Project 2: Final Project

Instructor: Yuan Yao   *Due: 23:59 Saturday 6 Dec, 2025*

1 Project Requirement

This project as a warm-up aims to explore basic techniques in machine learning.

1. Pick up ONE (or more if you like) favourite dataset below to work. If you would like to work on a different problem outside the candidates we proposed, please email course instructor about your proposal.

2. Team work: we encourage you to form small team, up to **FOUR** persons per group, to work on the same problem. Each team just submit

   (a) *ONE report, with a clear remark on each person's contribution*. The report can be in the format of either a *poster*, e.g.

   ```
   https://github.com/yuany-pku/2017_math6380/blob/master/project1/DongLoXia_
                                 poster.pptx
   ```

   or *technical report within 8 pages*, e.g. NIPS conference style (preferred format)

   ```
   https://nips.cc/Conferences/2019/PaperInformation/StyleFiles,
   ```

   with source codes such as Python (Jupyter) Notebooks with a detailed documentation.

   (b) *ONE short presentation video within 5 mins*, e.g. in Youtube or Bilibili link. You may submit your presentation slides together with the video link to help understanding.

3. For Kaggle contests, please register your team with name in the format of mafs5440_lastname, so that we could easily find your results on Kaggle. For example, a team with Shawn Zhu and Kate Wong would be named by mafs5440_Zhu_Wong.

4. In the report, show your proposed scientific questions to explore and main results with a careful analysis supporting the results toward answering your problems. If possible, you should include your Kaggle contest score or rating in the report. Remember: scientific analysis and reasoning are more important than merely the performance tables. Separate source codes may be submitted through email as a GitHub link, or a zip file.

5. Submit your report by email or paper version no later than the deadline, to the following address (aifin.hkust@gmail.com) with a title "Mafs5440: Project 2"

2   QRT Challenge: Learning Factors for Stock Market Returns Prediction

Description of this challenge can be found at

> https://challengedata.ens.fr/challenges/72#menu

Files are accessible when logged in and registered to the challenge.

3   Kaggle: Jane Street Real-Time Market Data Forecasting

3.1   Project Overview

Modeling problems in modern financial markets are inherently complex due to unique challenges such as fat-tailed distributions, non-stationary time series, and data that often violate assumptions of standard statistical methods.

In this competition, you are tasked with building a model using real-world data derived from Jane Street's production systems, providing a glimpse into the complexities of modern trading. The dataset includes features and responders related to markets where automated trading strategies are employed, highlighting the importance of robust modeling. To balance relevance and the proprietary nature of trading, some features and responders have been anonymized and lightly obfuscated. Despite these adjustments, the essence of the problem remains intact, offering a meaningful and challenging task that reflects the real-world work done at Jane Street.

> https://www.kaggle.com/competitions/jane-street-real-time-market-data-forecasting/overview

3.2   Data Description

You are provided real-time market data from Jane Street with anonymized and lightly obfuscated features and responders.

The competition will proceed in two phases:

1. A model training phase (11:59 PM UTC Oct 14, 2024 - 11:59 PM UTC Jan 13, 2025) with a test set of historical data. This test set has about 4.5 million rows.

2. A forecasting phase (after 11:59 PM UTC Jan 13, 2025) with a test set to be collected after submissions close. You should expect this test set to be about the same size as the test set in the first phase.

During the final weeks of the model training phase, the public test set will be extended to include data closer to the submission deadline. At the start of the forecasting phase, the unscored public test set will be extended up to the final day of the model training phase and the private set updated roughly every two weeks.

Since the Team Merger deadline (January 6, 2025) has passed, which is the last day participants may join or merge teams, you may not join as new participant teams. The final submission deadline (January 13, 2025) is also passed, then you may not be evaluated using the real time test data provided by the Kaggle contest. If you had not participated the contest before, you may only download the training data to play with your own algorithms.

If you choose this project but did not join the competition, you may download the data from: `https://drive.google.com/file/d/1kIB_ZP1dxEl7-YLc2BeWcAF1KBMHqNDR/view?usp=sharing` by courtesy of Mr. Tim Zetian Lu.

4 Kaggle: M5 Forecasting

There are two complementary competitions that together comprise the M5 forecasting challenge:

- Accuracy, Estimate the unit sales of Walmart retail goods. Can you estimate, as precisely as possible, the point forecasts of the unit sales of various products sold in the USA by Walmart?
  `https://www.kaggle.com/c/m5-forecasting-accuracy`

- Uncertainty, Estimate the uncertainty distribution of Walmart unit sales. Can you estimate, as precisely as possible, the uncertainty distribution of the unit sales of various products sold in the USA by Walmart?
  `https://www.kaggle.com/c/m5-forecasting-uncertainty`

How much camping gear will one store sell each month in a year? To the uninitiated, calculating sales at this level may seem as difficult as predicting the weather. Both types of forecasting rely on science and historical data. While a wrong weather forecast may result in you carrying around an umbrella on a sunny day, inaccurate business forecasts could result in actual or opportunity losses. In this competition, in addition to traditional forecasting methods, you're also challenged to use machine learning to improve forecast accuracy.

In this competition, you will use hierarchical sales data from Walmart, the world's largest company by revenue, to forecast daily sales for the next 28 days and to make uncertainty estimates for these forecasts. The data, covers stores in three US States (California, Texas, and Wisconsin) and includes item level, department, product categories, and store details. In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust dataset can be used to improve forecasting accuracy.

5 Large Language Models + Financial Analysis

5.1 Introduction to LLMs in Financial Analysis

Large Language Models (LLMs) utilize transformer architectures and are trained on massive datasets, enabling them to recognize, translate, predict, or generate text and other content. In the financial

domain, LLMs have demonstrated remarkable capabilities in automating and streamlining complex analysis tasks. These models can process vast amounts of heterogeneous data, including news articles, research reports, financial statements, earnings calls, and real-time market data, to extract valuable insights and identify patterns that might be difficult for human analysts to detect at scale.

5.2 Recommended Topic: AI-Powered Hedge Fund with Multi-Agents

For this project, we recommend exploring the AI Hedge Fund repository (`https://github.com/virattt/ai-hedge-fund`), a highly acclaimed open-source project with over 42,000 GitHub stars that demonstrates the practical application of LLMs in investment management.

The AI Hedge Fund project implements a multi-agent architecture where different AI agents simulate the investment philosophies and decision-making processes of renowned investors. The system employs twelve specialized investor agents, each embodying distinct investment philosophies:

- **Value Investing Agents**: Ben Graham (margin of safety focus), Charlie Munger (wonderful businesses at fair prices), Michael Burry (contrarian deep value), and Mohnish Pabrai (low-risk doubles)

- **Growth Investing Agents**: Cathie Wood (innovation and disruption), Phil Fisher (growth through research), and Peter Lynch (ten-baggers in everyday businesses)

- **Strategic Investors**: Bill Ackman (activist investing), Stanley Druckenmiller (macro and asymmetric opportunities), and Rakesh Jhunjhunwala (Indian market specialist)

- **Analytical Specialists**: Aswath Damodaran (valuation expert) and Warren Buffett (quality at reasonable prices)

Additionally, the system incorporates four analytical agents (Valuation, Sentiment, Fundamentals, and Technicals), a Risk Manager, and a Portfolio Manager that orchestrates final trading decisions.

5.3 Recommended Project Directions

5.3.1 Direction 1: Implement a New Investment Agent

Develop and integrate a new AI agent based on your favorite investor or a unique investment philosophy. Your task includes:

- Research and codify the investment principles of your chosen investor (e.g., Ray Dalio's principles-based approach, George Soros's reflexivity theory, or David Tepper's distressed debt strategies)

- Implement the agent's decision-making logic using appropriate prompting strategies

- Integrate the new agent into the existing multi-agent framework

- Conduct backtesting to evaluate the agent's performance across different market conditions

- Compare your agent's decisions with existing agents to understand philosophical differences

The teaching assistant has developed a Streamlit-based user interface `https://github.com/fuxiaoyi/ai-hedge-fund-plus` that allows you to visualize agent decisions, compare strategies, and analyze portfolio performance metrics, making it easier to evaluate and present your results.

5.3.2 Direction 2: Novel Financial Applications

Extend the AI hedge fund framework to create innovative applications in financial investments. Your creativity is encouraged, but potential directions include:

- **Sector-Specific Analysis**: Develop specialized agents for specific sectors (e.g., biotech, renewable energy, cryptocurrency) with domain-specific knowledge

- **Risk Management Systems**: Create advanced risk assessment agents that can identify portfolio vulnerabilities and suggest hedging strategies

- **ESG Integration**: Implement environmental, social, and governance factors into investment decision-making processes

- **Alternative Data Integration**: Incorporate satellite data, social media sentiment, or patent filings into investment analysis

- **Market Regime Detection**: Develop agents that can identify changing market conditions and adapt strategies accordingly

Previous student projects have included implementations of financial new analysis and fundamental analysis, of which presentations can be found below:

- `https://www.youtube.com/watch?v=4nkZgyifR_s`

- `https://youtu.be/Lk8iM29FGcI?si=Pgf9oBBw4hH_YwEd`

Your application should demonstrate both technical sophistication and practical relevance to financial markets.

5.4 Submission

You are required to submit your report, which should include a demonstration of your *demo application* and the necessary *analysis*. Additionally, please provide the source materials in a zip folder. The zip folder should contain the main logic codes and other essential files, excluding large data files such as checkpoint weights.

5.5 Additional References

- TradingAgents: Multi-Agents LLM Financial Trading Framework (Paper) `https://arxiv.org/pdf/2412.20138`

- TradingAgents Framework - `https://github.com/TauricResearch/TradingAgents`

- Financial Agent - GitHub Repository `https://github.com/virattt/financial-agent`

- Financial Analyst App - GPT-4, Streamlit, and Llama-Index Tutorial `https://youtu.be/c_LAIkkhKts?si=zbyANC91SymDJuV3`

- LlamaIndex starter tutorial `https://gpt-index.readthedocs.io/en/latest/getting_started/starter_example.html`

- HKUST Azure OpenAI API service `https://itso.hkust.edu.hk/services/it-infrastructure/azure-openai-api-service`

- Other API services: `https://api.bianxie.ai`, AI/ML API:`https://aimlapi.com`

- FinGPT-Forecaster Huggingface Space Demo `https://huggingface.co/spaces/FinGPT/FinGPT-Forecaster`

- Huggingface Generation with LLMs tutorial `https://huggingface.co/docs/transformers/llm_tutorial`

6 Paper Replications

6.1 *(Re-)Imag(in)ing Price Trends*

6.1.1 Background

We are targeting to replicate the following paper by Jingwen Jiang, Bryan Kelly and Dacheng Xiu: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3756587`.

This paper explores convolutional neural networks that flexibly learn price patterns as images that are most predictive of future returns. The raw predictor data are images – stock-level price charts, from which authors model the predictive association between images and future returns using a convolutional neural network (CNN). They claims that by using CNN they can automatically identify context-independent predictive patterns which can gave more accurate return predictions, translate into more profitable investment strategies and are robust to variations.

In the empirical designs, they first embeds 1D time series data in a higher dimensional space, representing it as a 2D image depicting price and volumes. Then they feed each training sample into CNN to estimate the probability of a positive subsequent return over short (5-day), medium (20-day) and long (60-day) horizons. Afterwards, they use CNN-based out-of-sample predictions as signals in a number of asset pricing analyses. Finally, they attempt to interpret the predictive patterns identified by the CNN.
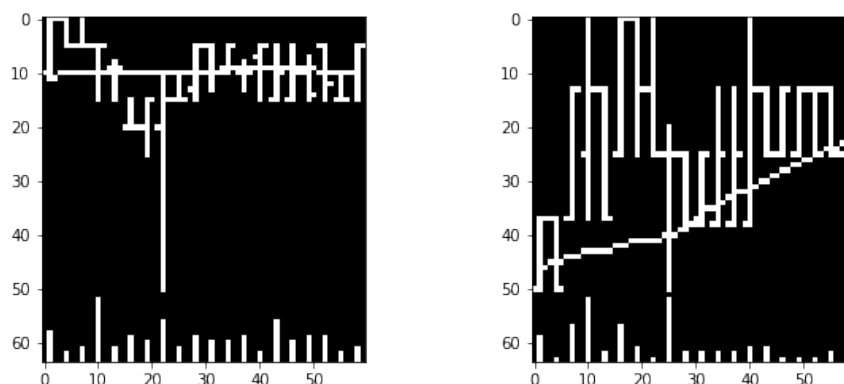
Figure 1: Examples of 20-day Image with volume bar and moving average line

6.1.2 Replication studies

In this reproduction process, we mainly focus on understanding the data preparation (how to transfer 1D time series data to 2D images representing historical market data), model design (CNN architecture design and mechanism behind it), workflow design (from training to model tuning and finally to prediction), performance evaluation and finally the interpretation part.

1. **Data**

   The sample runs from 1993-2019 based on the fact that daily opening, high, low prices. In the original paper, authors construct datasets consisting three scale of horizons (5-day, 20-day, 60-day), Here we just collect the 20-day version. The total size of data is 8.6G in a zipped file (802.9MB). The download link of data is:

   `https://dachxiu.chicagobooth.edu/download/img_data.zip`

   or a fast access

   `https://www.dropbox.com/s/njehqednn8mycze/img_data.zip?dl=0`

   with iPython image processing demo in

   `https://dachxiu.chicagobooth.edu/download/img_demo.html`.

   We already transformed the OHLC charts into images following the same procedures introduced in the paper (Section 2). Current images have the same resolution (64 * 60) and added with moving average lines(MA) and volume bars(VB). Some example figures is shown in Figure 1.

   Images labels take value **1** for positive returns and **0** for non-positive returns. In addition, we use **2** to mark the NaN value. In the simplest terms, you need to complete a two-class classification problem, and use the CNN model to predict whether the trend is 'down' or 'up' for the input image. For detail of data and label file, please refer to appendix.
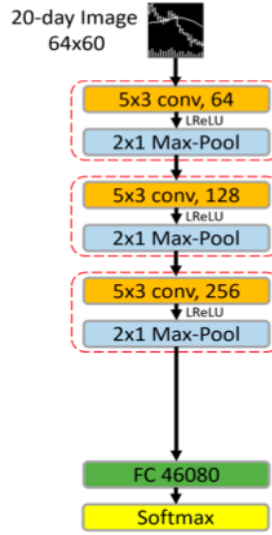
2. **Architecture Design**

Figure 2: Diagram of CNN model

Why use CNN? Since CNN impose cross-parameter restrictions that dramatically reduce parameterization and embed a number of tools that make the model resilient to deformation and re-positioning of important objects in the image. A core building block consists of three operations: convolution, activation and pooling. In the paper, for 20-day image, they build a baseline CNN architectures with 3 conv blocks and connected with a fully connected layer as a classifier head. You should refer to the design of the conv block in the original paper (including the selection of the size of the convolution kernel, the selection of the convolution method, the design of the pooling layer and the selection of the activation function, etc) Figure 2 shows a diagram of 20-day CNN model proposed in the paper, just for your reference.

3. **Working Flow**

   Data split First, consider dividing the entire sample into training, validation, and testing samples. In the original paper, they use the first seven-year sample (1993-1999) to train and validate the model, in which 70% of the sample is randomly selected for training and the remaining 30% for validation. The remaining twenty years of data comprise the out-of-sample test dataset. You should consider following the same format in case better compared with the original paper.

   Loss and evaluation You can simply treat the prediction analysis as a classification problem. In particular, the label for an image is defined as y = 1 if the subsequent return is positive and y = 0 otherwise. The training step minimizes the cross-entropy loss, which is the standard objective function for classification problem, which define as:

   $$L_{CE}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y)log(1 - \hat{y})$$

   where $\hat{y}$ is the prediction and y is the ground truth.

   To measure the classification accuracy, a true positive (TP) (true negative (TN)) occurs when a predicted probability of greater than 50% coincides with a positive realized

return (a probability less than 50% coincides with a negative return, respectively). False positives and negatives (FP and FN) are the complementary outcomes. We calculate classification accuracy as:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

For more evaluation metrics or methods, like Sharpe Ratio, please refer to the original paper.

Training process The author adopts several ways to combat the over-fitting issue and aid efficient computation. For example, they applied the Xavier initialization for weights in each layer, which guarantees faster convergence by scaling the initial weights. Other techniques like dropout, batch normalization, and early stopping may also improve performance. We recommend referring to the training details mentioned in paper 3.3 when training the baseline model.

4. **Extensions**

- For ablation studies and testing robustness, we suggest you follow what original paper mentioned in Appendix B. For example, you can perform the same sensitivity analysis of the CNN prediction model to alternate choices in model architecture (e.g. varying the number of filters in each layer or varying the number of layers, like the paper shows in Table 18)

- Another direction that can be used as an extension is exploring of the interpretability of the CNN model in Chapter 6 of the original paper. Though interpreting a CNN model is quite difficult due to its stacks of non-linear structures, you can imitate what the author did in Part 6.3, using a visualization method (Grad-CAM) to understand how different image examples activate the regions of the CNN to trigger 'up' or 'down' return predictions.

- What's more, we encourage you not being limited to simple binary classification task, since the label files we provided consist more meaningful attributes, containing both categorical and numerical values. For example, you can use the same 20-day horizon images to train your model to predict the return trend of different subsequent **y**-days even the detailed return values. (**y** can be 5, 20 even larger). In this way, you can prove more firmly that using CNN can automatically identity robust and transferable predictive features.

6.2 Empirical Asset Pricing via Machine Learning

The fundamental goal of asset pricing is to understand the behavior of risk premiums. However, risk premium is difficult to measure: market efficiency forces return variation to be dominated by unforecastable news that obscures risk premiums. This paper predicts the expected return and identifies informative predictor variables via machine learning methods, which facilitates more reliable investigation into economic mechanisms of asset pricing. Now you are required to replicate some results of this paper based your understanding of it, and write a report about your work.

The requirements of this paper replication project are as follow:

- The machine learning methods used in this paper include linear regression (OLS, elastic net), dimension reduction (PLS, PCR), generalized linear models, trees (gradient boosting trees, random forest) and neural networks. Please try to replicate **at least 6 methods** of them (e.g., OLS, elastic net, PLS, PCR, random forest, neural networks, etc. Please note that if you choose OLS, OLS-3 should also be included; and if you choose neural network, NN1 to NN5 are included. Besides, robust loss function should also be considered. See details in the paper), and analyze your results specifically. Hints on parameter choice are presented in the paper.

- Include the variable importance (section 2.3 of the paper) in your analysis. You do not need to replicate all the figures in section 2.3, but you are encouraged to investigate it carefully.

- Note that this paper uses a 'recursive performance evaluation scheme'. You are also required to evaluate your result by this method. For more details of this method, please refer to the paper and its supplementary material. The PPT presented in class about this project may also be helpful to you.

- As you can know from the paper (section 2.1), predictive characteristics include firm characteristics, sic code and macroeconomic predictors. Firm characteristics and sic code are provided in the original dataset of this paper, and the 8 macroeconomic predictors are constructed following Welch and Goyal (2008), which are not directly provided in the original dataset of this paper. Hence, you may construct the predictors by yourself according to the description in Welch and Goyal (2008), for instance, see
`https://christophj.github.io/replicating/r/replicating-goyal-welch-2008/`.

- The portfolio forecast part of the paper (section 2.4) is not compulsory for you to replicate.

You may access the paper and the supplementary material via:
`https://dachxiu.chicagobooth.edu/download/ML.pdf`
or
`https://academic.oup.com/rfs/article/33/5/2223/5758276`.
Meanings of characteristics of the data are provided in the supplementary material.

The original dataset (4.05GB)[1] can be obtained at
`https://dachxiu.chicagobooth.edu/download/datashare_OLD.zip`.
The zip file is about 1.64GB. Please be patient since it may take you about 6 hours to download the data. Another fast access can be via
`https://www.dropbox.com/s/zzgjdubvv23xkfp/datashare.zip?dl=0`

7   Old Kaggle Contest in Project 1: Home Credit Default Risk

Many people struggle to get loans due to insufficient or non-existent credit histories. And, unfortunately, this population is often taken advantage of by untrustworthy lenders.

---

[1]An updated dataset was posted on Jun 23, 2025, which seems missing some factors and return values and just for your reference: `https://dachxiu.chicagobooth.edu/download/datashare.zip`

Home Credit strives to broaden financial inclusion for the unbanked population by providing a positive and safe borrowing experience. In order to make sure this underserved population has a positive loan experience, Home Credit makes use of a variety of alternative data–including telco and transactional information–to predict their clients' repayment abilities.

While Home Credit is currently using various statistical and machine learning methods to make these predictions, they're challenging Kagglers to help them unlock the full potential of their data. Doing so will ensure that clients capable of repayment are not rejected and that loans are given with a principal, maturity, and repayment calendar that will empower their clients to be successful.

Visit the following website to join the competition.

`https://www.kaggle.com/c/home-credit-default-risk/`