Aalto University
Industrial Engineering and Management
TU-E2231 Machine learning in financial risk management
Spring 2025
Kaila, Toepfer

Exercise 3
This exercise is graded passed/failed. The early-bird deadline is Friday, 28.2.2025. The final deadline is 16.4.2025. You can answer in any main European language.

Return 2 files:

1. A PDF file with
    - Individual work (one page)
    - Answers to questions in the Python exercises 3.1, 3.2 and 3.3 by dePrado and 3.4 (fraud detection). Do not answer directly in the Python-files.
    - A screenshot of the last picture in the Python exercise 3.4 (fraud detection)
2. A PDF file of the entire Python file 3.4. (see the instructions at the end for guidance on converting Python files to PDF)

1. Questions based on Lecture 3
individual work, length about one page. If you do the 3-credit version, you can choose all questions or focus on just a few. If you do the 5 or 6 credit version, answer every question.

1. What is the ADF test?
2. Read from de Prado 4.2 Overlapping outcomes and 7.3 Why K-fold CV fails in finance. Explain your own words and examples when the outcomes might be overlapping and why the information might leak from training group to testing group. dePrado's e-book Advances in financial Machine Learning is available at the Aalto learning center.
3. Give examples on using bagging and boosting in finance (stock price prediction, portfolio management, credit risk assessment, algorithmic trading). Think independently first. Then, instead of asking ChatGPT, search Google. And finally, you can complete your answer with ChatGPT (in your own words).
4. What are Exhaustive cross-validation methods? When are they used in finance, and what are their advantages?
5. How do you choose between Lasso and Ridge regularization in finance?

2. Python exercises 3

3.1 An exercise on stationary and non-stationary time series. (de Prado) In this exercise, make sure in the beginning that you understand how we can test the stationarity of a time series with the ADF test and p-values. Then, you can read on fractional differencing in the end of this paper.
3.2 A short exercise on ensemble methods (de Prado)
3.3 An exercise on cross-valuation (de Prado)
3.4 An exercise on fraud detection.
———————————-

Theory needed in Python exercise 3.1

Fractional differencing

Fractional differencing is an alternative to integer differencing, which unnecessary removes memory. You have used the exponential moving average EMA, which gives different weights to old observations. Similarly, in the fractional differencing, old observations are given different weights.

Let B denote the lag operator, i.e. $BX_t = X_{t-1}$ for $t > 1$ and some time series $X = \{X_1, X_2, ...\}$. Element-wise differencing of first order can then be expressed as

$$(1 - B)X_t = X_t - BX_t = X_t - X_{t-1}.$$

Let d define the amount of influence past prices have on the current fractionally differenced value. We can expand the series using Binomial coefficients:

$$(1 - B)^d = 1 - d * B + \frac{d(d-1)}{2!}B^2 - \dots,$$

where $B^2X_t = X_{t-2}$. Note that d is a single value to define all historical price weights.

We are interested in seeing how a real positive d preserves memory. Let

$$\hat{X}_t = \sum_{k=0}^{\infty} \omega_k X_{t-k},$$

with weights ω

$$\omega = \Big\{1, -d.\frac{d(d-1)}{2!}, \frac{d(d-1)(d-2)}{3!}, \dots\Big\}.$$

When d is a positive integer, $\Pi_{i=0}^{k-1} \frac{d-i}{k!} = 0$ for all $k > d$, and the memory beyond that point is cancelled. For example, when $d =, \omega = \{1, -1, 0, 0, \dots\}$.

We can approximate the weights ω iteratively by

$$\omega_k = \omega_{k-1}\frac{d-k+1}{k}.$$

We will use the Augmented Dickey Fuller (ADF) stationarity test to choose the optimal d for a stationary time series.

If you are interested in this important topic, have a look at https://medium.com/rapids-ai/rapid-fractional-differencing-to-minimize-memory-loss-while-making-a-time-series-stationary-2052f6c060a4, and maybe at https://towardsdatascience.com/preserving-memory-in-stationary-time-series-6842f7581800.