NHH

TERM PAPER
FIE463

Spring, 2025

Start: Wednesday, April 2, 09:00

End: Wednesday, April 16, 12:00

THE TERM PAPER SHOULD BE SUBMITTED IN **CANVAS**

Your candidate number will be announced on StudentWeb. The candidate number should be noted on all pages (not your name or student number). In case of group examinations, the candidate numbers of all group members should be noted.

Collaboration between individuals or groups on submission preparation, as well as exchange of self-produced materials between individuals or groups is prohibited. The answer paper must consist of individual's or the group's own assessments and analysis. All communication during the home exam is considered cheating. All submitted assignments are processed in Ouriginal, a plagiarism control system used by NHH. Use of artificial intelligence (AI) in this exam is determined by the course coordinator. Information on AI usage and guidelines can be found on the course's Canvas page.

SUPPLEMENTARY REGULATIONS FOR EXAMINATIONS

You can find supplementary regulations under the headline “Regulations”

https://www.nhh.no/en/for-students/regulations/

Find more information under chapter 4.0 in the Supplementary provisions to the regulations for fulltime study programmes

Number of pages, including front page: 10

Number of attachments: 1

Group term paper 2

FIE463: Numerical Methods in Macroeconomics and Finance using Python

Richard Foltyn
NHH Norwegian School of Economics

Deadline: Wednesday, April 16 at 12:00 (noon)

About the group term paper

- Your solution needs to be submitted on Canvas (not Wiseflow).
- Submissions are done in groups of 2-3 students. Cooperation across groups is not allowed.
- You are allowed to use all online resources for help, including generative AI. You must include a statement on how you used these resources to solve the tasks.
- Deadline: Wednesday, April 16 at 12:00 (noon). Late submission will not be accepted.

Requirements

- You should submit a *single* Jupyter notebook which contains your solution.
- Make sure your notebook runs without errors (`Restart` and `Run all`).
- State the approximate total run time of your notebook at the top.
- Your implementation *must* work with the Anaconda environment FIE463 created from the environment file `environment.yml` in the GitHub repository. See the guide on how to create this environment if you haven't already done so.
- Your notebook should be well structured and visually appealing, e.g.,:
  - Use markdown cells with headings and other formatting elements (see here and here).
  - Each part should be in a clearly distinguishable sub-section.
  - Each function should be defined in its own code cell.
  - Explanations should be written into markdown cells, not as Python comments.
  - Each graph should be in its own cell.
  - Graphs should contain legends and labels so they are easy to comprehend.
- Your code should satisfy the following criteria:
  - Use a seed of 1234 whenever RNG is involved.
  - Write code that is reusable and avoid duplication. An excellent submission should contain only one implementation of each function that is called repeatedly to solve different tasks.

    For example, parts 2 and 4 require a lot of plotting which should be performed by repeatedly calling the same one or two functions.
  - Write code that is efficient, e.g., it uses vectorization where applicable.
  - Your code should be commented to help the reader understand what it is doing.

- Be sure to provide an explanation for your results where applicable.

Please read the section on Performance issues **before** starting.

Assessment

Your term paper will be assessed according to the following criteria:

Component	Points
Part 1	10
Part 2	15
Part 3	30
Part 4	15
Part 5	25
Notebook runs without errors	5

Note that a correct result is not sufficient to obtain full points: code efficiency, code reuse, elegance of the solution, and the presentation of the results are also factored in.

Portfolio allocations in the Survey of Consumer Finances

Earlier in the course, you computed the optimal risky share following from the Merton-Samuelson model of portfolio choice. In this project, you are asked to analyze actual portfolio allocations observed in the data using the US Survey of Consumer Finances (SCF), a standard data set used in household finance research.

The SCF is administered by the Federal Reserve Board every three years. We will focus on the "modern" SCF waves from 1989–2022. The SCF is a repeated cross-section survey, i.e., households are not followed over time but each wave contains a potentially different set of households drawn from the US population. For most variables in the SCF, the unit of analysis is the household, so all income and wealth components are reported at the household level. Selected variables such as age, sex, education, employment status, and race are reported for the reference person (household head).

Because the aim of the SCF is to enable research on the wealthiest Americans, the sample contains an additional set of very wealthy respondents selected from the Forbes list and tax records. Because of this oversampling of the wealthy, any analysis of the SCF needs to use sampling weights (stored in the variable `weight`), as otherwise the statistics will not be representative for the US population.

In this project, you are asked to address two questions:

1. What fraction of US households *participate* in the stock market, either directly or indirectly? Which groups of the population are more likely to participate, and can we predict participation using classification models?
2. Conditional on participation, what *fraction* of their financial wealth do households hold in stocks and stock mutual funds (the so-called conditional *risky share*). Does the risky share vary within the population, and can we use regression models to predict it?

Part 1 — Data preprocessing

The data for this project comes in the form of 12 CSV files named `SCF_XXXX.csv` where `XXXX` is the survey year 1989, 1992, ..., and 2022. The last section of this document contains the detailed list of the 48 variables included in this data. Most variables are available for all survey years, but a few have been asked only in later years and should thus not be used in the analysis. Note that these files contain only a condensed and harmonized version of the variables available the original SCF files.

As a first step, write code to read in, process, and merge these files. For *each* year, perform the following steps:

1. Read in the data.
2. Create the variable `year` which stores the current survey year.
3. Restrict the sample according to the following criteria:
    1. Compute the 0.1 and 99th percentiles of the net worth distribution. Drop observations with a net worth outside of this range.

       Note: You need to compute *weighted* percentiles using the weight variable, e.g., using np.percentile().
    2. Only keep observations with ages between 20–80.
    3. Drop observations with negative total gross assets or wage income.
4. Because many functions don't support sampling weights, you need to create a representative sample suitable for unweighted analysis. To this end, draw 10,000 observations with replacement from the original sample for each survey year using the sampling weights. You can use `DataFrame.sample()` for this step.

5. For the variables `assets`, `networth`, `income`, `wageinc`, `liqassets`, `debt`, `houses`, and `finassets`, create a corresponding variable with the suffix `_rank` which contains a household's *rank* within the distribution for that particular survey year. This can be achieved using `rank()`.

   For example, for the `assets` variable, `assets_rank` should assign 0 to the household with the lowest total gross assets, 0.5 to the household with median assets, and 1 to the household with the highest level of assets in a given survey year.

   You can use these rank variables later when fitting models since these are numerically better behaved than dollar values.

Once you have processed all years, merge them into a single data set. This data should contain 120,000 observations.

Finally, create the following additional variables:

- An indicator variable `college` which is 1 whenever the reference person has a bachelor's degree or higher.
- An indicator variable `white` which is 1 whenever the reference person is white non-Hispanic.
- An indicator variable `part` which is 1 whenever the households holds more than 100 USD in `stocks` or `stkmutfnd` (stocks held via mutual funds).

  In this project, we'll refer to participation via directly held stocks or stocks held via mutual funds as *direct stock holdings*.
- An indicator variable `part_any` which is 1 whenever the household holds more than 100 USD in `equity`.

  Unlike `part`, the indicator `part_any` in addition takes into account indirect stock market participation via individual retirement accounts (IRAs), life insurances, or other managed financial assets where the household often does not make a conscious portfolio allocation decision. We'll refer to this broader participation concept as *any stock holdings*.

Part 2 — Participation: Exploratory data analysis

Before fitting a model of stock market participation using scikit-learn, you should first perform exploratory data analysis to get a rough idea about how different variables relate to stock market participation.

First, compute the correlations of all variables in the data set with the indicators `part_any` and `part`, and tabulate the 20 most correlated variables (don't include the variables `part`, `part_any`, `stocks`, `stkmutfnd` and `equity` in this analysis as these are obviously correlated and part of what we are trying to predict).

Second, create a series of graphs that plot the average `part_any` and `part` by:

1. The survey year.
2. The deciles of net worth (you can use the variable `networth_rank` to bin observations into deciles).
3. The deciles of total household income (you can use the variable `income_rank` to bin observations into deciles).
4. 5-year age bins (20-24, 25-29, . . . , 75-80).
5. Education level of the reference person.
6. The four combinations of marital status and sex of the reference person (single/male, single/female, couple/male, couple/female).
7. Race of the reference person.
8. Employment status of the reference person.
9. Willingness to take financial risk.

Each of these plots should contain both the average `part_any` and `part` within each subgroup (don't create separate plots for `part_any` and `part`).

Note that because you created a representative sample earlier, you don't need to take into account the survey weights to perform any of the tasks in this or later parts.

Part 3 — Predicting participation

After conducting the exploratory data analysis above, you should have a good idea which variables are most relevant to predict stock market participation. In this part, you are asked to use classifiers from scikit-learn to fit models to the data and evaluate the model performance.

Part 3.1 — Train-test split

As a first step, split the data into a training sample which will be used for model fitting and a test sample which will be used for model evaluation.

Assign 80% of the data to the training sample and the remainder to the test sample. To ensure that all survey years and outcomes are proportionally represented in both the training and the test data, you need to perform the train-test split using *stratification*. The strata in this case are defined by the combination of survey year and the `part` indicator, i.e., you should have a total of $12 \times 2 = 24$ unique strata. For example, an observation from 1989 with `part=0` should be in stratum 0, whereas an observation from the same year with `part=1` is in stratum 1. An observation from 1992 with `part=0` should be assigned to stratum 2, etc.

Create a new variable which contains the stratum for each observation, and tabulate the number of observations in each stratum. Then call `train_test_split()`, passing this variable as the argument to `stratify`.

Part 3.2 — Baseline classifier

With the test and training samples in hand, fit baseline models for *both* outcomes `part` and `part_any` which only use demographic information (`female`, `married`, `race`, `empl`, `educ`, `age`) and the survey year. You can add `age` and `year` as ordinal variables for this analysis.

- Fit the following models:
  1. Logistic regression without a penalty term
  2. Random forest
- For each model, answer the following questions and take your answers into consideration when fitting the model:
  - Does the model handle categorical variables or do these need to be encoded as dummy variables?
  - Does it make sense to include interactions of variables in the feature set?
  - Does the model require feature standardization?
  - Does the model have hyperparameters which need to be tuned?
- Report the hyperparameters used for fitting, if applicable. If cross-validation is required, choose a suitable metric (scoring function) and motivate your choice.
- For each outcome, evaluate the model performance on the test sample using at least the following metrics: accuracy, precision, recall, and F1.
- For each outcome, plot the confusion matrix computed on the test sample.
- How would you rate the models' ability to predict stock market participation? Which of the two outcome variables is harder to predict?

Part 3.3 — Classification with additional features

You are now asked to augment the baseline models from the previous part with any additional features from the data you find suitable (you are not allowed to use variables that directly imply stock market participation: `stocks`, `stkmutfnd`, `equity`, `part`, `part_any`).

- Rerun the estimation for both the logistic regression (this time with regularization) and the random forest. Depending on which features you chose, it might make sense to run cross-validation for the logistic regression.
- Report the hyperparameters used for fitting, if applicable. If cross-validation is required, choose a suitable metric (scoring function) and motivate your choice.
- For each outcome, evaluate the model performance on the test sample using at least the following metrics: accuracy, precision, recall, and F1.
- For each outcome, plot the confusion matrix computed on the test sample.
- How would you compare the predictive ability of the augmented models to the baseline specification? Which model performs best?

Provide a table which reports the above metrics for all classification models you estimated to give the reader an overview of your results.

Part 4 — Risky share: Exploratory data analysis

You are now asked to examine the risky share, i.e., the fraction of financial assets invested in stocks (directly or indirectly) *conditional on participation*.

For this analysis, retain only the sub-sample of observations with `part = 1`, i.e., households who directly hold stocks or stock mutual funds. We focus on these households since they are more likely to make a conscious risky allocation choice compared to households who only hold risky assets via retirement accounts or similar. This should leave you with approximately 25,000 observations (depending on the RNG when drawing the initial sample).

Create a new variable `risky_share` which is the ratio of equity (`equity`) and total financial assets (`finassets`).

The perform the same exploratory data analysis you did in Part 2:

- Tabulate the correlations with the 20 most correlated variables (exclude `risky_share`, `equity`, `stocks`, and `stkmutfnd`)
- Create the same sequence of graphs as you did in Part 2. Each graph should show the average conditional risky share split by the same grouping variables used in Part 2.

Part 5 — Predicting the conditional risky share

After conducting the exploratory data analysis above, you should have a good idea which variables are most relevant to predict the conditional risky share. In this task, you are asked to fit regression models from scikit-learn and evaluate the model performance.

As you will have noticed, the conditional risky share varies much less than participation, so don't expect the models in this part to perform overly well.

Part 5.1 — Train-test split

Assign 80% of the data to the training sample and the remainder to the test sample. To ensure that all survey years are proportionally represented in both the training and the test data, perform a stratified train-test split using the survey year as the stratifying variable.

Part 5.2 — Baseline predictor

To establish a baseline against which to evaluate future models, fit the intercept-only linear regression model to the data. Compute and report the RMSE on the test sample.

Part 5.3 — Prediction with additional features

Augment the baseline model from the previous section with any additional features from the data you find suitable (you are not allowed to use variables that directly predict the risky share: `stocks`, `stkmutfnd`, `equity`).

- Fit a linear regression model without any regularization.
- Fit at least *two* additional models implemented in scikit-learn. Restrict you attention to regression models we have covered in course, or those that are closely related:
  - Principal component regression
  - Ridge regression
  - Lasso
  - Elastic net (we haven't covered elastic net, but it is a straight-forward combination of Ridge regression and Lasso)

  Note that for Lasso and Elastic net, you might need to increase the maximum number of iterations if you get warnings about convergence, e.g., `max_iter=10_000` or even higher.
- You can use any (polynomial) interactions of variables.
- Report the hyperparameters used for fitting, if applicable.
- Evaluate the model performance on the test sample using the RMSE and R^2.
- Which model performs best in terms of prediction? How does it compare to the baseline specification? Add a table which contains the RMSE and R^2 for all regression models you estimated to give the reader an overview of your findings.

Performance issues

With scikit-learn it is easy to run into performance issues, for example when doing cross-validation. If you are unable to solve tasks because they take too long or you run out of memory, try the following:

- Use 5 folds for cross-validation instead of 10.
- Reduce the number of candidate values considered for cross-validation.
- For `LassoCV` and `ElasticNetCV`, experiment with the setting `selection='random'` instead of the default value as this might lead to faster convergence.
- Use fewer features (e.g., fewer polynomial interactions, lower maximum polynomial degrees, or fewer ordinal variables that are encoded as dummies).
- Use simpler categorical variables with fewer possible values that retain most of the information, e.g., `white` instead of `race`, `college` instead of `educ`, `lfpart` instead of `empl`.
- Temporarily lower the number of drawn samples from 10,000 per survey year to 5,000 while you are working on the project, and switch back to 10,000 for the final submission (allow for enough time to recompute the entire notebook with 10,000 observations).
- Set the `n_jobs` argument accepted by many scikit-learn function according to your hardware:
  - `n_jobs=K` executes `K` tasks in parallel, which speeds up computation. `K` should not be larger than the number of cores on your CPU, e.g., `K=4` or `K=8` for most modern hardware.
  - `n_jobs=-1` uses *all* your CPU cores, if possible.

  However, larger `K` increase the memory consumption considerably, so if you run your of memory (RAM), you need to lower `K`.
- Don't waste time trying to fit models with regularization (Logistic regression, Ridge regression, Lasso, Elastic net) to a regressor matrix that contains only categorical variables in the form of dummy variables. Regularization doesn't do much for these types of features (just like we don't regularize the intercept).

Data description

Variables

Legend for variable type:

`C` – continuous; `I` – indicator (0 or 1); `N` – nominal categorical; `O` – ordinal categorical

Variable	Type	Description
weight	C	Survey weight
age	C	Age of reference person
female	I	Indicator: reference person is female
married	I	Indicator: married/living with partner
educ	N	Education of reference person (1 = no high school/GED, 2 = high school or GED, 3 = some college or Assoc. degree, 4 = Bachelors degree or higher)
kids	C	Number of children
race	N	Race/ethnicity of reference person (1 = White non-Hispanic, 2 = Black/AA non-Hispanic, 3 = Hispanic, 4 = Other)
lfpart	I	Indicator: Labor force participation
empl	N	Employment status of reference person (1 = working for someone else, 2 = self-employed, 3 = retired/disabled, 4 = out of the labor force and other non-working)
income	C	Household income

Variable	Type	Description
wageinc	C	Household income from wages
ssretinc	C	Household income from SS/retirement
intdivinc	C	Household income from interest/dividends
busfarminc	C	Household income from business/farm
capgainsinc	C	Household income from capital gains
othinc	C	Other household income, incl. transfers
owner	I	Indicator: Homeowner
rent	C	Monthly rent: all housing types
hpayday	I	Indicator: Has payday loans
bankrupt5y	I	Indicator: Bankruptcy in last 5 years
latepay60d	I	Indicator: Payments more than 60 days due in last year
creditapplied	I	Indicator: Applied for credit (in the last 12 months/5 years)
creditdenied	I	Indicator: Credit application turned down
takefinrisk	O	Willingness to take financial risk (0 = None, 1 = Average/above average, 2 = substantial)
finknowledge	O	Knowledge of personal finance (-1 = not knowledgable at all, 10 = very knowledgable)
finlit	O	Financial literacy score (number of correct answers, 0-3)
equity	C	All directly and indirectly owned equity (shares of publicly traded companies)
finassets	C	Financial assets
liqassets	C	Liquid assets
houses	C	Value of primary residence
business	C	Value of businesses
vehicles	C	Value of vehicles
assets	C	Total gross assets
networth	C	Net worth
stocks	C	Value of directly held stocks
stkmutfnd	C	Value of stock mutual funds
othresre	C	Other residential real estate
netnresre	C	Net equity in other non-residential real estate
othnfin	C	Other non-financial assets
mortages	C	Mortgages excluding HELOC
heloc	C	Home equity line of credit (HELOC)
othloc	C	Other lines of credit
instloanveh	C	Installment loans: vehicles
instloaneduc	C	Installment loans: education
instloanoth	C	Installment loans: other
ccbalance	C	CC balance after last payment
debt	C	Total debt
totloanpay	C	Total monthly loan payments

Reference

- URL: https://www.federalreserve.gov/econres/scfindex.htm
- DOI Identifier: https://doi.org/10.17016/8799
- Creator: Board of Governors of the Federal Reserve Board
- Name: 2022 Survey of Consumer Finances
- Description: The Survey of Consumer Finances (SCF) is normally a triennial cross-sectional survey of U.S. families. The survey data include information on families' balance sheets, pensions, income, and demographic characteristics.
- Publisher: Board of Governors of the Federal Reserve System
- Publication Year: 2023