# Machine Learning and the Optimal Choice of Asset Pricing Model

Corresponding Author:
Daniel Broby, Department of Acc, Finance  Economics,
Ulster University

Other:
Aleksadner Bielinski, School of Computing,
Edinburgh Napier University

# Contents

# List of Figures

# List of Tables

**Abstract**

This chapter evaluates the traditional methods for price prediction and examines, what we believe, are the most promising machine learning techniques for that task. Asset price forecasting is one of the fundamental problems in the financial field. Traditional forecasting methods include Capital Asset Pricing Theory (CAPM) or Factor Models to estimate stocks' excess returns. More recently, an increasing number of researchers and financial practitioners began to explore the role of machine learning in asset pricing. We show how these methods have been already applied in practice and discuss their results. We also explore the potential use of neural networks in asset pricing as we believe that their capacity to process large amounts of data together with the ability to accurately capture non-linear relationships among the variables makes them a great tool for price prediction.

**Keywords**: machine learning, asset-pricing, neural networks, factor models

# Section 1

# Introduction

Asset pricing is a major area of interest within the field of quantitative finance. The forecasting of the asset price is one of the main fundamental challenges for quantitative finance practitioners and academics alike. With the rapid development of technology, the computing power increased, thus making more investment firms and managers point their attention to machine learning techniques. Data is central to the modern digital economy and with humans generating and capturing more and more of it each year, there was a need to apply modern computer science techniques to deal with such a large volume of this resource.

Machine learning is defined as a mechanism used to train machines to perform a specific task while handling the data in the most efficient way. Machine learning techniques are designed to handle highly dimensional, large volumes of data, which make them a great tool for estimating asset prices. While traditional asset-pricing models are largely linear, machine learning techniques allow to utilise of new data sources and incorporate non-linear interactions among variables in making the predictions. With a large body of documented stock-level factors (Green, Hand, and X. F. Zhang, 2013), (Harvey, Y. Liu, and H. Zhu, 2016), the question remains which ones to use and how to best capture the ongoing relationships between them and expected return. Furthermore, Harvey and Y. Liu (2021), argues that traditional statistical techniques used in evaluating the explanatory power of these factors are redundant given the multiplicity issues arising from such methods. Machine Learning techniques offer a wide range of approaches to deal with the evaluation of the predictive power of factors, which were proven to be more effective compared to traditional statistics methods.

In this chapter, we will discuss the issues with traditional factor models and identify the main constraints when designing an asset-pricing model. From explaining the main principles of machine learning methods described, to showing their practical application in the asset-pricing field, we will show the disruptive potential of machine learning tech-

niques in finance. Moreover, the discussion will also highlight the role of neural networks in asset-pricing as they have been one of the fastest-growing sub-fields of machine learning recently. Neural networks have been successfully applied in many fields of study and their ability to capture complicated non-linear relationships in a variable rich environment makes them a perfect tool for designing an asset-pricing factor model.

# Section 2

# Empirical Asset Pricing Models

In this section, we will focus on examining the most popular empirical asset pricing models in financial markets. These models are the Capital Asset Pricing Model (CAPM), the Arbitrage Pricing Theory (APT) model and Fama-French 3 & 5 factors model.

## 2.1 Overview and Rationale

Traditionally, investors were referring to income statements, balance sheets and other publicly available information on a company to perform their investment choices. With the increased access to high-quality fundamental data, investors and academics have begun employing statistical, behavioural and machine learning techniques to facilitate the asset pricing methods, which gave birth to systematic value investing, first mentioned by Graham and Dodd (1951). The fundamental property of empirical asset pricing models is that not all risks should affect the performance of an asset, therefore it is important to distinguish key factors influencing asset price (Pástor and Stambaugh, 2000). Investors all over the world continue to use such models to aid their investment decisions.

## 2.2 Capital Asset Pricing Model (CAPM)

Developed independently by Treynor (1961), Sharpe (1964), Lintner (1965) and Mossin (1966), Capital Asset Pricing Model (CAPM) is considered the first comprehensible asset pricing model (Perold, 2004). CAPM builds directly on Markowitz's Modern Portfolio Theory (MPT) in which achieving higher yields is possible only though taking on more risky investments (Markowitz, 1952), which is addressed by including market risk premium in the model's equation. According to MPT, the risk of an asset consists of systematic (market) and unsystematic risk (company-specific). Since non-systematic risk can be fully diversified away, though reducing correlation between returns of the

assets, CAPM assumes that the only relevant metric in determining the expected return on the asset is market risk, commonly referred to as beta. Therefore the CAPM can be described as follows:

$$R_i = R_f + \beta_i * (R_m - R_f) + e_{it} \tag{2.1}$$

where:

$R_i$ = the expected return on the investment
$R_f$ = the risk-free rate
$\beta_i$ = the market risk of the investment
$R_m$ = the expected return on the market
$e_{it}$ = the standard error of the linear regression

### 2.2.1 CAPM Limitations

Although CAPM is one of the most widely taught theory on MBA (Master of Business Administration), (Womack and Y. Zhang, 2005) and financial economics courses (Dempsey, 2013), the model has its limitations, mostly resulting from its unrealistic assumptions and difficulties in beta estimation. Roll (1977), argued that the CAPM model cannot be tested as creating a market portfolio would require collecting all of the information about the market from many different industries and sectors which in practice is impossible. Moreover, Banz (1981), found that the average yields are contingent on the size-capitalization of the companies, which is especially visible among small-cap stocks with higher average returns than large-cap ones, further highlighting the ineffectiveness of CAPM. However, surveys such as that conducted by Partington et al. (2013), have shown that empirical test performed on CAPM are not so much proving its validity but rather highlighting the important correlations between variables in respect to the cross-section of realized returns.

## 2.3 Multifactor Models

### 2.3.1 Arbitrage Pricing Theory (APT)

Developed by Ross (1977), Arbitrage Pricing Theory (APT), was created as an alternative to CAPM. It is a multifactor model that builds on the existence of a linear

relationship between an asset's expected return and multiple possible factors influencing systematic risk. In APT the return of an asset is influenced by a range of macroeconomic factors such as unemployment, GDP growth, inflation or interest rates. If the APT holds, there would be no arbitrage opportunities (i.e. creation of riskless profits by taking positions that are based on security "mispricing"). The more efficient market is, the quicker the arbitrage opportunities will disappear. The model, due to its simplicity and flexibility, is commonly used in asset management, cost of capital estimation, portfolio diversification as well as evaluation of collective investment schemes (e.g. ETFs, Mutual Funds, Hedge Funds) performance (Huberman, 2005). The mathematical representation of the model is:

$$r_i = \alpha_j + b_{i1}F_1 + b_{i2}F_2 + ... + b_{in}F_n + e_{it} \tag{2.2}$$

where:

$r_i$ = the total return of individual asset i
$F_s$ = the factors affecting the asset's return
$b_{ik}$ = the sensitivity of the i$_{th}$ asset to the factor k
$e_{it}$ = the standard error of the linear regression

**The limitations of APT**

The main weakness with this theory is the fact that it does not specify what factors should be chosen. However, the analyst can decide on what factors to choose by regressing historical portfolio returns against the chosen range of macroeconomic factors. By doing so they can identify the statistical significance of any of these factors, thus tailor the model to the specific asset or group of assets. Nevertheless, Dhrymes, Friend, and Gultekin (1984), found that with the increasing number of securities, the number of determining factors increases making it gruelling to distinguish between "priced" and "non-priced" risk factors.

### 2.3.2 Fama-French 3 & 5 Factor Models

Following Roll's and Banz's critique of CAPM, Eugene and K. French (1992), developed a new asset pricing model, which introduced two new variables in explaining the

expected asset returns. The *size factor* adapted from Banz and *book equity / market equity (BE/ME)* ratio building on Chan, Hamao, and Lakonishok (1991), who found that book-to-market (BE/ME) plays a huge role in explaining the cross-section of average returns. The formula for Fama-French 3 Factor model can be described as follows:

$$R_{it} - R_{Ft} = \alpha_i + \beta_i(R_{Mt} - R_{Ft}) + \beta_s SMB_t + \beta_h HML_t + e_{it} \tag{2.3}$$

where:

| | |
|---|---|
| $R_{it}$ | = the total return of individual asset i |
| $R_{Ft}$ | = the risk-free rate |
| $R_{Mt}$ | = the total market portfolio return |
| $R_{it} - R_{Ft}$ | = the expected excess return |
| $R_{it} - R_{Ft}$ | = the excess return on a market portfolio index |
| $SMB_t$ | = the difference between the return of a diversified portfolio of small and big stocks (size premium) |
| $HML_t$ | = the difference between the returns on a diversified portfolios of high and low BE/ME stocks (value premium) |
| $\beta_{i,s,h}$ | = factors' coefficients |
| $\alpha_i$ | = Fama-French three factor alpha |
| $e_{it}$ | = the standard error of the linear regression |

To evaluate the effectiveness of their model, Fama and French used a sample of monthly stock returns from July 1963 until December 1991. Using the data from the Center for Research in Security Prices (CRSP) they have created 25 separate equity portfolios based on the size (i.e. Small, 2, 3, 4 and BIG) and BE/ME (i.e. Low, 2, 3, 4, High) factors with Treasury Bill rate as the risk-free rate. The results have shown the negative correlation between size factor and average yields as well as a positive relationship between BE/ME indicator and average returns, with the latter being persistent in all 25 portfolios. Their research showed that this three-factor model was capable of explaining a significant portion of stock return variation, eventually becoming a basis for evaluation of other asset classes and different markets.

In 2015, having investigated the profitability and investments of the companies, Fama and French developed their model by adding two more factors:

$$R_{it} - R_{Ft} = \alpha_i + b_i(R_{Mt} - R_{Ft}) + \beta_s SMB_t + \beta_h HML_t + \beta_r RMW_t + \beta_c CMA_t + e_{it} \quad (2.4)$$

where:

$R_{it}$ = the total return of individual asset i
$R_{Ft}$ = the risk-free rate
$R_{Mt}$ = the total market portfolio return
$R_{it} - R_{Ft}$ = the expected excess return
$R_{Mt} - R_{Ft}$ = the excess return on a market portfolio index
$SMB_t$ = the difference between the returns on a diversified portfolio of small and big stocks (size premium)
$HML_t$ = the difference between the returns on a diversified portfolio of high and low BE/ME stocks (value premium)
$RMW_t$ = the difference in returns between the most and least profitable companies (profitability risk factor)
$CMA_t$ = the difference in returns between conservatively and aggressively investing firms (investment factor)
$\beta_{i,s,h,r,c}$ = factors' coefficients
$\alpha_i$ = Fama-French five factor alpha
$e_{it}$ = the standard error of the linear regression

Similar tests were performed to evaluate the effectiveness of the model, with additional 22 years of return data from the same source. Although the five-factor model was superior to the three-factor model when it comes to forecasting asset prices, the researchers have highlighted the redundancy of *value premium* factor as it was largely explained by *profitability risk* and *investment* factors (E. F. Fama and K. R. French, 2015). Additionally, their study showed that, within the sample used, small-cap stocks achieved similar performance to low profitable but highly investing firms.

### Fama-French 3 & 5 Models Limitations

Despite satisfactory results of Fama and French models in their research and the wide adaptation of them among both academics and investment professionals, according to Blitz et al. (2016), the models fail to account for low-volatility and momentum premiums and does not attempt to address robustness issues. Even though Fama and French claim that in long run, the low beta anomaly is addressed by their five-factor model (E. F. Fama and K. R. French, 2016), there is a lack of significant evidence confirming that

higher market beta exposure is rewarded with increased returns. Moreover, later studies by Dichev (1998) and Campbell, Hilscher, and Szilagyi (2008), have demonstrated the negative relationship between distress risk and return, confirming the existence of low-risk premium. Fama-French models do not attempt to account for the momentum premium and with momentum profits becoming increasingly more important in asset pricing, many researchers began adding momentum factors resulting in 4- (Carhart, 1997) and 6-factor variants (Roy and Shijin, 2018). In addition to the lack of robustness of two newly added factors, the economic rationale for their addition to the updated model is also unclear. While size and value factors in the three-factor model were justified from the risk-based perspective (E. F. Fama and K. R. French, 2021), in the five-factor model it is unclear whether the observed return premiums are associated with systematic risk or behavioural anomalies.

## 2.4 Discussion

Although multifactor models are commonly used by investment managers, there is a discussion regarding their performance in respect to machine learning methods. Despite multifactor models being able to explain the historical correlation matrix relatively well (Chan, Karceski, and Lakonishok, 1999), they deliver poor predictions (Simin, 2008). Understanding the behaviour of risk premium is crucial in asset pricing. Traditionally, differences in expected returns were estimated using cross-sectional regressions, which in addition to the Ordinary Least Squares (OLS) method involved sorting assets into individual portfolios based on their underlying characteristics (Lewellen, 2014). Whereas, time-series forecasts of returns were obtained using time-series regressions of entire portfolio returns, with few macroeconomic predictors tested (Rapach and Zhou, 2013). While such methods are relatively simple and easy to implement, they pose substantial limitations in contrast to modern machine learning solutions. Evidence suggests that the main weakness of such methods is their inability to handle a large number of predictors (Gu, Kelly, and Xiu, 2018), which considering the large body of currently documented possible predictor variables, is not desirable.

# Section 3

# Machine Learning in Asset Pricing

In this section, we will explore the potential use of machine learning techniques in asset price forecasting. We will show the rationale behind the application of machine learning techniques in asset price forecasting. By synthesizing the machine learning methods with modern empirical asset pricing research, we will show why this particular financial field has the potential for a successful machine learning application.

## 3.1 Machine Learning- Overview

Although the definition of machine learning can vary from one scientific field to another, according to Dey (2016), machine learning is generally used to train machines to perform a specific task while handling the data in the most efficient way. Regardless of the definition used, the fundamental property of machine learning is its high-dimensional nature, which is the main reason why its suitable for asset pricing. Machine learning techniques provide more flexibility compared to traditional econometric methods, thus allowing to better capture the complexity of the asset pricing problem. However, the increased flexibility offered by machine learning comes at the cost of a higher probability of overfitting (Mullainathan and Spiess, 2017). Therefore, it is important to perform adequate refinements while applying machine learning that would reduce the chance of overfitting (Cawley and Talbot, 2010).

The machine learning algorithms work by extracting the patterns from historical data, in the process known as "training" and therefore applying these findings to accurately predict new data. After the process is completed, the created predictions need to be tested, allowing for theirs performance evaluation.

## 3.2 The Case for Machine Learning in Asset Pricing

As shown in the first chapter the prevailing question in forecasting a future price of an asset was to predict the risk premium. The tests performed by Fama-French on their 5 Factor Model showed the $R^2$ ranging from 0.91 to 0.93 (E. F. Fama and K. R. French, 2015). However, even when the model can almost perfectly observe the expected results, the remaining issue is how well it explains its behaviour, which requires additional testing. Additionally, with market efficiency making the risk premium estimation limited to news headlines response, there is a need to update traditional asset pricing methods by exploring new predictor variables. Nevertheless, calculating the risk premium remains the conditional function of future expected excess return. Therefore, thanks to its predictive capabilities resulting from combining forces of statistics and computer science (Das and Behera, 2017), machine learning makes a perfect tool for this task. If applied correctly, it has the chance to revolutionize the asset pricing (Arnott, Harvey, and Markowitz, 2019).

Another issue with traditional approaches to factor models, as highlighted by Harvey, Y. Liu, and H. Zhu (2016), is the way the explanatory power of the factors is evaluated. Typically, the statistical significance of the factor explanatory power is reported as the t-statistics, with factors scoring t-statistics of at least 2.0 considered significant. Although when testing a single factor it is unlikely for the t-statistics of 2.0 or greater to occur by chance, with the number of factors and therefore the number of tests increasing, the probability of t-statistics achieving such levels by chance is significantly higher. The issue has been also investigated by Harvey and Y. Liu (2014) and Bailey et al. (2015), both rising concerns regarding the use of traditional significance criteria for newly discovered factors. Harvey and Y. Liu (2021), argue that given the test statistic multiplicity in a numerous factors environment, some of the factors found to be significant have only be deemed so because of luck rather than their actual predictive power (i.e. "Lucky Factors"). The authors of the paper suggest that to evaluate the significance of the factor it is important to perform the out-of-sample testing procedure which will prove whether the examined factor or group of factors are explaining the risk premium well enough to be included in the asset pricing model. Moreover, they propose a new method for evaluating the significance of tested factors based on the number of the variables that have been

tested.

Moreover, since the creation of CAPM, researchers and academics alike were testing various financial and economic predictors that show forecasting capabilities, with more than 300 stock level factors used to describe return on the asset (Green, Hand, and X. F. Zhang, 2013), (Harvey, Y. Liu, and H. Zhu, 2016). Although identifying the whole array of factors with high predictive power is relatively simple, traditional methods break down when the number of predictors is close to or higher, compared to the number of observations. Additionally, such models are also subject to failure resulting from high multicollinearity, which considering the similar nature of many possible predictors is inevitable. Thanks to the broad availability of dimension reduction tools (i.e. Principle Component Analysis, Random Forest, Factor Analysis), machine learning offers degrees of freedom optimization and condenses the variance among predictors (Fodor, 2002).

As mentioned before, in traditional asset pricing models the interactions among the predictors were linear, more precisely modelled using the OLS method, possibly due to its simplicity of application. However, the relationship among the independent variables can be also nonlinear. The lack of documented guidance regarding the functional form of the predictors poses a huge problem when designing an asset pricing model. Luckily, machine learning offers a broad range of unique techniques from generalised linear methods, thorough regression trees to neural networks, offering high diversity in creating the model (i.e. high number of functional forms). Together with highly controllable parameter penalization and strict criteria for model selection, the created models are less likely to overfit or be subject to false discovery. Moreover, some of the machine learning algorithms can help to eliminate unnecessary factors from the model and extracting the underlying relationships that are likely to be true in the future, thus reducing the estimation error.

To summarize there are four main challenges when it comes to designing a factor model:

- Predicting the Risk Premium
- Determining the Functional Form
- Variable Selection

- Managing the Estimation Error

### 3.2.1 Machine Learning and Modern Portfolio Theory

According to MPT, it is possible to construct an "efficient frontier" that would consist of optimal portfolios, offering the maximum possible expected return for a given level of risk. To determine optimal weights for the portfolio, MPT suggests using mean-variance analysis. However, the main issue with using this approach is the large numbers of estimates required even for a smaller portfolio (Hirschberger, Qi, and Steuer, 2010). The high number of estimates leads to high estimation error, effectively making it impossible to compute 100% accurate efficient frontier. While multi-factor models such as these described in section 2 address some of the MPT issues by including multiple sources of risk, thus in theory reducing the estimation error and improving the quality of the estimates. Nevertheless, as mentioned in the previous section, traditional approaches to factor models pose some limitations as well. We believe that by applying machine learning techniques, it is possible to design an effective factor model that, by solving the issues highlighted in 3.2, would more accurately estimate the risk premium. With the quality of the estimates improved, investors could make better investment decisions (i.e. superior portfolio allocation, better stocks picks etc.), hence moving the efficient frontier above the optimal level, see figure 3.1. The final result would be the portfolio with a higher Sharpe ratio (3.1) (i.e. move from red star to orange one on the figure 3.1).

$$SharpeRatio = \frac{E[R_i - R_f]}{\sigma_a} \tag{3.1}$$

where:

$R_i$ = the expected return on the asset i
$R_f$ = the risk free return
$\sigma_i$ = standard deviation of an asset i
$R_i - R_f$ = the risk premium

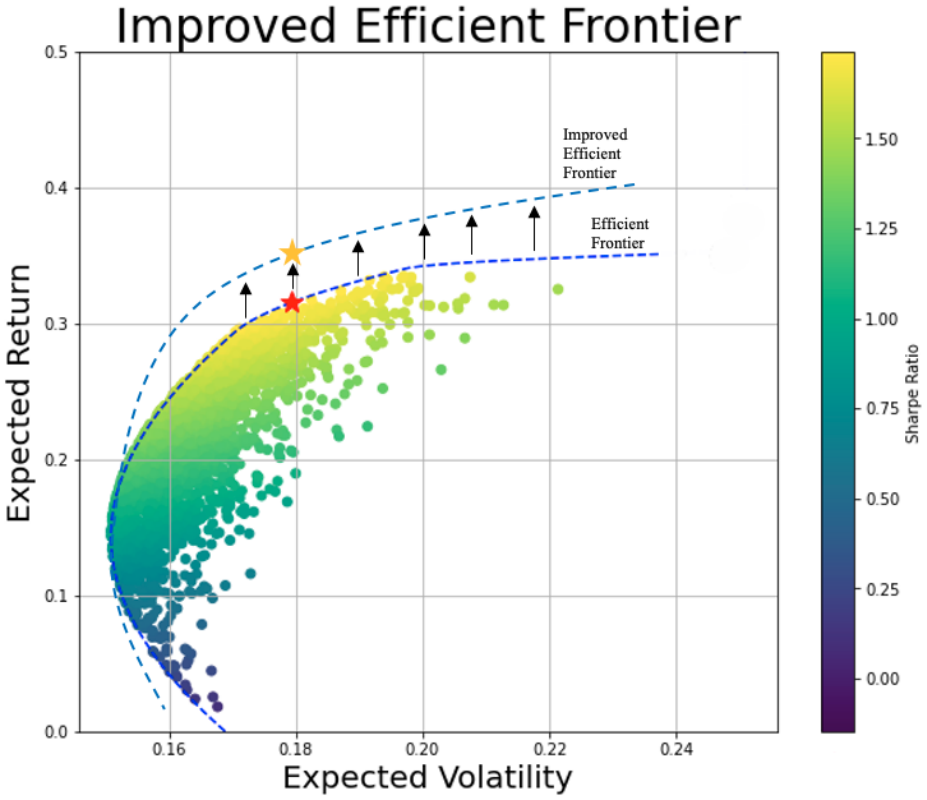


Figure 3.1: This figure shows the improved efficient frontier. We believe that incorporating machine learning techniques into the portfolio creation process would shift the efficient frontier up and improve the Sharpe ratio (move from red to orange star).

# Section 4

# Machine Learning Methods in Asset Pricing

There is a growing body of literature that recognizes the importance of machine learning in asset pricing (Ayodele, 2010). This results in a wide assortment of machine learning algorithms that can be applied to the asset pricing problem, which we will further explore in this section. Due to the nature of an asset pricing problem, most academics classify it as a supervised learning problem (Henrique, Sobreiro, and Kimura, 2019), (Krollner, Vanstone, and Finnie, 2010). In supervised learning parameters used for prediction need to be user-defined (i.e. labelled data), with both input and output data provided for training. Whereas unsupervised learning requires only input data with the purpose of finding the unknown patterns.

This chapter will describe the possible machine learning methods that are best suited to address issues from 3.2 in two fundamental areas. From providing a statistical background for each of the machine learning methods that can be used in risk premium estimation to discussing its possible application in the financial field through real-world applications.

## 4.1 Penalized Linear Regression

### 4.1.1 Statistical Overview

Although a simple linear model becomes inefficient when the number of predictors is close to or larger than the number of observations, by using penalization techniques the number of parameters can be limited. Considering that a large percentage of stock data consist of noise, an unmodified linear model with a large number of parameters will often overfit such noise rather than extracting valuable information. To avoid such a situa-

tion, machine learning proposes the introduction of a regularization parameter that will adequately penalize each factor by reducing the variance of the estimated regression parameters (P. Bruce, A. Bruce, and Gedeck, 2020). However, while this solution minimizes the error term, it also reduces the complexity of the model eventually adding the bias to the final estimation. The OLS regression finds the optimal value of the coefficients by minimizing the residual sum of squares (RSS) through finding the adequate coefficients:

$$RSS(\beta) = \sum_{i=1}^{N}(Y_i - \beta_0 - \beta_1 X_{i1} - \cdots - \beta_p X_{ip})^2 \tag{4.1}$$

where:

$RSS(\beta)$ = residual sum of squares for coeffcient $\beta$
$Y_i$ = the response variable
$X_{is}$ = the predictor variables
$\beta_s$ = coefficients

To prevent less contributive variables from impacting the forecast, penalizing methods reduces the coefficients' values towards zero, hence excluding such variables from the final model. There are several regularization methods that can be applied to the linear models. The general form of the penalization methods for linear models can be described as:

$$\overline{\beta} = arg\, min_\beta \left( \sum_{i=1}^{N}(y_i - (X\beta)_i)^2 + \lambda P(\beta) \right) \tag{4.2}$$

The solution for the vector of the regression coefficient, $\overline{\beta}$, is calculated through minimizing the RSS function based on the established penalty on the regression coefficients ($\lambda$P($\beta$)). The parameter $\lambda$, known as the shrinkage parameter sets the shrinkage on the regression coefficients. It is a non-negative number and the bigger it gets the more penalty is applied to the regression coefficients. Although, current academic research proposes a wealth of penalization methods, the most popular ones are LASSO (Tibshirani, 1996); Elastic net (Zou and Hastie, 2005) and Adaptive LASSO (Zou, 2006). The main difference between these methods lies down in the form that penalty takes, see table 4.1.

| **Regression Regularization Methods** | | |
|---|---|---|
| **Method** | **Penalty Equation** | **Description** |
| LASSO | $\sum_{j=1}^{p} \mid \beta_j \mid < \lambda$ | The penalty is placed on the L1 norm of the regression coefficient, indicating it reduces overfitting by estimating the median of the data. Each factor is adequately penalized. |
| Elastic net | $\sum_{j=1}^{p} \mid \beta_j \mid < \lambda_1 \, and \, \sum_{j=1}^{p} \beta_j^2 < \lambda_2$ | It combines the L1 penalty with the L2. The L2 penalty tries to estimate the mean instead of the median of the data to avoid overfitting. The method uses two shrinkage parameters (i.e. $\lambda_1 \, and \, \lambda_2$), which allows for more flexibility compared to the LASSO method. |
| Adaptive LASSO | $\sum_{j=1}^{p} \left( \mid \beta_j \mid / \mid \widehat{\beta}_j \right) < \lambda$ | Similarly to the LASSO, the penalty is placed on the L1 norm of the regression coefficient, however, in the adaptive LASSO, the user can assign different penalty weights to different coefficients. |

Table 4.1: Popular Regularization Methods

### 4.1.2 Application in Asset Pricing

In their paper, Kelly and Pruitt (2015), propose a method called three-pass regression filter (3PRF) in asset price forecasting using many predictors. By applying regularization techniques to OLS regression, 3PRF separates the factors that highly influence the target variable, while discarding irrelevant ones. While, high percentage of methods dealing with problems involving many predictors relied on Principal Component Regression (PCR) (Stock and Watson, 2012), (J. Bai and S. Ng, 2006), (Boivin and S. Ng, 2006), Kelly and Pruitt (2015) argue that 3PRF method delivers better performance. The main difference between 3PRF and PCR is that the former condenses the cross-section size in accordance with covariance with the forecast target, while the latter does it using covariance within the predictors. As a result, the 3PRF process estimates only relevant factors, hence making it more efficient while dealing with a large number of predictors. The superiority of 3PRF, compared to other regularization methods is especially noticeable when dealing with small samples. Moreover, Kelly and Pruitt (ibid.), provide empirical results that prove superiority of 3PRF estimating market returns compared to PCR from Stock and Watson (2002), Lest Absolute Residuals (LAR), proposed by De Mol, Giannone, and Reichlin (2008) as well as Quasi–Maximum Likelihood Approach, developed by Doz, Giannone, and Reichlin (2012). It is important to mention that all of the tests, evaluating the effectiveness of their method, were performed out-of-sample, demonstrating the competitive forecasting performance compared to other method tested.

Regarding to the issues stated in 3.2, the 3PRF method effectively addresses four issues with the designing factor model listed in 3.2. Apart from being an effective tool for risk premium estimation, it allows to improve models of linear functional form, helps to isolate the most influential variables which translate to fewer estimations required, thus decreases the estimation error. Furthermore, as the model is evaluated using out-of-sample forecasting, the probability of tested factors being "lucky" is relatively small. However, Kelly and Pruitt (2015) have applied 3PRF only to linear models, therefore, its performance when dealing with non-linear models is yet to be tested.

## 4.2 Regression Trees

### 4.2.1 Statistical Overview

Although we previously showed how regularization techniques can improve the linear forecasting models, parameters penalization fails to deliver desirable results when the number of estimators is larger than the sample size. Considering a large number of documented possible predictors as highlighted in 3.2, the issue lies not only in choosing the most appropriate factors but also to account for interactions among them. In statistics, the interaction effect refers to a situation in which the effect of the predictor on the dependent variable is subject to changes in one or more other regressands. Traditional statistics methods offer variance analysis (i.e. one-way and two-way ANOVA) to capture interaction effects among independent variables. However, creating a model using ANOVA methods, considering a large body of available factors, would be infeasible as all combinations of parameters would have to be tested. The machine learning alternative to incorporate multi-way predictor interactions are regression trees. Given their intelligibility and simplicity, they become one of the most popular machine learning techniques in data mining[1] (Wu et al., 2008). The regression tree starts by identifying the groups of observations that show some similarities to each other. Therefore through a process of binary recursive partition, that splits the data into partitions (i.e. branches), the tree is built until the split that further reduces impurity cannot be found. In the Classification And Regression Tree (CART) algorithm, proposed by Breiman et al. (1984), when it comes to regression trees the impurity methods that can be used are:

- Mean Square Error (MSE)- where the split is based on the minimized value of the residual sum of squares between the observation and the mean in each node.
- Least Absolute Deviation (LAD)- in which the mean absolute deviation is minimized from the median within a node.

Figure 4.1 shows the basic regression tree example with two variables (i.e. size and investment factor) in accordance with the CART algorithm. The yellow rectangles are

[1]Data mining- the process of finding anomalies, patterns and correlations within large data sets to predict outcomes.

the factors used in the regression tree, with teal, blue and green rectangles representing the terminal nodes (i.e. leafs) of the tree. In this case, the sample of individual stocks is divided into three categories based on the value of only two characteristics. Each terminal node is defined as a simple average of all observations within the subset. Before each split the algorithm would minimize the impurity, using the equation that can be written as:

$$minimise : J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right} \tag{4.3}$$

where:

$k$ = the feature in a subset
$t_k$ = the threshold for a split
$G_s$ = the impurity of a subset s
$m_s$ = the number of instances in each subset

Note that $k$ and $t_k$ are chosen as to produce the purest subsets.

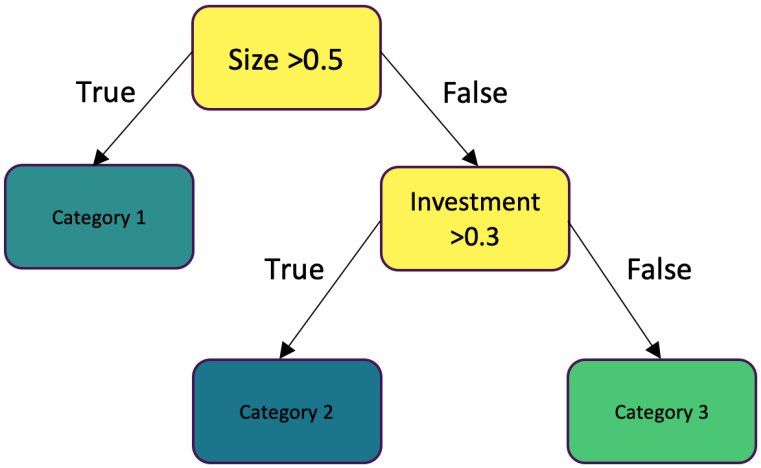


Figure 4.1: This figure shows the diagram of the regression tree using two variables. The category 1,2 and 3 rectangles are the terminal nodes (i.e. leafs). In this case, the algorithm divided the sample of stock data into three categories based on the values of size and investment factors.

Ideally, we would grow the tree until the best predictor variable and its value is found so that the forecast error is minimized. However, as Bramer (2007) points out, the decision trees are prone to overfitting, as excessive growing could lead to each of the leafs containing only one instance. For that reason, regression trees need to be highly regularized so that the final model will also be optimal, apart from being accurate. As Wolpert (1996), suggest without assumptions about the data, there is no reason to prefer

one model from another. Instead of relying on the results of one algorithm, it is desirable to aggregate the results of multiple models in the technique known as Ensemble Learning.

### Regression Tree Boosting

First proposed by Schapire (1990), as a way to improve the performance of weak learners for the classification problem, boosting techniques were extended to the concept of gradient boosted regression tree by Friedman (2001). The operating principle behind "boosting" is to combine forecast from multiple over-simplified trees. In theory that should lead to the creation of a "strong-learner" that would not only be superior to the single complex tree in terms of predictive power but also will be characterized by greater stability and less computational cost. The algorithm starts by fitting the shallow tree with only two branches. Therefore, the second shallow tree is used but this time to fit the residuals from the first regression tree. The procedure is repeated by adding other trees fitting the residuals from previous models, however, at each step the estimated values from the new tree are penalized by a tuning parameter, to prevent overfitting the residuals. The algorithm stops when the pre-specified number of trees is reached. The final model can be described as:

$$\widehat{g_B}(B, v, L, z) = \sum_{b=1}^{B} v\widehat{f}_b(\cdot) \tag{4.4}$$

where:

$\widehat{g_B}$ = the final ensemble predictor
$b$ = the step of the algorithm
$z$ = data used for the regression
$B$ = the total number of trees in a ensemble
$L$ = the depth of each tree
$v$ = the tuning parameter
$\widehat{f}_b$ = the single over-simplified regression tree function

### Random Forest Regression

Similarly to regression tree boosting, random forest is an ensemble method used to improve the accuracy of the model by aggregating the forecasts from many different trees. However, unlike the boosting, the random forest method builds on bootstrap aggregating,

commonly known as bagging. In bagging, the regression tree is trained on various subsets of data, which are sampled with replacement (i.e. multiple subsets can include the same instance). Random forests build on bagging, however, instead of searching for the best feature when splitting a node it tries to find it among the random subset of features (Breiman, 2001). By doing so it addresses the limitations of simple bagging by producing more diverse trees with weaker correlation among bootstrap samples. Moreover, the method trades higher bias for a lower variance (see figure 4.2), which addresses the biggest drawback of regression trees which is overfitting. Bias refers to errors resulting from simplifying assumptions made by the model to make the target function easier to approximate (i.e. underfitting). Variance error comes from too much sensitivity to the fluctuations in data (i.e. overfitting). The bias-variance trade-off refers to the inability to reduce both variance and bias at the same time, hence the most optimal solution is to minimize the total error, which is roughly at the intersection of bias and variance (Von Luxburg and Schölkopf, 2011).

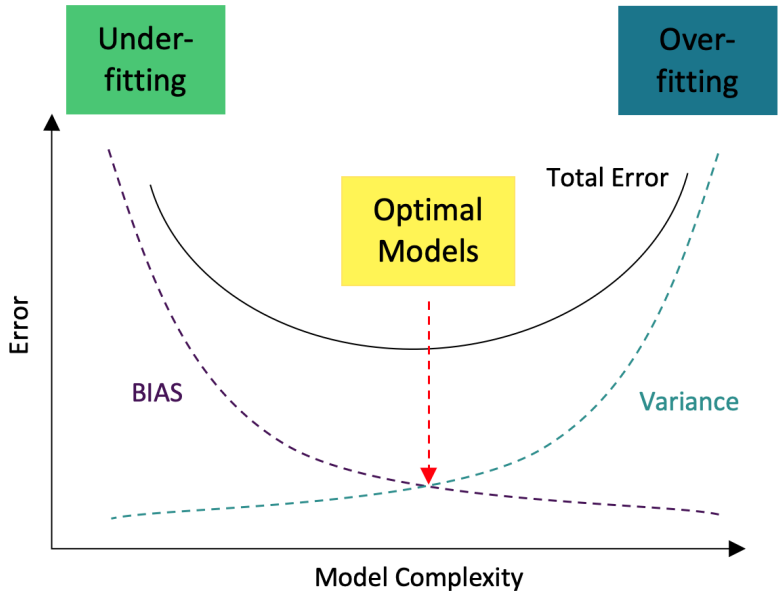


Figure 4.2: This figure is the graphical representation of the bias/variance trade-off. The optimal model will be the one that minimizes the bias error (i.e. underfitting) and at the same time minimizes the variance error (i.e. overfitting). At that point, further minimization of either bias or variance errors will cause one of them to increase significantly as they are negatively correlated.

The final output of random forest is the average output of all trees used for the ensemble, which can be described as:

$$\widehat{g_B}(L, B, z) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}_b(\cdot) \tag{4.5}$$

where:

$g_B$ = the final ensemble predictor
$b$ = the current number of bootstrap sample
$B$ = the total number of trees in a ensemble
$z$ = the data for the regression
$L$ = the depth of each tree
$\widehat{f}_b$ = the single regression tree

### 4.2.2 Application in Asset Pricing

E. Fama and MacBeth (1973), introduced the multivariate regressions to address the issue of cross-predictor interactions. Their model allows exploring the marginal effect on each predictor with other variables being controlled for. However, the issue arises when variables are considered jointly (i.e. multi-way interactions). For example, even in a 50 variable model, accounting for only two-way interactions would result in 1275 regression coefficients, which is significantly larger than sample sizes splits proposed by E. F. Fama and K. R. French (2008). Moreover, the results from their model can be extremely vulnerable to outliers (i.e. extreme returns) in the data. To address these issues Moritz and Zimmermann (2016), proposed the random forest regression approach to establish portfolio sorts and therefore combined all estimates from each tree into final prediction. As shown in 4.2.1 random forests allow producing de-correlated trees, which in that respect allow spotting many different, yet related predictors. Additionally, the problem of overfitting is addressed as each tree is trained only on the subset of data. Apart from classic accounting variables (i.e. book-to-market ratio) they also test their framework using return-based variables, arguing for the importance of momentum factors. By testing 126 return-based factors from many different time horizons, on various company sizes (i.e. large, small and micro firms according to E. F. Fama and K. R. French (2008)) and conclude that more recent past returns are more relevant than intermediate past

returns in return forecasting. Therefore, they combine the most influential return-based indicators into the one model using random forest method with a total of 200 independent trees (i.e. 200 different portfolio sorts) each using 8 out of 25 possible regressors (roughly 30%) as suggested by Breiman (2001). To test their method, they employ a simple strategy of going long on the stocks with the highest decile of predicted returns and shorting the lowest decile of predicted returns, using the CRSP data from 1963 to 2012. To test the out-of-sample performance of their model they employ a pseudo-out-of-sample procedure, which works by training the model using the data only available at a given time t and then computing all of the forecasts outside of the training set (i.e. $t + 1, t + 2, \cdots, t + n$). Additionally to confirm the importance of momentum factors they supplement their model with 86 different accounting factors such as book-to-market, leverage, gross profitability etc. with the momentum factors continuing to be the most influential. Their result shows that, when applied to the tested data, their strategy would deliver a positive annual return for the past 45 years from 1967 to 2012, see figure 4.3. Moreover, despite most of the momentum strategies delivering negative returns during the Global Financial Crisis (GFC), their strategy would not lose money in that period either. Their algorithm was able to detect the reversal in momentum factors soon enough, thus avoid the drawdown in 2009. Overall their strategy delivered a superior information ratio[2] compared to the standard Fama-Macbeth framework (2.9 vs 1.3 per month). However, its performance begins to deteriorate from early 2000, suggesting that the algorithm was not able to capture momentum movements soon enough to match its previous performance.

[2]Information ratio is the metric used to compare excess active return (i.e. above the benchmark) of the investment, considering the overall volatility of those returns in a given time period.

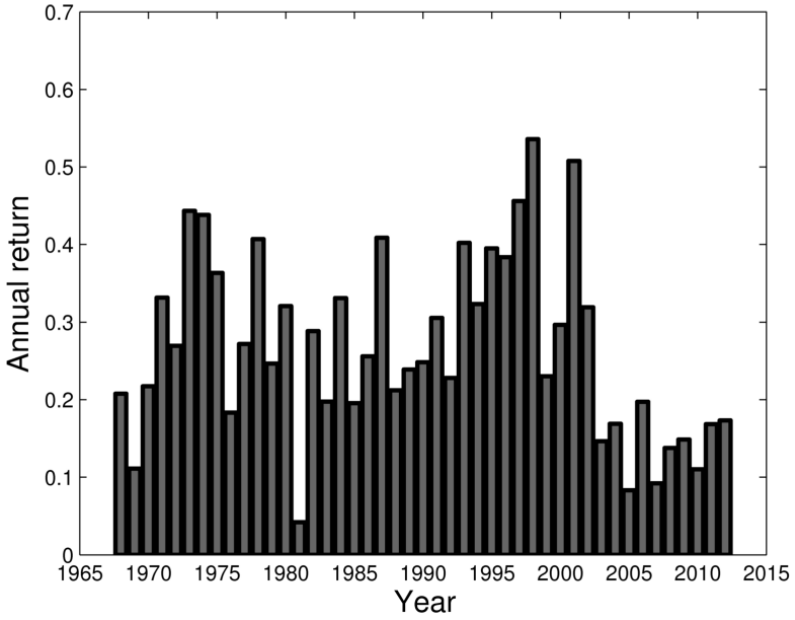


Figure 4.3: This figure shows the annual percentage return of researchers strategy. The algorithm used to create an investing strategy was random forests based on momentum factors. It delivered positive returns, even during the 1980's crisis. However, its performance deteriorated noticeably from the beginning of the 21st century and failed to match the previous performance in a given time period.

Source: Moritz and Zimmermann (2016)

In respect to the issues highlighted in 3.2, Moritz and Zimmermann (2016) model does generally a good job at predicting the risk premium and managing the estimation error. Furthermore, the effectiveness of their method is supported by the out-of-sample testing, limiting the probability of "lucky factors". However, as regression trees are non-linear models, the issue of functional form remains as the model cannot be simply summarized in a linear equation, thus complicating the interpretation of the results. Additionally considering the decrease in performance of the model from the early 2000s onward, the question remains whether the momentum factors proposed by them will continue to have high predictive power in the future. Nevertheless, with the framework constructed around tree-based conditional portfolio sorts, they hope for an increased level of scientific discovery regarding asset pricing in the years to come.

## 4.3 Support Vector Regression (SVR)

### 4.3.1 Statistical Overview

According to Efficient Market Hypothesis (EMH), it is impossible to consistently achieve above-market returns. However, the theory has been questioned since its introduction (Malkiel, 2003). Moreover, considering the computational advancements over the past decade, the use of machine learning techniques in asset pricing is constantly growing (Gerlein et al., 2016). Studies by Ballings et al. (2015) or Nayak, Mishra, and Rath (2015) show the successful application of Support Vector Machines (SVM) in asset pricing. However, as SVM are primarily used for classification problems, the methods above were focusing on determining only the direction of asset prices, rather than estimating their exact value. The goal of SVR algorithm is to find function $f(x)$, with at most $\epsilon$-*deviation* from the target $y$. The problem can be written as:

$$\begin{aligned} & min\ \frac{1}{2} \parallel w \parallel^2 \\ s.t.: \ & y_i - w_1 * x_i - b \leq \epsilon; \\ & w_1 * x_i + b - y_i \leq \epsilon \end{aligned} \tag{4.6}$$

where:

$y_i$ = the target variable
$w_s$ = the weighted coefficients
$b$ = the intercept
$x_i$ = the factor used for the regression
$\epsilon$ = the precision (tolerance) level

The graphical representation of linear SVR in a two-variable environment can be seen in figure 4.4. The SVR tries to fit as many instances (purple dots) within the decision boundaries (yellow lines), as possible, while limiting the number of margin violations. The green line is the final equation used to predict continuous output (i.e. hyperplane). Errors are ignored as long as they are within the earlier set decision boundaries. The decision margin can be either soft or hard. While the soft decision boundaries allow for some margin violations, hard margins strictly impose that all instances have to be within decision boundaries. The hard decision boundaries are infeasible where data consists of

a large number of outliers.

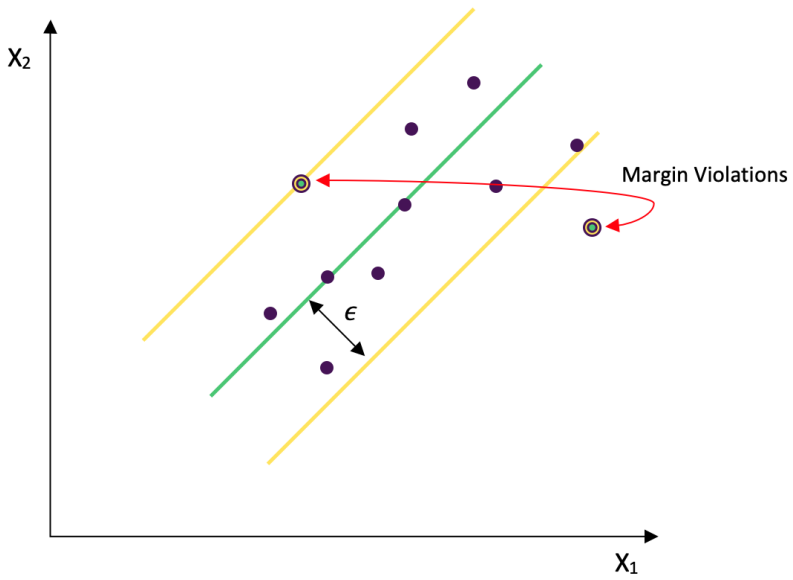


Figure 4.4: This figure is the graphical representation of how Support Vector Regression works. The purple dots are the data points and the algorithm tries to find the optimal hyperplane (green line) that will predict the final output while keeping the number of margin violations outside of the decision boundaries (yellow lines) to the minimum.

However, SVRs are not only limited to describe linear relationships. Using the so-called kernel trick allows for the data to be transformed into a higher-dimensional space without the need for data transformations. By doing so the explicit mapping needed for linear algorithms to capture non-linear interactions is avoided. Kernels allow finding a hyperplane in the higher dimensional space without a huge increase in computational cost. Table 4.2 shows three most commonly used kernels along with their mathematical functions and brief descriptions.

| **Types of Kernels for SVR** | | |
|---|---|---|
| **Kernel** | **Mathematical Function** | **Description** |
| Linear Kernel | $K(x, y) = (x^T y)$ | With $x$ and $y$ as the vectors computed from training the model |
| Polynomial Kernel | $K(x, y) = (x^T y + 1)^d$ | With $d$ being the degree of the polynomials. In addition to examining the features of the given sample, the polynomial kernel allows for exploring their combinations, known as interaction features. |
| Radial Basis Function (RBF) Kernel | $K(x, y) = exp(-\lambda \parallel x - y \parallel^2)$ | With $\lambda$ being a free parameter, that cannot be calculated precisely and must be estimated and $\parallel x - y \parallel^2$ being the squared euclidean distance between the two feature vectors. The RBF kernel works by transforming the data in accordance with the similarity between instances in the range from 0 (far away) to 1 (identical). |

Table 4.2: Popular SVR Kernels

### 4.3.2 Application in Asset Pricing

According to Awad and Khanna (2015), one of the main advantages of SVR is that its computational complexity does not depend on the dimensionality of the data. Thanks to the earlier described kernel trick, SVRs can easily deal with a large number of variables, maintaining high prediction accuracy. Henrique, Sobreiro, and Kimura (2018), used SVR to predict stock prices for firms of various sizes coming from different markets. Similarly

to the methods described in 4.2.2, the authors focus on momentum-based factors. Using the SVR's ability to capture high dimensional non-linear interactions among the variables they evaluate the significance of technical analysis (TA) indicators including Simple Moving Average (SMA), Weighted Moving Average (WMA), Relative Strength Index (RSI), the Accumulation/Distribution Oscillator (ADO) and the Average True Range (ATR). They employed their model on Brazilian, American and Chinese stocks with three blue-chip and three small-cap stocks for each country, resulting in 18 assets total. The model was tested in three different scenarios. The first one considered daily price changes over 15 year period (2002-2017). Second used up-to-the-minute price changes from 3 months (03/2017-05/2017). Lastly as 3 months period is unlikely to include all possible market conditions, they evaluated their model using 2-year up-to-the-minute prices, this time using solely Brazilian stocks. Although the second environment might seem too short to determine the effectiveness of any asset-pricing model, note that as the test considered one-minute price data, there are over 33000 observations, making it computationally expensive even for SVR. For the up-to-the-minute data sets, the prediction model starts 10 minutes after the beginning of the trading session. Firstly, they run multiple SVRs using normalized TA indicators to determine the optimal degree of polynomials for the polynomial kernel as well as the optimal $\lambda$ parameter for the radial kernel. Therefore, by applying the optimal parameters they run the test in each of the earlier defined environments using three different kernels (i.e. linear, polynomial and radial). To determine test and training sample they used k-fold cross-validation. This process differs from bootstrapping as resampling is done without replacement, hence surrogate datasets are smaller than the original. One of their key findings was that the linear kernel performed the best across all three tests with the smallest Mean Absolute Percentage Error (MAPE). They compared their method with the random walk theory of E. F. Fama and Malkiel, 1970. Although in a fixed training period their SVR model delivered worse predictions than random walk model, when periodically updated[3] up-to-the-minute models predictions, using the linear kernel, were more accurate than random walk model ones for the majority of the stocks. They claim that missing data was the main obstacle to their models

[3]Each of the tested models was updated in accordance to the periodicity of its data e.g. for up-to-the-minute observations the model was updated every minute, every time with next minute's closing price serving as a test observation.

achieving better results, especially in the one-minute data sets.

Considering the issues listed in 3.2, the models proposed by Henrique, Sobreiro, and Kimura (2018) do not address any of the four issues well enough. Although the authors showed their models have some predictive power, considering the small sample period it is impossible to validate that claim. Moreover, the variables proposed were strictly momentum-based and the fact that their model failed to deliver consistent outcomes across all of the stocks, remains a huge issue especially if the model was to be used by other investors. Finally, the fact that their model uses TA indicators in estimating asset price, poses the concern as TA methods lack substantial empirical evidence. Overall, while SVRs have the potential to be utilised in asset pricing, thanks to their numerous advantages, in this case, it is impossible to determine whether the models discussed are indeed good asset pricing tools.

## 4.4 Markov Switching Models (MSM)

### 4.4.1 Statistical Overview

Traditional factor models assume that the comovement among the variables is constant over time. In other words, regardless of the state of the economy or business cycle, factors within these models are assumed to have the same effect on the estimated asset price. However, in reality, markets fluctuate between regimes of growth characterised by low volatility as well as periods of economic downturn often accompanied by high volatility and negative returns. Therefore, it is crucial to not only be able to identify these regime changes but also be able to adequately adjust the asset-pricing models to exploit such events. Clarke and Silva (1998), showed that by adjusting the investment exposure in accordance to the present regime, investors can improve their efficient frontier. To model these state changes, the Markov model assumes that the future state depends solely on the current state (Gagniuc, 2017). Following that assumption, one can estimate the probability distributions of future shifts in non-stationary predictor variables and adequately update their coefficients with respect to the regime identified. In 1989, Hamilton introduced Markov switching model of regime change in which he rec-

ognized the presence of periods of faster and slow economic growth in the US economy. Therefore, using autoregressive process 1 (AR1), he modelled long-term economic trends incorporating the transition between the states. The transition between the states is governed using a first-order Markov chain in which the probability of subsequent state ($S_{t+1}$) is based only on the immediately preceding state ($S_t$), see 4.7. Note that there are higher-order Markov chains in which the probability of the transition depends on more than one preceding states.

$$P(S_{t+1}, \mid S_0, \cdots, S_t) = P(S_{t+1} \mid S_t) \tag{4.7}$$

The Markov chains own their popularity mainly thanks to their simplicity. Apart from providing a model that is easy to specify, they also allow for network extension. Each of the modelled future states can be used to perform the additional test in data considering various environments.

### 4.4.2 Application in Asset Pricing

Building on the model of Hamilton (1989), J. Chen and Kawaguchi (2018), applied Markov Regime Switches on multi-factor asset pricing models. The two states they recognized were bull and bear market and they used Hamilton's framework to model the transition between these states. The rationale for their model was the fact that as size factor (SMB) and value factor (HML) originate for the stocks in the market and return series come from the market itself, it is sensible that these factors may vary over time. Therefore, they extend the Fama-French three-factor model shown in 2.3.2 with Markov Switching, creating the MR-FF3 Model. They assume that the betas for the three factors in the model are regime-dependent and adequately estimate them for bull and bear markets using Markov chains. To test their model they use Chinese stock market data from 1995 to 2015. One of their key findings was that in the bear market the risk premiums of SMB and HML factors are higher than in the bull market. Such phenomenon is possible since, during bear periods, investors seek a higher return on size- and value-related risks (Cochrane, 2009). Additionally, they found that in a bear market, betas for SMB

and HML[4] factors increased. It not only highlights the ability of the size factor to capture the risk-return relationship but also the capacity of the value factor to explain the return dispersions between low and high BE/ME stocks in a bear market. Furthermore, their tests using time-series regression on stocks proved the presence of the regime-dependent risk exposure pattern in the data. Finally, to confirm their findings they performed one-step-ahead out-of-sample forecasting on all of the tested portfolios. The average root means squared error (RMSE) in out-of-sample tests was 0.0188, indicating the high predictive ability of the MR-FF3 model across the Chinese stock market.

This particular model proved not only to explain risk premiums well but also is clear on its functional form and the variables required. By building on the already existing Fama-French three-factor model, MR-FF3 creates a powerful and relatively simple to implement method for asset price forecasting. It allows capturing bull and bear cycles effects on asset price while providing the explanation between investigated relationships. Moreover, by keeping the number of factors low the estimation error is kept at an acceptable level. However, to ultimately confirm the predictive ability of the model further test are required. Possibly testing it on different markets and using bootstrapping procedure would allow to test its effectiveness and also highlight other regime-dependent relationships between the variables.

[4]Only for the high BE/ME firms

# Section 5

# Artificial Neural Networks (ANNs) in Asset Pricing

While Neural Networks are an indispensable part of machine learning, their extraordinary ability to deal with a large number of variables makes neural networks potentially very useful in asset pricing. There are many types of neural networks, however, in this section, we will discuss, what we believe, are the two most prominent types when it comes to asset price forecasting.

## 5.1 Artificial Neural Networks- Overview

The main idea behind Artificial Neural Networks is to teach computers to process data the way humans do. Traditionally potential predictors in a factor model were tested on basis of the hypotheses, which results determined whether to include given variables in the final model. In contrast, the ANNs' output is algorithmically engineered, meaning that thousands of different combinations of trainable parameters are tested to finally maximize the explanatory power of the network. Thanks to their ability to model non-linear processes, neural networks (NNs)[1] have been successfully applied in medical diagnosis, see Jiang, Trundle, and Ren (2010) and Sengupta, Sahidullah, and Saha (2016), automated trading, see Azzini and Tettamanzi (2008), speech Abdel-Hamid et al. (2014) and handwritten text Maitra, Bhattacharya, and Parui (2015), recognition as well as finance, see J. French (2017), to name a few. According to Hornik, Stinchcombe, and White (1989), NNs are one of the most powerful modelling techniques in machine learning.
There are four main components of every ANN:

- **Neurons**- the main component of all NNs. Neurons are divided into input and

[1]For the purpose of this chapter the terms Neural Networks (NNs) and Artificial Neural Networks (ANNs) will be used interchangeably.

output neurons. The former consist of either feature from the training set or outputs from the previous layer of neurons, while the latter is simply a successor of input neurons.

- **Connections and Weights**- Each NN consist of connections that connect input with output neurons. Every connection will have assigned weight based on the algorithm's learning.

- **Propagation Function**- It is used to compute the input of a neuron based on its predecessor neurons. It establishes the initial connections' weights by minimizing the observed errors considering sample observations.

- **Learning Rule**- It is used to adjust the connections' weights, by compensating for each error found during the learning process. It uses stochastic gradient descent or other optimization methods to compute gradient descent with respect to weights, thus improve the accuracy of the NN.

The ANN consist of input and output neurons, connected by weighted synapses. Figure 5.1 shows the simple ANN with only one output layer (i.e. one layer NN). Neural Networks are usually divided into shallow networks with 1 to 3 layers and deep networks with more than 3 layers. The learning process involves adjusting the weights of the synapses in accordance to minimized observed errors. Usually the more data available (i.e. bigger data sets), the better NN will learn, thus more accurate predictions. The learning process starts from forward propagation in which data is transformed from the input to the output layer, with each of the neurons adequately processing the input data. The next step is backpropagation in which the weights of the connections are adjusted so that the errors will be minimized.

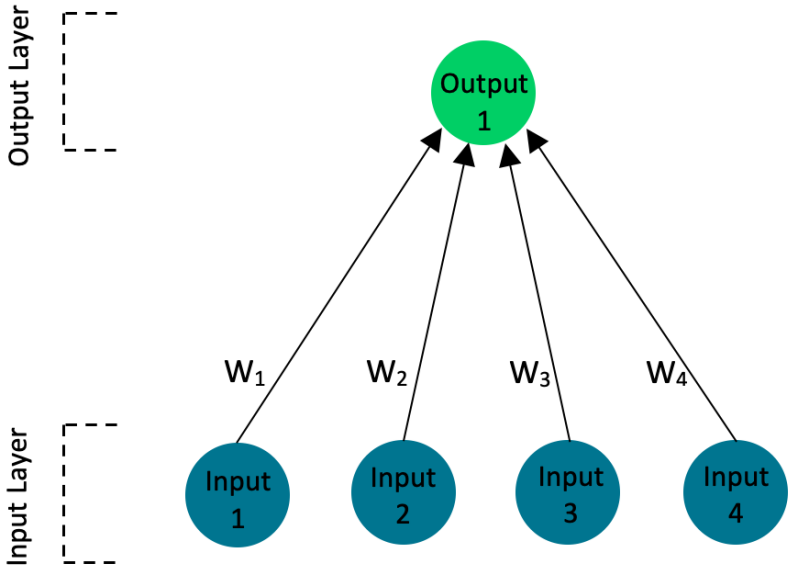


Figure 5.1: This figure is the graphical representation of a simple neural network in which the inputs are transformed into the outputs in accordance with weights, ($W_s$), associated with each neuron.

## 5.2 Feed-Forward Neural Networks (FFNs)

The FFNs are considered one of the simplest type of neural networks. In addition to the input layer of raw predictors, they also have one or more hidden layers that interact with each other and perform nonlinear transformations of the data. The output layer in FNNs is aggregating hidden layers into the final prediction, hence capturing more predictive associations within the data set. Figure 5.2 shows a simple FNN with only one hidden layer between inputs and output. The hidden layer consists of additional neurons that take the set of weighted inputs and returns the output through the activation function. For example, the second neuron from the hidden layer in figure 5.2, ($S_2$), is the weighted sum of all of its input neurons:

$$X_{S2} = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 \tag{5.1}$$

where:

$X_{S2}$ = the output variable for $S_2$
$w_{is}$ = the weight for each of the input neurons
$x_{is}$ = the inputs from input neurons

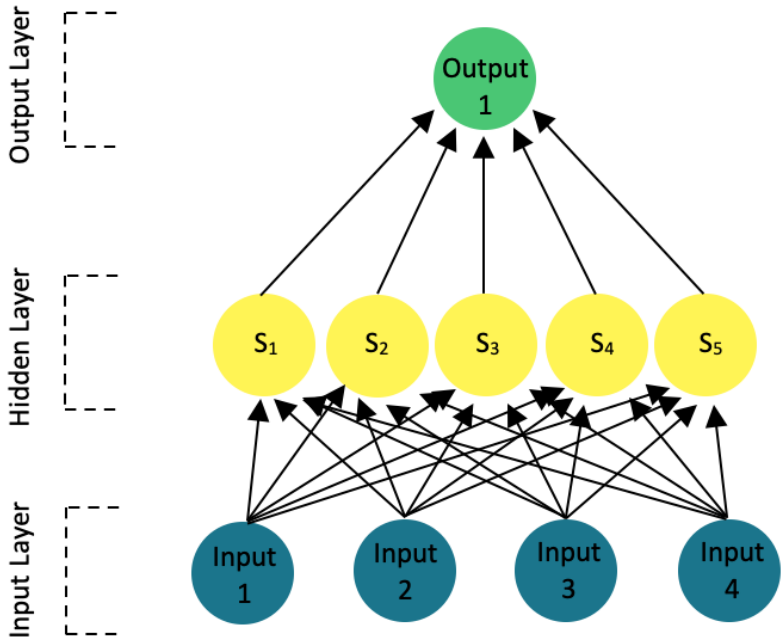


Figure 5.2: This figure shows the neural network with a single hidden layer. Each of the hidden layer's neurons has an activation function upon which the algorithm decides whether to activate the neuron. The final output is the weighted sum of all active neurons.

Therefore, the final output forecast consists of results from aggregated neurons, similarly to the formula 5.1, however this time using all active neurons. Using the example from the figure 5.2 there are a total of $31 = (4+1)*5+6$ parameters, with five parameters to reach each neuron and six different weights used to aggregate the neurons into the single forecast.

The person responsible for structuring the FNN has to decide on the number of hidden layers, the number of neurons within each layer and finally how the neurons are connected. The results from Eldan and Shamir (2016), show that deepening the FNNs (i.e. more hidden layers) is more valuable than increasing width (i.e. adding more neurons in the layer), with algorithms achieving similar accuracy using fewer parameters. It is important to mention that each of the hidden layer's neurons is subject to the activation function which determines whether it is used for the final prediction or not. If the activation function is not applied, the output would be a linear function, thus the more complex non-linear interactions would not be captured. Table 5.1 shows three main activation functions along with their mathematical functions and brief descriptions.

| **Penalized Regression Methods** | | |
|---|---|---|
| **Activation Function** | **Mathematical Function** | **Description** |
| Rectified Linear Activation Function (ReLU) | $max(0.0, x)$ | If the input value $x$ is negative the value 0.0 is returned (i.e. dead neuron) if not, the value x is returned. The main advantage of ReLU is that it does not activate all neurons at the same time being more computationally efficient. However, during the backpropagation process, the weights of such neurons will not be updated, resulting in never activated neurons. |
| Sigmoid Hidden Layer Activation Function | $\frac{1.0}{(1.0+e^{-x})}$ | The $e$ is the mathematical constant, which is the base of the natural logarithm. The sigmoid activation function returns the output between 0 and 1, therefore it is especially useful when predicting probability. However, for large negative or positive numbers, the function flattens, resulting in a small gradient. If the local gradient becomes too small, the backpropagation process will not work properly. |
| Tanh Hidden Layer Activation Function (TanH) | $\frac{(e^{x}-e^{-x})}{e^{x}+e^{-x})}$ | The function takes and real value and returns the output value between -1 and 1. The main advantages of TanH are that strongly negative inputs will be mapped strongly negative and near-zero inputs will be mapped near zero. However, similarly to the sigmoid function, it is subject to the vanishing gradient problem. |

Table 5.1: Popular Activation Functions For FNNs

The choice of activation function is data specific and will determine the learning rate of FNN. The higher the learning rate the shorter the training time, however, at the cost of diminished accuracy (Y. Bai, H. Zhang, and Hao, 2009).

## 5.3 Recurrent Neural Networks (RNN)

While in FNNs allows for the signals to travel only one way (i.e. from input to output), RNNs can have signals travelling in both directions. It is possible through the introduction of loops in the network. The main assumption of forward NNs is that the input and outputs are independent of each other. By introducing self-loops, the RNNs have the capability to memorize the data. The memorized data is affecting the activation of other neurons within the hidden layer, therefore affecting the ultimate forecast. Figure 5.3 shows the simple RNN, with the recurrence process highlighted. Note that recurrence can involve all neurons from the hidden layer.

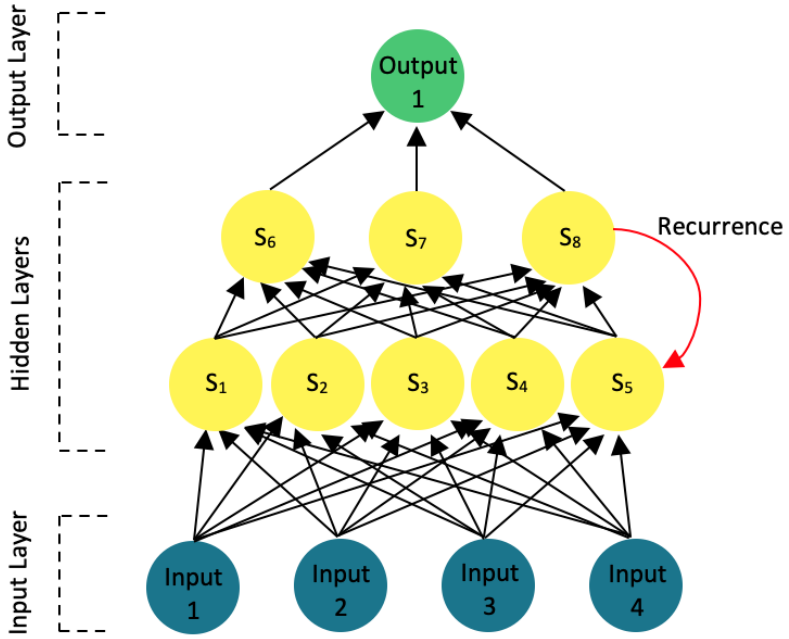


Figure 5.3: This figure shows the recurrent neural network with two hidden layers. In RNNs the signal can travel in both directions, therefore allows for the network to capture underlying relationships over time. This allows for the NN to build up memory which can affect the activation of other neurons, thus the final output.

RNNs can be viewed as multiple copies of the same network, each time with the signal passed onto the successor. To define values of hidden units, RNNs often use the following equation:

$$h^t = f(h^{(t-1)}, x^t; \theta) \tag{5.2}$$

where:

$h^t$ = the state of current neuron in a hidden layer at the time step t
$x^t$ = the variable value at time step t
$\theta$ = the weight parameter for current synapse determined by the activation function and learning process

As explained in 4.4, many macroeconomic variables are non-stationary. Thanks to its ability to recognize past sequences of data, RNNs can find the adequate stationary transformation of the variables so that their dynamics can explain asset prices (L. Chen, Pelger, and J. Zhu, 2020). In contrast to Markov Model, RNNs can learn important variable interactions across different states using information from both current and past states. Additionally, RNNs can detect changes over time, which is impossible with FNN (Gencay and T. Liu, 1997). Nevertheless, all that extra information used in RNNs, makes them more computationally expensive and often complicates the training as such NNs are prone to problems of gradient vanishing (Li et al., 2018).

## 5.4 Ensemble Neural Networks

One of the major drawbacks of neural networks, similarly to other non-linear models, is that they are prone to high variance. Recall the 4.2 in which we discussed the use of ensemble methods to deal with excessive over-fitting. Such methods can also be applied to NNs. By combining the predictions from numerous NNs, the variance can be reduced, thus a smaller chance of over-fitting. Table 5.2 shows the three main ensemble types used in ANNs. However, the choice of ensemble method is problem dependant. Usually, the constants of the problem can help in determining the optimal ensemble method. For example, if we were dealing with low amounts of data, the varying models or combinations methods would be probably a better choice than varying training data.

| **Neural Network's Ensemble Methods** | | |
|---|---|---|
| **Ensemble Type** | **Popular Methods** | **Description** |
| Varying Training Data | Bootstrap Aggregation (bagging), K-fold Cross-Validation, Random Training Subset | Varying Training Data ensembles, as the name suggests, use different techniques to divide the data into different subsets and then use these subsets to train the model. The final output is the weighted sum of all the single networks' forecasts. |
| Varying Models | Multiple Training Run, Snapshot, Horizontal Epochs, Hyperparameter Tuning | This group of ensembles is used to train the same data set using different variations of the neural network. Everything from the activation function to the number of neurons in the hidden layer. |
| Varying Combinations | Model Averaging, Stacked Generalization (Stacking), Boosting, Weighted Average | In this family of ensemble methods, the way the forecasts from single models are combined is altered. These methods are used to update the weights from each prediction model. |

Table 5.2: Popular Ensemble Methods for Neural Networks

## 5.5 Neural Networks in Asset Pricing

In traditional factor models, each factor is tested in terms of its explanatory power of the target variable (i.e. expected return). When it comes to neural networks, especially the ones with an extensive amount of hidden layers (i.e. deep neural networks), the trainable parameters used, are adjusted through the learning process to maximise the explanatory power of the network's forecast. Constructing a factor model requires testing

various combinations of variables, NNs are effectively performing multiple hypotheses at the same time, thus facilitating the entire process. Neural networks allow the discovery of the important relationships within the data, without the need for extensive feature engineering. Moreover, recurrent neural networks allow to model these relationships using the data from various points in time or even incorporate the changes in the structure of the specific variables over time. In other words, NNs are capable of finding qualities and sequences of a company's data that have the most predictive power.

The factor model can be used to construct portfolios, sort stocks and produce a cross-section of return analysis. Messmer (2017), uses a deep feed-forward neural network (DFN) to predict the US cross-section of stock returns. Using the data from the CRSP database from 1970-2014, he tested 68 individual firm characteristics, which are believed to possess information to explain differences in expected cross-sectional returns as suggested by Harvey, Y. Liu, and H. Zhu (2016) or Green, Hand, and X. F. Zhang (2017). He recognizes the problem of over-fitting as the main concern when training the neural network to predict excess returns. To deal with the excessive variance he employs various regularization techniques such as bagging[2] or early stopping[3]. Additionally, he performs stochastic gradient descent on mini-batch of data which decreases computational cost and proved to have regularization benefit (Wilson and Martinez, 2003). On the other hand to control under-fitting he evaluates the created model on independent data sets, derived using k-fold cross-validation, similarly to methods shown in 4.3.2, however, given the large time period of data, k-fold splits are more likely to bring desirable outcome. To find optimal hyper-parameters for his network, such as number of neurons per layer, number of hidden layers, activation algorithm etc. he employs random-search in which the hyper-parameters are drawn randomly. To determine which hyper-parameters to choose, he tested them on a sub-sample of data from 1970 to 1981 and therefore combines the random set of 75 predictions (from 150 available) as well as all 150 predictions for each stock to construct value-weighted portfolios for mid- and large-cap stocks separately. The performance of the value-weighted portfolio is compared to the equally-weighted

[2]Bagging is the ensemble method that aggregates the multiple predictions from various models to provide a final weighted forecast.

[3]Early stopping is another regularization strategy that stops the training process after the validation error is not improved for a certain amount of iterations.

one, Fama-French 5 factor model and Fama-French 5 factor model plus momentum factor, using the remaining time period (from 1981 to 2014). Finally, Messmer (2017), tests his strategy results in terms of the Sharpe ratio using a linearly computed benchmark. Overall, 16 different median, max, min, linear, value-weighted and equally-weighted portfolios are compared. Figure 5.4 shows the results from his tests across large- and mid-cap stocks.

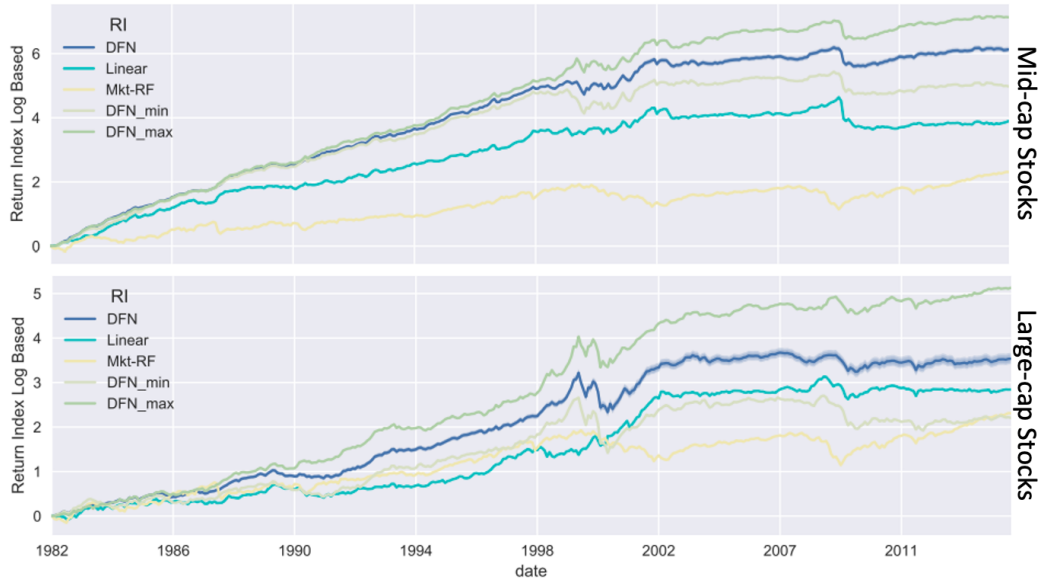


Figure 5.4: This figure shows the return from mid- and large-cap stocks using the strategy based on a deep forward network. The results are compared with the linear benchmark and the Fama-French 5 factor model with a momentum factor. The DFN's based strategy delivers on average superior returns compared to other methods. However, the fact that there is a significant difference between the best and worst-performing DFN portfolios, represent the uncertainty arising from the model estimation.

Source: Messmer (2017)

The difference between the best and worst performing DFN portfolio reflects the uncertainty arising from model estimation. Although the DFN used delivered a higher return than linear based benchmark and other tested methods, the fact that there is a huge difference between worst and best-performing DFN model's deciles, highlights, the uncertainty arising from model estimation. Additionally, when testing the impact of trading cost on strategies' return, he found that every month rebalancing prevents any of the DFD's strategies from achieving a positive mean return. Reducing the rebalancing

frequency to every five months yields the best results when accounting for transaction costs. Nevertheless, the superior performance of DFN compared to the linear benchmark, highlights the importance of non-linear relationships between firm characteristics and expected return. Furthermore, Messmer (2017), identifies the short-term reversal and the twelve-month momentum factors as the main drivers of expected return.

The model created by Messmer (ibid.) performs well, considering the issues stated in 3.2. It is not only in line with finance theory but also showed high explanatory capability in estimating excess returns of the stocks. It clearly specifies what factors, along with their predictive power, are to be implemented to the model and employ modern regularization techniques to ensure the most optimal structure of his network. As the model is still being improved, we expect that in the near future it could become a useful tool that will aid investment professionals in portfolio creation.

# Section 6

# Limitations

## 6.1 Limitations of Machine Learning

In this chapter, we showed multiple applications of machine learning in asset price forecasting. Although some of the methods discussed shown promising results in practical application, designing an asset-pricing model requires more than accurate predictions. The machine learning algorithms are designed to improve the way the model can fit the data, however, they do not disclose underlying economic mechanisms or equilibria. In addition to the data scientist responsible for designing and adjusting the model, there is a need for economists who can provide an underlying structure for the estimation problem. Only that way the created models can be implemented into the finance theory and further contribute to the asset-pricing field.

### 6.1.1 Machine Learning and Regulatory Environment

Apart from concerns regarding the model's predictive power and constraints coming from traditional econometric and financial theory, investors have to also consider the legal implications while using machine learning in asset pricing. Although algorithmic forecasts are on average 10% more accurate than human forecasters, across numerous domains, people continue to have a very low tolerance to machine learning's errors (i.e. algorithm aversion) (Dietvorst, Simmons, and Massey, 2014). Since the 2008-GFC, regulators undertook a more proactive approach when designing a financial regulation. For example 2018's E.U. General Data Protection Regulation (GDPR), grants investors the right to ask companies about their machine learning practices. Such a decision prompted private institutions to make their machine learning algorithms more transparent as well as ensure they are in line with finance theory (Kou et al., 2019). Furthermore, the Markets in Financial Instruments Directive (MiFID) II also introduced in 2018, allows

regularity authorities to require detailed information about the machine learning algorithm company uses, including the details of trading strategies or limits of the systems employed (Sheridan, 2017).

## 6.2 Limitations of This Study

The main purpose of this chapter was to explain the potential use of machine learning techniques in asset pricing. Although the results from individual studies were discussed, it is impossible to determine the effectiveness of any of the models presented in sections 4 and 5, without performing additional tests. Possibly all of the discussed methods would have to be evaluated over the same time period using the same dataset. Only such test would allow evaluating the predictive power of each model in comparison to the other methods including traditional factor models discussed in section 2.

Moreover, some of the investigated models such as these described in 4.2.2 or 4.3.2 diverge from traditional finance theory by leaving the estimation of the stock prices only to the momentum-based factors. However, more recently, finance practitioners and academics have focused on making the loss functions of machine learning algorithms more in line with the finance theory. For example, modern machine learning techniques are being used to reduce the number of features in the model (Feng, Giglio, and Xiu, 2020), (Kelly and Pruitt, 2015). Additionally, the implementation techniques used for model explainability, such as LIME Ribeiro, Singh, and Guestrin (2016) or SHAP Lundberg and Lee (2017) into the programming languages allows researchers to explain their algorithms better.

# Section 7

# Conclusions

In conclusion, this chapter first provided the theoretical framework for traditional asset-pricing methods and highlighted their limitations, which we believe can be addressed by machine learning techniques. The current machine learning techniques show promising results in dealing with high dimensional data of large volumes, such as today's financial data. Although, in this chapter we have not performed any tests that would allow for evaluation of machine learning models described, identifying main obstacles in designing a factor model, served as the main discussion point in the assessment of the models. From showing the statistical overview of each examined methods, to discussing their current application in asset-pricing, we were able to show the disruptive power of machine learning in the financial field. Machine learning methods such as penalized linear models, regression trees, support vector regressions or Markov switching models can be used to model expected return in respect to finance theory but also aid its development thanks to innovative solutions they provide. Traditionally, the modelling process involved testing numerous predictor variables in terms of their t-statistics and therefore combine findings into the linear regression model. Machine learning techniques can handle large amounts of data relatively easily, thus allowing for more possible factors to be tested. Moreover, they also introduce more appropriate evaluation methods such as bootstrapping or cross-validation, which let researchers assess their models more deeply without the significant increase in computational cost. Additionally, some of the methods that are able to capture non-linear interactions among variables, are yet to be fully documented when it comes to financial data. Furthermore, we believe that the one of the most rapidly developing subset of machine learning, neural networks are especially fit for the asset-pricing problem. Thanks to their ability to deal with a massive amount of variables and highly adjustable properties, neural networks make a great tool for estimating excess returns of stocks. NNs can learn from data and use the knowledge to adjust their structure. In that respect, the output from the ANNs can be viewed as a factor itself, mainly because of the fact

that it can be used to construct portfolios, sort the stocks or perform a cross-section of return analysis. The issue remains with carefully constructing a network itself, deciding on its width and depth as well as other hyperparameters. It is crucial for the network effectiveness, to structure it the best way possible. However, the task is highly problem dependant, thus calls for experts from the desired field to contribute. Luckily the recent developments in the open-source software community open doors for many researchers outside of the data science field to exploit the use of neural networks in other disciplines. The finance theory has not yet confirmed whether the neural networks are indeed a good tool in estimating asset price. Nevertheless, increasing availability of NNs frameworks will help to address this question.

# Bibliography


Abdel-Hamid, Ossama et al. (2014). "Convolutional neural networks for speech recognition". In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10, pp. 1533–1545.

Arnott, Rob, Campbell R Harvey, and Harry Markowitz (2019). "A backtesting protocol in the era of machine learning". In: *The Journal of Financial Data Science* 1.1, pp. 64–74.

Awad, Mariette and Rahul Khanna (2015). "Support vector regression". In: *Efficient learning machines*. Springer, pp. 67–80.

Ayodele, Taiwo Oladipupo (2010). "Types of machine learning algorithms". In: *New advances in machine learning* 3, pp. 19–48.

Azzini, Antonia and Andrea GB Tettamanzi (2008). "Evolving neural networks for static single-position automated trading". In: *Journal of Artificial Evolution and Applications* 2008.

Bai, Jushan and Serena Ng (2006). "Confidence intervals for diffusion index forecasts and inference for factor-augmented regressions". In: *Econometrica* 74.4, pp. 1133–1150.

Bai, Yanping, Haixia Zhang, and Yilong Hao (2009). "The performance of the backpropagation algorithm with varying slope of the activation function". In: *Chaos, Solitons & Fractals* 40.1, pp. 69–77.

Bailey, David H et al. (2015). "Mathematical Appendices to:'The Probability of Backtest Overfitting'". In: *Journal of Computational Finance (Risk Journals)*.

Ballings, Michel et al. (2015). "Evaluating multiple classifiers for stock price direction prediction". In: *Expert systems with Applications* 42.20, pp. 7046–7056.

Banz, Rolf W (1981). "The relationship between return and market value of common stocks". In: *Journal of financial economics* 9.1, pp. 3–18.

Blitz, David et al. (2016). "Five concerns with the five-factor model". In: *Available at SSRN 2862317*.

Boivin, Jean and Serena Ng (2006). "Are more data always better for factor analysis?" In: *Journal of Econometrics* 132.1, pp. 169–194.

Bramer, Max (2007). "Avoiding overfitting of decision trees". In: *Principles of data mining*, pp. 119–134.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Breiman, Leo et al. (1984). *Classification and regression trees*. CRC press.

Bruce, Peter, Andrew Bruce, and Peter Gedeck (2020). *Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python*. O'Reilly Media.

Campbell, John Y, Jens Hilscher, and Jan Szilagyi (2008). "In search of distress risk". In: *The Journal of Finance* 63.6, pp. 2899–2939.

Carhart, Mark M (1997). "On persistence in mutual fund performance". In: *The Journal of finance* 52.1, pp. 57–82.

Cawley, Gavin C and Nicola LC Talbot (2010). "On over-fitting in model selection and subsequent selection bias in performance evaluation". In: *The Journal of Machine Learning Research* 11, pp. 2079–2107.

Chan, Louis KC, Yasushi Hamao, and Josef Lakonishok (1991). "Fundamentals and stock returns in Japan". In: *The journal of finance* 46.5, pp. 1739–1764.

Chan, Louis KC, Jason Karceski, and Josef Lakonishok (1999). "On portfolio optimization: Forecasting covariances and choosing the risk model". In: *The review of Financial studies* 12.5, pp. 937–974.

Chen, Jieting and Yuichiro Kawaguchi (2018). "Multi-factor asset-pricing models under markov regime switches: Evidence from the Chinese stock market". In: *International Journal of Financial Studies* 6.2, p. 54.

Chen, Luyang, Markus Pelger, and Jason Zhu (2020). "Deep learning in asset pricing". In: *Available at SSRN 3350138*.

Clarke, Roger G and Harindra de Silva (1998). "State-dependent asset allocation". In: *Journal of Portfolio Management* 24.2, p. 57.

Cochrane, John H (2009). *Asset pricing: Revised edition*. Princeton university press.

Das, Kajaree and Rabi Narayan Behera (2017). "A survey on machine learning: concept, algorithms and applications". In: *International Journal of Innovative Research in Computer and Communication Engineering* 5.2, pp. 1301–1309.

De Mol, Christine, Domenico Giannone, and Lucrezia Reichlin (2008). "Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components?" In: *Journal of Econometrics* 146.2, pp. 318–328.

Dempsey, Mike (2013). "The capital asset pricing model (CAPM): the history of a failed revolutionary idea in finance?" In: *Abacus* 49, pp. 7–23.

Dey, Ayon (2016). "Machine learning algorithms: a review". In: *International Journal of Computer Science and Information Technologies* 7.3, pp. 1174–1179.

Dhrymes, Phoebus J, Irwin Friend, and N Bulent Gultekin (1984). "A critical reexamination of the empirical evidence on the arbitrage pricing theory". In: *The Journal of Finance* 39.2, pp. 323–346.

Dichev, Ilia D (1998). "Is the risk of bankruptcy a systematic risk?" In: *the Journal of Finance* 53.3, pp. 1131–1147.

Dietvorst, Berkeley J, Joseph Simmons, and Cade Massey (2014). "Understanding algorithm aversion: forecasters erroneously avoid algorithms after seeing them err". In: *Academy of Management Proceedings*. Vol. 2014. 1. Academy of Management Briarcliff Manor, NY 10510, p. 12227.

Doz, Catherine, Domenico Giannone, and Lucrezia Reichlin (2012). "A quasi–maximum likelihood approach for large, approximate dynamic factor models". In: *Review of economics and statistics* 94.4, pp. 1014–1024.

Eldan, Ronen and Ohad Shamir (2016). "The power of depth for feedforward neural networks". In: *Conference on learning theory*. PMLR, pp. 907–940.

Eugene, Fama and Kenneth French (1992). "The cross-section of expected stock returns". In: *Journal of Finance* 47.2, pp. 427–465.

Fama, Eugene and James MacBeth (1973). "ARisk, Return and Equilibrium: Empirical Tests". In: *Journal of Political Economy* 81.3, p. 607.

Fama, Eugene F and Kenneth R French (2008). "Dissecting anomalies". In: *The Journal of Finance* 63.4, pp. 1653–1678.

— (2015). "A five-factor asset pricing model". In: *Journal of financial economics* 116.1, pp. 1–22.

— (2016). "Dissecting anomalies with a five-factor model". In: *The Review of Financial Studies* 29.1, pp. 69–103.

Fama, Eugene F and Kenneth R French (2021). *Common risk factors in the returns on stocks and bonds*. University of Chicago Press.

Fama, Eugene F and Burton G Malkiel (1970). "Efficient capital markets: a review of theory and empirical work". In: *The Journal of Finance* 25.2, pp. 383–417.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu (2020). "Taming the factor zoo: A test of new factors". In: *The Journal of Finance* 75.3, pp. 1327–1370.

Fodor, Imola K (2002). *A survey of dimension reduction techniques*. Tech. rep. Citeseer.

French, Jordan (2017). "The time traveller's CAPM". In: *Investment Analysts Journal* 46.2, pp. 81–96.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Gagniuc, Paul A (2017). *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.

Gencay, Ramazan and Tung Liu (1997). "Nonlinear modelling and prediction with feedforward and recurrent networks". In: *Physica D: Nonlinear Phenomena* 108.1-2, pp. 119–134.

Gerlein, Eduardo A et al. (2016). "Evaluating machine learning classification for financial trading: An empirical approach". In: *Expert Systems with Applications* 54, pp. 193–207.

Graham, Benjamin and D Dodd (1951). *Security Analysis: Principles and Technique*. New York, NY: McGraw-Hill.

Green, Jeremiah, John RM Hand, and X Frank Zhang (2013). "The supraview of return predictive signals". In: *Review of Accounting Studies* 18.3, pp. 692–730.

— (2017). "The characteristics that provide independent information about average US monthly stock returns". In: *The Review of Financial Studies* 30.12, pp. 4389–4436.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu (2018). *Empirical asset pricing via machine learning*. Tech. rep. National bureau of economic research.

Hamilton, James D (1989). "A new approach to the economic analysis of nonstationary time series and the business cycle". In: *Econometrica: Journal of the econometric society*, pp. 357–384.

Harvey, Campbell R and Yan Liu (2014). "Evaluating trading strategies". In: *The Journal of Portfolio Management* 40.5, pp. 108–118.

— (2021). "Lucky factors". In: *Journal of Financial Economics*.

Harvey, Campbell R, Yan Liu, and Heqing Zhu (2016). "... and the cross-section of expected returns". In: *The Review of Financial Studies* 29.1, pp. 5–68.

Henrique, Bruno Miranda, Vinicius Amorim Sobreiro, and Herbert Kimura (2018). "Stock price prediction using support vector regression on daily and up to the minute prices". In: *The Journal of finance and data science* 4.3, pp. 183–201.

— (2019). "Literature review: Machine learning techniques applied to financial market prediction". In: *Expert Systems with Applications* 124, pp. 226–251.

Hirschberger, Markus, Yue Qi, and Ralph E Steuer (2010). "Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming". In: *European Journal of Operational Research* 204.3, pp. 581–588.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5, pp. 359–366.

Huberman, Gur (2005). *Arbitrage pricing theory*. Tech. rep. Staff Report.

Jiang, Jianmin, P Trundle, and Jinchang Ren (2010). "Medical image analysis with artificial neural networks". In: *Computerized Medical Imaging and Graphics* 34.8, pp. 617–631.

Kelly, Bryan and Seth Pruitt (2015). "The three-pass regression filter: A new approach to forecasting using many predictors". In: *Journal of Econometrics* 186.2, pp. 294–316.

Kou, Gang et al. (2019). "Machine learning methods for systemic risk analysis in financial sectors". In: *Technological and Economic Development of Economy* 25.5, pp. 716–742.

Krollner, Bjoern, Bruce J Vanstone, and Gavin R Finnie (2010). "Financial time series forecasting with machine learning techniques: a survey." In: *ESANN*.

Lewellen, Jonathan (2014). "The cross section of expected stock returns". In: *Forthcoming in Critical Finance Review, Tuck School of Business Working Paper* 2511246.

Li, Shuai et al. (2018). "Independently recurrent neural network (indrnn): Building a longer and deeper rnn". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5457–5466.

Lintner, John (1965). "Security prices, risk, and maximal gains from diversification". In: *The journal of finance* 20.4, pp. 587–615.

Lundberg, Scott and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *arXiv preprint arXiv:1705.07874*.

Maitra, Durjoy Sen, Ujjwal Bhattacharya, and Swapan K Parui (2015). "CNN based common approach to handwritten character recognition of multiple scripts". In: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, pp. 1021–1025.

Malkiel, Burton G (2003). "The efficient market hypothesis and its critics". In: *Journal of economic perspectives* 17.1, pp. 59–82.

Markowitz, Harry (1952). "The utility of wealth". In: *Journal of political Economy* 60.2, pp. 151–158.

Messmer, Marcial (2017). "Deep learning and the cross-section of expected returns". In: *Available at SSRN 3081555*.

Moritz, Benjamin and Tom Zimmermann (2016). "Tree-based conditional portfolio sorts: The relation between past and future stock returns". In: *Available at SSRN 2740751*.

Mossin, Jan (1966). "Equilibrium in a capital asset market". In: *Econometrica: Journal of the econometric society*, pp. 768–783.

Mullainathan, Sendhil and Jann Spiess (2017). "Machine learning: an applied econometric approach". In: *Journal of Economic Perspectives* 31.2, pp. 87–106.

Nayak, Rudra Kalyan, Debahuti Mishra, and Amiya Kumar Rath (2015). "A Naıve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices". In: *Applied Soft Computing* 35, pp. 670–680.

Partington, Graham et al. (2013). "Death where is thy sting? A response to Dempsey's despatching of the CAPM". In: *Abacus* 49.1, pp. 69–72.

Pástor, L'uboš and Robert F Stambaugh (2000). "Comparing asset pricing models: an investment perspective". In: *Journal of Financial Economics* 56.3, pp. 335–381.

Perold, André F (2004). "The capital asset pricing model". In: *Journal of economic perspectives* 18.3, pp. 3–24.

Rapach, David and Guofu Zhou (2013). "Forecasting stock returns". In: *Handbook of economic forecasting*. Vol. 2. Elsevier, pp. 328–383.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). "" Why should i trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Roll, Richard (1977). "A critique of the asset pricing theory's tests Part I: On past and potential testability of the theory". In: *Journal of financial economics* 4.2, pp. 129–176.

Ross, Stephen (1977). "Risk, return and arbitrage". In: *Risk and return in Finance* 1, pp. 189–218.

Roy, Rahul and Santhakumar Shijin (2018). "A six-factor asset pricing model". In: *Borsa Istanbul Review* 18.3, pp. 205–217.

Schapire, Robert E (1990). "The strength of weak learnability". In: *Machine learning* 5.2, pp. 197–227.

Sengupta, Nandini, Md Sahidullah, and Goutam Saha (2016). "Lung sound classification using cepstral-based statistical features". In: *Computers in biology and medicine* 75, pp. 118–129.

Sharpe, William F (1964). "Capital asset prices: A theory of market equilibrium under conditions of risk". In: *The journal of finance* 19.3, pp. 425–442.

Sheridan, Iain (2017). "MiFID II in the context of Financial Technology and Regulatory Technology". In: *Capital Markets Law Journal* 12.4, pp. 417–427.

Simin, Timothy (2008). "The poor predictive performance of asset pricing models". In: *Journal of Financial and Quantitative Analysis*, pp. 355–380.

Stock, James H and Mark W Watson (2002). "Forecasting using principal components from a large number of predictors". In: *Journal of the American statistical association* 97.460, pp. 1167–1179.

— (2012). "Generalized shrinkage methods for forecasting using many predictors". In: *Journal of Business & Economic Statistics* 30.4, pp. 481–493.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Treynor, Jack L (1961). "Market value, time, and risk". In: *Time, and Risk (August 8, 1961)*.

Von Luxburg, Ulrike and Bernhard Schölkopf (2011). "Statistical learning theory: Models, concepts, and results". In: *Handbook of the History of Logic*. Vol. 10. Elsevier, pp. 651–706.

Wilson, D Randall and Tony R Martinez (2003). "The general inefficiency of batch training for gradient descent learning". In: *Neural networks* 16.10, pp. 1429–1451.

Wolpert, David H (1996). "The lack of a priori distinctions between learning algorithms". In: *Neural computation* 8.7, pp. 1341–1390.

Womack, Kent L and Ying Zhang (2005). "Core finance trends in the top MBA programs in 2005". In: *Available at SSRN 760604*.

Wu, Xindong et al. (2008). "Top 10 algorithms in data mining". In: *Knowledge and information systems* 14.1, pp. 1–37.

Zou, Hui (2006). "The adaptive lasso and its oracle properties". In: *Journal of the American statistical association* 101.476, pp. 1418–1429.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the royal statistical society: series B (statistical methodology)* 67.2, pp. 301–320.