# Machine Learning for Quantitative Investment

Fintelligence

2022-01-23

# Contents

**Preface** **5**

**1 Introduction** **7**

**2 Elements of Machine Learning** **11**

2.1 Linear Regression . . . . . 12

2.2 Assumptions of Linear Regression and Violation Implications . . 14

2.3 Conclusion . . . . . 16

**3 Methods** **17**

3.1 math example . . . . . 17

**4 Applications** **19**

4.1 Example one . . . . . 19

4.2 Example two . . . . . 19

**5 Final Words** **21**

# Preface

Welcome to the first edition of Machine Learning for Quantitative Investment (ml4quant)! This is an online book all about machine learning and data science methods in quantitative investment presented by Fintelligence Academy: https://fintelligence-academy.github.io/.

The adoption of Machine Learning and Artificial Intelligence continues to progress from a niche activity to mainstream applications at an ever-accelerating pace in this Big Data era. Here at Fintelligence, we are embracing the ML in financial application.

In this book, we aim to walk through the ….

Having quantitative research in mind, the intended audience is presumed interdisciplinary, fluent in mathematical notation, familiar with basic data science concepts. You may be surprised at the neglect of econometric/finance elements - we are confident to teach you the foundational knowledge of financial market along the reading journey. In addition, this book could be also interesting to those readers who are thinking of joining the quant workforce or data science community.

In another vein, this book also brings the mathematical foundations of basic machine learning concepts to the fore and collects the information in a single place so that a reader could learns both modern quantitative finance and machine learning. We assimilate and benefit core ideas from the two classical machine learning textbooks *An Introduction to Statistical Learning (ISL2)* and *The Elements of Statistical Learning (ESL)*, which are the best to-go reference books to learn machine learning. Here for ml4quant, we strive to strike a balance of difficulty in the middle level between the two books and season the topic with a new flavor of financial data science.

We genuinely hope that both the quantitative researchers and data scientists could enjoy reading this interdisciplinary book and find it helpful.

> "The best time to plant a tree is twenty years ago. The second best time is now." - Chinese Proverb by Poet Wang Bo
> "　　"_

# Chapter 1

# Introduction

Machine learning is the latest in a long line of attempts to distill human knowledge and reasoning into a form that is suitable for constructing computer automated systems. The enthusiastic practitioner who is interested to learn more about the magic behind successful machine learning algorithms currently faces a daunting set of pre-requisite knowledge:

- Programming languages and data analysis tools
- Large-scale computation and the associated frameworks
- Mathematics and statistics and how machine learning builds on it

Machine learning builds upon the language of mathematics to express concepts that seem intuitively obvious but that are surprisingly difficult to formalize. Once formalized properly, we can gain insights into the task we want to solve.

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 1. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter 3.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 1.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 1.1.

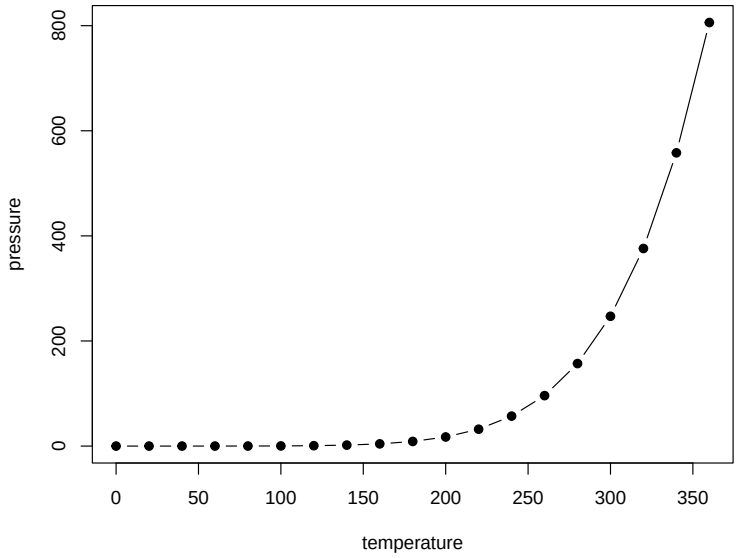


Figure 1.1: Here is a nice figure!

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2021) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

Table 1.1: Here is a nice table!

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| 5.7 | 4.4 | 1.5 | 0.4 | setosa |
| 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| 5.1 | 3.8 | 1.5 | 0.3 | setosa |

# Chapter 2

# Elements of Machine Learning

In a minimalism style, machine learning refers to how computers can "learn" by finding patterns in data and using them to make predictions. Mathematically, given a real-valued output $y$ and predictors vector $X$ containing $p$ variables, we assume a general function $f$ to describe $y$:

$$y = f(X) + \epsilon \tag{2.1}$$

Where, $f$ is a fixed but unknown function, and $\epsilon$ is a zero-mean random error term, which is supposed to be independent of $X$.

The core of machine learning is a suite of data-driven algorithms for estimating $f$. ML is based on *statistical learning theory* to design models to understand patterns and employs *optimization algorithms* to train the model to "learn" the pattern using input data.

The foundation of practical machine learning is the data. Data drives everything else. The model can not learn much pattern without enough data and could even have biased behaviour if the data quality is poor. In contrast, with substantial data, the machine learning model could achieve impressive results beyond expectation. A vivid example is to check the google's search engine - it is machine learning algorithm under the hood.

"All models are wrong, but some are useful." - George E.P. Box

## 2.1 Linear Regression

Linear regression adopts a linear function $f^{Linear}$ to equation (2.1) with learnable model parameters $\beta^T = (\beta_0, \beta_1, ..., \beta_p) \in (p, 1)$:

$$y = \beta^T X + \epsilon \tag{2.2}$$

Here we assume the first column of $X$ is all ones as an "intercept feature" and thus $\beta_0$ corresponds to the intercept term.

The model given by (2.2) defines the *population regression line*, which is the best linear approximation to the true relationship between $X$ and $u$. The population regression line is unobserved and we have access to a set of sample observations to compute an sample-based estimate line. Fundamentally, we apply a standard statistical approach of using information from a sample to estimate characteristics of a large population. Here the linear regression focus on the estimate of coefficient $\hat{\beta}$.

The most popular estimation method is *least squares*, also known as Ordinary Least Squares (OLS). OLS regression is an estimated model based on sample data in which we pick the coefficients $\hat{\beta}$ to minimize the residual sum of squares (RSS):

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - f(x_i))^2 = (y - X\beta)^T(y - X\beta) \tag{2.3}$$

To minimize (2.3), we differentiate the term with respect to $\beta$ and obtain:

$$\frac{\partial RSS}{\partial \beta} = -2X^T(y - X\beta) \frac{\partial RSS}{\partial \beta \partial \beta^T} = 2X^T X$$

Assuming that $X$ has full column rank, hence $X^T X$ is positive definite leading to optimality, we thus could set the first derivative to zero to get the $\hat{\beta}$:

$$X^T(y - X\beta) = 0 \Rightarrow X^T y = X^T X\hat{\beta}$$

We thus solve the unique solution:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.4}$$

Please note all this derivation makes no assumptions about the validity of model - it simply finds the best linear fit to the data except from ensuring $X$ is of full-rank. The non-full-rank case occurs most often when there is redundancy.

A natural way to resolve the non-unique representation is to drop redundant columns in $X$.

Lastly, since $\hat{y} = X\hat{\beta}$, we could form the predictions

$$\hat{y} = X\hat{\beta} = (X(X^T X)^{-1}X^T)y \tag{2.5}$$

We denote $P = X(X^T X)^{-1}X^T$ which is projection matrix. Moreover, we could represent the residual $\hat{\epsilon}$ with $P$:

$$\hat{\epsilon} = y - X\hat{\beta} = y - (X(X^T X)^{-1}X^T)y = (I_n - P)y \tag{2.6}$$

Note $Q = I_n - P$ is also a projection matrix. There are nice properties of *projection matrices* that we will use in the subsequent sections:

$$Q^T = Q;\ Q^2 = Q$$

### 2.1.1 Sampling Uncertainty and Statistical Inference

We infer from (2.4) that both $\hat{\beta}$ and $\hat{Y} = X\hat{\beta}$ are linear transformations of $y$ as random variables. If we had collected different sample data of size $n$, we would to be sure have different estimates, due to *sampling uncertainty*. We want to quantify this uncertainty by using *sampling statistics* and essentially get more control of our estimate coefficient $\hat{\beta}$. Furthermore, the inclusion of uncertainty measure such as standard error enables us to conduct statistical inference.

To pin down the sampling properties of $\hat{\beta}$, we include supplementary assumptions including *uncorrelated residual*, *constant variance of residual* $\sigma^2$, and additionally $X$ *is non-random*. Based on the assumption, we infer the variance of $y$:

$$Var(y) = \sigma^2 I_n \tag{2.7}$$

We could then derive the expectation of residual sum of squares (RSS):

$$RSS = Y^T(1-P)Y \Rightarrow E[Y^T(1-P)Y] = (n-p)\sigma^2$$

Hence we get the unbiased estimate of variance $\hat{\sigma}^2 = \frac{RSS}{n-p}$. We next derive the variance-covariance matrix of the $\hat{\beta}$ using equation (2.7):

$$\begin{aligned} Var(\hat{\beta}) &= Var(X(X^T X)^{-1}X^T) \\ &= X(X^T X)^{-1}X^T Var(y)(X(X^T X)^{-1}X^T)^T \\ &= \sigma^2(X^T X)^{-1} \end{aligned}$$

Therefore, the uncertainty in individual estimate $\beta_j$ is quantified by its *standard error (SE)*

$$SE(\beta_j) = \sqrt{\frac{\sigma}{(X^T X)^{-1}_{jj}}}$$

### 2.1.2 Rethinking the Unbiasedness of OLS

We require a property of unbiasedness for estimator $\hat{B}$. Unbiased estimator does not systematically over- or under-estimate the true parameter $B$.

Consider the decomposition of mean squared error of an estimator $\tilde{\theta}$ in estimating $\theta$:

$$\begin{aligned} MSE(\tilde{\theta}) &= E[\tilde{\theta} - \theta]^2 \\ &= Var(\tilde{\theta}) + [E(\tilde{\theta}) - \theta]^2 \end{aligned}$$

Here the first term $Var(\tilde{\theta})$ is the variance, while the second term $[E(\tilde{\theta}) - \theta]^2$ is the squared bias.

The Gauss-Markov (GM) theorem implies that the least squares estimator has the smallest mean squared error (MSE) of all *linear estimators with no bias* ( pay attention to the "no bias" here). However, there may well exist a biased estimator with smaller mean squared error. Such an estimator would *trade a little bias for a larger reduction in variance*. Put bluntly, *biased estimates are commonly used* for better out-of-sample performance (i.e., less MSE in test set). In reality, any method that shrinks or sets to zero some of the least squares coefficients may result in a biased estimate.

## 2.2 Assumptions of Linear Regression and Violation Implications

The question for the assumption of linear regression is ill-posed and requires more context as we need to specify the desired properties that we want the linear model to hold. Hence, we start with the standard assumptions that guarantee the above GM theorem to hold for OLS regression estimate: 1. There exists an additive linear model for $(X, Y)$ as equation (2.1),

$$Y = X\beta + \epsilon$$

Where, $\beta$ and $X$ are *non-random* while the randomness stem from $\epsilon$

2. Zero Expected Mean of Residual: $E[\epsilon_i] = 0$
3. Homoscedasticity: Constant Variance of Residual: $Var(\epsilon_i) = \sigma^2 < \infty$
4. Uncorrelated Residual Error: $Cov(\epsilon_i, \epsilon_j) = 0,\ \forall i \neq j$

Furthermore, if the error terms are 5. normally distributed and 6.identically and independently distributed (i.i.d.), we could infer that the OLS estimator becomes the Maximum Likelihood Estimation (MLE).

## Problem 1: Non-linearity of the Data

The linear regression model assumes that there is a straight-line relationship between the predictors and the response. If the true relationship is far from linear, then all of the conclusions that we draw from the fit are suspect.

*Residual plots* are a useful visualization tool for identifying non-linearity. Given a fitted linear regression model, we can plot the residuals, $e_i = y_i - \widehat{y}_i$, versus the predictor $x_i$. In the case of a multivariate regression model, we can plot the residuals versus the fitted values $\widehat{y}_i$. Independence assumption implies no discernible pattern. The presence of a pattern may indicate a problem with some aspect of the linear model. For instance, if residuals exhibit a clear U-shape, which provides a strong indication of non-linearity in the data.

## Problem 2: Correlation of Error Terms

An important assumption of the linear regression model is that the error terms $\epsilon_i$ are uncorrelated. Intuitively, if the errors are uncorrelated, then the fact that $epsilon_i$ is positive provides little or no information about the sign of $\epsilon_{i+1}$. One counterexample is to think about double a set of training observations $(X, y)$, as $X_{Copy} = [X; X], y_{Copy} = [y; y]$.

The standard errors that are computed for the estimated regression coefficients or the fitted values are based on the assumption of uncorrelated error terms. If in fact there is correlation among the error terms, then the estimated standard errors will tend to underestimate the true standard errors. As a result, confidence and prediction intervals will be narrower than expected.

Such residual correlations frequently occur in the context of time series data, which consists of observations for which measurements are obtained at discrete points in time. There is the issue of residual autocorrelation - tracking in the residuals that adjacent residuals may have similar values.

In general, the assumption of uncorrelated errors is extremely important for linear regression as well as for other statistical methods, and good experimental design is crucial in order to mitigate the risk of such correlations.

## Problem 3: Outliers

An outlier is a point which is far from the value predicted by the model. Outliers can arise for a variety of reasons, such as incorrect data collection. They are quite common in real dataset.

Residual plots can be used to identify outliers. If we believe that an outlier has occurred due to an error in data collection, then one solution is to simply remove the observation. Shrewd care and revised assumptions should be taken when removing outliers, since an outlier may instead indicate a deficiency with

the model, such as a missing predictor. 2008 global financial crisis is a data point that needs justification to be discarded as outlier.

### Problem 4: Collinearity

Collinearity refers to the situation in which a group of variables are closely related to one another. The presence of collinearity can pose problems in the regression context, since it can be difficult to separate out the individual effects of collinear variables on the response. Collinearity reduces the accuracy of the estimates of the regression coefficients, it causes the standard error for $\widehat{\beta}_j$ to grow.

Correlation matrix of the predictors is a simple way to detect collinearity. Unfortunately, not all collinearity problems can be detected by inspection of the correlation matrix: it is possible for collinearity to exist between three or more variables even if no pair of variables has a particularly high correlation - this is called *multicollinearity*.

There are in two practical solutions to the problem of collinearity. The first, and more straightforward one, is to *drop* one of the problematic variables from the regression. The second solution is to *aggregate* the collinear variables together into a single composite predictor. For example, apply PCA (Principal Component Analysis) approach to model the highly correlated features group and extract the first PC component as the representative indicator.

## 2.3 Conclusion

Linear models were largely developed in the precomputer age of statistics, but there are still robust baseline model to apply in today's computer era. They are simple and often provide an adequate and interpretable description of how the inputs affect the output. For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data.

# Chapter 3

# Methods

We describe our methods in this chapter.

Math can be added in body using usual syntax like this

## 3.1 math example

$p$ is unknown but expected to be around 1/3. Standard error will be approximated

$$SE = \sqrt{(\frac{p(1-p)}{n})} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

You can also use math in footnotes like this[1].

We will approximate standard error to 0.027[2]

[1] where we mention $p = \frac{a}{b}$

[2] $p$ is unknown but expected to be around 1/3. Standard error will be approximated

$$SE = \sqrt{(\frac{p(1-p)}{n})} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

# Chapter 4

# Applications

Some *significant* applications are demonstrated in this chapter.

## 4.1 Example one

## 4.2 Example two

# Chapter 5

# Final Words

We have finished a nice book.

# Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2021). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.24.