Machine-Learning in Finance
Project Instructions
Deadline : 11th of June 23:59

1 Introduction

You are hired as a team of external consultants for a hedge fund that wants to create a new machine-learning-based fund. Your task is as follows: you are given two types of datasets, **return datasets** and **predictor datasets**. The Hedge Fund wants you to find a Machine Learning model to predict **one** of the **returns datasets** using **one or many** of the predictor datasets.

2 Guidelines

You are free to use any machine learning approaches that we have studied in the class. You can also use other approaches if you wish, but **at least one of the approaches in the report has to use deep learning.** For each dataset, you can either use the following.

- **Regression Task**: Let $R_{i,t+1}$ be the return on stock $i = 1, .., P$ at time $t + 1$. Using some predictors X_t, available at time t, build a model such that

$$R_{i,t+1} \approx f(X_t; \theta)$$

  Note that you could also want to predict the whole cross-section of returns directly. Building

$$R_{t+1} \approx f(X_t; \theta) \in \mathbb{R}^P$$

- **Classification task**: Let $R_{i,t+1}$ be the return on the stock $i = 1, .., P$ at time $t + 1$. Create $c = 1, ..., C$ bins of returns and, using some predictors X_t, available at time t, build a model such that

$$P(R_{i,t+1} \in c) \approx f(X_t; c) \tag{1}$$

3 Datasets

We provide many different datasets to use to carry out your projects. They can be downloaded from this Google drive.

3.1 Returns dataset

First, here are the five return datasets that you can use. Those would be the targets you are trying to predict.

- **Monthly CRSP**: This dataset contains US monthly stock returns from December 1925 to December 2024. It contains 10 important columns:
  - *PERMNO*: Unique permanent identifier for a security assigned by CRSP.

  - *HdrCUSIP*: Header CUSIP code identifying the issuer of the security.
  - *CUSIP*: 8-character identifier for the specific security, often includes issuer and issue.
  - *Ticker*: The ticker symbol of the stock representing the security on the exchanges.
  - *TradingSymbol*: Trading symbol used in CRSP, sometimes more precise than the standard ticker.
  - *PERMCO*: Permanent company identifier in CRSP, it links together all PERMNOs belonging to the same firm.
  - *SICCD*: Standard Industrial Classification (SIC) code used to classify the firm's industry.
  - *NAICS*: North American Industry Classification System code, a modern classification for industries.
  - *MthCalDt*: Calendar month-end date for the observation in format YYMMDD. Note that the date corresponds to the return for the same month. For example, the return associated with 1986-12-31 is from the end of November to the end of December.
  - *MthRet*: Monthly return for the security (not adjusted for dividends or splits unless specified).
  - *sprtrn*: S&P 500 return for the same calendar month, used as a market benchmark.

- **Daily CRSP**: This dataset contains US daily stock returns from January 2000 to December 2024. It contains 10 important columns:
  - *PERMNO*: Unique permanent identifier for a security assigned by CRSP.
  - *HdrCUSIP*: Header CUSIP code identifying the issuer of the security.
  - *CUSIP*: 8-character identifier for the specific security, often includes issuer and issue.
  - *Ticker*: Stock ticker symbol representing the security on exchanges.
  - *TradingSymbol*: Trading symbol used in CRSP, sometimes more precise than the standard ticker.
  - *PERMCO*: Permanent company identifier in CRSP, links together all PERMNOs belonging to the same firm.
  - *SICCD*: Standard Industrial Classification (SIC) code used to classify the firm's industry.
  - *NAICS*: North American Industry Classification System code, a modern classification for industries.
  - *MthCalDt*: Calendar month-end date for the observation in format YYMMDD.
  - *MthRet*: Monthly return for the security (not adjusted for dividends or splits unless specified).
  - *sprtrn*: S&P 500 return for the same calendar month, used as a market benchmark.

- **10-minute frequency stock returns**: This dataset contains one month of intraday stock prices at the 10-minute frequency. It contains 4 columns that are important for your project:
  - *DATE*: Date in format YYYYMMDD.
  - *SYMBOL*: Stock symbol to which the other data are associated.
  - *TIME*: Intraday time of the data observation in format HH:MM:SS.
  - *MID_OPEN*: Mid-price (average of the best bid and best ask) at *TIME*

- **Daily Futures returns**: This dataset contains daily prices of various futures contracts.[1] The date column is in format YYYY-MM-DD and is the date of the associated price (at close).

[1] As you might know, futures expire. This dataset circumvents this problem by creating a continuous price using the front-month contract returns to create a "synthetic" price history.

3.2 Predictor datasets

Then, there are plenty of predictors that you can use to try to achieve your prediction task:

1. **Quarterly Compustat Firm Characteristics**: This dataset contains firm characteristics from income statements, balance sheets, cash-flow statements, and so on. The dataset has 256 columns, but not all of them are stock characteristics. As you learned, some dataset have many missing values, so you will have to create a way to pick the columns you believe to be meaningful and to take cares of duplicates and NaNs accordingly. Also, you might want to investigate the *dtypes* of each column, as some are strings. Here is a list of the important columns you might need :

   (a) *cusip*: This is a stock identifier that you can use to merge with the others datasets.

   (b) *datadate*: This is the date in format DD.MM.YYYY. The data on a given line are available **from this date on**.

   You can find the description of each column in the PDF on the Google Drive.

2. **Jensen, Kelly, and Pedersen (JKP) factors**: This dataset contains the returns of the 153 JKP factors at each date. It contains 3 important columns :

   - *date*: The date to which the other data are associated. The date has a "same line" approach. This means, for example, if the date is 31.05.1926 and the associated return is 5%, then it means that the return of the factor from the 30.04.1926 that ended on the 31.05.1926 had a 5% return.
   - *name*: This is the name of the given factor.
   - *n_stocks*: This represents the number of stock in the factor portfolio.
   - *ret*: This is the return of the given factor

3. **Chen-Zimmerman data** : This dataset contains monthly long-short return for 205 predictors. You can get the data from Monthly long-short returns for 205 predictors and get the relevant informations using the data dictionnary. Additional data might be found on the Fed's website

4. **Earnings Calls**: Earnings calls are quarterly conference calls hosted by publicly traded companies to discuss their financial results with investors and analysts. They usually follow the release of earnings reports (10-Q or 10-K) and include management commentary on performance, strategy, and guidance, followed by a Q&A session.

5. **10-K Reports**: A 10-K is an annual report filed by public companies with the SEC, providing a comprehensive summary of the firm's financial performance, business operations, risk factors, and management discussion. It includes audited financial statements and is more detailed than quarterly 10-Q filings. The part we are most interested in for this project is the *Management Discussion and Analysis (MD&A)* section, where executives provide qualitative insights into the company's past performance, strategic outlook, and perceived risks.

6. **Bonus: 8K Report**: An 8-K is a report that publicly traded companies in the United States must file with the Securities and Exchange Commission (SEC) to announce major events that shareholders should know about. Unlike quarterly (10-Q) or annual (10-K) filings, the 8-K is an unscheduled report, filed as needed, usually within four business days of the event. If you want to use those reports, you will have to scrape them yourself. You can get them from here.

4 Report instruction

Each group, consisting of 3 to 5 people, is required to submit a concise and well-structured report in PDF format, detailing the experimental setup, methodology, results, and conclusions of your project. The report must adhere to the following guidelines:

- The report must not exceed **7 pages**, excluding references and appendices (if any). Exceeding this limit will results in penalties.
- The document should be clear and logically organized, including (but not limited to) the following components:
  - Introduction and problem statement
  - Description of data preprocessing and feature engineering
  - Explanation of the predictive models used
  - Evaluation methodology and performance metrics
  - Discussion of results, including tables and figures where appropriate
  - Conclusion and possible extensions
- Figures and tables should be used effectively to illustrate key findings, and must be properly labeled and referenced in the text.
- Any external libraries or tools used must be clearly stated.

In addition to the report, students must submit a **well-structured and documented codebase**. The code should be clean, modular, and reproducible. It should be possible for the teaching team to run your experiments using minimal setup instructions (ideally from a README file). The quality of the codebase will be taken into account during grading and will have a non-negligible impact on the final evaluation.

5 Small projects examples

Here are small descriptions of projects you could do. Those descriptions are **not representative of what a whole project should be**, and just reproducing what is below will not give you full credit for the project. The goal is to give you some ideas of how one could proceed, and you should go from there to refine your analysis and models.

5.1 10-minute frequency stock returns

For this high-frequency dataset, the low-frequency (monthly/quarterly) predictors that we provide will not be of great help. Hence, you are forced to use past returns to try to predict future ones. One way could be to build a deep neural network (DNN) that uses the whole cross-section of past returns R_t to predict the whole section of future returns R_{t+1}. You would train your model by fixing a lookback window T and use $R_{t-\tau}, \tau = 1, ..., T$ as training data. Now, even if your model works, it might very well be that it can not generate any real-life money because of transaction costs or the bid-ask spread. Hence, one cool idea would be to implement a custom loss function that would refrain from excessive trading when using the model predictions.

5.2 Textual embeddings as predictors

Financial documents such as 10-K reports or earnings call transcripts contain valuable forward-looking information that traditional numerical predictors do not always capture. In this project, you could extract textual data from these documents and use natural language processing (NLP) techniques to obtain document-level embeddings. Embeddings are numerical vector representations of text that capture its semantic meaning. Unlike simple keyword counts, embeddings place similar texts close together in vector space, allowing machine learning models to understand context and tone better. For instance, two earnings call segments that express optimism may have similar embeddings even if they use different wording. A simple version could involve regression or classification with embeddings at the firm-month level, but you could also explore how these textual signals interact with known risk factors or whether traditional variables subsume them.

6 Miscellaneous Advice

Here are some recommendations to think about before starting:

- Investigate your data before training your models. Are there outliers? Are there values that do not make sense? Are all of your predictors of the same magnitude? Try to get a deep grasp of how you can make your data as clean as possible before training your model.
- Your goal is to provide predictions that would **actually** work. In that sense, data mining should be avoided as much as possible: please always split the data into train and test components. Implement techniques that can help you assess whether your model is robust. Being critical of your own work is the best way to avoid catastrophic outcomes when using your models on live data ![2]
- Recall that if something is easy to discover, then it is likely that people might have started trading on it and made the anomaly disappear. So be creative in your approach, it can be a real asset when looking for trading strategies.
- Some of those datasets are very large. You might want to try your methods on a subset of the data before launching your model for a big run. Moreover, when you use Python and load some data, it is loaded into your RAM (Readily Available Memory). This is a part of your computer that can be accessed rapidly to perform operations. Because some of those datasets are large, it might be that your RAM is overloaded (the dataset size is larger than your RAM). If this is the case you can use a subset of your whole dataset for you analysis. The function *pd.read_csv()* has an argument *nrows* that you can use to load the first *nrows* of the CSV file.
- An alternative would be to use Parquet files instead of CSV. Parquet is a columnar storage format that is much more efficient for large datasets, especially when you're only interested in a few columns or rows. It allows for faster reading and smaller memory usage compared to CSV. You can convert a CSV to Parquet using *pandas* with `df.to_parquet()` and read it back using `pd.read_parquet()`. If your dataset is large and you're facing memory constraints, this can significantly affect performance.
- The following Pandas function might be super useful when merging different datasets: *pd.merge_asof()*. It allows you to merge dataframes of various frequencies by appending the **most up-to-date** data from the right dataset to the left dataset.
- To merge the different datasets, you will need to use appropriate linking tables. For example, to connect SEC filings (such as 10-K reports) to Compustat data, you can use the Central Index Key (CIK), which is the unique identifier assigned by the SEC to each company. To link Compustat with CRSP data, you should use the CRSP/Compustat Merged (CCM) linking table available on WRDS. A helpful overview can be found at: WRDS Overview of CRSP/Compustat Merged CCM.

[2]There are many histories of funds that went bust because they had too much trust in what they found without proper risk management: See, for example, `https://en.wikipedia.org/wiki/Long-Term_Capital_Management`