**The Chinese University of Hong Kong**
**Econ5821 Data Science for Economists**
**Final Examination**

# 1 Instructions

This is the final exam for Econ5821. All communication should be sent to `zhentao.shi@cuhk.edu.hk`.

Each student joins a group of at most 4 people, and a group designates one person as the **group captain**. The group captain must inform the instructor before **11:59 pm, May 1** about his/her group members.

You can use any computing language(s) to complete the tasks. Group members are allowed to use all available non-human resources (including but not limited to books, papers, Internet, and chatGPT). However, a group must work independently. Communication with other human beings outside of the group (except the instructor if you have questions about the data) about the final question is strictly prohibited. Detected violations will be reported to University's disciplinary panel.

A remote Zoom presentation will be conducted on the afternoon of **May 12** (Friday). All group members must show up in the 20-minute Zoom session. Each group decides which student(s) speak in the presentation.

The codes and reports should be prepared in a private `Github` (or any other distributed git system) repository shared among its group members only. The grader will consider the last commit in the repository before **May 18, 11:59 pm** as the final training/validation outputs. No change of the code and the model should be made after that.

The instructor will post a new testing dataset online on the early morning of May 19, and each group will append the report with the out-of-sample forecast. The group captain must turn their working repository from a private one into a public one, and send the link of the public repository to the instructor by **May 19, 11:59 pm**.

The final code should be able to run with no errors from the start to the end. A human-readable report should accompany the code by describing the procedures and summarizing the results. The report can be merged with the code by literature programming, for example in the `ipynb` or `Rmd` format.

**Important dates**:

- Report group membership: May 1, 11:59 pm.
- Online presentation: May 12, afternoon (The exact time and Zoom link will be sent to the group captain by May 5).
- Send the instructor the URL link of the public repository: May 19, 11:59 pm.

## 2 Question

*This is an open question for you to get exposure to big data and machine learning methods.*

In this exercise, we want to use the Chinese macroeconomic data to forecast China's inflation. Inflation is about the price level. In China, the two major indices of the price level is either CPI (consumer price index) or PPI (Producer Price Index).

### 2.1 Data

All data are contained in `dataset_inf.Rdata`, which can be imported to your session by

```
load(url("https://github.com/zhentaoshi/Econ5821/raw/main/data_example/dataset_inf.Rdata"))
```

In this Rdata set there are 4 objects. The first column of all objects is `month`, indicating that the time dimension is recorded in monthly frequency from 1 (earliest) to 168 (latest in the revealed data). The same number in `month` in each data frame represents the same calendar month.

- `cpi`: A $168 \times 2$ data frame. The second column is the raw CPI level.
- `ppi`: A $168 \times 2$ data frame. The second column is the raw PPI level.
- `X`: A $168 \times 152$ data frame. Columns 2 to 152 are the potential predictors. The column names are in Chinese characters. In case your computer environment cannot correctly display Chinese characters, you can find the column names at https://github.com/zhentaoshi/Econ5821/blob/main/data_example/X_colnames.csv or load into your session by

```
read.csv("https://github.com/zhentaoshi/Econ5821/raw/main/data_example/X_colnames.csv")
```

This is a forecasting exercise. That is, $y_t$ should be predicted using the data happened in the past (up to period $t-1$). For example, if we want to predict the inflation rate at `month=168`, we can only use the information with `month<=167`. The inflation rate is computed as

$$\text{inflation\_rate}_t = \log y_t - \log y_{t-12}$$

where the gap of 12 months is used for year-on-year inflation rate, and $y_t$ is either the level of CPI or PPI.

### 2.2 Training

You will use the data of 168 months to train and validate machine learning models. The best performing algorithm for the inflation rate by CPI or by PPI may be different. Therefore, it is expected at you will produce at least two models—one tailored for CPI and the other for PPI. You have the freedom to choose any machine learning methods, no matter covered in class or not.

Since no single method fits all scenarios, you can also combine forecast methods if you want. If you experiment with many algorithms, you may choose to include and summarize some of them in the report and make comparisons.

You can rely on your prior knowledge and beliefs to choose a set of predictors before conducting statistical computing. You can transform the predictors as you wish, for example, with scaled normalized values, differenced values or lagged values; the lagged value of inflation rate can serve as predictors too. You have the freedom to choose the loss function, the length of estimation windows, length of validation windows, and cross validation schemes.

### 2.3 Testing

The 4th object in `dataset_inf.Rdata` is an artificial out-of-sample testing dataset named `fake.testing.X`, which is a $30 \times 152$ data frame with the first column as `month=169~198`. However, the other variables are all fake. This dataset is included as a placeholder for you to write the code to carry out the out-of-sample forecast; you do not need to care about the performance of your algorithm in this fake dataset at all.

The ultimate output of this exercise will be a function with the input `fake.testing.X` and/or `X,` and it will return 30 values as its forecast of the inflation rates according to CPI and another 30 values as its forecast of inflation rates according to PPI.

The `true.testing.X` will be shared with you in the early morning of May 19, to replace `fake.testing.X`. Then you will update the 60 forecast values (30 by CPI, and 30 by PPI).

## 3 Grading

Grading will based on the history, structure and information of the repository, the code, the written report, the presentation, and the forecast performance with the real testing data. The grader will rank the forecast performance by the out-of-sample R-squared (OOS $R^2$)

$$\text{OOS } R^2 = 1 - \frac{\widehat{\text{var}}_{\text{testing}}\left(y_t - \widehat{y}_t\right)}{\widehat{\text{var}}_{\text{testing}}\left(y_t\right)},$$

where $\widehat{\text{var}}_{\text{testing}}\left(\cdot\right)$ is the sample variance of the testing data.

Have fun with Machine Learning!