Data preparation

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

References

Beyond technical articles and books on data cleansing, we recommend simple papers:

- **A Backtesting Protocol in the Era of Machine Learning** by Arnott, Campbell and Markowitz (2018).
- **Being Honest in Backtest Reporting: A Template for Disclosing Multiple Tests** by Fabozzi and Lopez de Prado (2018)

One common theme: the fight against data snooping and overfitting $\rightarrow$ reproducible finance!

+ **Advances in financial machine learning** by Lopez de Prado.

Taxonomy / Nomenclature

Usual ML terms

$$\boldsymbol{y} = f(\boldsymbol{X}) + \epsilon,$$

where

- $\boldsymbol{y}$ is the **dependent** / endogenous / predicted / explained variable, or the label
- the columns of $\boldsymbol{X}$ are the **predictors**, the independent / exogenous variables, the inputs, the features. In factor investing, they will be chosen to be factors, characteristics or attributes (from firms)
- ϵ is the **error** or residual (sometimes, the innovation in time-series settings)
- f is the model (or possibly the data generating process)

The present session is about $\boldsymbol{X}$ and $\boldsymbol{y}$.
Not very appealing, but incredibly important.

Data: know it

Crucial step: descriptive statistics, with plots if necessary/possible (visual synthesis).

GIGO: Garbage in, garbage out

- Which **features**? Do you feed all the data or do you select some preferred variables (i.e., do you let the data talk, or do you have priors stemming from economic intuition/empirical work)?
- Are there **redundancies**? Is the risk of collinearity a problem or not?

Which **label**? $\rightarrow$ more on that soon!

About colinearity

Imagine 2 standardized predictors with $\boldsymbol{X}$ such that $\boldsymbol{X}'\boldsymbol{X} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Then $(\boldsymbol{X}'\boldsymbol{X})^{-1} = \frac{1}{1-\rho^2}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix}$ and $\boldsymbol{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{v}$ is such that

$$\beta_1 = \frac{v_1 - \rho v_2}{1-\rho^2}, \quad \beta_2 = \frac{v_2 - \rho v_1}{1-\rho^2}$$

Thus, as ρ gets closer to 1, the magnitude of the β_i increases and the signs depend on the relative importance of the "covariance" terms v_i.

- Basically, one coefficient hedges the other.
- But because the 2 variables are redundant, only one should have been included in the first place.
- **Penalization** (ridge, LASSO, elastic-net) is one way to solve this.

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

Missing data (1/2)

A lot of papers deal with the handling of missing data. There are mainly two possibilities when facing a missing point in one occurrence:

1. **remove**/delete the occurrence: agnostic but costly
2. **replace** the missing point with a value (imputation): keeps the data, but relies on some assumption

Imputation: naive approaches

- Replace with a '**median**' (e.g., cross-sectional) value: the value lies in the bulk of the distribution. Possible problem if the *true* corresponding point is extreme: you lose some information
- Replace the value using some **parametric** or non-parametric assumption (distribution, Bayesian prior, etc.) / interpolation, extrapolation, nearest neighbor
- In a time-series context, replace by the **preceding** value (but be careful)

Missing data (2/2)

Caveats

- In **chronological** data, do not **extrapolate** between two values (tempting for quarterly accounting data). Example: the February value is not the average between January and March! This is **forward-looking** and impossible from a backtest standpoint.
- In **chronological** data, be careful when replacing by the previous value. It can be ok for accounting data (released quarterly), maybe less so for other variables (e.g., trading volume).
- In the same spirit, it is ok to replace missing price with past price (no movement), but it is not ok to do so with returns! (replace with zero return, i.e., no movement). Overall, if the variable is **persistent** (highly autocorrelated), using the previous value is not an awful proxy.

Imputation: tough choices

Hybrid problem for **fundamental data**: the dividend yield example. Usually, dividends are paid on a **quarterly** basis.

Date	Original yield	Replacement value
2015-02	NA	(preceding (if it exists)
2015-03	0.02	untouched (none)
2015-04	NA	0.02 (previous)
2015-05	NA	0.02 (previous)
2015-06	NA	$\leftarrow$ **Problem**!

Do you continue the imputation, knowing that the firm must have paid a dividend or do you consider a zero yield?
Is the value missing because of internal data problems **or** is it missing because the firm did not pay any dividend?

Outlier management

A complicated topic: it's hard to discern a true **outlier** from an **error** in the dataset. (Ref: **Outlier Analysis** by Aggarwal (2016))

Hard thresholds

- a very classical method is to set a multiple of the standard deviation around the mean. All points outside the interval $[\mu - m\sigma, \mu + m\sigma]$ are considered outliers (often $m = 2, 3, 5, 6, 10$, etc. - arbitrary!).
- in the same spirit, if the largest value is larger than m times the second-to-largest, it can also be considered an outlier.
- extreme points in the **distribution** of one variable can be categorised as outliers, even though this is overly simplistic: points outside the $[q, 1 - q]$ quantile range are often instructive.

Outliers depend on **subgroups**: it is useful to compute statistics chronologically and across firms. A 900B$ market cap is an outlier in the cross-section, but the market cap of Apple is pretty consistent through time. Sometimes, a closer look is useful.
$\rightarrow$ And keep in mind: 'true' outliers are very insightful!

Winsorisation

A popular practice in Finance.

Still thresholds!

- Given a sample x_i, $i = 1, \ldots, n$, and a **quantile threshold** q, we write $x^{(q)}$ for the point located exactly at q on the empirical distribution of x (cdf), i.e., such that $P_x[x \leq x^{(q)}] = q$.
- Winsorising amounts to setting to $x^{(q)}$ all values below $x^{(q)}$ and to $x^{(1-q)}$ all values above $x^{(1-q)}$. The winsorised variable $\tilde{x}$ is:

$$\tilde{x}_i = \begin{cases} x_i & \text{if } x_i \in [x^{(q)}, x^{(1-q)}] \quad \text{(unchanged)} \\ x^{(q)} & \text{if } x_i < x^{(q)} \\ x^{(1-q)} & \text{if } x_i > x^{(1-q)} \end{cases}$$

- The range for q is usually $(0.5\%, 5\%)$ with 1% and 2% being very often used.

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

The problem

Scale!

Financial data comes in lots of scales and ranges:

- **returns** are usually smaller than one in absolute value
- stock **volatility** lies between 5% and 80% most of the time
- **market capitalisation** is expressed in million or billion $
- **accounting values** as well
- **accounting ratios** have inhomogeneous units
- **synthetic attributes** (sentiment) also have their idiosyncrasies

Feeding all this data to a regression would result in estimates with very different scales, which is not a problem per se.

Other tools (e.g., neural nets) often work better when data scales are **homogeneous**. They also require *numerical* inputs (categorical features are excluded or re-coded via one-hot for instance).

Normalisations

Several options...

Below, $\tilde{x}$ denotes the normalised version of the raw data x.

- classical standardising: $\tilde{x} = \frac{x-\mu}{\sigma}$
- 0-1 reduction: $\tilde{x} = \frac{x-\min(x)}{\max(x)-\min(x)}$
- uniformisation: $\tilde{x} = F_x(x)$, where F_x is the empirical cdf of x.

The way **normalisations** are performed *can* matter a big deal. In order to avoid any forward-looking bias, we recommend to proceed as follows:
$\rightarrow$ for each date and each **attribute** (feature), normalise the values in the cross-section.
This means: at each point in time and for each **characteristic**.

Augmenting the feature space

It may desirable to consider 'derivatives' of original features.

A snapshot of possibilities...

- **Lagged** variables: x_{t-1}. It is possible that memory effects play a role in the determination of future returns.
- **Variations** in variables: it may not be the level of a particular variable that matters, but its recent variation: $\Delta x_t = x_t - x_{t-1}$. Examples: variation in earnings, profits, volatility, sentiment, etc.
- **Macroeconomic shifts**: If z_t is a macroeconomic variable (dividend yield/growth, inflation, term spread, credit spread, etc.), use $x_t \times z_t$. Data can be accessed from the Fed of St Louis even though update times are sometimes long!

The numerical 'upside' is a higher number of exogenous variables. The risk of **overfitting** increases nonetheless (just like in a simple linear regression adding variables mechanically increases the R^2). The choices must make sense, economically.

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

An important question...

What exactly do you want to explain or predict?

Many possibilities

- **future returns**
- future **relative** returns (versus some benchmark: market-wide, or sector-based for instance)
- the **probability of positive return** (or of return above a specified threshold)
- the **probability of outperforming a benchmark**
- the binary version of the above: YES (outperforming) versus NO (underperforming) $\rightarrow$ label!

Normalising or not? What's the **horizon**?

Categorical data (1/4)

Sometimes, the output (or dependent variable) will not be a real number, but a **category**. In an investment context, this can for example be: buy, hold, sell.
Nonetheless, most algorithms require numbers as inputs (trees are one exception).
Hence, categories must be recoded into numbers! Two solutions:

- either categories are ordered (**ordinal**), in which case a simple mapping is possible. Example: -1 for sell, 0 for hold and +1 for buy;
- either categories are unordered (**nominal**), and we must resort to one-hot encoding.

Categorical data (2/4)

One-hot encoding

When dealing with nominal variables, one way to **recode** the data is to create new binary columns: one for each class in the variable. The value is then the indicator function of the class (1 if the variable takes the value of the class, 0 if not).

Initial	One-hot encoding		
Position	**Pos_sell**	**Pos_hold**	**Pos_buy**
buy	0	0	1
buy	0	0	1
sell	1	0	0
hold	0	1	0
sell	1	0	0

In classification tasks, the output is usually the **probability of each class** (a vector of size equal to the number of classes)

Categorical data (3/4)

Categories often stem from numbers!

One classical example is the following. The manager sets a confidence threshold r.

- when a forecast is below $-r$, he decides (or tells the algo) to sell
- when a forecast is above r, he decides to buy
- when a forecast is in $[-r, r]$, he does nothing

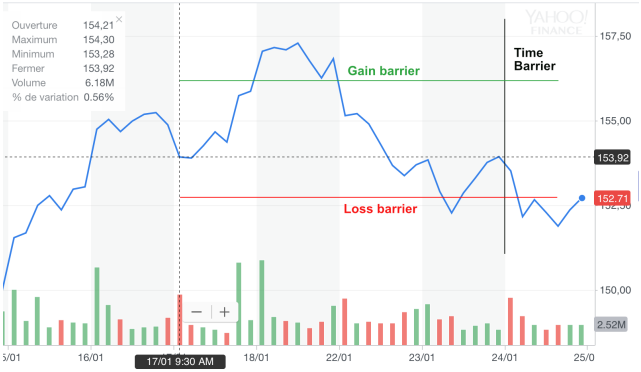
This gives

$$y_{t,i} = \begin{cases} -1 & \text{if} \quad \hat{r}_{t,i} < -r \\ 0 & \text{if} \quad \hat{r}_{t,i} \in [-r, r] \\ +1 & \text{if} \quad \hat{r}_{t,i} > r \end{cases}$$

Some researchers (Lopez de Prado) advise to resort to so-called *meta-labelling*, whereby the direction is separated from the bet size in a two-stage ML process.

Categorical data (4/4)

Categories from the triple barrier method: see Lopez de Prado's book.



Three barriers on PRICE data (dynamic!): stop either when you reached

- target **profit** (green) → **+1**
- target **loss** (red) → **-1**
- target **horizon** (black) → **0** (or proxy for distance to barriers)

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

Refinements

Meta-labelling

Lopez de Prado advises to resort to so-called *meta-labelling*, whereby the direction is separated from the bet size in a two-stage ML process.

Conditional labelling

Not all periods are equal! In times of high volatility, labelling can be made more conservative. For instance, barriers and thresholds can be asymmetric so that losses are more penalised.

One final comment

Consistency!

- One important property of labels (and features) is their **chronological stability** - or lack thereof (autocorrelation).
- Many features (accounting-based, like P2B, or price-based, like 12M momentum or volatility) are auto-correlated at the daily or monthly frequency.
- Hence, if the dependent variable is **highly oscillatory** (as monthly returns are), it is likely that no model will be able to link the features to y in a robust fashion.
- what will happen is that the algorithm will find spurious arbitrages between variables that will fail to provide valid predictability out-of-sample.

Agenda

1. Data
2. Cleaning: missing data, outlier detection
3. Feature engineering
4. Labelling
5. Extensions
6. Wrap-up

Key takeaways

Engineering!

- data preparation is often overlooked, but it is crucial (**GARBAGE IN, GARBAGE OUT**!)
- there are different was to scale and normalise features and one choice can be impactful
- labelling (i.e. defining what it is we try to predict) is incredibly important and is often a matter of experience/craft because the degrees of freedom are numerous
- be careful to **autocorrelations** patterns in labels versus features

One final **tip**: do not think data preparation is done only once! $\rightarrow$ use scripts that you can recycle if need be!

Thank you for your attention

Any questions?