Netspar

Network for Studies on Pensions, Aging and Retirement

NETSPAR ACADEMIC SERIES

Predictive Returns using Machine Learning Techniques

Raimondo Grova

MSc 08/2019-011

Maastricht University

School of Business and Economics

Quantitative Economics

Predictive Returns using Machine Learning Techniques

Author:
Raimondo Grova

Supervisor:
Dr. Antoon Pelsser

August 2019

A thesis submitted for the degree of

MSc Econometrics and Operations Research

Abstract

In the paper, the econometrics and machine learning field in asset pricing are summarized. A complete review and comparison of techniques including Ordinary Least Squares, Elastic Net, Random Forest and Neural Networks for stock return prediction is performed. All approaches are tested using a predictive out-of-sample R^2 to observe their behavior under an unstudied sample. Furthermore a portfolio using long-short strategy is constructed. Results show that machine learning algorithm are raising hope in the asset field but remain inaccurate for the moment.

Key words: Machine Learning, Econometrics, Return Prediction,Portfolio, OLS, Elastic Net, Random Forest, Neural Networks.

List of Figures

2.1 Underfit vs Overfit 12

2.2 Bias-Variance Tradeoff 13

2.3 Ridge Regression 15

2.4 Lasso Regression 16

2.5 Penalized Regression 17

2.6 K-fold Cross Validation 18

3.1 Regression Tree 20

3.2 Random Forest 21

3.3 Biological Representation of a Neuron 22

3.4 The simple perceptron 23

3.5 Comparison of a biological representation versus a computer-based approach 24

4.1 Stocks representation 30

4.2 Stocks Boxplot 31

5.1 OLS R^2_{oos} 34

5.2 Elastic Net R^2_{oos} 35

5.3 Random forest R^2_{oos} 35

5.4 Neural Network R^2_{oos} 36

5.5 Rolling Window 38

5.6 Predictive Returns on Neural Networks 39

5.7 Predictive Returns on VAR 40

5.8 LS Strategy on Approaches 42

5.9 Gains/losses with L-S 44

List of Tables

3.1 Approaches Summary 26

4.1 Stocks Summary 31

5.1 Elastic Net on MMM 34

5.2 R^2_{oos} in percentage 36

5.3 Neural Network 12 months returns prediction 38

5.4 VAR 12 months returns prediction 38

5.5 12 months real returns 39

5.6 VAR Portfolio Strategy 41

5.7 Neural Network 12 months returns prediction 42

5.8 L-S on 12 months real returns 43

Contents

1 Introduction **6**

1.1 Literature 8

I Theoretical Section **9**

2 High Dimensional Econometrics Models **10**

2.1 Ordinary Least Square 10

2.1.1 Bias-Variance Trade-off 11

2.2 Penalized Linear Regression 14

2.2.1 Ridge Regression 14

2.2.2 Lasso Regression 15

2.2.3 Elastic Net Regression 16

2.3 Cross Validation 17

3 Machine Learning techniques **19**

3.1 Trees 20

3.1.1 Random Forest 21

3.2 Neural Networks 21

3.2.1 The Structure 24

3.2.2 Regularization 26
3.3 Approaches Summary 26

II Empirical Section **28**

4 Methodology and Data Processing **29**
4.1 Data Set 29
4.2 Accuracy Measures 31

5 Findings **33**
5.1 Out of sample R^2 33
5.1.1 Linear regressions 33
5.1.2 Non-linear regressions 35
5.1.3 First Test Conclusion 36
5.2 Portfolio Construction 37
5.2.1 Second Test Conclusion 44

6 Conclusion **46**

References **48**

Chapter 1

Introduction

Every investor, bank or hedge fund have the same objective of maximizing their returns while minimizing the risk they are taking. For decades, researchers have studied several new techniques to create a robust, profitable alternative to beat the market and maximize profits. With the emergence of machine learning and big data, they have built an extraordinary ability to manage a massive amount of data, leading to researchers seeking for new machine learning approaches in the asset field.

The objective of this paper is to study predictive returns using both basic and more advanced techniques in the field of machine learning and econometrics. Several advanced machine learning approaches are tested against basic regression modeling approaches. Furthermore, some researchers such as Gu, Xiu and Kelly (Gu, Kelly, & Xiu, 2018) provide evidence that machine learning and especially neural networks are leading to positive returns when a portfolio is built. By building an equivalent strategy and including market regulations (i.e transaction fees, 100-30, etc.), the goal is to test whether machine learning approaches are viable in such an environment or not. In the next subsection, a complete literature review is provided for lecturer to be aware of the following context.

The paper will be split into two main sections. First of all, the theoretical study presents the studied model in their technical environment through the literature review in the first chapter. The second chapter provides a full exploration of OLS, Ridge, Lasso and Elastic Net regression techniques. Their most common strengths and weaknesses for regression purposes will be presented. Furthermore, intuition on the Bias-Variance trade-off will be introduced. Chapter three depicts the theoretical framework of machine learning approaches, starting with the exploration of tree-based methods and concluding with neural networks.

Afterwards, an empirical study section is presented. In this last part, chosen models are tested using real stock historical data. A data processing section is dedicated to the pure data analysis and cleaning in chapter four. After this essential process, each model is tested out-of-sample to see their predictive R^2. In chapter five, a portfolio construction is

established to determine whether investing in the selected stocks is profitable or not. The last step aims at evaluating the performance of two opposite approaches on a real-world case. Finally, a conclusion is drawn in the last chapter.

1.1 Literature

To have a deep literature understanding of stock return prediction, it is relevant to go way back in time. In 1952, Markowitz (Markowitz, 1952) explored portfolio theory and underlined the importance of portfolio selection and the supreme objective of any investor, which consists of maximizing the profit while minimizing risks. In 1993, (Fama & French, 1993) designed a three-factor model for stock return. In a more recent 2015 paper, (Fama & French, 2015) developed a more robust version called the five-factor model.

At that time, machine learning was still unknown for most of the academic population, giving that most of the papers dealt with cross-sectional regressions of stock returns on lagged stocks characteristics.

After the emergence of big data, (Gu et al., 2018) tried to show the importance of machine learning techniques in the asset pricing field. By having a large panel of different methods, from basic OLS, shrinkage, and dimension reduction to advanced tree-based model and neural networks, the main goal of the paper was to predict the return of assets including more than 30,000 shares, and the stock predictive characteristics, to mention a few. To compare the methodologies and to assess their accuracy, a predictive R-squared is performed out-of-sample. Furthermore, they built a portfolio that is buying the 100 best stocks and selling the worst 100. Every month the time window is shifted and all the models re-assessed. In conclusion they found that the neural network and tree-based models outperformed all other models. Of course, their findings are not the only ones present in the literature. Since the discovering of neural networks and the emergence of big data, many researchers have tried to predict the return of assets. (Welch & Goyal, 2008) is examining the predictors in stock market returns. They affirm that since 1920 the number of researchers are primarily focused on trying to beat the market.

More recently, with machine learning, (Zhang, 2003) is trying to forecast time-series using hybrid modeling that combines both ARIMA and neural networks. (Yoo, Kim, & Jan, 2005) studied the stock market prediction using various machine learning techniques finding neural network as the best model, (Fung, Yu, & Lam, 20-23 March 2) have explored text mining techniques and real-time news in order to predict returns, (Freitas, De Souza, J N Gomes, & R De Almeida, 2019) has explored the neural networks in portfolio selection. (Chinco, Clark-Joseph, & Ye, 2017), has studied the sparse signals in the cross-section return world. Previously, (Schumann & Lohrbach, 1993) had detailed a comparison of statistical and machine learning methods for stock prediction.

From statistical to big data models, older to newer papers, practitioners continuously research the asset field. Different approaches can be explored. As illustrated in the literature, machine learning is powerful and can lead to excellent performance, outperforming the older findings. However, uncertainty remains in the field. Hopefully, the next sections will help in a better understanding of the subject.

Part I

Theoretical Section

Chapter 2

High Dimensional Econometrics Models

In this chapter, all the econometric models to estimate returns of stocks are presented in their theoretical background. The first model explored is the most simple. The approach sections are ranked by their complexities, with the most complex coming last.

2.1 Ordinary Least Square

In a linear regression[1], a response Y is predicted based on the predictor variables $X^T = (X_1, X_2, ..., X_p)$. This relationship can be written as:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{2.1}$$

Where the β are the unknown parameters. To solve such a regression, the estimation method is the least squares where the unknown parameters $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ use to minimize the residual sum of squares(RSS).

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_{ij})^2. \tag{2.2}$$

The main objective of the linear regression is to minimize the RSS. Let $\boldsymbol{X}$ be a $Nx(p+1)$ matrix with each row as an input vector and $\boldsymbol{y}$ be an $N-$vector of outputs, then

[1]Information about this chapter as well as the figures can be retrieved in (James, Witten, Hastie, & Tibshirani, 2014) chapter 2,3,6 and (Hastie, Tibshirani, & Friedman, 2001) chapter 2 and section 3.2,3.4

RSS can be re-written as:

$$RSS(\beta) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta) \tag{2.3}$$

As the above 2.3 is a quadratic function in the $p + 1$ parameters, the function can be differentiating with respect to β to obtain:

$$\frac{\partial RSS}{\partial \beta} = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) \tag{2.4}$$

$$\frac{\partial^2 RSS}{\partial \beta \partial \beta^T} = 2\boldsymbol{X}^T\boldsymbol{X} \tag{2.5}$$

If it is, then assumed, that $\boldsymbol{X}^T\boldsymbol{X}$ is positive definite, the first derivative can thus be equal to 0. Solving the equation will lead to a unique solution:

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

Once the parameters are estimated and the response predicted, it is wise to test the accuracy of the model. In order to do so, the R-squared(R^2) is explored. R-squared will assess the variability in the response explained by the variable X can be calculated by computing the following:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{2.7}$$

Where TSS, the total sum of squared is obtained via $\sum(y_i - \bar{y})^2$ and RSS, the residual sum of squared and can be solved using the equation 2.2:

R^2 is a double lying between 0 and 1. An R^2 close to 0 indicated that there is no explanation of the variability. Contrary to a very small R^2, a value close to 1 is showing that, a large proportion is explained by the regression itself. Depending on the application, the interpretation of the R-squared may vary.

2.1.1 Bias-Variance Trade-off

The most common problem when assessing the quality of fit of a specific model is the Bias-Variance Trade-Off dilemma.

Due to the comprehensive information processed, a poor selection of parameters can push the model to overfit the data. The model will produce a peak in accuracy with the

training data. However, as the data are split into training and test dataset, the model will result in reduced accuracy once tested out-of-sample.

Financial stocks are complex time-series composed of signals (useful) and various noise (useless). It is thus primordial to avoid overfitting in our model. An illustration of the difference between underfitting and overfitting can be visualized in figure 2.1. On the left picture, three estimates are shown: the linear regression in orange and two spline fits for the example. On the right figure, the corresponding MSE of the methods is provided.

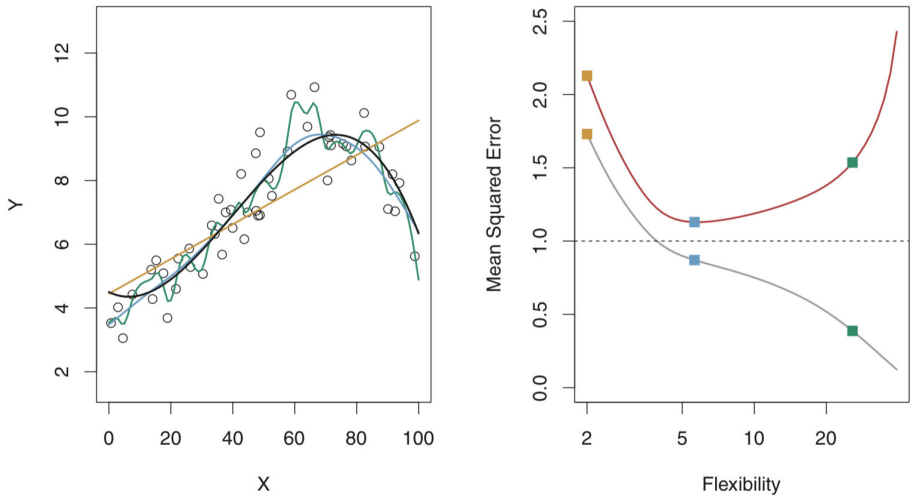


Figure 2.1: Underfit vs Overfit

When considering a regression case, based on training data $(x_1, y_1), ..., (x_n, y_n)$, a function $f(x)$ is built to approximate y at future observations of x. To be precise, $D = (x_1, y_1), ..., (x_n, y_n)$, thus the previous function $f(x)$ can be written as $f(x; D)$. Given x and D, the mean square error is:

$$E[(y - f(x; D))^2 | x, D] = E[(y - E[y|x])^2 | x, D] + (f(x; D) - E[y|x])^2 \tag{2.8}$$

Thus, the mean squared error of f as an estimator of the regression $E[y|x]$ is $E_D[(f(x; D) - E[y|x])^2]$ Where E_D is the expectation with respect to the training sample D The error

decomposition by S.Geman (Geman, Bienenstock, & Doursat, 1992):.

$$\begin{aligned}
E_D[(f(x;D)\text{–}E[y|x])^2] &= E_D[((f(x;D) - E_D[f(x;D)]) + (E_D[f(x;D)] - E[y|x]))^2] \\
&= E_D[(f(x;D) - E_D[f(x;D)])^2] + E_D[(E_D[f(x;D)] - E[y|x])^2] \\
&+ 2E_D[([f(x;D)] - E_D[f(x;D)])(E_D[f(x;D)] - E[y|x])] \\
&= E_D[(f(x;D) - E_D[f(x;D)])^2] + (E_D[f(x;D)] - E[y|x])^2 \\
&+ 2E_D[f(x;D) - E_D[f(x;D)]].(E_D[f(x;D)] - E[y|x]) \\
&= \underbrace{(E_D[f(x;D)] - E[y|x])^2}_{bias^2} + \underbrace{E_D[(f(x;D) - E_D[f(x;D)])^2]}_{Variance}
\end{aligned}$$

As seen in the decomposition above, the expected square difference can be expressed as the sum of the square bias and the variance. The bias represents: "average prediction over all data sets differs from the desired regression function." The variance: "measures the extent to which the solutions for individual data sets vary around their average, and hence this measures the extent to which the function $f(x;D)$ is sensitive to the particular choice of data set."(Bishop, 2007)

The goal is thus to minimize the expected loss, expressed as:

$$L(x) = B^2 + \sigma^2 + \epsilon \tag{2.9}$$

Where B is the bias, σ^2 the variance and ϵ the noise. There exists a trade-off between the bias and the variance that is essential when elaborating the model. As illustrated in figure 2.2, on one side, a weak bias will result in underfitting of the model, and on the other side, a high variance leads to overfitting. The objective to find the balance between both the bias and variance can be achieved by having both low variance and bias to minimize the expected loss.

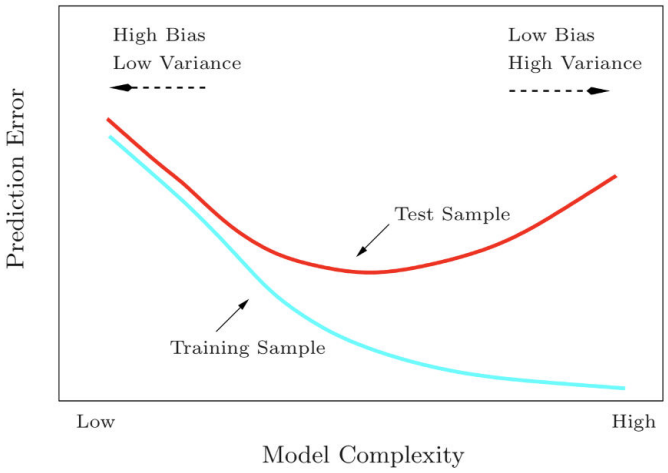


Figure 2.2: Bias-Variance Tradeoff

2.2 Penalized Linear Regression

Standard linear models perform poorly when they are facing complex time-series and financial data. An alternative to the problem that faces linear methods is to allow the regression models to have a constraint in the equation. The constraint will lead to penalties in the regression model. These sets of methods are also called "Shrinkage Methods." In this section, ridge, lasso and elastic net are detailed.[2]

2.2.1 Ridge Regression

By adding a penalty to control the magnitude of their size, the ridge regression (Hoerl & Kennard, 1970) shrinks the regression parameters. Instead of minimizing the RSS as the linear regression is accomplishing, ridge is minimizing the penalized RSS.

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \Big\} \tag{2.10}$$

where $\lambda \sum_{j=1}^{p} \beta_j^2$ is the function. λ is the coefficient or parameter that will decide on the shrinkage intensity, in this case $\lambda \geq 0$. Of course, if λ is large, the shrinkage will follow the same pattern and the β converges to zero. The ridge regression can also be written as an optimization problem where:

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$
$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 \leq t$$

Ridge is also alleviating one complication that the linear regression cannot handle. In OLS, if a large number of correlated variables are estimated, their parameters can poorly be identified, which will lead to having high variance. By imposing a penalty and having a size constraint, the ridge regression is able to surmount such an obstacle. After having centered the input values, the RSS can be written (in matrix form):

$$RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda \beta^T \beta \tag{2.11}$$

[2]Can be retrieved in (James et al., 2014) section 6.2 and (Hastie et al., 2001) section 3.4, (Tibshirani, 1994), (Hoerl & Kennard, 1970), (Zou & Hastie, 2005)

Solving using the same procedure as previously executed in 2.4 and 2.5, the estimated β is:

$$\hat{\beta}^{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{2.12}$$

Where $\boldsymbol{I}$ is the identity matrix. It can be visualized that compare to $\hat{\beta}$ in 2.6, in the ridge case, the β will vary and leads to OLS if the lambda selected is equal to zero. This makes the model more powerful and more flexible while keeping the same interpretability. However, as the predictors in the model are not equal to zero but only converge through that value, the model is keeping all the predictors which will not lead to a parsimonious model. An illustration of the ridge regression can be interpreted in figure 2.3

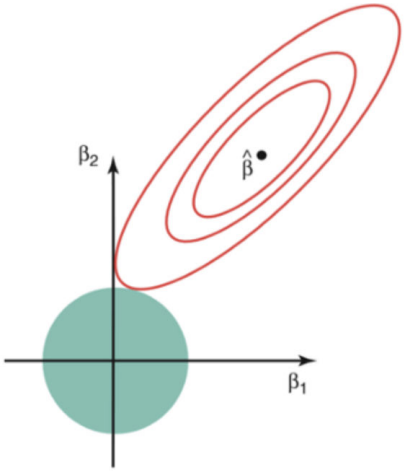


Figure 2.3: Ridge Regression

2.2.2 Lasso Regression

Lasso (Tibshirani, 1994) can be referred to the Least Absolute Shrinkage and Selection Operator. This method is another variation of the linear regression. Lasso is the same as ridge but with a subtle difference in penalizing. Following the same principle as ridge, the objective is to minimize the penalized RSS with below function 2.13

$$\hat{\beta}^{lasso} = \underset{\beta}{\operatorname{argmin}}\Big\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\Big\} \tag{2.13}$$

For an easier overview, the function is written with the constraints:

$$\hat{\beta}^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t$$

Instead of using the squares, the lasso will adopt the modulus of β_j (also called L_1 norm) in the constraint function.

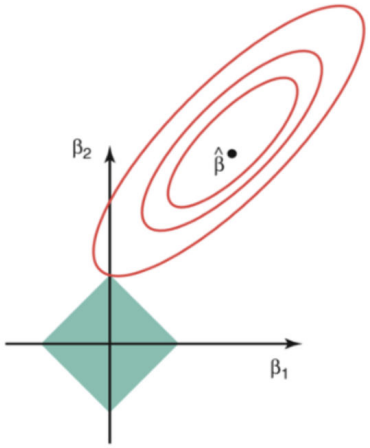


Figure 2.4: Lasso Regression

To see the difference graphically, as illustrated in 2.4 and 2.3, the green areas represent the constraint functions. A visual representation is helpful in the sense that the RSS ellipse (in red) enters in contact with the constraint function is different in both methodologies. With a sphere, the ellipse will enter in contact the constraint region outside of the axis which will lead to non zero coefficients, while in the lasso regression with such a sharp shape, the ellipse will arrive in the constraint at the axis. Once such an event happens, the coefficients will then be equal to zero. This event makes the lasso to have an automatic variable selection, which is very useful in big data. Despite its ability to automatically select variables, lasso has some inconveniences. For example, if variables are highly correlated, lasso will select one of them without taking care of the others in the group. Another limitation of the lasso is that if $p > n$, it will select at most the n. In other words, lasso is bounded by the sample no matter the number of predictors.

2.2.3 Elastic Net Regression

Elastic Net (Zou & Hastie, 2005) is the newest version of penalized linear regression. With its ability to overcome the inconvenient of the lasso regression while keeping the strengths

of both ridge and lasso, elastic net is the most powerful shrinkage model. The predicted β is equal to:

$$\hat{\beta}^{ElasticNet} = \underset{\beta}{\operatorname{argmin}} \Big\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \Big\} \tag{2.14}$$

The equation 2.14 represents the way of minimizing $RSS(\beta)$ plus the constraints of lasso and ridge.

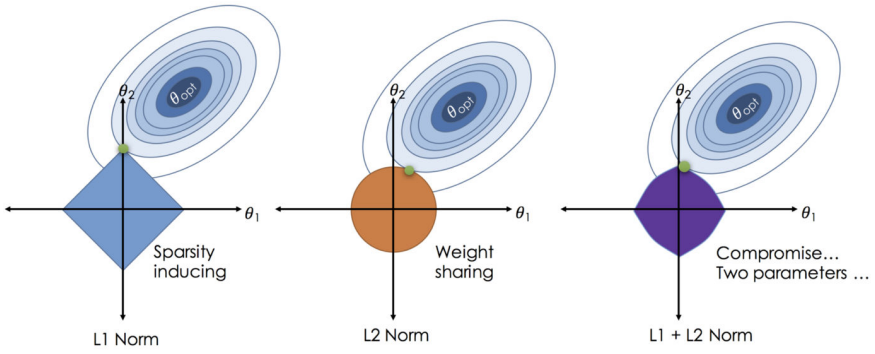


Figure 2.5: Penalized Regression

As illustrated in figure 2.5, Elastic Net is combining both L_1 and L_2 norm. Geometrically, the shape of the constraint is thus a combination of sharp and circular shape. Elastic Net is removing the group selection effect as well as the bounding issue of Lasso. Furthermore, it will avoid sparsity (zero values) and density (non-zero values) in the model.

2.3 Cross Validation

Cross-validation is one of the most widely used method in the literature for error prediction estimation. Having a considerable amount of data for modeling is excellent. Indeed, having much information will undoubtedly help the model to predict accurately. However, as seen in the bias-variance trade-off, having the right balance is not that simple. Having only the training set and force-feed the model with data will lead to overfitting of the model. On the contrary, having only a few training samples will lead to miss important information and thus conduct the model to underfit. A possible way of bypassing this issue is the k-fold cross-validation.

The principle of the k-fold cross validation is roughly simple. The algorithm will use part of the data to fit the model and the rest to testing it. It is dividing the sample into

K subsets (equally-sized). The principle is looped k times such that each time, $k-1$ subsets will be used to train the model and the last inch to test it. The error estimation is then the average of all k trials.

Mathematically, the k-fold cross-validation can is expressed as follows. Suppose, $k : 1, ..., N \to 1, ..., K$ is function that indicate which partition is allocated randomly. Then the estimating error is calculated as:

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{2.15}$$

Where $\hat{f}^{-k}(x)$ is the fitted function.

To have a visual illustration on how the procedure works, figure 2.6 is a graphical representation of the k-fold cross validation method.

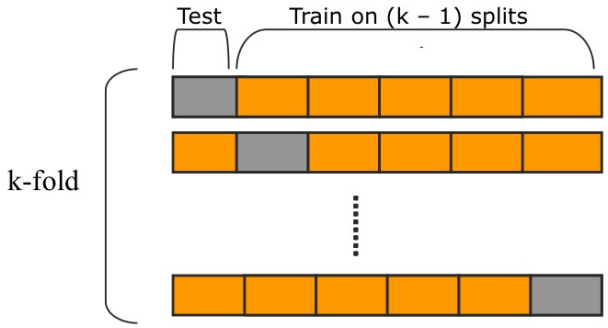


Figure 2.6: K-fold Cross Validation

Chapter 3

Machine Learning techniques

In order to solve a problem with a computer, an algorithm is employed. An algorithm is a series of instructions for the computer to follow to achieve a particular goal. In other words, it is a process that will transform input into an output. Of course, nowadays, there exist various algorithms that can achieve the same goal. However, the difference lies in which of them is the most efficient in terms of minimizing the number of instructions, robustness, or memory allocation, to mention a few. As covered by Alpaydin (Alpaydin, n.d.), it may be required to have other possibilities to solve a specific case. Indeed, there might be problems (such as the difference between conventional email and spamming (Alpaydin, n.d.)) that are unsolvable by the algorithm at first sight. Despite this lack of knowledge, thanks to the daily signs of progress of artificial intelligence and computers, a vast amount of data can be extracted from the internet. This amount of data will help the machine to learn and automatically solve the obstacle that they are currently facing. From an artificial intelligence point of view, machine learning is now capable of adapting themselves to the most modern environment.

From health care to finance, the range of applications of machine learning is vast and flexible. It can be used to detect a tumor in the scanner as well as being exploited in digital marketing analysis. It is conventional to separate the machine learning section into two classes namely Supervised and Unsupervised learning. While the supervised task will give priority to what the output should be and what is the best approximation of it, on the other approach, the unsupervised learning will be to infer the structure of the data without having any output

In the investigation of stock returns, supervised learning is exploit. Two main tasks can be differentiated from the supervised learning tool, namely classification and regression. In this particular environment, regression is applied.

3.1 Trees

Tree-based methods[1] for regression are widely used because they are simple and have a more pronounced interpretation than linear methods. Before going in depth, some basic properties need to be mentioned. A decision tree is a procedure with a tree-like structure. The data set will be split into subsets, and then this procedure is recursively repeated to each subset. A graphical representation can be found in figure 3.2.

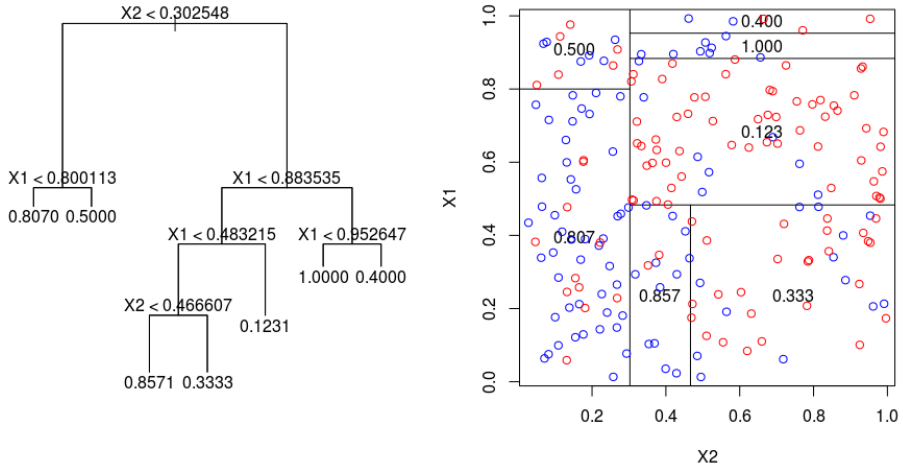


Figure 3.1: Regression Tree

Mathematically speaking, the possible values for $X_1, X_2, ..., X_p$ are divided into J regions (no overlap) $R_1, R_2, ..., R_j$. Once the regions are set up, the same prediction is made for the observations that fall into the same region. Obviously, again, the goal is to minimize the residual sum of squares here given in 3.1.

$$RSS = \sum_{j=1}^{J} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_j})^2 \tag{3.1}$$

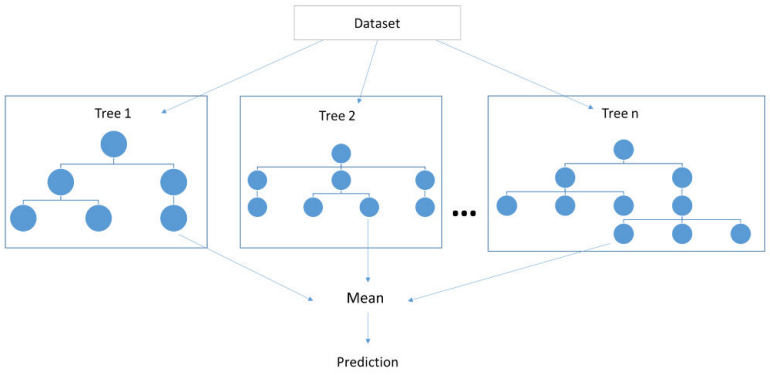
However, if a massive data set is used and split into n regions, it is computationally too expensive to consider all possibilities. To avoid overfitting of data and greedy approaches, it is recommendable to grow a large tree and then prune it back to subtrees for better test performances.

[1] Can be retrieved in (James et al., 2014) Chapter 8

3.1.1 Random Forest

After having given a clear definition of decision trees and how it is computed, it might be essential to improve the predictions given by the decision trees. Bagging and Boosting are two other well-known tree-based methods. In this paper, neither of the two will be used for several reasons. Boosting is sensitive to noisy data, which makes it difficult in the case of stocks where the time-series is composed of only small signals. Furthermore, in boosting, the tree is grown sequentially based on previously grown trees, which lead to slow computation. Due to the data available and the weaknesses mentioned above, the random forest is preferred over the two others.

Random Forest (Breiman, 2001) [2] is an improved version of bagging where random subsets from the training data are constructed. After that, each subset is training their own decision trees using bootstrapping. Finally, an average of all predicted trees is made. Random forest is following the same structure as the bagging method, which will build an ensemble of decision trees trained using bootstrapping. However, one crucial difference is made once the decision trees are built. Instead of using all the features that are growing the tree, only a random selection is selected. The term of random forest is employed when several random trees are built.



Forest.png

Figure 3.2: Random Forest

3.2 Neural Networks

Machine learning techniques[3] have become more and more popular this past decade in finance. Many researchers and practitioners have tried to use these models to predict the return of stocks, to predict the market, etc. Supervised learning algorithms such as the neural network can be robust but needs to be used with precaution. To understand how the

[2]Can be retrieved in (Breiman, 2001), (Blockeel, 2010) chapter 4 and 13, (James et al., 2014) chapter 8

[3]This section can be retrieved in (Hastie et al., 2001) chapter 11, (Bishop, 2007),(Zhang, 2003), (Blockeel, 2010) chapter 10

neural network is modeled and used by practitioners, it is relevant to refer to the biological history of neural networks.

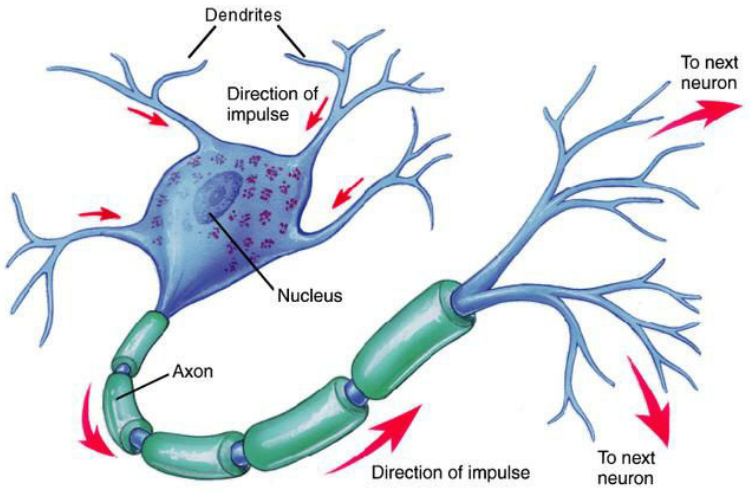


Figure 3.3: Biological Representation of a Neuron

The biological neuron has several characteristics that are worth mentioning. The central core of the neuron is the nucleus which is surrounded by several branches that are themselves attached by sub-branches named dendrites. These dendrites are the input of the neuron and, thus, is the part where the impulse is sent. This impulse was given from the dendrites and will run across what is called the axon. The axon is referring to the tail covered by an isolated substance from the rest of the environment. At the end of this axon, sub-branches that will connect to the next neuron can be identified. The biological representation of the neuron gives sufficient insight on how to construct an artificial neural network. At a high level and without going into too many details, the dendrites are the input of the neuron. Once they receive enough chemicals in their environment, the compact nucleus sends an electrical pulse that goes through the axon and passes the information to the next neuron affixed to it. Moreover, if the dendrites do not receive enough, the nucleus refuses to send a pulse and, thus, no output is given. Artificial neural networks follow the same principle over an ensemble of neurons. An illustration of a single neuron provides the same characteristics as the biological overview. A signal is sent from the inputs, run across the process, and deliver a final output. A single artificial neuron is called a "perceptron."

The simplest perceptron represents a single neuron. Mention by (Alpaydin, n.d.), each input nodes $n_i, i = 1, ..., D$ is affixed to a numerical value x_i. Each input is then, associated to a weight connection w_i. Finally, the output y can be seen as a transfer function of a weighted sum of the inputs:

$$y = f(\sum_{i=1}^{D} w_i.x_i) \tag{3.2}$$

A simple illustration of the basic structure is illustrated in the below figure 3.4 .

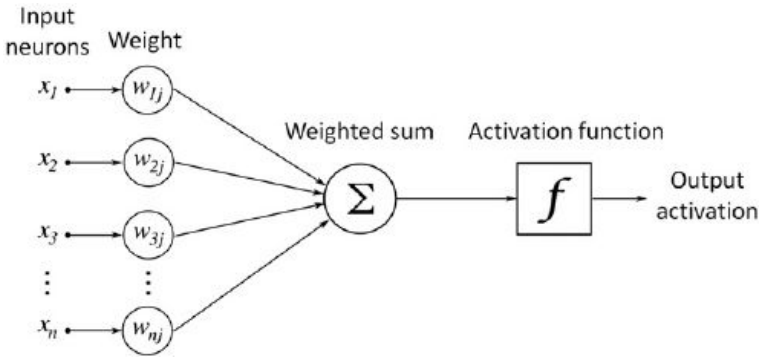


Figure 3.4: The simple perceptron

The transfer function f can be selected from various activation functions. The most widely used is the logistic function defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

The main alternative to the logistic function that is also mainly used in the literature is the Relu activation function. This activation function can be defined as:

$$f(x) = max(0, x) \tag{3.4}$$

The key characteristic of this function is that when using a massive number of layers and nodes, it will enable the model to be computationally less expensive. Of course Relu has some downsides, such as the dying Relu problem. In most of the papers, if the number of layers and nodes is vast, Relu is preferred to the logistic. In this study, the architecture of the neural network will remain at a low level. Other activation functions such as *linear*, *tanh* or $softmax$ can also be considered.

In practice, however, only a very few application suggests a single perceptron. Due to the limitations of a single perceptron that will only represent the linear component between classes, complex applications suggests working with a more robust and sophisticated perceptron by adding more neurons to form a network. The principal structure of the multilayer perceptrons remains the same as a simple perceptron. The structure will now consist of inputs of predictors, followed by one or several hidden layers to finally the output.

An illustration of the differences between one neuron and a group of such can be analyzed in figure 3.5.

Adding neurons to the group forces the output to be:

$$y_t = w_0 + \sum_{j=1}^{q} w_j . g(w_{0,j} + \sum_{i=1}^{p} w_{i,j} y_{t-i}) + \epsilon_t \tag{3.5}$$

where, $w_{i,j}(i = 0, ..., p)(j = 0, ..., q)$ and $w_j(j = 0, ..., q)$ are connection weights; p is the number of input nodes, and q the number of hidden units. In the hidden layer, the weights are summed and then the logistic function, which will transform the function into a non-linear one, is often used as the activation function (Zhang, 2003).

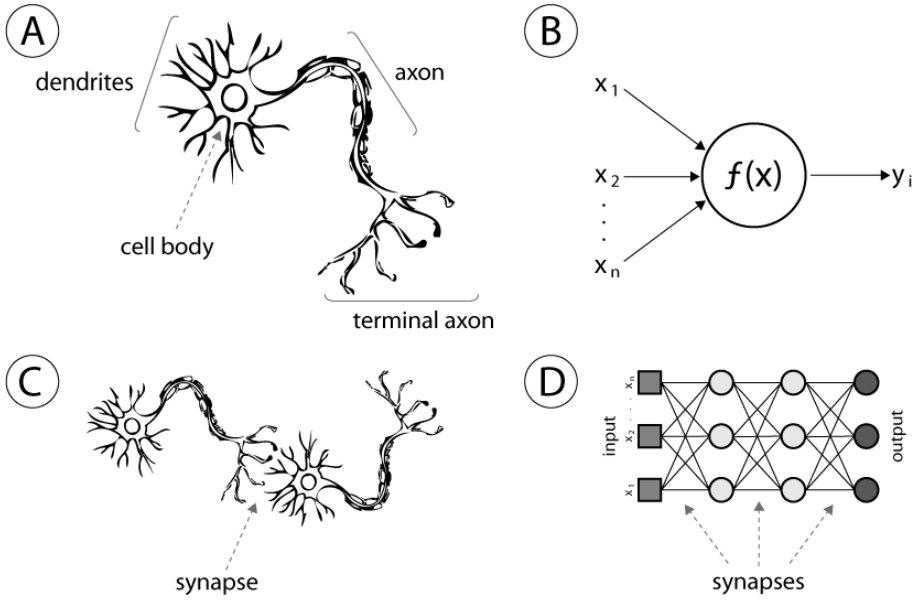


Figure 3.5: Comparison of a biological representation versus a computer-based approach

3.2.1 The Structure

The purest form of the neural network - a multi-layer feed-forward network with the logistic activation function - is modeled in this study. The structure defined here, however, delivers some uncertainties. As such, the structure of the network will be a crucial player in the model to success or failure.

The number of hidden neurons or layers, as well as the number of nodes is difficult to handle at first sight. Although there are no appropriate formulas or general guidelines for nodes and layers selection, Timothy Masters (Masters, 1993) suggests using some basic rules of thumb for choosing an appropriate number during the structure building.

The first element to choose when building the neural network is the number of layers. Many old pieces of research suggested using very few hidden layers. However, recent studies delivered new promises. With the emergence of deep-learning, studies by Eldan and Shamir (2016), Hinton et al. (2006), He et al. (2016) have shown that it is more valuable to increase the depth of the layers. There are no correct answers between shallow or deep learners. The choice depends on the data studied and explored.

The last element to decide when engineering the structure of the neural network is the number of nodes inside the layers. This last step is crucial for the model to be adequate to the data. Indeed, having too many nodes will be computationally intensive, which will result in too much waiting time, causing the training to be impossible to exploit. Furthermore, another cause of intensive computation is overfitting. The next subsection is devoted to the overfitting problem that faces the neural network. For the two reasons mentioned above, it is of high importance to aim for the minimum number of nodes.

By following a triangle or pyramid progression from the base (the input) until the summit, which represents the output, the number of nodes will decrease toward the outcome. If the neural network is represented by only one hidden layer, then the number of nodes is obtained by:

$$NumberOfNodes = \sqrt{n * m} \tag{3.6}$$

Where n are the inputs and m the outputs. If the number of hidden layers is two, then the principle remains the same, but the formula of the nodes is slightly more advanced. Indeed a new variable has to be defined as:

$$r = \sqrt[3]{n/m} \tag{3.7}$$

Then the number of nodes by hidden layer can be calculated using the formulas below:

$$NumberOfNodes_{H1} = m * r^2 \tag{3.8}$$

$$NumberOfNodes_{H2} = m * r \tag{3.9}$$

Although this guideline applies to most of the cases, this is not a general rule for building neural networks. If the number of inputs is the same as the outputs, then an auto-associative neural network is faced, and the rule of thumb is not possible. The same rule applies if the number of inputs and outputs is meager. Thus, it is essential to use these rules with care.

3.2.2 Regularization

To avoid overfitting, several methods can be exploited. The first one is the Early Stopping regularization. Too much training is leading the model to overfit while on the opposite, too few training is underfitting. Avoiding poor results on test sample, early stopping will stop the training procedure as soon as the performance of the validation is decreasing. To be concise, the model is trained based on the training sample. At the end of, for example, E epochs, the performance(on test sample) is assessed. Note that one epoch is a manner to define the process that the dataset has been throughout the entire network back and forth one time. If the model is outperforming the best current model, then the model is saved at that epoch.

Based on the same process of avoiding overfitting and understanding more data, it is not possible to pass the entire dataset at once in the network. To do so, dividing the data set into batches is helping the network to overcome this obstacle. Batch learning is just diving the all data set into parts to complete an iteration. To be more detailed, supposed the sample is composed of 4000 elements. These elements can be decomposed into batches of 500 elements. It will, thus, take 8 iterations to complete one single epoch.

3.3 Approaches Summary

In this last chapter of the theoretical part, a complete comparison of all approaches presented in the previous chapters is performed. The objective of this chapter is to synthesize the strengths and weaknesses of all models as well as exposing why such approaches are used in the asset field.

For illustration purpose, the table 3.1 provides the strengths and weaknesses of every model.

	Strengths	**Weaknesses**
OLS	Very Intuitive Easy interpretation	Sensitive to outliers Not suited for high-dimensional problems
Elastic-Net	Suited for high-dimensional problems Combine both L1 and L2 regularizations Avoid overfitting	Perform poorly if there is non-linear relationships Not flexible enough to capture complex problems
Random Forest	Can deal with massive amount of data Highly flexible Robust to outliers	Sensitive to parameters Overfitting
Neural Network	Can deal with massive amount of data Can approximate non-linear relations Highly flexible	Hard to interpret Sensitive to parameters Overfitting

Table 3.1: Approaches Summary

Predicting stock return is a complex task. Indeed, stocks are following a random walk. Furthermore, they are composed of only few signals which renders the task even more complicated. All models have their own strengths and weaknesses.

OLS is the easiest approach. Based on predictors, the approach is trying to get a quantitative response. Once the data are trained to produce the estimated coefficients, the response can be predicted. This approach is the most intuitive technique presented above. However, OLS have drawbacks that are not suited for stocks prediction. In stocks, outliers have an economic interpretation such as economic crisis, political news, environmental catastrophe, etc. For this reason, outliers are not removed from the sample. One of the big weaknesses of the linear regression is that the model is very sensitive to outliers. To improve the linear regression results, the elastic-net is proposed. This approach is more suited to big data as it will combine both shrinkage ridge and lasso methods. By using both L_1 and L_2 norms, the technique will work on the bias-variance tradeoff by reducing variance and increase bias. By working on the tradeoff, Elastic-net is preventing overfitting. However, as it is still a linear approach, the model is not able to capture non-linear relationships of the stocks.

Machine learning approaches are overcoming most of the weaknesses of the linear models. However, their weaknesses are much more rough to handle. Even if they should produce more accurate results due to their capacity to handle huge amount of data and being very flexible, they suffer from overfitting and parameters sensitivity. Indeed, selecting a specific parameter can totally change the outcome of the prediction.

In conclusion, while OLS and Elastic Net are intuitive and have an easy interpretation, the machine learning approaches (Random Forest and Neural Network) are much more rough to handle due to their parameters selection. Despite their intuitive characteristics, both the Elastic Net and OLS are unable to correctly capture non-linear patterns. This drawback is easily outperform by the machine learning techniques that deal with a massive amount of data and non-linear components. However, even if the random forest and neural network tends to be more suited to complex time-series such as stocks, they are remaining very sensitive to overfitting.

Predicting stocks resolves more into an art than purely science. As there is no reference model for prediction stock prices, comparing approaches that differs in their strengths and weaknesses gives an overview on what model to select.

Part II

Empirical Section

Chapter 4

Methodology and Data Processing

In the empirical section, the stocks gathering and assessing processes are established. The first section is presenting the dataset and detailing the inputs of the models. Afterwards, to avoid overfitting, data is split into subsets. Then, due to the complex selection process, an entire section on how the parameters of the neural network are chosen is explored. Finally, performance evaluation is detailed.

4.1 Data Set

The data is gathered on several websites. The importance and relevance of the data are crucial to accurately feed models. SP500[1] stocks are directly downloaded in R using the *"GetBatchSymbols"*. The monthly data analyzed is taking the window from 1970 to December 2018, collecting 588 observations per stocks. Due to the entry of tech companies in the list, some data is unavailable in the oldest periods. To overcome this obstacle, processing is performed to delete all companies with NA values in it. In the end, only 25 companies remain in the sample. From that sample, five are chosen as a basis. To calculate the return of the stock, two options are established. In the first case, only the lagged prices are taken as input. In the second one, according to the literature, it is relevant to include predictors in the sample. Indeed, based on (Welch & Goyal, 2008), historical prices of SP500, Dividend-Price Ratio (d/p), Earning-Price Ratio (e/p), Treasury Bills (tbl) are used as input.

- The first input to be analyzed is the Dividend-Price Ratio(d/p). Data for each stock is gathered from the Robert Shiller is website. The dividend-price ratio is the logs dividend of a certain share divided by the price of the same share.
- Earning Price Ratio(e/p) are also downloaded from the Robert Shiller is website. This ratio can be calculated by computing the difference between the logs of earnings and

[1] https://www.slickcharts.com/sp500

prices of the SP 500 index

- Treasury-Bills data are from the Federal Reserve Bank at St Louis (FRED). It represents the 3-months treasury bills in this particular case.
- The last predictor to be used is the historical SP500 index price.

However, due to academic rights, it is impossible to gather more predictors.

The five stocks studied are 3M Company (MMM), Altria Group Inc(MO), American Electric Power (AEP), Arconic Inc(ARNC) and Boeing Company(BA). The five time-series are shown in figure 4.1.

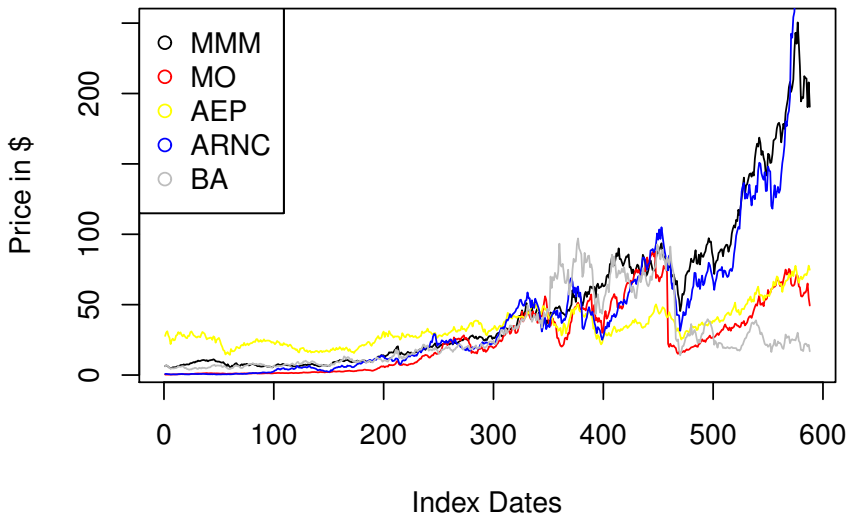


Figure 4.1: Stocks representation

	company	tickers	GICS.Sector	Date.first.added	CIK	Founded
1	MMM	3M Company	Industrials		66740	1902
2	MO	Altria Group Inc	Consumer Staples		764180	1985
3	AEP	American Electric Power	Utilities		4904	1906
4	ARNC	Arconic Inc.	Industrials	1964-03-31	4281	2016
5	BA	Boeing Company	Industrials		12927	1916

Furthermore, a table summarizing the dataset exploited in the study is developed in the table below. For visual representation purposes, boxplots are illustrated as well.

	MMM	MO	AEP	ARNC	BA
Min	4.71	0.32	14.38	3.76	0.40
1st Q	9.59	2.13	25.00	8.89	5.38
Median	27.28	20.27	31.12	19.57	24.56
Mean	50.10	25.09	34.08	27.81	46.59
3rd Q	77.45	41.56	41.43	37.11	62.95
Max	250.50	87.81	77.74	97.03	371.90

Table 4.1: Stocks Summary

The table 4.1 shows that each stock has highly fluctuated in 40 years. Even if they had a constant small increase in the first 300 months, as depicted in figure 4.1, the oscillation of the prices is volatile which makes the time-series more rough to handle and manipulate.

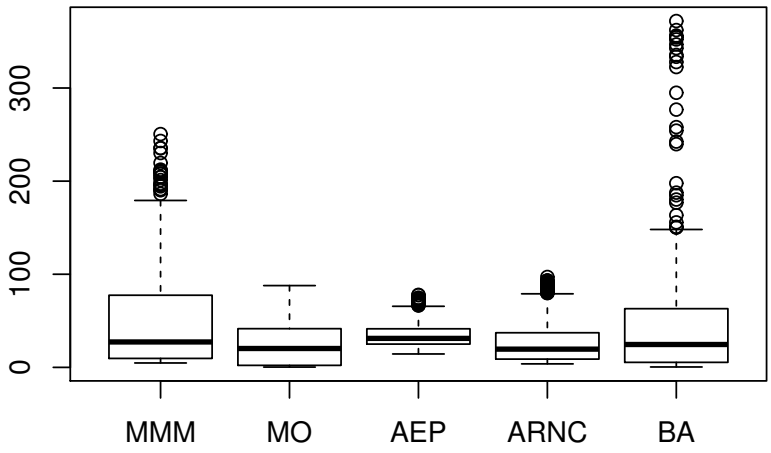


Figure 4.2: Stocks Boxplot

4.2 Accuracy Measures

To assess a model performance, a two-step approach can be used.

The first one, which is the most standard, is the R-squared. As described in section 2.1, this method is simple. An out-of-sample R^2 is estimated for the OLS, Elastic-Net, Random Forest, and Neural Network models. The out-performer model is then selected to be part of the second performance evaluation process.

Once the best model chosen, a portfolio is constructed by shifting rolling window. This approach is more realistic to assess performances. Suppose an investor decides to invest in the five chosen stocks, what return will he be able to get after one year? By forecasting one month ahead and build a zero net investment portfolio that buys and sells shares based on their predictive returns, the investor is going to have a prediction on how much money he can make. In this assessment process, the out-performer of the first evaluation is compared to a basic Vector Auto-Regressive model (VAR). Based on the literature (Gu et al., 2018), the neural network should impose himself as the out-performer model. Considering this outcome, it is wise to compare this predictive non-linear modeling to a linear model. While the linear VAR is capturing only very few components, the non-linear models will do the opposite. Financial time series are volatile and hard to predictable with accuracy. Furthermore, they are composed of few signals and the majority is composed of noise. Those aspects make it challenging to model.

Chapter 5

Findings

Over the years and the scientific literature, many different approaches have emerged. In the theoretical part, each models has been exposed to a number of advantages and disadvantages. In practice it is much more rough to handle models that are robust theoretically and empirically. In addition, the models are subject to a lot of criticisms from the researchers. Thus, it is of high importance to select approaches that differs in their characteristics and make our own opinion on the subject.

All of these reasons explain why it is necessary to interpret also the results with caution. One indicator hardly takes into account all the aspects that can influence the stocks.

In this section, the models developed in the theoretical framework are tested using the accuracy measures. The five stocks are compared and the out-performer model will be selected to perform the second test phase.

5.1 Out of sample R^2

The first part to be explored is the out-of-sample predictive R^2 (or R^2_{oos}). Each stock is fitted using different methodologies. Afterwards, the trained models are used to make prediction on a new sample. This technique allows the models to be faced by sensed data. For consistency purposed, all the models are tested using the same inputs.

5.1.1 Linear regressions

The first modeling approach to be tested is the OLS. Surprisingly, OLS achieves good results on both MMM and BA with more than 15% r-squared. For the other stocks, OLS performs poorly with only 2.8% for AEP. On average, on the five stocks studied, OLS is having a 9.5%

R^2_{oos}. By imposing a penalty when fitting the model, Elastic-Net should improve the basic linear regression developed earlier.

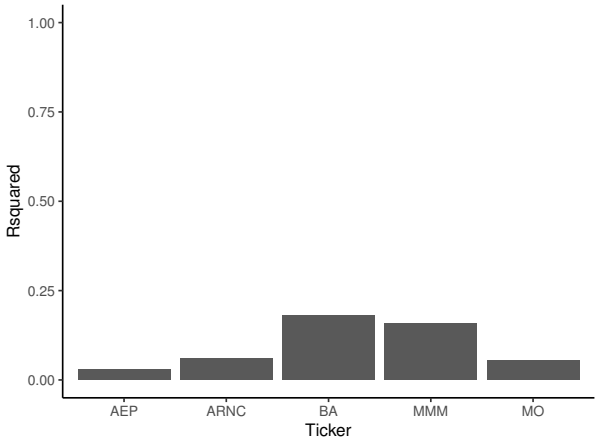


Figure 5.1: OLS R^2_{oos}

In Elastic-net a 10-fold Cross Validation (see section 2.3) is processed. On each subset created by the cross validation, the results are summarized in the table below (for the MMM stock). The subset that produces the lowest Mean Absolute Error is then exploited for the prediction. As it can be seen in table 5.1, for the MMM, the best model is using $\alpha = 0.55$, which means that the model lies between Ridge and Lasso.

alpha	λ	RMSE	MAE
0.10	0.00038	0.03131	0.02378
0.10	0.00381	0.03198	0.02445
0.10	0.03813	0.04422	0.03420
0.55	**0.00038**	**0.03130**	**0.02374**
0.55	0.00381	0.03332	0.02491
0.55	0.03813	0.04951	0.03583
1.00	0.00038	0.03137	0.02377
1.00	0.00381	0.03571	0.02584
1.00	0.03813	0.05789	0.04481

Table 5.1: Elastic Net on MMM

As illustrated in table 5.1, the best model in terms of the lowest RMSE and MAE is with $\alpha = 0.55$ and $\lambda = 0.000381$. After prediction, the R^2_{oos} of MMM is only improving by 0.3%. However, for the other stocks, the difference is a bit more significant. For AEP, for example, the R^2_{oos} is enhancing by 1.3%. As expected Elastic Net is generally more powerful compared to the OLS. In average the Elastic Net method has a 9.7% R^2_{oos}. Even though an improvement can be seen, Elastic Net fails to enhance the results significantly. For a better overview, the graphical representation of the stocks and their R^2_{oos} are illustrated below.

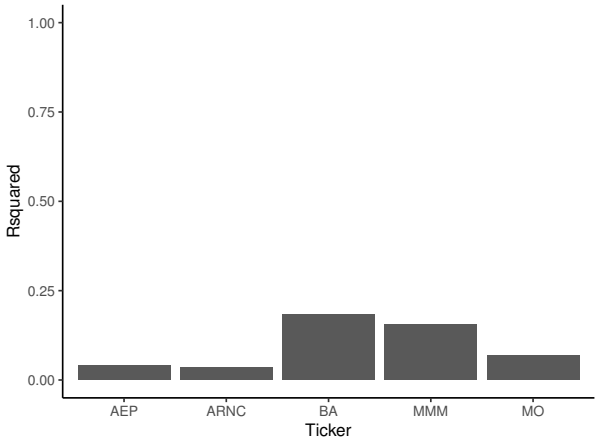


Figure 5.2: Elastic Net R^2_{oos}

Now that linear models have been tried out, it is interesting to see how well the non-linear models perform against the linear models.

5.1.2 Non-linear regressions

In the machine learning section, two models are assessed. In the random forest, the number of variables tried at each split is equal to $n_variables/3$ as a standard method. Thus, two variables are tried at each split. By definition, 500 trees are grown in the model. Because the random forest is building decision trees using a different bootstrapping sample at each run, the results differs at each iteration. To be precise in the out-of-sample R^2, fifty R^2 are performed and then averaged.

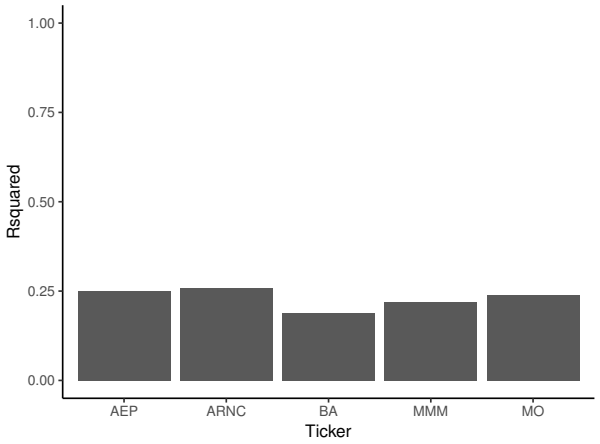


Figure 5.3: Random forest R^2_{oos}

The random forest is drastically improving the performance of the linear models.

Indeed, in figure 5.3, for most of the stocks, the R^2_{oos} is constant with approximately 25%. Only for BA, it only surpasses the Elastic-Net by 0.7%. In conclusion, compared to the linear methods, the random forest outperforms its competitors. On average, the random forest has an 23.1% R^2_{oos}, which is an average improvement of 13.4% compared to Elastic-Net.

Due to the random weights selection at the beginning of the neural network, one iteration can have a different outcome as the previous one as in the random forest. For this reason, the same test is performed on the neural network. The algorithm used is a feed-forward neural network with the logistic activation function. Following the literature, the geometric pyramid rule is explored ; no more than two hidden layers are exploited, leading, in this case, at the architecture of 8-4-2-1 where $\sqrt{8 * 2} = 4$. In terms of R^2_{oos}, the neural network is also outperforming the linear approaches. On average the NN has a 27.6% R^2_{oos} which clearly raises the bar compared to the 23.1% of the random forest.

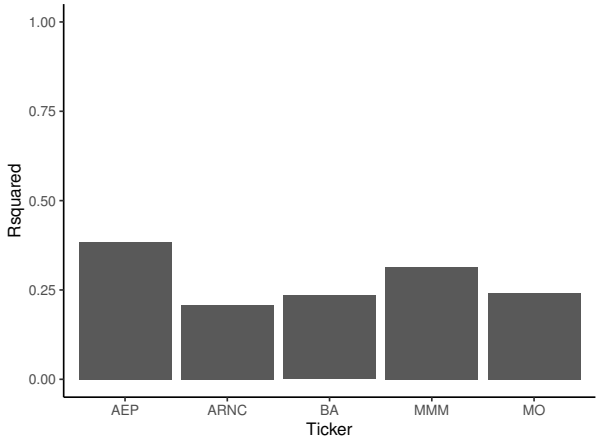


Figure 5.4: Neural Network R^2_{oos}

5.1.3 First Test Conclusion

An out of sample R^2 is computed for the different approaches studied in the theoretical section. As illustrated in the summary table 5.2, the non-linear approaches outperformed the linear approaches.

	MMM	**MO**	**AEP**	**ARNC**	**BA**
OLS	15.4	5.4	2.8	5.9	17.9
Elastic - Net	15.7	6.9	4.1	3.6	18.2
Random Forest	22.4	23.8	24.9	**25.8**	18.9
NN(8-4-2-1)	**31.5**	**24.1**	**38.4**	20.8	**23.5**

Table 5.2: R^2_{oos} in percentage

Recalling some strengths and weaknesses of the models, it can be visualize that

performing an R^2_{oos} is a good indicator of their performances. As predicted, OLS the worst performer because the model is not able to capture non-linear relationships and very sensitive to outliers. A more advanced linear regression is showing improvements. Indeed, due to the combination of both L_1 and L_2 regularization, the Elastic-Net is outperforming OLS as expected. However, not enough flexibility is penalizing the model to be a good competitor compared to the machine learning approaches. The random forest and neural network R^2_{oos} gives the best results overall. Their ability to catch non-linear patterns as well as their flexibility renders them as the out-performers. In predicting stocks, the flexibility and the amount of data that can be feed by the machine learning approaches gives them a clear advantage. Even though, machine learning outperforms linear techniques, overfitting seems to be present in the results. Indeed, having a high R^2_{oos} does not necessarily mean that the model is the best and can be an indication of overfitting.

In conclusion, compared to linear models which generally have a low R^2_{oos}, the machine learning approaches are clearly outperforming. However, even if an out-performance is observed, selection of parameters as well as the randomness of the weights for the neural network, for example, are raising the doubt concerning its interpretability. Furthermore, an high R^2_{oos} could also be an indicator of overfitting. It is therefore crucial to interpret the above results with caution. Furthermore, even if the test above highlight the power of the non-linear methods, it is wise to compare two opposite models in a more realistic approach.

5.2 Portfolio Construction

The last test section consists in building a portfolio by rolling window.

The sample is composed of stocks from 1970 to the end of 2017. The rolling window is working as follow: the first window lasts from January 1970 till December 2017. The artificial neural networks and VAR are fitted once a month and provide a one-month ahead prediction. Then the window is shifted by one month (i.e. February 1970 - January 2018) including the actual stock price. In total, the window is rolling for one year. The goal of this portfolio is to assess the models as if an investor invested in the market to maximize the return.

In this first step, it is crucial to select a good number of lags as input. For the VAR, up to 10 lags are tested on the model and the best lag is selected based on the lowest Bayesian Information Criterion (BIC). The lowest BIC suggests only one lag on the VAR. Concerning the neural network, half of the initial above stated inputs are taken as lagged input. The architecture of the neuron is thus 4-2-1 with only one hidden layer containing two nodes. Visualization on how the rolling window is constructed can be found in illustration 5.5.

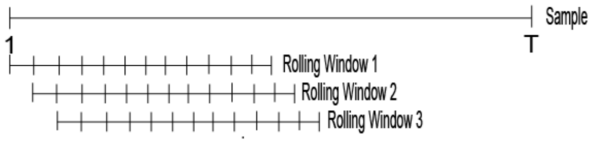


Figure 5.5: Rolling Window

The forecast for the prices is computed using the methods stated above for one year ($h + 12$). After the prices are predicted using the rolling approach, the prices are converted into returns. In table 5.3 and 5.4, the $h + 12$ prediction returns are summarized. In the last table 5.5, the real data returns of 2018 are exposed.

	MMM	**MO**	**AEP**	**ARNC**	**BA**
h+1	-0.00662	-0.00210	0.00803	0.00220	0.02010
h+2	0.14032	-0.01262	-0.06256	0.09703	0.35148
h+3	-0.10468	-0.10758	-0.05796	-0.16088	-0.01544
h+4	-0.06732	-0.01385	0.04458	-0.05568	-0.15423
h+5	-0.16749	-0.11321	0.01096	-0.26074	0.08243
h+6	0.06258	0.00892	-0.01315	0.01652	-0.02466
h+7	0.00787	0.01534	0.01597	-0.07903	-0.04053
h+8	0.08599	0.04444	0.01456	0.32501	0.03175
h+9	-0.00950	-0.00085	0.01108	0.02021	-0.01664
h+10	-0.02084	0.03287	-0.00660	0.02071	0.03172
h+11	-0.09614	0.07537	0.03113	-0.09616	-0.00724
h+12	0.09218	-0.16104	0.04213	0.05124	-0.01818

Table 5.3: Neural Network 12 months returns prediction

	MMM	**MO**	**AEP**	**ARNC**	**BA**
h+1	-0.26762	0.01302	0.03833	-0.00183	0.01858
h+2	0.50046	-0.01962	-0.07409	0.11029	0.23329
h+3	-0.07929	-0.13437	-0.06206	-0.23576	0.01784
h+4	-0.08243	0.01352	0.04718	-0.04246	-0.11321
h+5	-0.13569	-0.13532	0.01799	-0.26470	0.00625
h+6	0.01805	-0.00724	-0.03322	-0.00800	0.06282
h+7	0.00488	0.03482	0.01989	-0.01612	-0.05572
h+8	0.09155	0.04215	0.03025	0.31525	0.06873
h+9	-0.00531	0.00190	0.00668	0.04122	-0.04275
h+10	-0.00357	0.02822	-0.00719	-0.01151	0.09083
h+11	-0.10610	0.08645	0.03050	-0.04983	-0.05251
h+12	0.10773	-0.15960	0.05892	0.01176	-0.02327

Table 5.4: VAR 12 months returns prediction

	MMM	MO	AEP	ARNC	BA
h+1	0,0643	-0,0150	-0,0651	0,1031	0,2016
h+2	-0,0598	-0,1051	-0,0465	-0,1886	0,0221
h+3	-0,0679	-0,0100	0,0459	-0,0554	-0,0948
h+4	-0,1145	-0,0996	0,0203	-0,2270	0,0173
h+5	0,0146	-0,0066	-0,0290	-0,0090	0,0558
h+6	-0,0026	0,0188	0,0191	-0,0363	-0,0473
h+7	0,0793	0,0333	0,0273	0,2751	0,0620
h+8	-0,0066	-0,0027	0,0083	0,0318	-0,0379
h+9	-0,0010	0,0306	-0,0119	-0,0165	0,0849
h+10	-0,0971	0,0784	0,0350	-0,0763	-0,0458
h+11	0,0928	-0,1570	0,0597	0,0566	-0,0228
h+12	-0,0836	-0,0992	-0,0386	-0,2151	-0,0700

Table 5.5: 12 months real returns

For illustration purposes, figures 5.6, and 5.7 provides a visual interpretation of the tables.

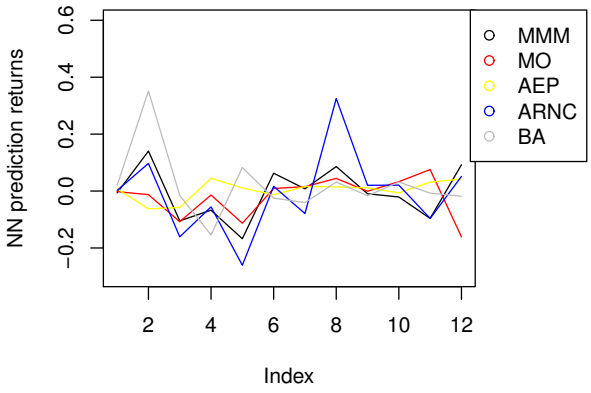


Figure 5.6: Predictive Returns on Neural Networks

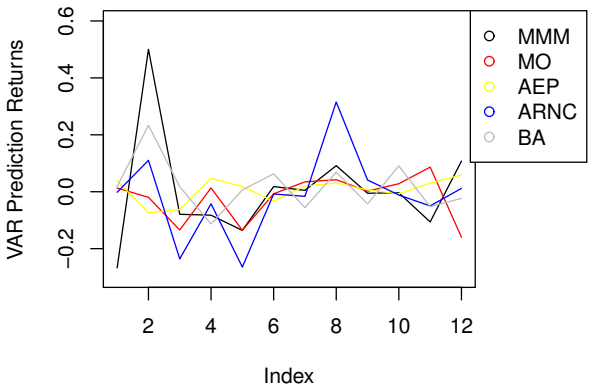


Figure 5.7: Predictive Returns on VAR

The goal of these forecasts is to see if an investor is making profit or not. Thus forecasts of each month and for each approach are sorted from the worst to the best. Once the sorting is computed, the best-expected returns is long and the worst is short. The investor is assumed to hold the positions for the whole month and repeat the strategy the next month. After one year using this zero net investment portfolio, the objective is to check if any approaches are profitable.

In the tables 5.7, 5.6 and 5.8, the highest and lowest return for each month are highlighted in red and yellow respectively. It is essential to remind that the investor is not aware of the next month prices. Each month strategy is thus based on the previous month long-short strategy. Furthermore, in real-world trading, 100%long - 100%short strategy does not exist. Due to regulatory issues, in most of hedge funds and quantitative trading firms, 130-30 strategies are applied. However, these strategies result in a high risk level on the market. The strategy selected in this paper is the 130-30. Each month 130% of long position and 30% of short position are exploited. In the portfolio construction strategy, the transaction costs are excluded. In practice, an average of 1% fees is assumed to be deducted monthly from the strategy's return.

	MMM	MO	AEP	ARNC	BA	L-S	Up to Date Gains
h+1	-0.26762	0.01302	0.03833	-0.00183	0.01858		1000.00€
h+2	0.50046	-0.01962	-0.07409	0.11029	0.23329	-0.2465	753.54 €
h+3	-0.07929	-0.13437	-0.06206	-0.23576	0.01784	-0.0845	689.89 €
h+4	-0.08243	0.01352	0.04718	-0.04246	-0.11321	-0.1344	597.14 €
h+5	-0.13569	-0.13532	0.01799	-0.26470	0.00625	0.0215	609.99 €
h+6	0.01805	-0.00724	-0.03322	-0.00800	0.06282	-0.0408	585.11 €
h+7	0.00488	0.03482	0.01989	-0.01612	-0.05572	-0.0784	539.22 €
h+8	0.09155	0.04215	0.03025	0.31525	0.06873	-0.1765	444.07 €
h+9	-0.00531	0.00190	0.00668	0.04122	-0.04275	0.0516	466.98 €
h+10	-0.00357	0.02822	-0.00719	-0.01151	0.09083	-0.0422	447.27 €
h+11	-0.10610	0.08645	0.03050	-0.04983	-0.05251	-0.0533	423.42 €
h+12	0.10773	-0.15960	0.05892	0.01176	-0.02327	-0.2398	321.88 €

Table 5.6: VAR Portfolio Strategy

To deepen the understanding of the portfolio construction, it is assumed that the investor invest 1000€at the end of the first month. After having located the highest and lowest return, the investor is doing 130% long on the highest and 30% short on the lowest. In table 5.6 at the end of h+1, the investor is investing long in AEP and short in MMM. The investor is then closing his positions at the end of h+2. looking at the data, the position at h+2 results in: $(130\% * -0.07409) - (30\% * 0.50046) = -24.09\%$ of the initial investment. The balance at h+2 is thus $1000.00 - 24.09\% = 753.54$€. The returns observed at the end of h+2 are then highlighted (i.e the investor is long on MMM and short on AEP) and the process is repeated in the next period.

As illustrated in portfolio using VAR modeling, most of the returns are negative with a peak of 3.9% and a drop at -24.09%. On average using the Long-Short strategy and a VAR(1) for prediction, the investor would lose on average -9.3% per month. This result exclude the transaction costs that would put down the returns even more. In practice, the investor would have lost almost all his money if he had followed this strategy. Indeed, with an initial investment of 1000.00€, there exist a loss of 678.12€in total after one year.

	MMM	MO	AEP	ARNC	BA	L-S	Up to Date Gains
h+1	-0.00662	-0.00210	0.00803	0.00220	0.02010		1000.00 €
h+2	0.14032	-0.01262	-0.06256	0.09703	0.35148	0.4148	1 414.83 €
h+3	-0.10468	-0.10758	-0.05796	-0.16088	-0.01544	-0.0027	1 411.02 €
h+4	-0.06732	-0.01385	0.04458	-0.05568	-0.15423	-0.1838	1 151.68 €
h+5	-0.16749	-0.11321	0.01096	-0.26074	0.08243	-0.0105	1 139.61 €
h+6	0.06258	0.00892	-0.01315	0.01652	-0.02466	-0.0370	1 097.42 €
h+7	0.00787	0.01534	0.01597	-0.07903	-0.04053	0.0224	1 121.99 €
h+8	0.08599	0.04444	0.01456	0.32501	0.03175	-0.0786	1 033.84 €
h+9	-0.00950	-0.00085	0.01108	0.02021	-0.01664	0.0229	1 057.56 €
h+10	-0.02084	0.03287	-0.00660	0.02071	0.03172	0.0174	1 075.97 €
h+11	-0.09614	0.07537	0.03113	-0.09616	-0.00724	0.1268	1 212.44 €
h+12	0.09218	-0.16104	0.04213	0.05124	-0.01818	-0.2247	939.97 €

Table 5.7: Neural Network 12 months returns prediction

Compared to the VAR approach, the neural network results in an average of 0.6% return per month. At the end of the year the investor would have lost only 60.03€of his investment. If the fees were included then, in a real situation, the investor would have lost even more. Thanks to the first period, where the return is about 41.48%, the neural network returns remain almost positive.

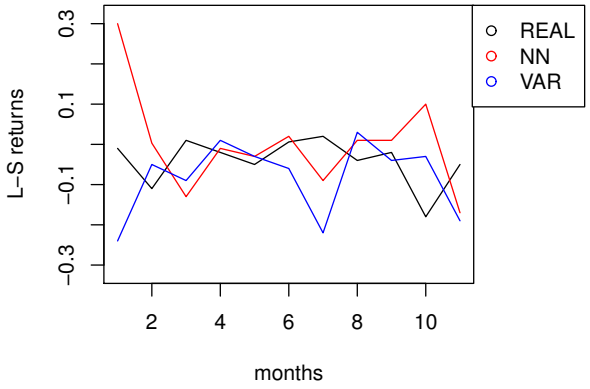


Figure 5.8: LS Strategy on Approaches

	MMM	MO	AEP	ARNC	BA	L-S	Up to Date Gains
h+1	0,0643	-0,0150	-0,0651	0,1031	0,2016	1 000.00 €	
h+2	-0,0598	-0,1051	-0,0465	-0,1886	0,0221	-0.0022	997.83 €
h+3	-0,0679	-0,0100	0,0459	-0,0554	-0,0948	-0.1370	861.14 €
h+4	-0,1145	-0,0996	0,0203	-0,2270	0,0173	0.0211	879.35 €
h+5	0,0146	-0,0066	-0,0290	-0,0090	0,0558	-0.0350	848.56 €
h+6	-0,0026	0,0188	0,0191	-0,0363	-0,0473	-0.0672	791.53 €
h+7	0,0793	0,0333	0,0273	0,2751	0,0620	0.0169	804.90 €
h+8	-0,0066	-0,0027	0,0083	0,0318	-0,0379	0.0389	836.19 €
h+9	-0,0010	0,0306	-0,0119	-0,0165	0,0849	-0.0470	796.91 €
h+10	-0,0971	0,0784	0,0350	-0,0763	-0,0458	-0.0367	767.69 €
h+11	0,0928	-0,1570	0,0597	0,0566	-0,0228	-0.2319	589.65 €
h+12	-0,0836	-0,0992	-0,0386	-0,2151	-0,0700	-0.0789	543.13 €

Table 5.8: L-S on 12 months real returns

To see if the predictions made by the opposite approaches are close-to-reality, a comparison with the real 2018 returns is proposed. The sam strategy is applied to the real returns, as summarized in figure 5.8. The long-short strategy is performing bad overall. Indeed, at the end of the year the balance of the investor would have been of 543.13€. Of course, the strategy is only applied to five stocks which is restrictive compared to the number of stocks available on the market for the investor. However, it is shown that the proposed strategy is not efficient and lead to consequent losses. Compared the the real data, the prediction of both approaches are not close-to-reality. While the VAR is predicting an after one year balance of 321.88€, the neural network is underestimating the reality by predicting a balance of 939.97€. For illustration purpose, the gains and losses can be visualized in the figure 5.9. The figure is showing that both approach is showing a decrease in the investment. However even if none of the approaches seems to be suited for returns prediction, the VAR is the model that is the most constant showing the same pattern as the real returns.

Finally, a benchmark portfolio is built to investigate if being passive (i.e. do not apply strategies) would lead to better results than L-S. An equally weighted portfolio is built and held as it is for a complete year. At the end of the year, all the positions are sold. On average a -15.5% loss per stock is observed (i.e -19% MMM, -31% MO, 2% AEP, -39% ARNC, 9% BA). At the end of the year, if a equally weighted portfolio was built using a benchmark strategy with 1000.00€(200.00€on each stock), the investor would have loss 155.26€. After one year the balance would be 844.74 €. This outcome shows that being passive on these stocks is more profitable than using a long-short strategy.

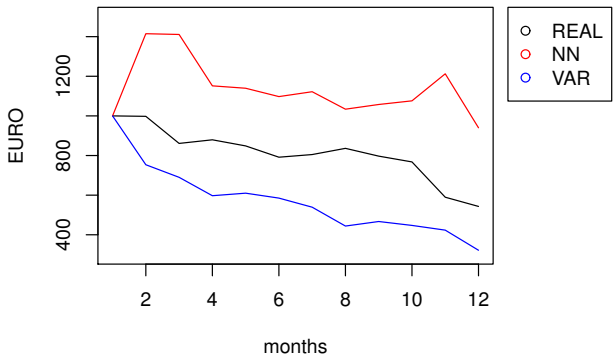


Figure 5.9: Gains/losses with L-S

5.2.1 Second Test Conclusion

These results lead to two main conclusion. The first one is that the strategy of buying the highest expected stock return and shorting the lowest is risky and does not lead to optimal results. Indeed, being passive using a benchmark portfolio is leading to a better investment. Furthermore, in a real-case approach, with a 130L-30S strategy plus transaction fees, investors should not invest in this portfolio. The second point is that, compared to the average loss of 5.1% per month with the strategy, the VAR overestimates the losses. On the opposite, the neural network is underestimating the losses with an average of 0.6%. In conclusion, none of the models proposed leads to close-to-reality results.

To conclude, none of the above models are suited for stock returns prediction. Indeed, none of the models is able to correctly capture the signals out of the noises. While the neural network will be more flexible and capture non-linear relationship, it will overfit the data by too much training and have poor results in forecasting. On the other hand, VAR is a linear approach that have an easy interpretation but is unable to capture the complex pattern of the stock time-series. The stock market remains unpredictable. In a close-to-reality situation, neither the neural network and VAR are significantly better. Because the neural network is certainly overfitting and its parameters selection too random, it is safer to use a model that is fully interpret able in the stock prediction environment. It is worth mentioning that the current sample is only including five stocks. In practice the investor has the possibility to invest in thousands of stocks. Furthermore, the results obtained can not lead to foregone conclusion on the subject. A different size sample, data granularity and other stocks might have lead the models to better outcomes.

Nowadays, it is still impossible to beat the market and know the price of a stock in advance. Because stocks data are too volatile and sensitive to any environmental chaos, politics and many other factors, it is not possible to have any control on the outcome.

Furthermore investors are influenced by rumors and other financial news which make the stock price unforeseeable. In the financial markets the human factor still plays a big role which can not be control by the historical data. Indeed, approaches can modeled the past but are unable to predict the future. If a major regulation is taken by the government for example, prices may drop or raise but the model is unable to capture that for now. This is also the reason why quantitative funds are practicing high frequency trading. Noticing that predicting the future with long term data (monthly) is more rough than focusing on short term data (nano seconds). By trading millions of stocks every second, they ensure to lower their risks and to have more control on the price that will vary of only a couple of cents.

In practice, neural networks are powerful in a field of image recognition and AI games for example, where the neuron can decompose a certain image and extract information that is needed to recognize a pattern. On the other hand, the VAR are useful in less complex prediction such as macroeconomic research (unemployment rates, GDP, etc.). In financial markets and asset field, time-series are too complex and signals can not be extracted without interfering the model.

Chapter 6

Conclusion

The main objective of this study was to test whether machine learning algorithm could be used in practice for portfolio construction or not. It also aimed at examining if neural network were better in regression modeling compared to more interpretative techniques such as VAR and Elastic Net.

Five stocks from SP500 were analysed and their historical monthly data from 1970 until December 2018 are gathered. After data cleaning had been processed, the out-of-sample predictive R^2 was computed. Each model was composed of following inputs: risk free rate, SP500 price index, d/p and e/p. The outcome of this R^2_{oos}test evidenced that machine learning, with neural networks and random forest, surpasses elastic net and OLS by more than 10% on average. Afterwards, the out-performer model was selected and his performances were evaluated in a new context against a simple VAR. A shifting rolling window portfolio was constructed over a one year horizon. Finally, the twelve-point forecast using NN and VAR was analysed using a Long-Short strategy.

Even if neural network proved to be better in terms of R^2 and equivalent in portfolio construction compared to linear approaches, neural networks would toughly be used by hedge funds or banks. Indeed, three main reasons are driving us to this conclusion. The first argument is that the choice of parameters in the network are undefined. There are no preset rules of parameters (i.e nodes, layers, inputs, output, activation function), which makes the model data-depending. The second reason is the overfitting that occurs in machine learning approaches. Even if early stopping, batch learning and other techniques can be used to regularize the model, there is a high chance of overfitting. Finally, the neural network is a "black box", where the combination of all above make its behavior inexplicable.

Since the 2008 economic crisis, more regulations are controlling hedge funds and banks; They guide them into safer strategies and impose stricter rules on models. In the current environment, neural network is not viable because not enough knowledge is available. Furthermore, to use it the investor has to accept that he has not the entire control over the model. Also, if an audit had to be executed, neural network should be explained in detail

which is impossible at this day.

Out of the strategy proposed, none of them is optimal and none of the models provides close prediction to the real data. Nevertheless, linear approaches are still safer for private investors and banks. In general, stocks follow a random walk. Therefore it is nearly impossible to beat the market. The emergence of machine learning is rising hope for the asset field and trading but knowledge has to be deepened for real applications.

For deeper inquiries on the subject, risk measures should be included in the portfolio. As mentioned in the beginning of the thesis, the ultimate objective of every investors is to maximize the return while minimizing the risk. If this study is only focusing on the return, it could be of a great interest to include risk measures. Concerning the return analysis performed on stocks, only monthly data were considered. Though challenging, using high frequency minute/daily data would have been brain-feeding and could lead to less change in volatility compared to monthly data. Including more stocks from the SP500 in the portfolio could enhance the result precision. Finally, only historical data from the stocks were considered. It is well known that investors are influenced by several financial news and rumors that influence the stock price. By including text-mining in the neural network, the accuracy of the predictions could have been improved.

References

Alpaydin, E. (n.d.). *Introduction to machine learning* (3rd ed.). Cambridge, MA: MIT Press.

Bishop, C. M. (2007). *Pattern recognition and machine learning, 5th edition*. Springer. Retrieved from http://www.worldcat.org/oclc/71008143

Blockeel, H. (2010). Machine learning and inductive inference.

Breiman, L. (2001, October). Random forests. *Mach. Learn.*, *45*(1), 5–32. Retrieved from https://doi.org/10.1023/A:1010933404324 doi: 10.1023/A:1010933404324

Chinco, A. M., Clark-Joseph, A. D., & Ye, M. (2017, October). *Sparse Signals in the Cross-Section of Returns* (NBER Working Papers No. 23933). National Bureau of Economic Research, Inc. Retrieved from https://ideas.repec.org/p/nbr/nberwo/23933.html

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

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, *116*(1), 1-22. Retrieved from https://EconPapers.repec.org/RePEc:eee:jfinec:v:116:y:2015:i:1:p:1-22

Freitas, F., De Souza, A., J N Gomes, F., & R De Almeida, A. (2019, 05). Portfolio selection with predicted returns using neural networks.

Fung, G. P. C., Yu, J. X., & Lam, W. (20-23 March 2). Stock prediction: Integrating text mining approach using real-time news. *Computational Intelligence for Financial Engineering, 2003. Proceedings. 2003 IEEE International Conference on*, 395-402. doi: 10.1109/CIFER.2003.1196287

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, *4*(1), 1-58. doi: 10.1162/neco.1992.4.1.1

Gu, S., Kelly, B., & Xiu, D. (2018, December). *Empirical asset pricing via machine learning* (Working Paper No. 25398). National Bureau of Economic Research. Retrieved from http://www.nber.org/papers/w25398 doi: 10.3386/w25398

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York, NY, USA: Springer New York Inc.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in r*. Springer Publishing Company, Incorporated.

Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, *7*(1), 77-91. Retrieved from `https://EconPapers.repec.org/RePEc:bla:jfinan:v:7:y:1952:i:1:p:77-91`

Masters, T. (1993). Practical neural network recipies in c++. *Academic Press Inc.*, *18-04*.

Schumann, M., & Lohrbach, T. (1993, Jan). Comparing artificial neural networks with statistical methods within the field of stock market prediction. In *[1993] proceedings of the twenty-sixth hawaii international conference on system sciences* (Vol. iv, p. 597-606 vol.4). doi: 10.1109/HICSS.1993.284239

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, *58*, 267–288.

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, *21*(4), 1455-1508. Retrieved from `https://EconPapers.repec.org/RePEc:oup:rfinst:v:21:y:2008:i:4:p:1455-1508`

Yoo, P., Kim, M., & Jan, T. (2005, 12). Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In (Vol. 2, p. 835 - 841). doi: 10.1109/CIMCA.2005.1631572

Zhang, P. (2003, 01). Zhang, g.p.: Time series forecasting using a hybrid arima and neural network model. neurocomputing 50, 159-175. *Neurocomputing*, *50*, 159-175. doi: 10.1016/S0925-2312(01)00702-0

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, *67*, 301–320.