary objective is to forecast the forward return of the stock following the earnings release. In addition, auxiliary targets are provided; these are designed to be more directly predictable using machine learning models and are significantly correlated with the forward return. While the use of these auxiliary targets is optional, leveraging them may help improve the accuracy of the final return prediction.
- `DummyX_test`: This dataset mirrors the structure of the final undisclosed `X_test` and serves as a template for evaluating your model. It does not include the real features, but a similar dataset will be used during the final evaluation phase. Your submission must enable us to generate predictions for this dataset without manual intervention. If you add new features, you must explicitly write code that adds them to `DummyX_test` for evaluation purposes.
- `all_prices`: This dataset contains historical data on prices, trading volumes, and market capitalizations for all stocks considered in the project. It can be used to engineer additional features, although its use is not required. If you choose to generate new features from this dataset, you must clearly provide the code or function used to construct them, ensuring reproducibility on the test set.
- `all_data`: This dataset contains historical values for the features used in this project. It can serve as a reference for normalizing features across different industries or sectors. To extract `X_train` from this dataset, one simply selects the rows where `MASK_EARNINGS` equals 1. Note, however, that using this dataset is optional. Please note that from the date corresponding to the end of `X_train`, the values are completely random.

Variable	**Definition**
`piot_norm`	Normalized Piotroski F-score indicating financial strength.
`dsi`	Short interest divided by the number of shares.
`dtcn`	Short interest divided by the average trading volume over the past 15 days.
`ddtcn`	Difference between the current `dtcn` and the one from the previous year.
`short_interest`	Percentage of shares shorted, indicating negative market sentiment.

Variable	Definition
`asset_turnover_ratio`	Efficiency of asset use in generating revenue.
`current_liabilities`	Company's short-term obligations.
`ev_to_ebit`	Enterprise Value to EBIT – a valuation metric.
`gross_profit_margin`	Ratio of gross profit to revenue.
`interest_expenses_net`	Net interest expense (interest expenses minus interest income).
`long_term_debt`	Total long-term borrowings.
`net_cash_flow_oper`	Cash flow from operating activities.
`net_debt_to_equity`	Net debt divided by equity – a leverage ratio.
`net_income_before_extr`	Net income excluding extraordinary items.
`price_to_book`	Ratio of market value to book value.
`total_assets`	Total assets held by the company.
`total_curr_assets`	Total current (short-term) assets.
`epsp`	Realized Earnings per Share divided by closing price.
`epsf`	Forecasted Smart Earnings per Share divided by closing price.
`reps1`	Difference between current and previous period's EPS-to-close ratio.
`repsf4`	Difference between the realized EPS-to-closing-price ratio and the forecasted ratio from one year earlier.
`sue`	Standardized Unexpected Earnings – surprise component of earnings.
`inesp`	Indicator for whether the return on the previous earnings day was less than -7%.
`inesn`	Indicator for whether the return on the previous earnings day was greater than 7%.
`reps41`	EPS-to-close ratio of the previous year minus that of two years ago.
`repsfs`	Normalized growth of Estimated Smart EPS over the past quarter.
`repsfl`	Difference between current Smart EPS and the average Smart EPS of the same quarter in the previous year (normalized).
`nspc5`	Percentage of times in the last 5 years when the earnings release day return was below -7%.
`value_mean_eps`	Average EPS estimate.
`value_smart_eps`	Smart estimate of EPS.
`deps`	Difference between `value_smart_eps` and `value_mean_eps`.
`value_split_adj_mean_eps`	Mean EPS adjusted for stock splits.
`value_split_adj_smart_eps`	Smart EPS estimate adjusted for stock splits.
`MASK_EARNINGS`	Indicator used to indicate if there is an earning the next following trading day.
`gics_sector`	Sector classification based on GICS.
`gics_group`	Industry group classification (subcategory of sector).
`gics_industry`	Industry classification within GICS.
`gics_subindustry`	Most detailed subindustry classification.

3 Grading Criteria

This project addresses a complex and realistic forecasting problem. As such, do not be discouraged if your model's predictive performance is limited. In financial markets, even marginal predictive power can provide a competitive edge. Therefore, a model that performs only moderately may still represent meaningful progress.

Your project will be assessed according to the following three criteria:

1. **Quality of the documentation:** You must thoroughly explain the reasoning behind your methodological choices. The documentation should include:

   – A clear description of your overall modeling pipeline.

   – Justification for preprocessing steps, with particular attention to the temporal structure of the data (e.g., avoiding future data leakage, maintaining chronological order).

   – Details on feature engineering and selection strategies.

   – Discussion of validation methods, such as walk-forward validation or expanding window approaches, appropriate for time series prediction.

   – Insights into any challenges encountered and how they were addressed.

   – The choice of the ML model, you can explain it if you want.

   – Performance metrics and your backtest on your validations split.

   The goal is to allow a technically competent reader to understand, replicate, and critically evaluate your approach.A PDF is necessary. You may create it using LaTeX or Microsoft Word.

2. **Quality of the Python code:** Your code must be clean, modular, and well-documented. It should follow good software engineering practices, be reproducible, and allow for straightforward testing to new data. Clarity and maintainability will be valued alongside functionality.

3. **Performance of the approach:** Your final model will be evaluated on a undisclosed `X_test` dataset constructed by the instructors, which follows the same structure as the publicly available one. While you will not have access to the actual test features or targets, your model must be able to generate valid predictions for this data without manual intervention.

   Predictive performance will be assessed through a backtest conducted on the undisclosed `X_test`. Therefore, it is essential that you clearly specify how your model's outputs should be interpreted and used in the evaluation. For example, if you are using a binary classification approach and only wish to act on predictions where the estimated probability exceeds a certain threshold, you must explicitly communicate that threshold and decision rule in your submission.

4 Proposed Approach

This section outlines a general methodology that you may consider for tackling the forecasting task. Following these steps is not mandatory; they are merely intended as guidance to help structure your workflow.

1. **Data preparation:** Load the `X_train` dataset and split it into multiple training and validation subsets using a walk-forward validation scheme. This will allow you to simulate a realistic evaluation of your model's performance over time.

2. **Feature engineering:**

   – Analyze each feature and apply appropriate transformations (e.g., normalization, winsorization, log-scaling) to ensure consistency in scale and reduce the impact of outliers.

   – Optionally, create additional features derived from historical data in the `all_prices` dataset. These may include technical indicators (e.g., moving averages, momentum signals) or cross-sectional factors.

3. **Target definition:** Select the target variable you want to predict. While the forward return is the primary objective, you may choose to include auxiliary targets to improve model training or stability.

4. **Model training and feature selection:**

   – Within each walk-forward split (e.g., every 6 months), select the most relevant features using statistical or model-based techniques.

   – Train a machine learning model on the training set and optionally perform hyperparameter optimization via grid search or similar methods.

5. **Evaluation and backtesting:**

   – Aggregate predictions on all validation (out-of-sample) periods and evaluate their performance.

   – Conduct a backtest using a simple long/short strategy: go in equal weighted long on stocks predicted to have high forward returns and short on those predicted to have low forward returns. This will help assess the practical utility of your forecasting model in a portfolio context.

Warning: Data Leakage Ahead!

When creating new features for financial models, **always ensure you only use information that would have been available at the time of prediction.** Using future data—even accidentally—introduces *data leakage*, which leads to unrealistically high model performance and poor generalization in live environments.

– ✓ Use lagged values, historical indicators, and only past data.

– ✗ Do **not** include future prices, returns, or variables that would be unknown at prediction time.

Golden Rule: *If you wouldn't have known it back then, don't use it now!*

5 Conclusion

This project is intentionally ambitious and reflects the type of complex, uncertain challenges faced in real-world quantitative finance. Forecasting stock returns around earnings announcements is a notoriously difficult task—even for experienced professionals with access to advanced infrastructure and proprietary data.

As such, we do not expect perfect predictions or highly profitable backtests. Instead, our main objective is to evaluate your ability to approach a challenging problem with rigor, creativity, and methodological discipline. What matters most is the clarity of your reasoning, the quality of your implementation, and your capacity to derive insights from data—even when the results are inconclusive.

We encourage you to treat this project as an opportunity to apply the skills you've developed throughout the course, explore new ideas, and reflect critically on the limitations of data-driven approaches in finance. A thoughtful, well-documented exploration is far more valuable than a black-box solution with uncertain validity.

Good luck—and enjoy the process!

References

- Snow, D. (2017). *Financial Event Prediction using Machine Learning.* Available at: https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=3481555
- Zura Kakushadze. (2016). *101 Formulaic Alphas.* The Journal of Financial Markets, 34, 59–74. Available at: https://ssrn.com/abstract=2701346
- *TA-Lib: Technical Analysis Library.* A widely used open-source library for technical indicators. Documentation available at: https://mrjbq7.github.io/ta-lib/
- Ferhat Akbas, Ekkehart Boehmer, Sorin M. Sorescu (2001) *Short Interest, Returns, and Fundamentals.* Documentation available at: https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID2327365_code358101.pdf?abstractid=2216919&mirid=1
- Andreas Disch (2001)*Dispersion in Analyst Forecasts and the Profitability of Earnings Momentum Strategies* Documentation available at: http://momentum.technicalanalysis.org.uk/Disc01.pdf