Supervised Learning in Factor Investing: Foundations

supervised **learning** **factor** investing

Agenda

Why?

Four core purposes for this course

- Gain knowledge and insights on some mainstream asset pricing results and methods
- Apply ML tools on financial datasets:
  1. what are these tools?
  2. why resort to them?
  3. how to use them?
- Understand the weaknesses of ML applied to asset management: **ML is not a magic wand**![1]
- Improve your coding skills (in R + ...!)

More broadly: ML culture/knowledge is more & more required in finance... both in buy side and sell side jobs.
My goal: that you **shine in interviews** (& get the jobs you want)!

[1] see Lopez de Prado: The 10 Reasons Most Machine Learning Funds Fail

What the course is **not** about

Other applications of ML in Finance

- Derivatives pricing & hedging
- Fraud detection
- Credit scoring

Related topics

- Alternative data use cases
- Deep mathematical developments in supervised learning + computer vision & CNN
- Algorithms in unsupervised/reinforcement learning
- NLP tools for finance (sentiment)
- ML for high frequency trading (order book dynamics)

If you like challenges: https://challengedata.ens.fr

We will focus on **insights** and **applications** for Factor Investing.

Why now?

A favorable nexus

- **Data availability** (cross-section of stocks $\sim$1000, characteristics $\sim$100, time points $\sim$100 + alt-data!) gives ML a favorable playground
- **Computational power**:
  - **Hardware**: storage & processing speed almost limitless (all major players provide cloud solutions: IBM, Google, Amazon, Microsoft + niche players)
  - **Software**: easily accessible thanks to the private sector (ML libraries funded by Google and Facebook) and to large academic research groups (INRIA, Stanford, UPenn etc.)
- **Economic grounding**: what this session is about.
  Computer scientists have used finance as playing field for decades (data is readily available). Applied CS is <u>not enough</u>: we need an economic/logical framing.

Layout of the course

Eight sessions

1. Introduction & foundations
2. Portfolio strategies in R
3. LASSO & sparse hedging
4. Data preparation, feature engineering, labelling
5. Decision trees & extensions
6. Neural networks
7. Tuning & validating
8. Extensions: SVM, ensemble learning, interpretability & backtest overfitting

Within the sessions

How this is going to work

Two (possibly three) parts

1. Theory (slides): 25%-40%
2. Practice (notebooks): 40%-70%
3. Exercises / questions (notebooks): remaining time

+ course work (ML backtest project)

One major reference: www.mlfactor.com

Agenda

References

Factor investing: the idea that (some) firm characteristics **drive** future profitability.

This topic is HUGE!

We refer to the monographs:

- **Asset Management: A Systematic Approach to Factor Investing** by Andrew Ang
- **Expected Returns: An Investor's Guide to Harvesting Market Rewards** by Antti Ilmanen

An important article on the subject is:
... and the cross section of stock returns, by Harvey et al.

+ chapter 3 in the book which we'll cover soon.

One important contribution

Fama French (1992)

Monthly portfolio sorts

The process is the following:

- Each month, assets (stocks) are sorted according to one (or two) of their **characteristics** (cap = size, B/M ratio, past return, etc.)
- Portfolios are constituted according to these sorts (e.g., quintile or decile portfolios). The allocation is usually cap-weighted or equally-weighted
- The performance of the portfolio is recorded for the month ahead, and then the portfolio composition is updated based on the new (current) values of the characteristics
- the corresponding vectors of returns can be analysed (average means, t-tests)

The results

Source: Fama French (1992).
The figures are equal to the average monthly returns of the EW-portfolios, in percent.

	Book-to-Market Portfolios										
	All	Low	2	3	4	5	6	7	8	9	High
All	1.23	0.64	0.98	1.06	1.17	1.24	1.26	1.39	1.40	1.50	1.63
Small-ME	1.47	0.70	1.14	1.20	1.43	1.56	1.51	1.70	1.71	1.82	1.92
ME-2	1.22	0.43	1.05	0.96	1.19	1.33	1.19	1.58	1.28	1.43	1.79
ME-3	1.22	0.56	0.88	1.23	0.95	1.36	1.30	1.30	1.40	1.54	1.60
ME-4	1.19	0.39	0.72	1.06	1.36	1.13	1.21	1.34	1.59	1.51	1.47
ME-5	1.24	0.88	0.65	1.08	1.47	1.13	1.43	1.44	1.26	1.52	1.49
ME-6	1.15	0.70	0.98	1.14	1.23	0.94	1.27	1.19	1.19	1.24	1.50
ME-7	1.07	0.95	1.00	0.99	0.83	0.99	1.13	0.99	1.16	1.10	1.47
ME-8	1.08	0.66	1.13	0.91	0.95	0.99	1.01	1.15	1.05	1.29	1.55
ME-9	0.95	0.44	0.89	0.92	1.00	1.05	0.93	0.82	1.11	1.04	1.22
Large-ME	0.89	0.93	0.88	0.84	0.71	0.79	0.83	0.81	0.96	0.97	1.18

The first row and first column show the trends!

One step further: "risk factors"

Inspiration: Fama French (1993)

Long-short portfolios

- Suppose you form one portfolio that is long small firms and short big firms (SMB) and one portfolio that is long firms with high B/M ratio and short firms with low B/M ratio (HML).
- Then, you can use these (dollar-neutral) portfolios to analyse/decompose the returns of individual asset or portfolios via a linear regression:

$$r_t^i - r_t^f = \alpha + \beta^M (r_t^M - r_t^f) + \beta^{SMB} r_t^{SMB} + \beta^{HML} r_t^{HML} + \epsilon_t$$

β^M is the exposure to the market, β^{SMB} and β^{HML} are those to the corresponding factors and α is the performance that cannot be explained by these 3 components.

The **SMB** factor is referred to as the Size factor and the **HML** one is the Value (vs Growth) factor.

More generally...

Academics and practitioners have tried to introduce new factors (i.e., based on new characteristics) in the game.

The most classical ones:

- the **Momentum** factor: the attribute here is the return between one year ago and last month (Winners minus Losers WML).
- the **Low-Vol** factor: the attribute is the volatility! (see also: low idioysinc. vol, low beta, BAB: Betting Against Beta)
- the **Profitability** factor (operating profitability: Robust Minus Weak profits) - RMW from Fama French (2015)
- the **Investment** factor (change in total asset: Conservative Minus Aggressive) - CMA also from Fama French (2015)
- the **Quality** factor (a mix!): Quality minus Junk - QMJ from Asness et al (2019)

Original factors look at the names of companies, or the number of marathons ran by the CEO! $\rightarrow$ **economic relevance**?
Nowadays, people investigate the **green** factor (ESG-based).

Identification: how to proceed?

You have to be careful!

The usual steps:

1. identify a firm **characteristic**
2. form monthly portfolios according to x-percentiles of this characteristic (e.g. five quintile portfolios or ten decile portfolios)
3. keep track of their returns over a sufficiently long period ($>$20Y)[2]
4. let's say r_t^+ and r_t^- are the returns associated with the top and bottom portfolios, perform a t-test on the series $r_t^+ - r_t^-$ (or a test for the mean on the two series - taking the difference in variance into consideration)
5. if the statistics is larger (in absolute value) to some threshold (2, 3?!), then you have yourself a factor! (more or less)

The amount of caution in such empirical design should be extreme because results are often the results of data-snooping / p-**hacking**. **Robustness checks** are compulsory!

[2] Hard for the ESG factors!

supervised **learning factor** investing

Theoretical groundings

'*Factor investing*' can be viewed as a special case of APT.

The APT (Ross 1976)

- the asset return follows a linear model:

$$r_t^j = \alpha^j + b_1^j F_t^1 + \dots b_n^j F_t^n + \epsilon_t^j,$$

- where the F_t^k are n factors driving stock returns.

BUT! It can also be argued that it is the (raw) characteristics that matter (Daniel and Titman (1997)).[3] In an ML-driven approach, it will be easier to rely on firm attributes (factors take time to compute and depend on many degrees of freedom). Also: for predictive purposes, we will need to add a lag in the predictors (more on that later).
$\rightarrow$ **Factor analysis** is nonetheless useful because it helps understand if one attribute is valuable.

[3] E.G.: firms with low market cap can load negatively on the size factor!

Agenda

1. Introductory remarks
2. Asset Pricing Anomalies
3. Advanced anomaly detection
4. Extensions
5. Supervised Learning
6. Wrap-up

Fama-MacBeth regressions (1/2)

For a modern view on **anomaly detection**, see Baker, Luo and Taliaferro (2018) and Harvey Liu (2021).

Step one: cross-section of time-series regressions

We consider n asset returns $r_{n,t}$ and m factors f_t^m. We start by estimating n equations, each with m loadings (and a constant):

$$
\begin{aligned}
r_{1,t} &= \alpha_1 + \beta_1^1 F_t^1 + \dots + \beta_1^m F_t^m + \epsilon_{1,t}, \quad t \in (1, T) \\
&\vdots \\
r_{n,t} &= \alpha_n + \beta_n^1 F_t^1 + \dots + \beta_n^m F_t^m + \epsilon_{n,t}, \quad t \in (1, T)
\end{aligned}
$$

This gives a matrix of **loadings** $\hat{\beta}_i^j$; i relates to asset and j to factor.

The $\hat{\beta}_i^j$ characterise the exposure of asset i to the factor j. The sign indicates the direction of the co-movement and the associated t-stat indicates whether the relationship is statistically significant.

Fama-MacBeth regressions (2/2)

Step two: time-series of cross-sectional regressions

Given the $\hat{\beta}_i^j$, estimate, for each date t,

$$r_{i,1} = \kappa^1 + \gamma_1^1 \hat{\beta}_i^1 + \dots + \gamma_1^m \hat{\beta}_i^m + \epsilon_i^1, \quad i \in (1, n)$$

$$\vdots$$

$$r_{i,T} = \kappa^T + \gamma_T^1 \hat{\beta}_i^1 + \dots + \gamma_T^m \hat{\beta}_i^m + \epsilon_i^T t, \quad i \in (1, n)$$

This gives a matrix of estimated coefficients $\hat{\gamma}_t^j$. The premium of factor j is estimated as the average of the $\hat{\gamma}_t^j$ over all $t = 1, \dots, T$. Assuming a large number of observations, the classical t-stat is

$$t_j = \frac{\frac{1}{T}\sum_{t=1}^{T} \hat{\gamma}_t^j}{\hat{\sigma}_m/\sqrt{T}},$$

where $\hat{\sigma}_m$ is the standard deviation of the $\hat{\gamma}_t^j$.

=> tells if the premium is strongly positive or negative in the **long run**!

Factor competitions

Academics spend a **LOT** of time trying to figure out new factors - or ways to test if factors are truly *factors*.

Singling out the *best* ones

We assume m factors F_t^m. We run m regressions: each factor is regressed against all other factors:

$$\begin{aligned}
F_t^1 &= \alpha^1 && +\beta_2^1 F_t^2 + \beta_3^1 F_t^3 + && \dots + \epsilon_t^1 \\
F_t^2 &= \alpha^2 + \beta_1^2 F_t^1 && + \beta_3^2 F_t^3 + && \dots + \epsilon_t^2 \\
F_t^3 &= \alpha^3 + \beta_1^3 F_t^1 && +\beta_2^3 F_t^2 + && \dots + \epsilon_t^3 \\
&\vdots
\end{aligned}$$

If the estimated $\hat{\alpha}^j$ is significantly different from zero, it means that factor j fails to be explained by the other factors. If $\hat{\alpha}^j$ is not statistically different from zero, then is **redundant** because it can be **captured** by **exposures** to the other factors.

Agenda

1. Introductory remarks
2. Asset Pricing Anomalies
3. Advanced anomaly detection
4. Extensions
5. Supervised Learning
6. Wrap-up

supervised **learning** **factor** investing

Other asset classes

Less clear...

It's all a matter of specificities/characteristics

- **Fixed income**: firm ratings and bond maturities
- **FX**: country and liquidity risk
- **Commodities**: price-based only (includes futures)
- **Exotic classes**: real-estate, crypto, NFTs $\Rightarrow$???

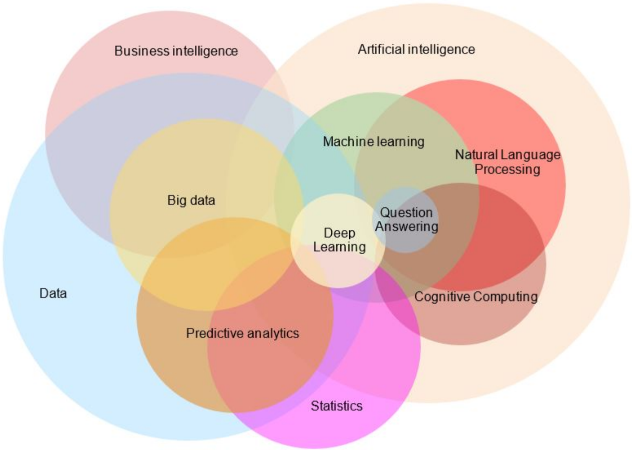
Academic research is scarcer compared to equity. The framework and the empirical results are not so well established.

A common factor across all classes is **momentum** (it only requires prices, which are most of the time available).

Agenda

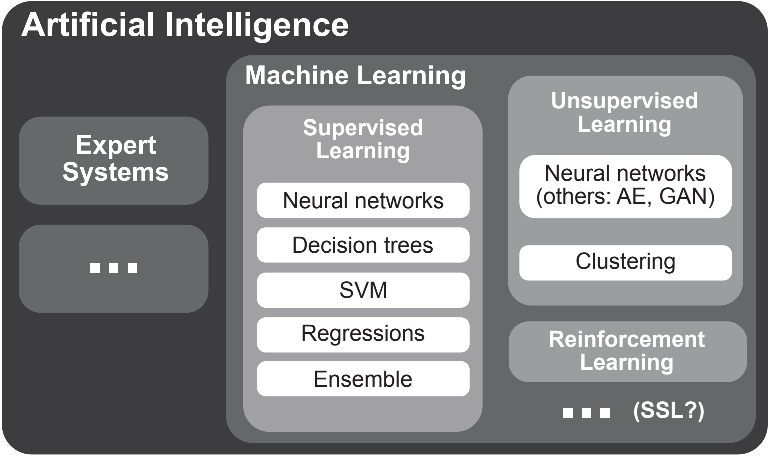
Where is it located? (1/2)

According to **IBM**...



Where is it located? (2/2)

The way I see things (fewer colours).



Natural Language Processing is related to several rounded rectangles.

The AI paradigm shift

Expert vs Data

Also: **symbolists** vs **connexionists**.

- Expert systems (popular in the 1990s) are rule based (e.g.: if that then that)
- Given the complexity of tasks, rules quickly become too limited
- An alternative route is to consider massive amounts of data from which to learn patterns (and hence infer rules)
  $\rightarrow$ connexionism (neural networks mostly)

François Chollet summarises this as:

- **Expert system**: Rules + Data $\rightarrow$ Answers
- **Supervised Learning**: Answers + Data $\rightarrow$ Rules

+ Reinforcement learning & unsupervised methods / SSL.

How does supervized learning work? (1/2)

A primer

Consider a large dataset (e.g., rectangular). You want to be able to understand (and then forecast) one column (y) as a function of the others (x). You can always represent the problem as follows:

$$y_i = f\left(x_i^1, x_i^2, \dots, x_i^n\right) + \epsilon_i,$$

where i denotes the occurrence number and ϵ_i the related error. Two essential (and related) questions in ML are:

1. How do I find/choose f?
2. Will the performance of my model (f) change if I test it on new data?

(in **unsupervised learning**, the above expression does not hold: the machine learns on its own because there is no y.)

The link to asset management

Characteristics!

As we saw previously, the firms' **characteristics** are likely to impact their future performance. This impact is probably nonlinear and time-varying. A very flexible way to evaluate this impact is to consider the above model:

$$y_i = f\left(x_i^1, x_i^2, \dots, x_i^n\right) + \epsilon_i,$$

where y is a proxy for future performance (many choices are possible, e.g., horizon) and the x^j are a set of characteristics. The degrees of freedom are numerous; a shortlist:

1. the set of characteristics
2. the investment set (data availability)
3. the family of functions f: which ML tool?
4. data preprocessing: labelling and feature engineering

In short: factors (**characteristics**) $\leftrightarrow$ **features** (inputs)!

Example

The typical **factor dataset** looks like that:

	Tick	Date	Close	Vol_1M	Mkt_Cap	P2B	D2E	Prof_Marg	ESG_rank	Forward_return
	<chr>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	F	2021-01-29	10.5	45.0	41893.	1.37	529.	-7.75	44.2	0.111
2	DIS	2021-01-29	168.	27.8	305105.	3.56	59.4	0.105	67.5	0.124
3	D	2021-01-29	72.9	19.0	59465.	2.48	142.	19.4	43.1	-0.0627
4	CVX	2021-01-29	84.0	30.3	164011.	1.23	33.7	-2.68	34.9	0.190
5	CVS	2021-01-29	71.6	21.6	93784.	1.35	122.	1.40	40	-0.0491
6	C	2021-01-29	58.0	40.2	120733.	0.671	327.	20.9	46.9	0.136
7	BAC	2021-01-29	29.6	32.9	256497.	1.03	196.	27.2	58.5	0.171
8	AAPL	2021-01-29	132.	35.3	2215357.	33.5	169.	25.8	79.9	-0.0797

So, basically, the aim is to explain $y =$ the forward return (i.e., predict the return) using $x =$ the other columns (the predictors)
NOTE: obviously, Tick and Date are not predictors...

How does supervized learning work? (2/2)

A primer

- Parametrising the model with the information at our disposal!
- Usually, f will depend on some parameters $\mathbf{\Theta}$.
- We define a 'loss' (or error) function $L(y_i, f_{\mathbf{\Theta}}(\boldsymbol{x}_i))$, often $L(y_i, f_{\mathbf{\Theta}}(\boldsymbol{x}_i)) = (y_i - f_{\mathbf{\Theta}}(\boldsymbol{x}_i))^2$ when working with continuous variables.
- Machine Learning is simply finding

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmin}} \sum_{i=1}^{N} L(y_i, f_{\mathbf{\Theta}}(\boldsymbol{x}_i)),$$

  where $i = 1, \dots, N$ are the occurrence indices within the training sample.
- this ensures that the model values $f_{\mathbf{\Theta}}(\boldsymbol{x}_i)$ are as close as possible to the 'true'/observed values y_i.

Sometimes, the parameters will not be numbers, but 'architectural choices', in which case they are often referred to as **hyper-parameters**.

ML 101: regressions

There are limitless choices for f.

One crucial building block!

- The simplest one is the linear form:
$$f(\boldsymbol{x}_i) = \beta_0 + \sum_{k=1}^{K} \beta_k x_i^{(k)}$$
the parameters are the betas and are usually estimated via OLS.
- Some ideas driving the regression are also behind more elaborate nonlinear tools (decision trees and neural networks).
- For more complex models, there are no closed-forms for the optimal parameters $\boldsymbol{\Theta}$.
- Often, the optimal solution can be iteratively approximated using gradient methods for instance.
- In other situations, hyper-parameter tuning will required some level of expertise.

Predictive regressions?

In a factor context

Suppose you have identified 'factors' f^k (not necessarily L/S portfolios) that are likely to drive future performance. As an investor, you want to be able to **predict** this performance.
Let's call it r for simplicity ($\rightarrow$ return!).
One very simple way to proceed is to estimate:

$$r_t = \alpha + \beta_1 f^1_{t-1} + \dots + \beta_n f^n_{t-1} + \epsilon_t$$

with data from the past up to the last know r_t. At time t, the f^j_t are known, thus, the conditional expectation (i.e., the forecast) is:

$$\hat{r}_{t+1} = \mathbb{E}[r_{t+1}|\boldsymbol{f}_t] = \hat{\alpha} + \hat{\beta}^1 f^1_t + \dots + \hat{\beta}^n_t f^n_t$$

supervised **learning** **factor** investing

Predictive panels!

In fact... does the cross-section of stocks matter?

Another way to look at it is:

$$r_{t+1,n} = a + \sum_{k=1}^{K} b^{(k)} x_{t,n}^{(k)} + u_{t+1,n},$$

the double index in time t and firm n calls for panel-like estimations.
NOTE: a, $b^{(k)}$ do not depend on t and n. The characteristics $x_{t,n}^{(k)}$ depend on everything. For technical reasons, we can decompose the error as

$$r_{t+1,n} = a + \sum_{k=1}^{K} b^{(k)} x_{t,n}^{(k)} + \mu_n + e_{t+1,n},$$

where μ_n is an average firm-specific term.[4]

Factor models that rely on ML are simply generalizations of this model to non-linear relationships.

[4]For more on that, see *Panel data econometrics in R: The plm package*

Agenda

1. Introductory remarks
2. Asset Pricing Anomalies
3. Advanced anomaly detection
4. Extensions
5. Supervised Learning
6. Wrap-up

Key takeaways (1/2)

Characteristics[5]

- academic research finds that characteristics/factors help understand/predict future returns
- there is no real **consensus**, except on a very small group of key features and the devil can be in the details[6]
- the foundation of our approach is very agnostic: characteristics drive profitability, but we do not necessarily know which ones, and if the relationships are stable through time (risk premia are notoriously time-varying)
- but we keep in mind that the inputs should make sense economically (first letters of firm names?)

[5] See Daniel & Titman (1997)
[6] Asness & Frazzini (2013)

Key takeaways (2/2)

Factors

- there are several ways to define and quantify an asset pricing factor/anomaly and test can help extract the variables that truly matter
- machine learning tools are expected to sort out the factors/features that matter **agnostically** (which has some pros & cons) - *even if the choice of inputs can/should be economically motivated (brute force data mining is not the best option)*
- nonetheless, expert guidance will probably improve results out-of-sample
- practice is key $\rightarrow$ train models & **train yourselves**!

Thank you for your attention

Any questions?

Bibliography

Asness, C., & Frazzini, A. (2013). The devil in HML's details. **Journal of Portfolio Management**, 39(4), 49-68.

Asness, C. Frazzini, A., Pedersen LH (2019) Quality minus Junk, **Review of Accounting Studies**, 24, 34- 112 Baker, M., Luo, P., &

Taliaferro, R. (2018). Detecting anomalies: The relevance and power of standard asset pricing tests. Working paper.

Daniel, K., & Titman, S. (1997). Evidence on the characteristics of cross sectional variation in stock returns. **Journal of Finance**, 52(1), 1-33.

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. **Journal of Finance**, 47(2), 427-465.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. **Journal of Financial Economics**, 33, 3-56.

Harvey, C. & Liu, Y. (2021). Lucky Factors. **Journal of Financial Economics** Forthcoming.