CHARLES UNIVERSITY
FACULTY OF SOCIAL SCIENCES
Institute of Economic Studies

On the Utilization of Machine Learning in Asset Return Prediction on Limited Datasets

Master's thesis

Author: Bc. Lukáš Petrásek

Study program: Economic Theories

Supervisor: doc. PhDr. Jozef Baruník, Ph.D.

Year of defense: 2019

Declaration of Authorship

The author hereby declares that he or she compiled this thesis independently, using only the listed resources and literature, and the thesis has not been used to obtain any other academic title.

The author grants to Charles University permission to reproduce and to distribute copies of this thesis in whole or in part and agrees with the thesis being used for study and scientific purposes.

Prague, July 31, 2019

Lukas Petrasek

Abstract

In this thesis, we conduct a comparative analysis of how various modern machine learning techniques perform when employed to asset return prediction on a relatively small sample. We consider a broad selection of machine learning methods, including e.g. elastic nets, random forests or recently highly popularized neural networks. We find that these methods fail to outperform a simple linear model containing only 5 factors and estimated via ordinary least squares. Our conclusion is that applications of machine learning in finance should be conducted carefully, because the techniques may not actually be as powerful as one might think when they are applied under unfavorable circumstances.

JEL Classification	C45, C52, C53, C58, G12
Keywords	asset pricing, machine learning, return prediction, regression, decision tree, random forest, neural network
Title	On the Utilization of Machine Learning in Asset Return Prediction on Limited Datasets

Author's e-mail	petrasek.lks@gmail.com
Supervisor's e-mail	barunik@fsv.cuni.cz

Abstrakt

V této diplomové práci provádíme komparativní analýzu toho, jak moderní metody strojového učení dokáží predikovat výnosy aktiv při použití malého množství data. Mezi použitými metodami jsou například elastic nets, náhodné lesy, nebo neuronové sít'ě. Konstatujeme, že tyto metody nedokáží predikovat lépe než jednoduchý lineární model obsahující pouze 5 faktorů. Moderní machine learning metody by se tedy měly aplikovat velmi opatrně, nebot' nemusí bý tak silné, jak by se mohlo zdát, pokud je aplikujeme za nepříznivých podmínek.

Klasifikace JEL	C45, C52, C53, C58, G12
Klíčová slova	oceňování aktiv, strojové učení, predikce výnosů, regrese, rozhodovací strom, náhodný les, neuronová sít'
Název práce	Využívání Strojového Učení pro Predikci Výnosů Aktiv při Použití Omezených Datasetů
E-mail autora	petrasek.lks@gmail.com
E-mail vedoucího práce	barunik@fsv.cuni.cz

Acknowledgments

Throughout the work on my diploma thesis, I have received plenty of valuable advice and support from my supervisor, doc. PhDr. Jozef Baruník, Ph.D. I would like to thank him for helping me formulate the research topic and for all the following guidance.

Typeset in LaTeXusing the IES Thesis Template.

Bibliographic Record

Petrasek, Lukas: *On the Utilization of Machine Learning in Asset Return Prediction on Limited Datasets*. Master's thesis. Charles University, Faculty of Social Sciences, Institute of Economic Studies, Prague. 2019, pages: 74. Advisor: doc. PhDr. Jozef Baruník, Ph.D.

Contents

List of Tables

List of Figures

Acronyms

B Benchmark

BATS Better Alternative Trading System

B/M Book to Market Ratio

CAPM Capital Asset Pricing Model

DJIA Dow Jones Industrial Average

EN Elastic Net

FSSS Fixed Sample Splitting Scheme

GBRT Gradient Boosted Regression Tree

IS In Sample

LASSO Least Absolute Shrinkage and Selection Operator

MSE Mean Squared Error

NASDAQ National Association of Securities Dealers Automated Quotations

NN Neural Network

NYSE New York Stock Exchange

OLS Ordinary Least Squares

OOS Out Of Sample

PCA Principal Components Analysis

PCR Principal Components Regression

PLS Partial Least Squares

ReLU Rectified Linear Unit

ReSSS Recursive Sample Splitting Scheme

RoSSS Rolling Sample Splitting Scheme

RGS Random Grid Search

RF Random Forest

S&P 500 Standard & Poor's 500

Master's Thesis Proposal

Author	Bc. Lukáš Petrásek
Supervisor	doc. PhDr. Jozef Baruník, Ph.D.
Proposed topic	Can machines learn how agents price assets?

Motivation Economists have been attempting to explain the way in which financial assets are priced for a long time. In the recent decades, the asset pricing theory advanced considerably. In their famous paper from 1993, Fama and French (1993) presented the so-called three-factor model which links asset returns to three variables – excess return of the market portfolio (like in the classical CAPM setting), market capitalization and book-to-market ratio. Since then, the empirical asset pricing literature has boomed rapidly. Hundreds of factors, among else momentum (Hou et al., 2015), profitability and investment factors, were proposed and tested by researchers. However, only very recently, more advanced deep learning methods have become to be applied to explaining asset returns (Gu et al., 2018, Krauss et al., 2017). These machine learning techniques can be a viable addition to the existing approach.

In this thesis, I intend to investigate the potential gains in using machine learning methods to the prediction of asset returns using the data provided on the websites of professors Kenneth French and Amid Goyal. Some of these methods have been successfully applied to factor models and were found to outperform the currently applied linear approaches, see e.g. Gu et al. (2018).

Hypotheses

Hypothesis #1: Machine learning methods better explain asset returns than the traditionally employed methods.

Hypothesis #2: Nonlinear effects are present in the prediction of asset returns.

Hypothesis #3: Predictions are more precise for blue-chip stocks than for small-cap stocks.

Hypothesis #4: The precision of the predictions is decreasing in time as markets are increasingly efficient.

Methodology My general intention is to the follow Gu et al. (2018) and Krauss et al. (2017) and perform a similar study on the dataset provided on prof. Kenneth French's website. Alternatively, similar dataset provided by prof. Amit Goyal can be used as well. The base factor models will then be enhanced (mainly for macroeconomic factors with potentially market-wide effects) and the importance of the added factors, some suggested by the literature (see e.g. Feng et al. (2017) for an inventory of factors used in the literature so far), will be tested. However, the core contribution of this thesis should lie in the comparison of the performance of the selection of recently developed machine learning methods to other techniques typically used in the literature. Where relevant, models of Fama and French (1993), Fama and French (2015) and Goyal and Welch (2007) will be used as benchmarks. A novel method of comparison between methods – the out-of-sample monthly return prediction R^2 developed by Gu et al. (2018) – will be used to decide which approach performs the best. Researchers who apply deep learning methods to financial data analysis point out the lack of transparency in the models, but only rarely do they investigate in a greater detail how individual factors contribute to their predictions. In this work, similarly to Nakagawa et al. (2018), layer-wise relevance propagation will be used to detect the importance of the studied factors. Finally, I will verify whether my results are in line with one of the observations of Krauss et al. (2017) that the predictions are less precise in time, likely due to markets behaving more efficiently.

Expected Contribution Efforts to find patterns according to which assets are priced have a long history in the economic literature. Albeit researchers are constantly developing a better understanding of these patterns, asset returns still haven't been explained in a satisfactory way (Fama and French, 2015) and hence there is still place for improvement. In most cases, the improvement is achieved by addition of new factors explaining the behavior of asset returns, and this is one way where this work could bring some contribution. Another, and recently increasingly pronounced, way to add to existing methods is the exploitation of machine learning techniques. Recently, Gu et al. (2018) and Krauss et al. (2017) have shown that the application of deep learning methods like, for example, neural networks, tree learning or forest methods can improve the predictive power of asset pricing models. However, they haven't included e.g. memory-based approaches (such as the long short-term memory network) in their portfolio of methods. Moreover, the methods can be tested on a variety of alternative sample splitting schemes. The contribution of this work will be in providing further information about how machine learning methods compare with the performance of existing linear methods and how they can be used in empirical asset pricing.

Apart from their contribution to the asset pricing literature, the results of this

work will potentially be attractive to entities that trade equity on financial markets as they can bring more insight into how different assets are priced.

Outline

1. Introduction: introduction to empirical asset pricing, factor models, and the role of machine learning
2. Literature Review: extensive review of the research that has been conducted in this area so far
3. Methodology: description of the methods (both linear and nonlinear) used in the thesis
4. Data Description: presentation of the dataset and its main characteristics
5. Results: description of the results, comparison of the performance of different methods, and the discussion of the outcomes in the context of outcomes of other papers
6. Conclusion: summary and concluding remarks

Core bibliography

Fama, Eugene F., and Kenneth R. French. "Common risk factors in the returns on stocks and bonds." Journal of Financial Economics 33, no. 1 (1993): 3-56.

Fama, Eugene F., and Kenneth R. French. "A five-factor asset pricing model" Journal of Financial Economics 116, no. 1 (2015): 1-22.

Fama, Eugene F., and Kenneth R. French. "International tests of a five-factor asset pricing model." Journal of Financial Economics 123, no. 3 (2017): 441-463.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu. "Taming the factor zoo." (2017).

Goyal, Amit, and Ivo Welch. "A comprehensive look at the empirical performance of equity premium prediction." The Review of Financial Studies 21, no. 4 (2007): 1455-1508.

Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu. "Empirical asset pricing via machine learning." (2018).

Hou, Kewei, Chen Xue, and Lu Zhang. "Digesting anomalies: An investment approach." The Review of Financial Studies 28, no. 3 (2015): 650-705.

Krauss, Christopher, Xuan Anh Do, and Nicolas Huck. "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500." European Journal of Operational Research 259, no. 2 (2017): 689-702.

Nakagawa, Kei, Takumi Uchida, and Tomohisa Aoshima. "Deep factor model". (2018).

Author

Supervisor

Chapter 1

Introduction

The field of empirical asset pricing which emerged decades ago has gone through an exciting period of growth recently. With growing computing capacity and availability of data, new possibilities open up in the analysis of financial markets. As companies operating on exchanges intensify their use of algorithmic trading, information become more accessible and markets become more efficient, arbitrage opportunities slowly disappear and the pressure increases for quantitative analysts to build more precise models. The need for better models is also apparent in academia. Recently, we have seen a dynamic advancement in the asset pricing theory. Narrowing our attention to the field of multifactor models, great progress has been made from the early contributions of, for example, Sharpe (1964) and Lintner (1965a;b) who developed the Capital Asset Pricing Model (known simply as the CAPM), or Banz (1981) who was the first to document effects of the size factor.

Since then, researchers have largely focused on shedding light on the behavior of financial markets by adding different factors to the returns equation. A highly valuable body of research follows the work of Fama & French (1993) who presented a model with three factors (excess return of the market portfolio, market capitalization and book-to-market ratio). Factor models have proved to describe prices of assets very well, and they have become widely accepted among the academic community and also in business. Conventionally, the authors assumed a linear relationship between asset returns and risk factors, however, as e.g. Gu *et al.* (2018) or Feng *et al.* (2018) have shown, the links between the variables might actually likely be of a nonlinear form which the traditional and widely used methods simply can not capture. This was also suggested by Black *et al.* (1972) as early as in 1972.

In the area of asset pricing, researchers often add new factors to the returns equation explaining the behavior of asset returns and then test their relevance. Another, and recently increasingly pronounced, way to enhance the existing knowledge in this field is the exploitation of machine learning techniques which often help uncover the potentially nonlinear nature of the risk-return relationship. A new branch of research in the asset pricing field has emerged which utilizes various machine learning techniques. After a long period of searching for (among else) new factors, some economists now begin to search for new estimation methods instead. Many of these new methods, in comparison with the previously applied ones, are able to capture the nonlinear and complex structure of the examined relationships.

Examples of these are Gu *et al.* (2018) and Krauss *et al.* (2017) who have recently shown that the application of deep learning methods like, for example, neural networks, or tree-based methods like gradient boosting or random forest can be highly beneficial for the predictive power of asset pricing models. Or Ban *et al.* (2016) who applied a highly specialized machine learning algorithm to portfolio optimization and were able to outperform some of the well-known Fama and French portfolios.

Most of the successful applications of machine learning in finance have one thing in common. They utilize enormous datasets compared to what a usual financial analyst or junior researcher is able to collect. But what happens when the size of the data is small? Because the modern machine learning techniques are rarely applied to datasets of limited size and dimension, we believe that the employment of these highly complex methods on small data should receive some attention.

In this thesis, we attempt to investigate the consequences of applying machine learning techniques which have been heavily promoted in the finance literature lately to a dataset of limited size. We choose the topic of asset pricing, because it has also received much attention in the recent decades. Our investigation will proceed similarly as a another comparative analysis on this topic, which is the work of Gu *et al.* (2018). We will examine the performance of several models built to make asset return forecasts utilizing a set of commonly used factors in the literature. The limitations of our data are two-fold. First we have gathered a moderate-sized dataset on asset returns of more than 3000 stocks traded in the United States equity market which is however almost ten times smaller by observation count than that of Gu *et al.* (2018). Second we consider only a small set of 19 predictor variables provided by Goyal (2019)

and French (2019) to which we add the lagged asset returns. Details about the data we employ can be found in Chapter 4.

When we apply some of the newly developed machine learning algorithms, we must account for their vulnerability with respect to potential overfit. This is why regularization techniques need to be employed when using these highly complex methods. We therefore implement several sample splitting schemes which allow us to control for the complexity of the given method via tuning the so-called hyperparameter of the model. The splitting schemes also enhance the challenging nature of our research by limiting the amount of data fed into an estimation algorithm or exposing the hyperparameter tuning process to a limited validation sample size. The sample splitting and hyperparameter tuning procedures are further detailed in Chapter 3 and Chapter 5.

Let us add a final note about the presentation of our results below. We always show only the Out Of Sample (OOS) predictive performance of the methods we apply. We do this in order to provide the reader with a realistic view of the actual accuracy of predictions. In the past, In Sample (IS) results showing contribution of many factors have been often presented without consideration of their true predictive ability out of sample. We restrain from this approach because we find it misleading.

The thesis is structured in the following way. Chapter 2 provides an extensive review of the asset pricing literature and other related works. In Chapter 3, theoretical framework is laid out and the selected methods used in our analysis are introduced. Chapter 4 provides description of our dataset. Chapter 5 discusses the implementation details of our methodology. In Chapter 6, the results of our analysis are presented to the reader. Finally, Chapter 7 concludes.

Chapter 2

Literature Review

During the past decades, economists have exerted a lot of effort to find patterns according to which assets are priced. Albeit researchers are constantly developing a better understanding of these patterns, asset returns still haven't been explained in a satisfactory way (Fama & French 2015). It would be overly optimistic to think that this thesis can fully explain how financial instruments are priced, nevertheless, in this work, we intend to at least help practitioners and the academic community understand another piece of the asset-pricing puzzle. An important first step prior to making any contribution in the field of asset pricing, and hence prior to the conduct of research on the topic of this thesis, is to develop a deep understanding of the current state of knowledge in the field of asset pricing, with special attention paid to multifactor models. It is this area of research on which our work is based. But first, let us review the origins of asset pricing in general. Before the beginning of the review, let us note that while we intend to embody the research in the field of asset pricing in the whole, most of the work is actually focused on stock pricing. This, however, does not represent any problem at all because the empirical part of this thesis will be centered around stock returns anyway.

This chapter contains two important parts. First, Section 2.1 presents the evolution of the central model in asset pricing, the Capital Asset Pricing Model (CAPM). Second, Section 2.2 builds upon Section 2.1 and introduces various modifications of CAPM which have appeared in the literature in the past. Models of this nature are commonly referred to as multifactor models.

2.1 The Capital Asset Pricing Model and Related Literature

We begin our survey of literature by examining the early work of Sharpe (1964) who began to develop the CAPM with his study of the behavior of asset prices under risk conditions.[1] He built a market equilibrium model for asset prices under assumptions about the preferences and possibilities of individual investors and studied its theoretical properties. A central idea behind the origin of the CAPM is that there is a negative relationship between the returns which investors expect for holding an asset and the uncertainty, i.e. risk, connected to holding this asset. This is reflected in the following statement from Jensen (1968) who argues that there exist two main tasks of a portfolio manager – make good predictions of asset prices to increase the portfolio's returns, and to minimize the portfolio's exposure to risk.

Simultaneously to Sharpe, Lintner (1965b) developed essentially the same model in a more rigorous way and presented additional implications. Fama (1968) in his short paper compares the models of Sharpe and Lintner and concludes that there are no practical differences in the models and that both approaches are equivalent. One year after publishing his paper, Sharpe (1965) empirically tested his own model under the assumption of market efficiency and concluded that the data support his model specification. Lintner (1965a) later elaborated on the model by publishing a similar but more advanced analysis. In his work he used a different set of assumptions and came to slightly different conclusions. Clarification of the previous conclusions was made by Mossin (1966) who constructed the model with more precision. The joined contribution of Sharpe, Lintner, and Mossin is often demonstrated by references to the Sharpe-Lintner-Mossin CAPM which is an expression for the well-known original version of the Capital Asset Pricing Model. Hirshleifer (1965) presented a valuable overview of different approaches in the field of investor decision modelling, briefly describing, among else, models characterizing production decisions or the introduction of risk-aversion and its impacts.

Since the times in which the CAPM originated, economists focus on modelling financial market equilibria using expected returns. Hirshleifer (1964) who appreciated Sharpe's contribution in the field and followed his work, mod-

[1] We must not forget to note that it is Jack L. Treynor who seems to be the first to develop the CAPM, but his work from 1961 has been unpublished for a very long time, see Treynor (1961) for an edited version of his CAPM manuscript.

elled investment decisions under time-state preferences which assume complete knowledge of the investors' beliefs about the probabilities of each state and pointed out that securities are of artificial nature, and that to be able to explain their behavior, researchers must also focus on the process of their creation. Hamada (1969) studied the propositions made by Modigliani & Miller (1958) and pointed out that the framework of Hirshleifer (1964) is not very feasible because it is difficult to be empirically tested due to the nonexistence of markets for each state.

Many other interesting applications and extensions of the so-called Sharpe-Lintner-Mossin CAPM model have been introduced in the field. Fama *et al.* (1969) built on the CAPM and examined the relationship between stock prices and the information coming from stock split announcements. They found that in the period which follows the announcement, markets adjust fast to the new information, and that they do so through adaptations of expectations regarding the future dividend payments. Levy & Sarnat (1970) investigated the implications presented by the interconnections among the world's financial markets and concluded that due to restrictions on international capital movements, we still observe inefficiency among financial markets in the studied developed countries. Jensen (1969) created a model for evaluating the performance of managed portfolios. Solnik (1974) built an international model for asset pricing and, among else, came to an interesting conclusion that forward exchange rates may be biased estimates of future exchange rates. Fama & French (1998) studied the influence of the book-to-market ratio in international financial markets.

Fama (1971) developed a micro-based risk-return model for the market equilibrium without relying on the assumption of riskless borrowing and lending. Black *et al.* (1972) presented a two-factor risk-return model and found evidence that the expected value of the market excess return beta may be evolving in time, and they also rejected the traditional Sharpe-Lintner-Mossin form of CAPM. Miller (1977) argued that the basic CAPM was built with the usage of a faulty assumption that each investor has equivalent estimates of returns for each security. He developed a model where the estimates can differ across investors. CAPM with the effect of taxes was introduced by Litzenberger & Ramaswamy (1979) who also integrated several borrowing restrictions into the model.

An intertemporal version of CAPM was introduced by Merton (1973a) who extended the model and observed that the demand for financial assets is susceptible to changes in the expectations about future investment opportunities.

He also argued that the expected returns might differ from the riskless interest rate even under no systematic risk. Galai & Masulis (1976) used a combination of the option pricing model developed by Black & Scholes (1973) and the classical CAPM setting to model the behavior of equity prices and the underlying risks. Merton (1980) investigated the expected market return and its relation to risk and presented several equilibrium models and asserted that the regression coefficients of models with realized market returns should be treated with caution if not being properly adjusted for heteroscedasticity. Hamada (1972) provides several extensions of the CAPM only by transforming the variables in the traditional equation to account for the given firm's capital structure. Mossin (1968) follows Tobin (1965) and introduces a multiperiod model for portfolio management and its theoretical characteristics.

On the other hand, economists have not always supported the CAPM, we will now go through several examples of views in asset pricing contrarian to the CAPM framework. Friend & Blume (1970) presented an analysis in which they questioned the traditional approach of Sharpe, Lintner and Jensen and concluded that CAPM-based estimates of portfolio performance are biased. Another work in which the value of the CAPM theory was challenged was Black *et al.* (1972) where the traditional Sharpe-Lintner-Mossin form of CAPM was rejected based on the observation that the expected excess return on a given security may not be linearly related to its beta. Merton (1973a) showed that under the absence of systematic risk, the expected returns might differ from the riskless interest rate, contrary to the CAPM implications. Galai & Masulis (1976) combined the CAPM with the option pricing formula of Black & Scholes (1973) and one of their conclusions was that betas may in fact be non-stationary. Similar remarks can be found in Bachrach & Galai (1979) who find that cheaper stocks carry more risk than those which are more expensive and in Gonedes (1973) who also postulates that the market beta could be a function of the given asset's price. Fama & Schwert (1977) examined the hedging nature of several assets and found that e.g. residential real estate was used to hedge against inflation during the 1953 – 1971 period, hence suggesting that there are other reasons to hold specific assets than the plausible risk-return tradeoff. Reinganum (1981) came to a conclusion that markets are either inefficient or the CAPM does not hold after observing other factors – firm size and the earnings-price ratio – to explain asset returns. Another critique of the CAPM may be found e.g. in Campbell (1996), or in Fama & French (2004) who provide a concise summary of the documented weak spots of the model.

Because of the amount of different approaches, methodologies and models and their countless adaptations and extensions, we suggest the reader to refer to papers which summarize the evolution in the field, such as Hirshleifer (1965) with a technical review of related frameworks used in the asset pricing literature. Another overview of the then-used approaches can be found in Fama & Malkiel (1970) who present several models along with the relevant empirical evidence and many different tests of market efficiency of all forms.[2]

Briefly after the CAPM was constructed, vast amounts of papers began to deal with the empirical tests of this framework and the relationship between risk and return has been frequently tested throughout the last couple of decades. Jensen (1968) empirically evaluated the performance of 115 mutual funds and concluded that these funds, despite their high expenditures on research, were not able to outperform a proposed simple trading strategy. He admitted, however, that he had not considered the contribution of funds in minimizing the market risk like e.g. Jensen (1969) did. The results of Fama & MacBeth (1973) who test the risk-return relationship on the New York Stock Exchange (NYSE) data support the efficient market hypothesis.

A body of research devoted to option pricing emerged during this pioneering period in the asset pricing field. Black & Scholes (1973) examined the relationship between risk and return for options and constructed their well-known option pricing formula. This model was extended by Merton (1973b) and then further explored by Merton (1976) who investigated the pricing formula under discrete option prices.[3] Merton (1974) developed and applied a model for pricing corporate liabilities and concluded that the model can be used to price essentially any type of security, the model was built in a similar way as Black & Scholes (1973) did build their option pricing model. Cox *et al.* (1979) constructed a more general option pricing model, special case of which is the famous Black-Scholes option-pricing equation.

Later on, Merton (1987) made an investigation into different anomalies caused by incomplete information and concluded that these anomalies might persist for a longer time even in markets with rational participants. Shleifer & Vishny (1997) contributed to the topic with their paper on arbitrage where they advocated that instead of many small market participants, who by exploiting the arbitrage opportunities push stock prices towards their fundamental values, the case is rather that a small number of specialized investors make large trades

[2]These are the weak, semi-strong, and strong forms of market efficiency.
[3]Black & Scholes (1973) assumed continuous option prices.

and they documented that under this setting, it is possible for prices to diverge from their fundamental values. As a response to the critique that the CAPM is not supported by empirical data, Jagannathan & Wang (1996) developed a model which allows the betas to be variable in time and concluded that the model fits the data well. Keim & Stambaugh (1986) studied seasonal effects in asset pricing. Ferson & Harvey (1991) studied the influence of additional factors, they found the market risk premium to be the most important factor. Two years later, Ferson & Harvey (1993) modelled equity market returns using both country-specific explanatory variables and variables with world-wide influence to study the risk-return relationship in equity markets on an international level.

More recently, Lettau & Ludvigson (2001) contributed to the family of conditional CAPM models and claim that their model performs about as well as that of Fama & French (1993). Campbell & Vuolteenaho (2004) split the market excess return beta into two betas to empirically estimate the market effect in terms of the future expected cash flows and the market discount rates. Barberis *et al.* (2015) examine a model which they call the extrapolative CAPM in which some investors are assumed to build their expectations of future asset returns based solely on the past returns.

Next, we will present a short overview of the historical evolution of the strand of literature devoted to multifactor models, i.e. extensions of the classical CAPM setting for various explanatory variables other than the market excess return. As we will show, many of these variables, i.e. factors, were found to significantly contribute to the explanation of variability among individual asset returns.

2.2 Multifactor Models Literature

Since the early 70's, when the validity of the original CAPM became to be questioned, researchers began to use other variables, i.e. factors, to describe the behavior of asset prices. Firstly, the significance of the newly added factors was primarily used to demonstrate that the CAPM is not complete and as such does not hold. Later, when it became widely accepted that the CAPM is not valid and that other variables than the excess market return influence the variability in asset returns, economists and financial analysts continued to add different factors to the risk-return relationship to enhance the accuracy of their models. This has soon become attractive not just from the academic

perspective, but also from the business perspective as investors were able to use the knowledge to increase the performance of their portfolios.

Among the first attempts to include other explanatory variables to the expected return equation, Arditti (1967) used the debt-equity and dividend-earnings ratios to explain asset returns. The dividend-earning ratio was found to be negatively related to returns, and the debt-equity ratio was, counterintuitively, found to be negatively related to returns as well. Arditti stated that the only explanation for the negative coefficients which he could offer was that there must exist explanatory variables relevant for the relationship but omitted in his regressions. Explanatory variables other than market excess return which, contrary to CAPM implications, helped to explain the variation in asset returns later began to be referred to as anomalies. Reinganum (1981) documented that the earnings-price ratio helps to explain variation in asset returns, but that its effect is rather contained in the firm size effect. He also asserted that in view of these findings, the CAPM is either misspecified, or the central assumption of market efficiency does not hold. Ferson & Harvey (1991) studied the influence of several additional factors for both stock and bond returns. Fama & French (1992) argued that size and book-to-market ratio describe the individual stock returns well.

A great breakthrough in the multifactor model field came with the work of Fama & French (1993) who developed the so-called three-factor model which linked asset returns to three variables – excess return of the market portfolio (like in the classical Sharpe-Lintner CAPM setting), market capitalization and book-to-market ratio. Since then, the empirical asset pricing factor literature has boomed rapidly. Hundreds of factors, among else momentum (Hou *et al.* 2015), profitability and investment factors, were proposed and tested by researchers, see e.g. Feng *et al.* (2019) who presented a survey of most of the factors suggested in the history of asset pricing literature. An interesting observation was documented by Schwert (2003) where it was claimed that the described multifactor models were implemented into trading strategies of investors and that this made several of the previously observed effects disappear.

Jegadeesh & Titman (1993) explored the influence of momentum factors and asserted that strategies based on momentum promise abnormal profits. Chan *et al.* (1996) argued that after inclusion of momentum factors, specifically lagged returns and lagged earnings, the size and book-to-market factor and the original market excess return factor do not contribute to the asset pricing process explanation anymore. A similar conclusion about momentum factors

and trading volume can be found in Brennan *et al.* (1998). Trading volume was also examined in Lee & Swaminathan (2000). Chan *et al.* (2001) found that stocks of companies with high ratio of research and development expenses to equity deliver higher returns. Jegadeesh & Titman (2001) analyzed the reasons behind the success of momentum strategies. Pástor & Stambaugh (2003) found their liquidity factor to bring contribution even under the presence of other factors, including momentum. Francis *et al.* (2005) suggested the quality of financial reports to be recognized as another important risk factor. The role of default risk was analyzed by Vassalou & Xing (2004). Diether *et al.* (2002) documented that stocks which have higher dispersion among the analysts' forecasts of their earnings earn abnormally low returns, but rejected the view that the dispersion in the forecasts is a good proxy for risk. Asness *et al.* (2013) provided yet another support for the momentum and size factors after they performed a study on different markets and across different classes of assets. In their paper from 2009, Fang & Peress (2009) carried out a cross-sectional analysis of the effects of media coverage in stock returns. They found that companies with higher media coverage tend to generate lower returns.

But not all new factors bring contribution in presence of the existing ones. Consider an example of the paper written by Fama & French (1996), where they claimed that their original three-factor model introduced in Fama & French (1993) covered most of the effects of several other variables which were previously found to be significantly contributing to the explanation of asset return behavior. Campbell (1996) studied the contribution of more than 20 different factors, but only few of them were found to be significant. Kothari *et al.* (1995) found a weaker relationship between the book-to-market ratio and asset returns than Fama & French (1992), and argued that the previous results describing a strong influence of the book-to-market ratio might suffer from selection bias. Lewellen *et al.* (2010) criticised the empirical methods used not just in the multifactor asset pricing literature and argued that many of the published models actually described the risk-return relationships worse than presented. MacKinlay (1995) even stated that the inclusion of other factors did not explain satisfactorily why did asset returns behave differently than what the CAPM would propose and suggested that the findings of such effects may be results of data fishing. Fama and French responded to these claims in Fama & French (1996) and albeit admitting that the presence of bias stemming from data fishing may not always be averted, they provided several arguments to defend the multifactor approach to asset pricing. Another evidence against

data fishing was presented in Jegadeesh & Titman (2001).

Lakonishok *et al.* (1994) asserted that the reason behind the success of trading strategies used by value investors was that they were capitalizing on suboptimal actions of other traders rather than being riskier and thus carrying higher returns. Gompers *et al.* (2003) found that traded companies with little shareholder rights provided lower returns than companies with more shareholder rights. Baker & Wurgler (2006) examined the role of investor sentiment in the pricing of securities. Lee & Swaminathan (2000) provided further information explaining the contribution of trading volume.

As we have shown, plenty of different anomalies were documented in the asset pricing literature over the recent decades. For easier orientation, see e.g. Green *et al.* (2013) who provided a review of the most important factors which have appeared in the literature. Similar inspections can be found also in Lewellen *et al.* (2015) who examined the performance of a model combining 15 stock-level characteristics, and in Feng *et al.* (2019) as mentioned above.

Other related papers followed the early accomplishments in the field of multifactor asset pricing. In 1998, Fama & French (1998) published a paper examining the role of the book-to-market ratio in international financial markets. Amihud (2002) proposed that the previously documented size effect may cover the effects of liquidity on stock returns. Another example of a study on the effects of liquidity in asset pricing is Acharya & Pedersen (2005) who identified three different channels for the influence of liquidity risk on asset returns and presented a form of CAPM adjusted for liquidity. Effects of several additional factors are examined in Fama & French (2008). Bai (2003) performed a theoretical study of a multifactor framework with large amount of explanatory variables, but low amount of time periods. Fama & French (2012) studied their multifactor framework on international data from four different regions of the world, but did not provide strong evidence for the integration of these markets.

Chapter 3

Theoretical Framework

To achieve the objectives of our research outlined in Chapter 1, i.e. examining the performance of modern machine learning techniques used to predict asset returns with limited datasets, we first have to set up clear theoretical grounds on which we can base our investigation. In this thesis, we perform a comparative analysis of how well different methods predict asset returns given datasets of various sizes. For that reason, we introduce the reader to several more or less advanced estimation techniques which represent the building blocks of our analysis.

Most of these methods are flexible for uncovering complex relationships in the data, but also susceptible to overfitting models and providing misleading results if not used wisely. Their susceptibility to overfit stems from the ability to encompass complicated nonlinear patterns which the standard linear techniques often used in asset pricing can not. Hence the feature that makes these methods so powerful is also the biggest threat for the researcher. To limit the chances of overfitting as much as possible, it is advisable to utilize regularization routines which restrict the method's complexity and make it less prone to fitting noise. Regularization is applied through the so-called hyperparameter tuning which is a name for the process of finding the parameters that determine the complexity of a given estimation method. It is important to emphasize yet again, that applications of machine learning methods must be performed carefully in the interest of obtaining reliable results. Therefore, each time we present the given method in Section 3.4 below, we provide suggestions for wrapping the method in a suitable regularization procedure.

The conduct of our estimations begins with the choice of a sample splitting scheme which governs the amount of data used for training our models, the

amount of data used for selecting the best hyperparameters, and the amount of data used for testing the model's predictions. The choice of points on which a dataset is split into training, validation, and testing sample determines exactly how will every observation be used and hence can severely influence the outcomes of the whole estimation procedure. Therefore, the subject of sample splitting must be addressed as well.

In this chapter, we first introduce the notion of hyperparameter tuning in more detail in Section 3.1. Second, Section 3.2 provides a brief discussion of sample splitting arrangements. Third, the general framework for predicting asset returns is laid out in Section 3.3. Section 3.4 contains a review of characteristics of selected machine learning methods later probed in our study.

3.1 Hyperparameter Tuning

When fitting models using highly sophisticated estimation methods, one must proceed with a certain level of caution. Sometimes, a model can be fitted so well that it loses its predictive capability. This may happen when the model is complex enough that it captures the delicate random variations, i.e. noise, in the specific data it is applied on, instead of capturing the underlying relationship between variables of interest. Such situation is known as model overfitting. When new data is fed into an overfitted model, the predictions it makes are then often wrong and misleading. To avoid overfitting, the complexity of the model can be restricted by regularization, i.e. by choosing such parameters that induce greater simplicity in the structure of the model. The parameters which control the model's complexity are referred to as hyperparameters. Consider the number of components used in principal components regression, tree depth in regression trees, or number of hidden layers in neural networks as examples of such hyperparameters.

The simplest approach to avoid overfit would be to use only a small number of predictors, fit a gradient boosted tree with only a few boosting iterations, or train a shallow neural network. However, the most parsimonious models do not necessary make the best use of our data and hence we need a more advanced approach for choosing hyperparameters to train our models with. The procedure of choosing appropriate hyperparameters is termed hyperparameter tuning.

In time series modelling, the usual way to approach hyperparameter tuning is to split the data available for building the model into two subsamples - one

used for training and the other for the so-called validation. The procedure involves repeatedly fitting the model, each time with different hyperparameter values, making predictions of the response variable values on the validation sample, and subsequently choosing the hyperparameter values under which the model performs best on the validation sample.[1] While the validation sample is in fact not used directly during estimation of the inner parameters of the model, it can not be perceived as out-of-sample data, because it is still used to search for optimal values of the hyperparameters.[2]

Regularization should help the model produce more stable out-of-sample results. The essential goal of regularization techniques is to cut down on fitting of the noise without deteriorating the fit of the signal. However, there is one important issue that comes with tuning our hyperparameters. Hyperparameter tuning can be a highly computationally intensive process and hence very expensive not only in terms of time. The hyperparameter space, i.e. the set of possible values that the hyperparameters can take, and the algorithm for searching the best hyperparameters must both be chosen carefully so that the computations are efficient, i.e. enough reasonable alternatives are covered, but the computing capacity is not wasted. For each searched point in the hyperparameter space, a model has to be fitted from scratch. This can consume a lot of computing capacity and take a lot of time, especially in the case of tree-based methods or artificial neural networks where fitting a single model itself can be very demanding.[3]

[1] It is important now to clearly establish the distinction between the hyperparameters and the inner parameters of the model, such as beta coefficients in regressions. The given modelling technique, be it e.g. the partial least squares or the random forest algorithm, always chooses the inner parameters in such a way that they provide the best fit of the training data under the given hyperparameters, but the hyperparameters must be determined before fitting the model. The validation data then serves as a reference sample on which the performance of the model with the already estimated inner parameters is evaluated. Based on the performance on the validation sample, the most accurate model, i.e. the full model specification along with the hyperparameter setting, is selected and later used to build out-of-sample forecasts.

[2] In contrast to modelling data with temporal ordering, modelling cross-sectional data often utilizes a different concept known as k-fold cross-validation. Under k-fold cross-validation, the part of data devoted to training is split into k equally sized subsamples. For a total of k times, the model is trained on k - *1* of the subsamples, and the remaining *kth* subsample is used for validation, each time changing the validation subsample so that each observation, falling into one of the k subsamples, is used exactly once for the validation purposes. Each training-validation round produces an estimate of the model parameters, the resulting single set of parameters is then the average of the k results. The usage of cross-validation in time series modelling is discussed e.g. in Bergmeir & Benítez (2012).

[3] In this sense, we could imagine even more costly scenario of running a hyperparameter optimization over a hyperparameter tuning procedure where we could optimize, for example,

Few sections below, we find additional details regarding regularization. Individual suggestions of regularization techniques for the selected methods are described in Section 3.4 below, the specific hyperparameter spaces that we cover and the choice of algorithm for hyperparameter tuning is depicted in Chapter 5.

3.2 Sample Splitting Schemes

As was stressed out in the previous section, it is very important for the performance of all regularized methods to select those hyperparameters under which the method is the least prone to overfitting the model. To appropriately tune the hyperparameters, we cannot measure the performance of the given technique on the same data on which it was fitted. Such approach would likely directly lead to choosing hyperparameter values which allow the method to fit as much noise as it can, and thus overfitting the model. Instead, we use data succeeding the training period which never enter the procedure of estimating the inner parameters of the underlying model, but which also cannot be used for making predictions. This part of data is referred to as the validation sample. After successfully selecting the tuning parameters, we produce predictions of the model on the held-out testing sample data which did not previously enter into the process of estimation of hyperparameter tuning.

In time series modelling, the analyst generally has three basic options when it comes to sample splitting. The first and most straightforward way of splitting the sample is using the fixed sample splitting scheme. Under the fixed sample splitting scheme, the whole dataset is divided into three disjoint subsamples, while respecting the temporal ordering of the data. The first subsample is the training subsample and the data from it are fed into an algorithm for estimating the inner parameters of the model. Based on the predictions on the second, i.e. validation subsample, the hyperparameters of the given algorithm are optimized. Lastly, predictions on the testing subsample are used to evaluate the model's out-of-sample performance. The reason why we cannot evaluate the performance on the validation subsample as well is that the data are not truly OOS, despite the fact that they are not directly involved in finding the inner parameters of the model. They still have to be considered IS, because they influence the choice of hyperparameters and therefore indirectly alter the resulting inner parameters estimated on the training subsample.

the dimensions and scope of the hyperparameter space of the lower-level hyperparameter tuning procedure.

The fixed sample splitting scheme has the advantage of providing space only for one round of estimation and hyperparameter tuning, and hence it is usually fast to employ it. However, the fixed scheme also comes with a few shortcomings. The first is that it is static, and because enough data must be used in training and validation, there sometimes is not many observations dedicated for OOS testing. We may overcome such an issue by limiting the amount of data in the training and validation samples, but this only leads to another drawback. When restricting the training and validation samples and extending the number of observations devoted to testing, we generate a situation in which the last points in our testing sample are located too far away from the last points used in estimation. Developing a model which is able forecast values many periods ahead requires the underlying relationship between the variables of interest to be very strong and stable.

An alternative to the fixed sample splitting scheme is the recursive scheme. Under the recursive scheme, we first choose the lengths of the first training period, the validation period, and the testing period. If we choose the sum of these lengths equal to the total size of the dataset, we get the fixed scheme described above as a special case of the recursive sample splitting scheme. But we rather choose the lengths in such way that some part of the available data is held aside for later use. We proceed to train and tune our models as usual, and then use the best performing model to produce OOS forecasts on the prespecified testing period. However, as opposed to the fixed scheme, the work does not end here. In the next step, the training sample window size is increased by the length of the testing period, so that it covers more data than in the previous phase. Both validation and testing windows are shifted forward by the same period. The familiar procedure of training, tuning, and testing our model is then applied again. In the next iterations, we continue to repeat all these steps and gradually increase the sizes of training samples, while keeping the sizes of the validation and testing windows constant.

The recursive scheme can hence utilize the same dataset, but compared to the fixed sample splitting scheme it does not suffer from too long forecast horizons as the training, validation and testing samples can be rolled forward until they reach the end of the whole dataset. A disadvantage of using the recursive scheme is that depending on the testing sample size, the number of iterations can make the computations run for a long time.

The last category of sample splitting schemes which we cover here is the rolling scheme. Similarly to the recursive scheme, the utilization of the rolling

scheme involves shifting the training, validation and testing sample forward by the length of the testing period. The only exception is that this time, the training sample does not necessarily contain the oldest observations in the dataset. Instead, every time the windows are rolled ahead, the last observations in the previous training sample are dropped and so the size of the training sample remains fixed. This way, we can exploit the most recent observations prior to prediction, but the computations can become very expensive as well.

Limiting the size of data used for training and validation via the choice of a sample splitting scheme is a mechanism for analyzing the performance of the newly introduced machine learning methods under adverse conditions while also allowing to capture the potential nonlinear and time-varying patterns in the data.

Finally, let us note that in order to enhance the performance of our methods, the data is standardized during every iteration over the given fixed, recursive, or rolling sample splitting scheme employed, i.e. in the first instance of the scheme and then each time the samples are rolled forward.[4] The standardization is performed in a usual way, described in Equation 4.1 below.

The sample splitting schemes presented here do not encompass all alternatives available for the researcher. There exist many variations, modifications and mutations of the above mentioned schemes, but for the sake of brevity, we only describe those which are employed in our work. Now we briefly describe the central predictive model of our analysis.

3.3 General Predictive Asset Pricing Model

In general, the predictive models which we will examine in this thesis take the following form

$$r_{i,t} = f(x_{i,t-1}) + \epsilon_{i,t} \tag{3.1}$$

where r stands for the excess return on the given asset, f is a function of predictor variables which are represented by a $P+1$-dimensional vector x, P is therefore the number of factors employed in the model, the remaining dimension of x is reserved for the intercept, ϵ is the error term, $i \in 1, ..., N$ indexes stocks (N is the total number of stocks) and $t \in 2, ..., T$ is an index of time (T is the total number of periods in the dataset).[5] The specific parameters of the

[4] There is only one iteration under the fixed sample splitting scheme.

[5] In our case, t is an index of months.

function f are estimated using various algorithms depending on the chosen estimation method. Notice that the factor values are lagged by one period, this is what creates the predictive nature of the model. The structure of the function f is usually assumed to be linear (see Section 2.1 and Section 2.2 of Chapter 2 for examples) as in the following equation

$$r_{i,t} = \theta * x'_{i,t-1} + \epsilon_{i,t} \tag{3.2}$$

where apart from the familiar variables, θ represents the parameters of the function f from Equation 3.1, i.e. the inner parameters of the model. Models represented by Equation 3.2 are then often estimated via Ordinary Least Squares (OLS). While some of the results of these linear approaches to asset pricing are of high quality, many researchers, e.g. Gu *et al.* (2018), have expressed their doubts about the linear nature of the estimated risk-return relationship. To allow for the potential nonlinearities, different methods than OLS must be utilized. We present some of these techniques in the following section.

3.4 Review of Machine Learning Methods

In what follows, we provide a short overview of machine learning methods which comprise the building blocks of our analysis. We will be concise by purpose, because most of the details and technicalities have been described many times in various scientific materials. For more information about the methods, consult a similar review in Gu *et al.* (2018) who provide a handy overview of several machine learning techniques, focusing on describing the modelling equations, objective functions, algorithms that efficiently solve them, and algorithms that also efficiently search for optimal hyperparameters. Alternatively, Hastie *et al.* (2009) provides a well-written textbook containing more elaborate definitions, explanations and examples.

3.4.1 Ordinary Least Squares

The first models we will encounter in our analysis will be linear models linking the excess stock returns to lagged factors. We begin the introduction of these methods with Ordinary Least Squares. OLS regression is a well-known method for quantifying linear relationships. By regressing the dependent variable on

the independent variables we obtain the model parameters – these are often referred to as betas. For the purpose of asset return prediction, our model has the same specification as the one provided in Equation 3.2. OLS algorithm has an *L2*-type objective function specified below

$$S(\theta) = \sum_{i=1}^{N} \sum_{t=2}^{T} (r_{i,t} - \theta * x'_{i,t-1})^2 \tag{3.3}$$

, i.e. the OLS assigns values to the regression coefficient vector θ such that the sum of squared residuals of the model is minimized.

A disadvantage of this model is that it simply is not able to capture higher levels of complexity in the relationships between the variables of interest. Another practical drawback of this method is that without regularization, the method is prone to overfit, especially if the number of independent variables is high. In what follows, we introduce a handful of methods for estimating linear relationships, which also provide some degree of protection against fitting the noise in the data.

3.4.2 Principal Components Regression

The Principal Components Regression is a well-known technique which combines the so-called Principal Components Analysis (PCA) and OLS regression. It is applied in two steps. First, the PCA is used to construct linear combinations of the predictor variables in such a way that they describe the most variability in the set of all predictor variables. Hence, it is able to condense a wide set of independent variables into several artificial factors which still retain most of the variability of the original data. The most important principal components are then used as regressors in a usual regression problem.

The key hyperparameter here is the number of components on which the response variable is regressed. It can be tuned using the standard training-validation procedure described above. The advantage of this method is that because the number of predictors can decrease substantially, the second-step regression should be less likely to produce an overfit. However, the choice of particular principal components may sometimes not be very fortunate, because the variable which is ultimately predicted does not play any role in the process. As we will see, also the following Partial Least Squares (PLS) method is based on a similar idea, but attempts to overcome this issue.

3.4.3 Partial Least Squares

Another popular dimension reduction technique is the Partial Least Squares. Similarly to the Principal Components Regression (PCR) described above, PLS combines the predictor variables into components and offers the researcher the option to use a chosen number of these components as independent variables in a regression estimated via OLS. Put simply, it runs simple linear regressions of the response variable on the predictors, taking the variables one by one. Then, coefficients from these regression are used to construct weights of the resulting components. Like with the PCR, the dependent variable can then be regressed on several of the constructed components and the number of components used is a possible tuning parameter for this method. As mentioned above, further details about this algorithm can be found e.g. in Gu *et al.* (2018).

3.4.4 Elastic Net

As we have established, OLS is highly likely to overfit our models, if we they include too many predictors. While PCR and PLS aim to reduce the dimension of the predictor variable set, so that only few components can be entered into the regression, there also exist a branch of penalization techniques which aim at reducing the complexity of the model through adjusting the objective function of the estimation method. The adjustment is pronounced via an inclusion of a so-called penalization term. The degree of penalization is dependent on the degree of the model's complexity. Parsimonious specifications are penalized less, while complex structures are penalized more. A popular choice of the penalization is the Elastic Net (EN) penalty which takes the following form

$$\lambda * (1 - \alpha) * \sum_{p=1}^{P} |\theta_p| + \lambda * \alpha * \sum_{p=1}^{P} \theta_p^2 \tag{3.4}$$

where $\lambda \leqslant 0$ and $\alpha \subset< 0, 1 >$ are hyperparameters of the EN method. They control the form of penalization and can be tuned using the approach described in Section 3.1. This penalty is simply added to the objective function described in Equation 3.4 and the coefficients are again chosen so as to minimize this function.

Least Absolute Shrinkage and Selection Operator (LASSO) and ridge regression are two popular penalization techniques which are also corner solutions of the hyperparameter optimization described above. When we choose α equal to

0, we obtain an *L1*-type LASSO penalty. Likewise, if α takes on the value of 1, the elastic net becomes an *L2*-type ridge penalty. The consequence of using EN is that some factor coefficients are set exactly to zero, which is a feature of the LASSO penalty, or at least shrunken towards zero, which is an effect that intensifies when the structure of the penalization term approaches the pure ridge penalty.

A potential drawback of using EN is that it can produce poor results when it is forced to select between variables showing strong mutual correlation. Of course, this holds for LASSO and ridge regression as well. In the following sections, we describe methods that do are not estimated with prior the assumption of linearity between the dependent variable and the predictors or predictor components.

3.4.5 Random Forest

Random Forest and also the next two algorithms introduced at the end of this section are modelling techniques which allow the functional form of the model specified in Equation 3.1 to be nonlinear. The methods also cover potential interactions among the predictors. Applying a Random Forest begins by growing a single regression tree. The regression tree is a supervised nonparametric method which can be also used for classification problems.

A single tree is build by making a series of decisions on how to split the given data into two parts. Each splitting decision considers only one variable, i.e. we are looking for an optimal combination of a value $c_{p,k}$ and variable p in the split k which divides the set of observations into those with p higher than $c_{p,k}$ and lower than $c_{p,k}$. Making *k - 1* of such decisions results in k terminal partitions of observations. Before making the splits, it is also important to determine the maximum depth of our tree. The full model specification under a regression tree can thus be written as

$$r_{i,t} = \sum_{k=1}^{K} 1_{\{x_{i,t-1} \in L_k\}} * \bar{r_k} + \epsilon_{i,t} \tag{3.5}$$

where K denotes the number of terminal nodes of a tree, i.e. the number of subsamples in which the whole training dataset is divided, 1 is a standard indicator function equal to one if the condition in the lower index is satisfied

and zero if not, L_k is the set of observations in the k^{th} terminal node, $\bar{r_k}$ is the average of response variable values falling into the k^{th} terminal partition.[6]

The decisions on splitting the data into partitions are made so that the impurity which is often an *L2*-type measure computed equivalently as the Mean Squared Error (MSE), i.e. by taking the mean squared difference between the realized and average values of the response variable r, is minimized. Let us note that when making predictions based on regression trees, we must always use the average dependent variable values from the training set.

Growing a whole random forest utilizes a process often referred to as bagging which is a method intended to make the final predictions more stable. The bagging routine involves bootstrapping, i.e. drawing [X] different subsamples of our data. In the next step, we fit a regression tree to each of these subsamples, and finally we average the forecasts from all of these regression trees. To secure that the individual tree structures are not alike, which could distort the positive effects of model averaging, we propose that only a randomly chosen subset of predictors having a prespecified size is selected prior to each splitting decision. This way, factors which would otherwise control most of the behavior of a tree, are forced to have lower influence in the final tree structure.

Single regression trees have several advantages, they are easy to interpret, they can also be used for classification purposes, they are fairly stable and can well approximate complex patterns in the data. However, the splitting decisions are myopic, because they are performed sequentially, and they can also easily lead to overfit, which requires every analyst to apply them carefully. Also, they produce forecasts which are not tailored to individual observations, but rather to a whole group of them. Next, we present the reader to another related tree-based technique.

3.4.6 Gradient Boosted Regression Tree

Gradient Boosted Regression Tree (GBRT), similarly to Random Forest (RF), is a very computationally intensive method, since the process of building a single boosted tree also requires building many individual regression trees. It is also nonparametric and supervised, but its structure is rather different from RF. It

[6]The tree structure doesn't have to be this simple, Breiman (2017) or Hastie *et al.* (2009) provide applications of more sophisticated tree methods. We avoid using these because our intention is to provide a general comparison across methods which posses divergent features on a basic level without delving deeply into the peculiarities of each of these modelling approaches.

too relies on building a series of regression trees, but rather than fitting them in a parallel way, the algorithm does that sequentially via a technique know as boosting. Boosting begins with fitting a common regression tree, this tree should be rather shallow in order for it not to fit the noise in the data. The residuals from the shallow tree, i.e. the differences between the realized and fitted values, are then used as a response variable and are fitted via another shallow tree. The algorithm then continues to iteratively fit residuals left out by previous regression trees by the same set of predictors.

Finally, when the maximum allowed number of trees in the GBRT model is reached, the predictions of the individual regression trees are added together. However, each of them is assigned a weight from the interval $(0, 1)$, so that the highest relevance is given to the first tree which predicts the actual response variable, while the other trees in the sequence are given a lower weight because the further we move from the first iteration, the further the trees are likely to fit noise instead of estimating the true underlying relationship in the data. The resulting predictions of boosted trees can be very accurate in-sample, but the researcher must make sure that this is due to fitting the signal rather than noise.

3.4.7 Neural Network

Neural Networks (NNs) represent another alternative among methods suitable for uncovering the potential nonlinear patterns in asset pricing. NNs are heavily parametrized estimation techniques which attempt to imitate the biological processes occuring in the human brain, hence the name Neural Network. The NNs became to be developed by Rosenblatt (1958) who also helped develop the first computer able to utilize NNs for learning.

Among the simplest and most popular artificial neural network architectures are the the so-called feed-forward NNs. A feed-forward neural network consists of neurons, the principal computation units of the network, organized in a series of layers. The first layer of neurons, or nodes, is the input layer. The input layer contains several neurons, each corresponding to one of the predictor variables. What follows are the hidden layer of a NN. Each hidden layer receives values from the previous layer, assigns each value with a weight, and passes the weighted sum of values along with an intercept into an activation function. The activation function of each neuron returns a single value which is then transferred into all neurons in the subsequent layer. Neurons in the subsequent

layer again weight the information the receive, and pass on the values of their activation functions to the next layer in the sequence. The last layer is the output layer. The output layer weights the values from the previous layer, but instead of feeding them into an activation function, it returns their weighted sum as the final fitted value. [7] An important feature of the feed-forward NNs is, as the name suggests, that the connections between individual layers (be it any of the input, hidden or output layers) have always the same direction. Based of their complicated design, it is clear that a big disadvantage of NNs is that they posses a low level of transparency. Because the structure of a Neural Network can be quite complex, we follow with a brief discussion of its individual elements.

Let us begin with the building stones of each NN, the individual nodes. The nodes are also often referred to as neurons or perceptrons. A single perceptron can itself be thought of as small NN, consisting only of the input layer and the output layer. And from another point of view, the whole Neural Network can be considered as a multilayer perceptron.

Another feature of a common NN is the intercept, i.e. constant term, pertinent to the input layer and all hidden layers of the network. Imagine the constant term in the aggregation step between individual subsequent layers A and B as being a weight of an additional node in the layer A whose activation function returns the value 1 for any possible input which is passed to it. Constant terms are also often referred to as bias nodes.

The activation function must be nonlinear, otherwise the whole artificial NN becomes a plain linear model with very high costs of estimation. The output layer usually has no activation function, however, we can also perceive it as a layer with an activation function in the form of identity.

Finally, let us mention that, Neural Network have been enhanced by researchers with multiple distinctive features in the past, creating various special types of NNs. Examples of these are multiple output nodes in the output layer, convolutional networks, or support vector machines. But even the simplest NN architectures which we consider here are in fact very complex structures. They also contain several regularization characteristics, the particular choices of which will be described in Subsection 5.2.7 below.

After establishing the key concepts in the theoretical background of this

[7] An interesting property of this architecture is that an ordinary linear regression model, which can be estimated via Ordinary Least Squares, can equivalently be estimated by a Neural Network with the simplest design, i.e. a NN with only the input and output layers, but without any hidden layers in between.

research topic, let us move to the introduction of the empirical framework. First, Chapter 4 describes the features of our dataset. Then, Chapter 5 presents the elements of our methodology, such as the specific sample splitting schemes used, or details about implementations of the given methods.

Chapter 4

Data

In this chapter, we describe the dataset which is later used in estimation. Generally, the problem of asset pricing is a problem of finding linkages between various stock-specific or macroeconomic variables, commonly called factors, and the asset prices or, equivalently (in this context, of course), asset returns. Hence we divide this chapter into two subsections, Section 4.1 contains description of data on the response variable, i.e. stock returns and Section 4.2 is dedicated to details about macroeconomic variables used in the analysis.

Before we move to the characteristics of our data, let us make several important notes. First, our dataset could be considered small, compared to datasets some of the other researchers, e.g. Gu *et al.* (2018) or Feng *et al.* (2019), work with in the field of empirical asset pricing. But this is not necessarily a bad think. We can think of our dataset as one which does not differ much from a typical dataset used by an average financial analyst or economist. Not every practitioner or academician works with enormous datasets gathering of which can be very costly in terms of time, effort, or money. Hence we perceive our average-sized data as a good approximation of the reality of other researchers or entities conducting analyses with financial data and also as a challenge for the recently introduced machine learning methods which are often applied on richer datasets.

Second, we never use our data in their raw form without standardization, not even for the simplest techniques for which the standardization would not make a big difference. Instead, we standardize our data during every iteration of every sample splitting scheme that we use. In other words, we perform standardization on the training and validation data on the first instance of the given sample splitting scheme and then each time the samples are rolled

forward.[1] Before the predictions are made by any model, the testing data is transformed as well via the same standardization parameters, i.e. using the same mean and standard deviation, as the respective training and validation samples. The same standardization procedure is performed under all fixed, recursive and rolling schemes employed. The standardization is carried out in a classical way, described in the following equation

$$s = \frac{x + \mu_X}{\sigma_X} \tag{4.1}$$

where x is the original raw value of the variable X, μ_X and σ_X are the average value and the standard deviation of the variable X in the subsample to be standardized, and s denotes the standardized value.

Third, data on all explanatory variables, i.e. factors, are shifted one period ahead, thus creating lagged values of these variables. This is important because we want to examine to predictive capacity of our models, hence, when making predictions, we always have to work with data that would actually be available at the time of our prediction. Finding linkages of contemporaneous nature between factors and excess returns is an interesting task, but it does not tell us anything about the predictive capabilities of our models.

Fourth, we add lagged excess returns to our dataset. Let us note that the lagged returns were one of the most important factors in the analysis of Gu *et al.* (2018). We further add interactions of this variable with all the other macroeconomic factors to our set of predictors. Squared value of the lagged excess stock return is added as well, making the total predictor count equal to 40. This setting was selected based on the Principal Components Analysis (PCA) performed on the set of all possible interactions amond our original variables.

Last, during computations, missing values are skipped rather than being given any special treatment. However, since the data are quite clean, missing observations do not pose a big problem. Now we can move to describing the variables we employ in estimation, we begin with the time series of asset returns.

4.1 Monthly Stock Returns

Stock returns comprise the first important part of our dataset. We managed to gather a moderate-sized dataset of monthly asset returns computed as simple

[1] There is only one iteration under the fixed sample splitting scheme.

returns from month-end closing prices accessed and downloaded via Refinitiv Eikon.[2] After computing the simple monthly returns, the risk-free rate, enclosed with the five-factor dataset of Fama and French, was used to compute the monthly excess returns for each stock and month in our dataset.

The excess return data span from December 1997 to November 2017, covering exactly 20 years, i.e. 240 months, of time. The dataset contains data on 3219 stocks traded on the largest exchanges in the United States of America, these are National Association of Securities Dealers Automated Quotations (NASDAQ), New York Stock Exchange (NYSE) and Better Alternative Trading System (BATS). The average count of monthly observations exceeds 1833, i.e. there are, on average, more than 1833 stock returns available for every month in our dataset. The total amount of observations hence amounts to over 440 thousand.

4.2 Macroeconomic Factors

The second essential part of our dataset consists of macro-level factors. These factors were downloaded from two different sources. The first is the notoriously known Fama and French five-factor dataset collected from Kenneth R. French's website (French 2019) which was used in Fama & French (2015). The second group of factors comes from the personal webpage of Amit Goyal (Goyal 2019) who gathered valuable data more than 15 variables commonly used as factors in the asset pricing literature together with Ivo Welch and used it to examine their empirical performance in asset return prediction (Welch & Goyal 2007). Datasets from both sources are frequently updated, hence we were able to accumulate data extending up to December 2017.

What follows is a list of the individual factors used and their brief description. We begin by the Fama and French factors, the details about the construction of these factors can be found in Fama & French (2015) or French (2019). Put simply, the last four of these factors are essentially returns provided by portfolios diversified via several selected features, while the first one is an excess return on a non-diversified market-wide stock portfolio. We must note that these factors are usually used to explain excess returns from the same time period. However, due to their extensive use in the literature, we

[2]The author also considered considered using logarithmic returns, but the values, predictably, did not differ enough to make any notable qualitative or quantitative differences in our results.

find those factors to be qualified for testing their predictive abilities and even if they eventually only add noise to our dataset, the advanced methods presented in Chapter 3 should be capable to handle these situations.

Market Excess Return Computed as the difference between the average return of a market portfolio and the 1-month Treasury bill rate.

Small Minus Big Computed as the difference between the average returns earned on a group of small stock portfolios and a the average return earned on a group of big stock portfolios. Whether a stock is considered small or big is governed by its market capitalization.

High Minus Low Computed as the difference between the average returns earned on a pair of value stock portfolios and a the average return earned on a pair of growth stock portfolios. The Book to Market Ratio (B/M) is used to distinguish between value (high B/M) and growth (low B/M) stocks.

Robust Minus Weak Computed as the difference between the average returns earned on a pair of robust stock portfolios and a the average return earned on a pair of weak stock portfolios. We distinguish between robust and weak stocks based on a measure of their operating profitability.

Conservative Minus Aggressive Computed as the difference between the average returns earned on a pair of conservative stock portfolios and a the average return earned on a pair of aggressive stock portfolios. The aggressivity of a stock is determined by how much does the given company invests.

To complete our portfolio of factors, let us briefly present factors provided by Goyal and Welch. Given that they were collected based on their proclaimed success in many different papers, these factors come from a variety of areas in finance and thus form a quite miscellaneous group. But generally speaking, they represent macroeconomic variables related either to the stock market, the debt market, or to consumer prices . See Welch & Goyal (2007) for details. Many of these factors are closely related to those introduced above, but because of the differences in how these factors are build and hence their actual realizations differ, we include all of these factors in our analysis, regardless of their similar nature.

Dividend/Price Ratio Computed as the difference between the natural logarithm of a yearly sum of dividends paid on the Standard & Poor's 500 (S&P 500) index and the natural logarithm of the index.

Dividend Yield Computed as the difference between the logarithm of the S&P 500 dividends and the logarithm of the lagged index values.

Earnings/Price Ratio Computed as the difference between the log of a yearly sum of earnings of companies in the S&P 500 index and the log of the index.

Dividend Payout Ratio Computed as the difference between the logarithm of the S&P 500 dividends and the logarithm of the S&P 500 earnings.

Stock Variance Computed as the monthly realized volatility (sum of squared returns made each day of the month on the S&P 500 index).

Book to Market Ratio Computed as the ratio of the book value of companies in the Dow Jones Industrial Average (DJIA) index to their market value.

Net Equity Expansion Computed as the ratio of the yearly sum of net equity issued on the NYSE and the market capitalization of stocks traded on NYSE.

Treasury Bill Rate Simply the rates on the 3-month securities issued by the United States Treasury.

Long-term Government Bond Yield Yields on the long-term bonds issued by the government of the United States.

Long-term Government Bond Return Returns on the long-term bonds issued by the government of the United States.

Term Spread Computed as the difference between the long-term government bond yields and the Treasury bill rate.

Default Yield Spread Computed as the difference between the corporate bond yields rated BAA and the corporate bond yields rated AAA.

Default Return Spread Computed as the difference between the long-term corporate bond returns and the long-term government bond returns.

Inflation Rate Reflects changes in the Consumer Price Index.

Now we can move to the next chapter which introduces the reader to our empirical framework.

Chapter 5

Methodology

In general, the methodology of Gu *et al.* (2018) will be followed to study the performance of several newly developed machine learning techniques in comparison with those that were traditionally used in the past. To increase the relevance and validity of the comparisons, all methods will be compared to a benchmark model represented by a linear regression of excess returns on the five factors comprising the well-known Fama and French five-factor model (Fama & French 2015). We will use the Out Of Sample monthly return prediction R^2 developed by Gu *et al.* (2018) to decide which approach performs the best.

The data that we employ in our analysis were described in Chapter 4. We have gathered a dataset consisting of 19 unique macroeconomic factors[1], these are used to explain the individual excess stock returns in our analysis. Moreover, the predictor set is extended for the lagged values of the dependent variable, squares of these values, and another 19 interactions of this variable with the macroeconomic predictors. All variables enter the models in the form of lags, we consider only values lagged one period behind, higher lag orders are ignored. While our dataset cannot be considered large with respect to other ones used in the literature, we see this as an advantage. The lower relative size of our dataset allows us to expose the advanced methods to unfavorable conditions in which we can test their performance.

We adopt a total of 11 sample splitting schemes for our research, details about them can be found in Section 5.1. In general, we attempt to achieve a high level of diversity among the schemes, hence we chose then in such way that each of them possesses unique features with respect to the amount of data

[1]While the five Fama and French factors included were not originally used to make forecasts of asset returns (Fama & French 2015), we feel that it is appropriate to add them among our regressors because of the proclaimed contribution.

used in training, validation, or testing. This reminds us to remind the reader of another important aspect of our estimation procedures, the act of standardizing training and validation data each time we roll the samples forward. Testing data are transformed using the same mean value and sample standard deviation before making predictions. Also, no missing values in the data are imputed, all of them are skipped instead. We restrain from imputing missing values because the excess return data is well known to contain a lot of noise and hence we consider imputing missing values not to be appropriate.

The collection of machine learning techniques applied in our research was introduced in Section 3.4. Further implementation details are explained in Section 5.2 below. We suggest the reader to address Section 3.4 of Chapter 3 or Gu *et al.* (2018) for more information about the techniques used.

In finance, the models are trained on the subsample of data devoted to training. Many methods then use a so-called validation sample to verify that the model parameters are not overfitted, i.e. that the performance of the model doesn't decrease too much when used on data it was not trained on. In case the performance on the validation sample drops substantially with respect to the performance on the optimization sample, the training process can be repeated with a different setting where some of the model's hyperparameters (do not confuse these with the inner parameters of a model such as regression coefficients or weights in neural networks) are changed in a way that it should be less simple for the model to result in an overfit. Examples of hyperparameters could be e.g. the number of variables in a regression or the depth of regression trees inside a random forest.

The example describes the use of a gradient descent algorithm for searching the best set of hyperparameter values. Instead of using this algorithm, we searching for optimal hyperparameters via a technique known as Random Grid Search (RGS). The algorithm is and adjustment of a standard grid search approach. Instead of searching the whole parameter space, it randomly selects a prespecified number of points in the space and chooses the best performing hyperparameters from this small subset. The advantage of this method is that it is much faster than the traditional grid search, while keeping a good overall performance. The procedure is discussed in Bergstra & Bengio (2012). During the search for optimal parameters, we pick the combination of hyperparameter values which minimizes the Mean Squared Error (MSE), this is a standard choice in the financial economics literature, as seen for example in Gu *et al.* (2018). The OOS predictions are then obtained by testing the model on OOS data.

Later, when presenting the results, we completely neglect the IS performance of our models, because in the field of asset-pricing, which is notoriously known for the broad presence of noise, the methods are often found to perform better in-sample than out-of-sample, and hence we find inclusion of IS results misleading for the reader.

Below we present further details about the individual methods we consider in our comparative analysis. The section devoted to OLS is short, because OLS is the simplest method used and it does not employ any regularization. Other methods are accompanied by a description of related regularization techniques. Most of these methods' hyperparameters are optimized, which is a computationally expensive process and it takes a lot of time. Because fitting artificial neural networks is the most computationally demanding technique that we use, we avoid searching a hyperparameter space to find optimal NN setting. Similarly to Gu *et al.* (2018), we instead design several neural network architectures and directly fit the prespecified models. We consider this as a reasonable approximation of the lengthy hyperparameter tuning process that would otherwise have to take place. The particular NN specifications that we use are described in below.

The next section contains the description of our choice of sample splitting schemes. Nextly, details about how the individual estimation techniques are depicted in Section 5.2. Section 5.3 concludes with a brief description of how we compare the performance of the individual methods.

5.1 Sample Splitting Schemes

As mentioned in Section 3.2, we generally have 3 options when it comes to choosing the appropriate sample splitting scheme – these are the Fixed Sample Splitting Scheme (FSSS), Recursive Sample Splitting Scheme (ReSSS), and Rolling Sample Splitting Scheme (RoSSS). Under each sample splitting scheme, the lengths of training sample windows in months were chosen to avoid situations where there was a low amount of observations with respect to the number of predictors. This was, however, not difficult because the total number of observations available amounts to more than 440,000. Nevertheless, the author experimented with training the models on very small samples, starting at 1 year of training data, but the results proved to be very poor and training sample sizes of less than 60 months were not included in the analysis.

When constructing all recursive and rolling sample splitting scheme, we

always use the same fixed amount of 12 months in the testing window. The one-year period was chosen in order to achieve computing efficiency at reasonable costs of prediction performance. We could simply use testing windows of size 1 month, i.e. always predict only one period ahead, but this approach would lead to long-running computations with what we believe is only a small potential increase in the precision of predictions. We could also set longer testing windows, the length of which could be e.g. 3 years, but that would probably create conditions too harsh for our methods which could not use data from the recent years to make predictions. Hence we have settled on a fixed 12 period size for the testing windows.

The situation is different when considering the fixed schemes, where the testing sample size directly determines how much data is left for training and validation. With the fixed scheme, the analyst faces the difficulty of predicting many periods ahead, without giving the recent observations the chance to influence the last predictions in the sample. We believe that this might be an issue especially in the problem of asset pricing, where finding stable and long-lasting patterns in the data proves to be a challenge.

Now let us describe the particular choices we made about the structure of splitting schemes used in our examination. Since we use a total of 11 schemes, we simplify the notation by abbreviating each of them. When we refer to a particular scheme, we provide the lengths of the training, validation, and testing windows in parentheses. For example, the following scheme is denoted by FSSS(5, 5, 10). We consider fixed schemes under two distinct settings, the first scheme utilizes 5 years of training data, uses the next 5 years for validation, and tests the results on the remaining 10-year period. Because the conditions are not very favorable for the methods under the previous scheme, we construct FSSS(10, 5, 5) as an alternative fixed scheme.

The recursive splitting schemes are chosen in the following specifications – ReSSS(5, 1, 1), ReSSS(5, 3, 1) and ReSSS(5, 5, 1). The first number in the brackets denotes the length of the initial training sample. We gradually increase the validation window sizes up to five years, increasing the demands on the validation sample performance for the training sample fits. There is no need to increase the initial training sample size, because the nature of recursive schemes makes it proceed such specifications anyway. We can hence simply start with 5-year training samples and add more data each time when the validation and testing windows move forward. If necessary, we could simulate the use of a 10-year starting training window by simply neglecting the first

5 years of predictions of the corresponding scheme with just a 5 years long starting training window.

The first 3 specifications for the rolling schemes – RoSSS(5, 1, 1), RoSSS(5, 3, 1) and RoSSS(5, 5, 1) – include five-year training window sizes, and the lengths of the validation windows are gradually increased from one to five years, just as with the recursive schemes. To give the methods more observations for training, we also consider the RoSSS(10, 1, 1), RoSSS(10, 3, 1) and RoSSS(10, 5, 1) sample splitting schemes.

Our choice of window sizes for all the sample splitting schemes exposes the methods to various circumstances. For example, the FSSS(5, 5, 10) scheme provides our methods with only 60 months of data for training, while the techniques enjoy a training sample the length of which totals 18 years under ReSSS(5, 1, 1).

5.2 Implementation of Machine Learning Methods

This section is devoted to the description of the details regarding our application of the methods presented in Section 3.4 of Chapter 3. We now briefly add some general notes on our methodology. First, whenever it is applicable, we always choose to fit an intercept when using our methods. This does not concern the tree-based methods. For optimizing the hyperparameters of our models, we utilize the random grid search algorithm during which we search for hyperparameter values which minimize the MSE on the validation sample. Many various hyperparameters were mentioned during the introduction of the given methods in Chapter 3. However, these are not the only hyperparameters we can think of. For example, the binary choice of either including an intercept to a regression or not including it is itself a hyperparameter. The minimum impurity decrease required for a node to be split in a regression tree or the choice of activation function in the NN architecture are other examples. Ultimately, every little detail which has the power to influence the resulting overall specification of the model even to a minimal extent can be considered a hyperparameter. Therefore, our specific estimation techniques implementations could actually be extended by dozens of their variations. Consider our portfolio of methods only as a representative sample of the whole universe of advanced machine learning estimation algorithms.

The implementation details of our approach follow.[2]

5.2.1 Ordinary Least Squares

We begin the overview of implementation details with Ordinary Least Squares. The OLS regression is a commonly applied method in the economic literature. In its simple form, it contains an important drawback, because it does not allow the analyst to apply it with any form of regularization. We utilize this technique in two ways. First, an OLS model using only the five factors used in Fama & French (2015) will be taken as a benchmark for the purpose of comparison to the performance of the other methods. Second, we will regress the excess returns on the whole group of 40 independent variables in our dataset, including the Fama and French 5 benchmark factors.

5.2.2 Elastic Net

Our implementation of the Elastic Net (EN) method is straightforward. The models are fitted on training samples and then we use the validation samples to select the best performing combinations of values of the two EN tuning parameters described in Subsection 3.4.4 of Chapter 3, namely α and λ which determine the form of penalization used. The hyperparameter spaces considered for EN are $< 0, 1 >$ for the parameter α and $< 0.0001, 0.1 >$ for λ.

5.2.3 Principal Components Regression

The Principal Components Regression (PCR) is applied as a series of two distinct steps. First is the choice of principal components based on the values of predictor variables. The components which explain the most amount of variance among the predictor data are then fed into a standard OLS regression. The hyperparameter which we want to optimize here is the number of the leading components used, i.e. the number of independent variables entering the

[2]During the conduct of our research, we considered an adapted version of the Fama-Macbeth regression technique. The original method is not build to provide predictions of asset prices, it rather allows to describe the contemporaneous relationship between the asset prices and the betas. Fama-Macbeth regressions have two steps, each of these steps consists of estimating a linear regression model via OLS, however, the overall nature of the technique is non-linear, a description of the method can be found in Fama & MacBeth (1973). In the second step of the adjusted Fama-Macbeth procedure, we took lagged values of betas as independent variables instead of using their contemporaneous values. The predictions from this model were however very inaccurate. Because the method was not developed for prediction purposes, we decided not to include it in our portfolio of methods.

regression equation. The hyperparameter space for this parameter is simply any natural number between 1 and the total number of the original predictor variables, i.e. 40.

5.2.4 Partial Least Squares

Similarly to PCR, the Partial Least Squares (PLS) regression method utilizes two steps. It first reduces the dimension of the set of predictor variables by constructing components which keep the highest degree of covariation between the original predictors and the response variable. What follows is a regression of excess returns on several of these components. The tuning parameter entering the estimation process is the number of components used in the regression in the second stage. Equivalently to the case of PCR, we consider any natural number between 1 and 40 as a potential value for the optimized hyperparameter.

5.2.5 Random Forest

A brief introduction to the RF algorithm and the underlying regression tree method was provided in Subsection 3.4.5. In our implementation of Random Forest, we use the *L2* impurity as the splitting criterion. Many regularization techniques are utilized in the process. Before building each regression tree, the observations are bootstrapped and only a subset of the full amount of datapoints is used in estimation. When building the individual trees, we control for their depth, which is a hyperparameter allowed to take integer values between 1 and 4. We also regulate the number of variables randomly sampled before each splitting decision to be between 3 and 40. Finally, the number of trees in a forest can either be 100, or 200. All hyperparameter spaces were carefully selected with consideration of their influence on the results, but also with respect to the computing capacity their application requires.

5.2.6 Gradient Boosted Regression Tree

The Gradient Boosted Regression Tree algorithm utilizes the fitting of a series of regression trees, which makes it closely related to the RF algorithm described above. The distinction between the two methods is that while RF uses a bagging procedure to construct the final specification of the model, GBRT utilizes the technique of boosting, which starts by fitting a simple regression tree, and than fits the residuals from the first tree using the same set of predictor variables.

The predictions made by the second tree are added to the predictions of the first tree shrunken by the so-called learning rate, which is a hyperparameter and we consider it to be equal either to 0.01 or 0.1. The procedure then continues by fitting residuals from the second model by another tree, etc. Regression trees in the model have a predefined depth, we consider shallow trees of depth 1 or 2. The total number of trees in the overall model is another tuning parameter, we estimate GBRTs composed of 1 to 100 regression trees.

5.2.7 Neural Network

In our application of the NN estimation technique which is described in **??**, there are many implementation details we have to consider. The most important is the overall architecture of the applied NNs, i.e. the number of layers which determine the depth of the network and the number of neurons in each layer. We find inspiration in Gu *et al.* (2018) and construct the following four different NN structures. The first is Neural Network with a single hidden layer consisting of 8 neurons, the second structure has two hidden layers containing 8 and 6 nodes, respectively. Third architecture extends the depth by one additional layer with 4 neurons. The last NN design we consider contains four layers, gradually decreasing the number of neurons in the following order –8, 6, 4, and 2.

We employ an activation function in the form of a Rectified Linear Unit (ReLU). The ReLU is equivalent to an identity function if its input is nonnegative and returns zero otherwise. For finding optimal values in the complicated structure of parameter, we utilize the 'Adam' solver introduced in Kingma & Ba (2014). The Adam's learning rate regularization parameter is set 0.001. Batch normalization introduced in Ioffe & Szegedy (2015) is used to improve the performance and stability of our NN predictions, the particular batch size we choose is 1000 observations. We also employ a technique called early stopping. If the prediction errors do not drop by at least 0.1% during the last 5 iterations of the solving algorithm, the estimation is stopped. Finally we always estimate the network 10 times and average the results to get the ultimate predictions. Because fitting a whole Neural Network is of random nature, each of the estimates is different and hence the model averaging makes sense.

The author also tried to apply dozens of other NN architectures, but their performance tended to be rather worse. For the sake of brevity, only a repre-

sentative selection of the relatively more successful NN designs is examined in this text.

5.3 Performance comparison

During the conduct of our research, we will the use the estimation techniques described in the previous section to predict asset returns. To compare the performance of individual methods with the performance of other methods, we use a simple measure of the so-called monthly Out Of Sample R^2 developed by Gu *et al.* (2018). The monthly OOS R^2 can easily provide us with a comparison of the given technique's prediction performance against alternative forecasts of zero excess returns. Not being able to beat a simple forecast of zero, which is equivalent to naive believes that the future asset price will be equal to the last observed price, essentially makes the model irrelevant to any rational financial analyst.[3]

Now we can move to Chapter 6 where the empirical results of our analysis are presented.

[3]See Section 2.8 in Gu et al. (2018) for the explanation of the choice of this metric. Roughly said, using a sum of squared returns in the denominator is better than using a sum of squared demeaned returns, because historical mean is a worse estimator of future returns than a zero constant.

Chapter 6

Results

In this chapter, we present the results obtained by applying the methods outlined in Section 3.4 of Chapter 3 and further detailed in Section 5.2 of the previous chapter. The more advanced nonlinear methods are commonly considered as being able to outperform the traditionally used linear models. But because these modern machine learning techniques are most often applied on rich data, we decided to conduct a comparative analysis to see whether these considerations hold when the new advanced techniques are challenged with a relatively small dataset.

A linear regression utilizing only the factors from the famous Fama and French five-factor model (Fama & French 2015) will be used as a benchmark model. The Fama and French 5 factors were originally used to explain excess asset returns realized in the same period, but we must use their lagged values to allow for making predictions. Although using them in a predictive fashion may be debatable, we consider their reputation as very successful factors as a qualification for being included in the analysis. It can turn out that their lags only add more noise to the underlying true risk-return relationship, but in that case, the regularization methods employed should be able to guard against these concerns.

In our study, the OOS performance is the key variable on the basis of which different approaches will be compared. We find it very important to consider Out Of Sample performance only and completely neglect the In Sample performance of our models in order not to create false impressions about how successful any of the methods actually is. Now that we have established the framework for demonstrating our results, we can move to describe the actual performance of all methods used in our analysis under all sample splitting

schemes examined.

6.1 Predictive Performance of Individual Machine Learning Methods

Table 6.1: Comparison of the Performance of Predictions on All Stocks via Monthly Out Of Sample R^2

	B	OLS	PCR	PLS	EN	GBRT	RF	NN1	NN2	NN3	NN4
FSSS(5, 5, 10)	0.009	-0.367	-0.003	-0.252	-0.030	-0.001	-0.021	-0.331	-0.404	-0.537	-0.685
FSSS(10, 5, 5)	-0.006	-0.014	-0.009	-0.005	0.004	0.006	0.005	-0.035	-0.038	-0.027	-0.033
ReSSS(5, 1, 1)	0.001	-0.201	-0.032	-0.160	-0.051	-0.005	-0.055	-0.258	-0.242	-0.213	-0.164
ReSSS(5, 3, 1)	0.000	-0.208	-0.064	-0.171	-0.051	-0.005	-0.057	-0.268	-0.245	-0.218	-0.165
ReSSS(5, 5, 1)	0.001	-0.209	-0.052	-0.174	-0.052	-0.004	-0.067	-0.275	-0.247	-0.220	-0.167
RoSSS(5, 1, 1)	-0.008	-0.352	-0.030	-0.100	-0.040	-0.030	-0.063	-0.490	-0.394	-0.274	-0.165
RoSSS(5, 3, 1)	-0.010	-0.151	-0.063	-0.120	-0.043	-0.004	-0.057	-0.247	-0.358	-0.243	-0.174
RoSSS(5, 5, 1)	-0.006	-0.208	-0.052	-0.172	-0.054	-0.003	-0.066	-0.256	-0.273	-0.200	-0.168
RoSSS(10, 1, 1)	-0.007	-0.081	-0.021	-0.041	-0.029	-0.002	-0.061	-0.162	-0.123	-0.086	-0.027
RoSSS(10, 3, 1)	-0.013	-0.010	-0.011	-0.013	0.003	0.005	0.007	-0.032	-0.044	-0.045	-0.026
RoSSS(10, 5, 1)	-0.006	-0.014	-0.006	-0.010	0.001	0.001	0.001	-0.005	-0.010	-0.013	-0.010

Values in the table represent the monthly OOS predictive R^2 computed based on the predictions made by the given estimation technique on the given sample splitting scheme. Positive values denote better performance that a naive forecast of 0 excess returns. The upper bound for the values (the case of perfectly accurate predictions) is 1. The compared estimation techniques are Benchmark (B), Ordinary Least Squares (OLS), Principal Components Regression (PCR), Partial Least Squares (PLS), Elastic Net (EN), Gradient Boosted Regression Tree (GBRT), Random Forest (RF), and four different artificial Neural Network (NN) specifications. The compared sample splitting schemes are denoted by their type – Fixed Sample Splitting Scheme (FSSS), Recursive Sample Splitting Scheme (ReSSS), or Rolling Sample Splitting Scheme (RoSSS), and the scheme's window sizes in years in the parentheses. The window sizes come in the following order: training, validation, testing. Data for all stocks in the dataset was used during estimations.

Table 6.1 presents the OOS monthly predictive R^2 statistics of our methods, please refer to Gu *et al.* (2018) for details about how they are computed. The values in the table are rounded to 3 decimal places. As can be seen, the performance of all methods is very poor.[1] The most successful of our models is clearly the benchmark model, i.e. the model including lagged factors from the Fama and French 5 factor model (Fama & French 2015) estimated via OLS. The benchmark model dominates the other models on the first Fixed Sample Splitting Scheme (FSSS) specification with the smaller training period, and also on all Recursive Sample Splitting Scheme (ReSSS) alternatives.

Under the Rolling Sample Splitting Scheme (RoSSS) modifications with

[1] Compare these results to a similar table provided in Gu *et al.* (2018) on page 26.

smaller training sample sizes and the RoSSS with only a 1-year validation period, none of the methods considered is able to beat a naive forecast of zero stock returns. The remaining RoSSS specifications, RoSSS(10, 3, 1) and RoSSS(10, 5, 1), show improved performance over RoSSS(10, 1, 1) or those RoSSSs with small training windows. We consider this as a sign of how important it is to feed the models with as much data as possible, and that the regularization is generally more successful when the validation window size increases.

An interesting observation is that all Neural Network architectures employed completely fail to produce reliable OOS forecasts.[2] This does not, however, mean that NNs are useless for asset price prediction, it can merely suggest that perhaps we did not provide them with enough data, or that the selection of factors we used was insufficient. Nevertheless, the results might be surprising due to the prominence the NNs achieved in the literature and their widely accepted qualities.

While the other advanced techniques were arguably more successful in predicting excess returns, only rarely did they obtain a positive OOS R^2. The bright exceptions share a common feature of having 10 years long training windows and at least 3 years long validation windows. Let us further note that the dimension reduction techniques, PCR and PLS never pulled their OOS R^2 measures from the negative territory.

We have also considered pairwise comparisons of the performance of the methods on all sample splitting schemes via an adjusted version of the Diebold-Mariano test (Diebold & Mariano 2002). The adjustment we applied was the same as e.g. Gu *et al.* (2018) have performed. Because of the high number of compared models, we cannot provide the results in a table format. That is however not an issue because all tests which we have considered failed to provide evidence that any method overperforms another at the 5% level of statistical significance, not even when comparing the best performing benchmark method against the NNs which perform the most poorly.

We can make several conclusions about the main results summarized in Table 6.1. Generally, all methods enjoy better performance when either the training sample size increases or similarly when we consider larger validation windows. While increasing the training sample can produce more stable results, larger validation window brings higher relevance to the optimal values

[2]Let us remind the reader that plenty of other NN specifications were considered during the conduct of this research. But only the most successful NN architectures were included for the sake of brevity.

of hyperparameters chosen during the RGSs. Furthermore, the advanced techniques considered in the analysis are not able to save the performance of the linear methods under the conditions of sample size limited in both the number of observations and the number of predictors. Because the dataset used her is relatively smaller, our findings are not directly in contrast with those of Gu *et al.* (2018) who assert that the newly developed nonlinear techniques overperform the predictions of traditional linear models. Our results instead suggest that a reasonable amount of skepticism should be held with respect to the advantages of the modern machine learning techniques and with respect to the value brought to asset pricing by various factors. Our findings also seem to agree with observations made by Welch & Goyal (2007) that many factors promoted in the literature actually perform badly as OOS predictors.

In the next section, we inspect the stability of our conclusions under some challenging circumstances.

6.2 Robustness Checks

In the sections below, we provide several robustness checks of our outcomes. When performing these checks, we are limited by our dataset and hence cannot provide complete answers as to why did the selected methods presented in the overview in Chapter 3 performed so poorly in our analysis. For example, we cannot fully examine whether the results are unsatisfactory due to the small amount of observations, or rather because of the limited number of factors used. However, we can, at minimum, provide several robustness inspections with which we can support our conclusions.

Firstly, we repeat the estimations above on a subset of the dataset which only contains stocks from the S&P 500 index. The models we use are usually more successful when applied on data for the largest, most liquid, and stable stocks on the market. We will investigate whether the machine learning methods are able to deliver better results under circumstances more favorable then in Section 6.1. In the same manner, we then attempt to help the modern methods by combining their predictions into a one big ensemble. Lastly, the time-varying performance of the methods is studied, attempting to find periods of time where these methods perform similarly to what is documented in Gu *et al.* (2018).

6.2.1 Standard & Poor's 500 Returns Prediction

As a check of robustness of the results presented in Section 6.1, we consider running the computations only on a specific subset of our data. In this section, we present the results obtained by performing the same procedure as above, i.e. using the same methods under the same sample splitting schemes and regularization techniques, only on stocks in the S&P 500 index. The S&P 500 is a group of 500 largest stocks traded in United States. Therefore, this subsample of stocks is highly selected and does not represent a random subsample of the data. Perhaps because the prices of stocks with greater market capitalization possess relative stability and enjoy high liquidity, asset pricing models considering only the largest stocks often show greater precision than those considering also small stocks. Hence, we believe that testing whether our conclusions hold if only S&P 500 stocks are included in the dataset is a suitable robustness check for our analysis.

Table 6.2: Comparison of the Performance of Predictions on the S&P 500 Stocks via Monthly Out Of Sample R^2

	B	OLS	PCR	PLS	EN	GBRT	RF	NN1	NN2	NN3	NN4
FSSS(5, 5, 10)	0.021	-1.046	0.007	-0.771	-0.088	-0.028	-0.041	-0.932	-1.373	-1.625	-1.704
FSSS(10, 5, 5)	0.015	0.016	0.026	0.011	0.033	0.037	0.035	-0.008	-0.052	-0.071	-0.021
ReSSS(5, 1, 1)	0.015	-0.618	-0.062	-0.508	-0.095	0.004	-0.096	-0.782	-1.057	-1.037	-1.023
ReSSS(5, 3, 1)	0.014	-0.669	-0.053	-0.564	-0.107	0.006	-0.085	-0.843	-1.132	-1.128	-1.092
ReSSS(5, 5, 1)	0.014	-0.727	-0.054	-0.610	-0.117	-0.022	-0.097	-0.912	-1.225	-1.235	-1.199
RoSSS(5, 1, 1)	0.009	-0.708	-0.112	-0.257	-0.069	0.000	-0.080	-0.825	-0.893	-0.699	-0.394
RoSSS(5, 3, 1)	-0.014	-0.365	-0.061	-0.429	-0.081	0.006	-0.076	-0.709	-1.118	-0.947	-0.588
RoSSS(5, 5, 1)	-0.009	-0.694	-0.058	-0.627	-0.126	-0.019	-0.098	-0.896	-1.135	-1.133	-1.166
RoSSS(10, 1, 1)	0.001	-0.259	-0.065	-0.110	-0.040	0.015	-0.094	-0.342	-0.390	-0.456	-0.117
RoSSS(10, 3, 1)	-0.003	0.016	0.024	0.012	0.029	0.016	0.030	-0.028	-0.056	-0.072	-0.035
RoSSS(10, 5, 1)	0.013	0.005	0.019	0.005	0.026	0.024	0.029	0.008	-0.025	-0.019	-0.006

Values in the table represent the monthly OOS predictive R^2 computed based on the predictions made by the given estimation technique on the given sample splitting scheme. Positive values denote better performance that a naive forecast of 0 excess returns. The upper bound for the values (the case of perfectly accurate predictions) is 1. The compared estimation techniques are Benchmark (B), Ordinary Least Squares (OLS), Principal Components Regression (PCR), Partial Least Squares (PLS), Elastic Net (EN), Gradient Boosted Regression Tree (GBRT), Random Forest (RF), and four different artificial Neural Network (NN) specifications. The compared sample splitting schemes are denoted by their type – Fixed Sample Splitting Scheme (FSSS), Recursive Sample Splitting Scheme (ReSSS), or Rolling Sample Splitting Scheme (RoSSS), and the scheme's window sizes in years in the parentheses. The window sizes come in the following order: training, validation, testing. Only data for the S&P 500 stocks was used during estimations.

In Table 6.2, we provide the monthly OOS prediction R^2 computed for each of the methods applied on each of the sample splitting scheme covered. The difference between these results and those summarized in Table 6.1 is that this

table contains values of Out Of Sample R^2 computed based predictions made exclusively on the subset of our data containing only stocks from the S&P 500 index.

The immediate observation we make is that the overall prediction accuracy increased. Compared to Gu *et al.* (2018), our models are however still underperforming heavily. Let us also note that Gu *et al.* (2018) also report higher accuracy of predictions for the largest stocks on the market. In this context, the modest increase of performance of our methods on the set of S&P 500 stocks is not overly satisfactory. Moreover, the NN technique recently heavily pronounced in the literature is once again not able to beat naive forecasts of zero returns.

Nevertheless, some estimation methods are now able to beat the benchmark model containing the 5 Fama and French factors on the fixed scheme with a larger training window as well as on RoSSS(10, 3, 1) and RoSSS(10, 5, 1). While the Out Of Sample predictive R^2 measures of some PCR, EN, or tree-based models' specifications are better that those of the benchmark model, the adapted version of the Diebold-Mariano test statistics does not allow us to reject the hypothesis of the predictions being equal even on the 15% level of statistical significance.

6.2.2 Ensemble Forecasts

Quantitative analysts often use ensemble methods to enhance their models. We have also used ensembling techniques in this analysis, e.g. when growing random forests from multiple regression trees or when constructing the final artificial neural network predictions as averages of several independent networks. But in this subsection, we consider a broad ensemble consisting of the final predictions of the individual modelling techniques. Using equal-weighted ensembles of the individual techniques is suggested e.g. by Krauss *et al.* (2017) who achieve better performance with these ensembles than when using the individual models alone. Additional support for using an ensemble prediction can be found in Gu *et al.* (2018). Therefore, we further use an equal-weighted average of the forecasts to find out whether it can outperform the predictive Fama and French five-factor benchmark.

?? includes the familiar OOS predictive R^2 measures. This time, we consider the prediction formed by three different ensembles of our methods, each of them is introduced in the note below the table. We use estimation outputs from

Table 6.3: Comparison of the Performance of Ensemble Predictions on the S&P 500 Stocks via Monthly Out Of Sample R^2

	All	All except B	PCR & PLS & EN & GBRT & RF
FSSS(5, 5, 10)	-0.434	-0.518	-0.108
FSSS(10, 5, 5)	0.027	0.027	0.035
ReSSS(5, 1, 1)	-0.326	-0.384	-0.099
ReSSS(5, 3, 1)	-0.353	-0.414	-0.105
ReSSS(5, 5, 1)	-0.391	-0.458	-0.123
RoSSS(5, 1, 1)	-0.226	-0.269	-0.063
RoSSS(5, 3, 1)	-0.256	-0.298	-0.079
RoSSS(5, 5, 1)	-0.375	-0.438	-0.126
RoSSS(10, 1, 1)	-0.109	-0.130	-0.041
RoSSS(10, 3, 1)	0.025	0.024	0.032
RoSSS(10, 5, 1)	0.029	0.028	0.027

Values in the table represent the monthly OOS predictive R^2 computed based on the predictions made by the given ensemble on the given sample splitting scheme. Positive values denote better performance that a naive forecast of 0 excess returns. The upper bound for the values (the case of perfectly accurate predictions) is 1. The compared ensembles contain either all methods All, All methods except the benchmark model (All except B), or an ensemble containing only the PCR, PLS, EN, GBRTand RF models. The compared sample splitting schemes are denoted by their type – Fixed Sample Splitting Scheme (FSSS), Recursive Sample Splitting Scheme (ReSSS), or Rolling Sample Splitting Scheme (RoSSS), and the scheme's window sizes in years in the parentheses. The window sizes come in the following order: training, validation, testing. Only data for the S&P 500 stocks was used during estimations.

the models considering only the S&P 500 stock data presented in the previous section. The reason why we examine the S&P 500 models instead of the original ones is that the predictive performance of the techniques is generally higher and thus we have better chances that some of the ensembles during will produce exceptional performance.

In the first column, all methods are combined into a single forecasting unit. The benchmark model is included solely out of curiosity, because it is one of the more successful models in our repertoire. The second column contains similar results, because differs from the first one only in that the benchmark model is absenting from the ensemble. We can see that the forecast accuracy did not improve much by combining the individual forecasts, this is probably due to inclusion of techniques which produce severely inferior predictions. In the third column, we further leave out OLS and NNs because these are the least accurate methods. The predictions in the last column are on average the best of the three ensemble alternatives. However, they fail to outperform forecasts of the individual models, both in terms of the monthly OOS R^2 and also when considering the Diebold-Mariano tests.

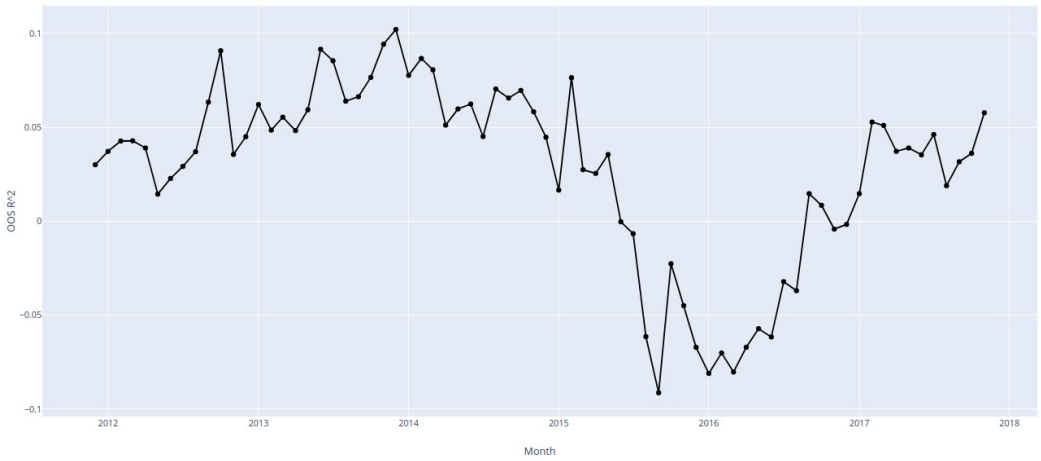
6.2.3 Rolling Out Of Sample R^2

In this section, we briefly examine how the performance of the estimation techniques evolves in time. For this purpose, we will utilize a simple 1-year rolling predictive OOS R^2 which is simply the ordinary OOS R^2 measured on the past 12 predictions. We again consider outputs from the S&P 500 model from **??**. Perhaps we will be able to find specific time periods during which our models achieve the same, or perhaps even greater performance than the models of Gu *et al.* (2018). Alternatively, we can identify months during which our performance deteriorates and analyze the potential causes of that.

As an example, the 1-year rolling OOS R^2 of the EN algorithm under the RoSSS(10, 3, 1) scheme is plotted in Figure 6.2. As we can see from the chart, the performance of the EN on the beginning of the testing sample reaches the value of 0.1 which is fairly close to the value measured in Gu *et al.* (2018). But the method fails to keep its high performance throughout the rest of the sample, which is documented by the large drop of the OOS R^2 into the negative territory around 2016.

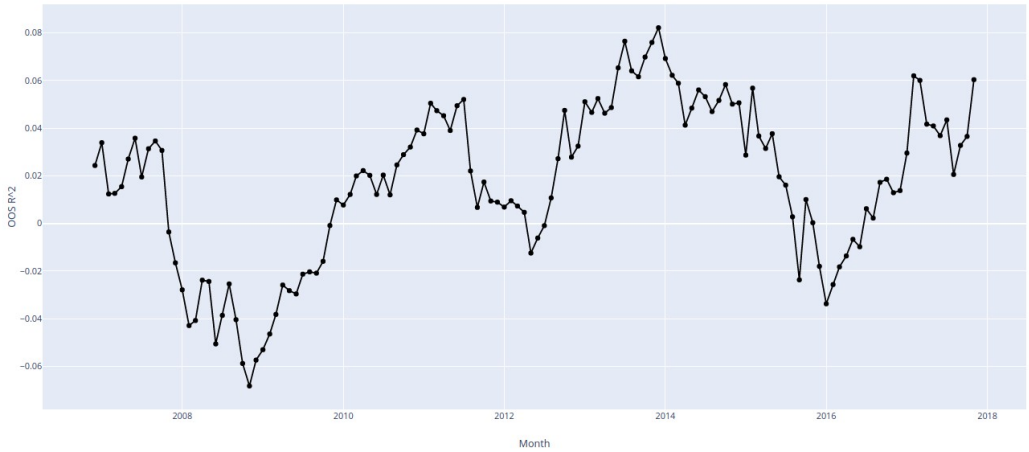
Next, we inspect the rolling OOS R^2 of the GBRT modelling technique used under RoSSS(5, 3, 1) to see how its performance evolves in time. Because

Figure 6.1: 1-Year Rolling Out Of Sample R^2 of Elastic Net under RoSSS(10, 3, 1)



The chart depicts the 1-year rolling OOS predictive R^2 measure for the computed on the EN estimation technique used under the RoSSS(10, 3, 1) scheme. The values of the 1-year rolling OOS R^2 are computed based on the past 12 predictions of the method. The x-axis denotes time in months and x-axis denotes the OOS R^2.

Figure 6.2: 1-Year Rolling Out Of Sample R^2 of Gradient Boosted Regression Tree under RoSSS(5, 3, 1)



The chart depicts the 1-year rolling OOS predictive R^2 measure for the computed on the GBRT estimation technique used under the RoSSS(5, 3, 1) scheme. The values of the 1-year rolling OOS R^2 are computed based on the past 12 predictions of the method. The x-axis denotes time in months and x-axis denotes the OOS R^2.

RoSSS(5, 3, 1) devotes only 5 years of the data to training, we have the opportunity, as opposed to the previous example, to uncover its performance prior to 2012. We notice that the performance drops again around 2016, but there is also even more significant drop in prediction accuracy timed exactly when the recent great financial crisis started. While having a decent performance during the year 2014, the method is not able to achieve these levels on the rest of the sample and its OOS predictive R^2 for the whole period averages at 0.006.

We have intentionally included charts only for well performing techniques on the given sample splitting schemes, to avoid artificially undermining the performance of our methods. Charts for other techniques and splitting schemes are not included for the sake of brevity. But they usually contain a similar behavior where the performance decreases around years 2015 and 2016. We fail to conclude what could be the reason for the sudden deterioration of our models performance. For schemes which are characteristic for their lower period, we often observe that the performance measure drops around 2009. We believe that this can be attributed to the recent financial crisis which was accompanied by increased volatility on the markets and could likely be the cause of this phenomena. This finding can perhaps be related to the observation of Welch & Goyal (2007) who find that several factors from their collection suddenly perform exceptionally well during the 1973 – 1975 Oil Shock. Let us also note that, with regards to for example NNs as well as the other advanced methods, in none of the cases examined has the 1-year rolling OOS R^2 got anywhere close to the values obtained by Gu *et al.* (2018) during the whole testing sample period. We conclude that the poor performance of the considered techniques under the selected sample splitting schemes can not be easily explained purely by the unfortunate location of the testing periods. Moreover, the rolling OOS R^2 on the recursive schemes did not show a clear positive trend, i.e. it did not suggest that the performance is gradually increasing in time which should be the case if the low number of observations in the training sample hold the performance back.

We can now proceed to the concluding remarks.

Chapter 7

Conclusion

In this thesis, we follow the empirical framework of Gu *et al.* (2018), and attempt to investigate how do some modern machine learning techniques perform when applied on a dataset of limited size. We have gathered a moderate sized dataset which we futher split into subsamples under a handful of sample splitting schemes considered. This way, each machine learning method employed has only a limited amount of data available for estimation. We consider the as a challenge for the newly developed complex nonlinear techniques.

We than compare the performance of these methods against a simple benchmark model. The results of our empirical procedure can be found in Chapter 6. We found that while several methods were able to slightly outperform the benchmark model containing the 5 factors from the well-known model introduced by Fama & French (2015), the improvement over the benchmark was not big enough for us to be able to assign it meaningful statistical significance via Diebold-Mariano tests.

We then proceeded to perform several robustness checks of our findings. By performing each of these checks, we attempted to undermine our conclusions from the previous chapter. But eventually, we failed to provide satisfactory evidence about a significant contribution of the examined machine learning methods over the traditionally used linear approaches.

Although the results are not promising in general, there are several observations we can take out after examining them more closely. First, the larger we set the training and validation windows, the greater results we achieve. While this can be troublesome for researchers working with small to moderately sized datasets, this finding promises that when there is an option to increase the number of observations in our dataset, there is also a chance to improve the

performance of our models. With increasing number of observations in the training sample, the fit of the given model to the data improves. On the other hand, an increased size of the validation dataset is especially important for methods which must be highly regularized in order to avoid overfit. Large validation windows give more relevance for the regularization hyperparameters which enjoy the best performance on the validation sample.

Based on these findings, we allow ourselves to express mild skepticism about the usefulness of various factors presented in the asset pricing literature in the past. While these factors are often able to help explain excess returns in a contemporaneous fashion, they failed in predicting these returns in our analysis. We however admit that this may also be the feature of the specific data we have used here. Another important thing to note here is that caution must be applied when inspecting discoveries which were not documented out-of-sample. Several of the factors we consider were not originally tested for OOS performance and still received a lot of attention. Without diminishing their contribution in the evolution of the finance literature, we maintain that researchers should proceed with increased scrutiny when declaring high usefulness that their factors bring.

Another consideration we take out from our examination is the practical versatility of the modern machine learning techniques in asset pricing. These techniques, represented e.g. by random forests or neural networks, enjoy great prominence both in the literature and in business. But as we have shown, they do not magically come to rescue the practitioner if he or she is not able to collect sufficiently enough data or an adequate number of predictor variables.

The results support the view presented in the paper of Welch & Goyal (2007), i.e. that the selected macroeconomic factors fail to predict the excess returns out of sample. Moreover, the conclusions are further strengthened because in this thesis, we utilized several advanced linear and nonlinear modelling techniques on top of the simple linear approach taken by Welch & Goyal (2007).

We provide mixed evidence as to whether the number of observations in the training sample is what limits the prediction performance of the modern machine learning methods the most. While increasing the training sample size by a huge leap of 5 years when specifying the sample splitting schemes proved to have some influence on the prediction accuracy of the methods, closer investigation via observing the prediction performance in time did not lead to any similar conclusion.

Bibliography

Acharya, V. V. & L. H. Pedersen (2005): "Asset Pricing with Liquidity Risk." *Journal of Financial Economics* **77(2)**: pp. 375–410.

Amihud, Y. (2002): "Illiquidity and Stock Returns: Cross-Section and Time-Series Effects." *Journal of Financial Markets* **5(1)**: pp. 31–56.

Arditti, F. D. (1967): "Risk and the Required Return on Equity." *The Journal of Finance* **22(1)**: pp. 19–36.

Asness, C. S., T. J. Moskowitz, & L. H. Pedersen (2013): "Value and Momentum Everywhere." *The Journal of Finance* **68(3)**: pp. 929–985.

Bachrach, B. & D. Galai (1979): "The Risk-Return Relationship and Stock Prices." *Journal of Financial and Quantitative Analysis* **14(2)**: pp. 421–441.

Bai, J. (2003): "Inferential Theory for Factor Models of Large Dimensions." *Econometrica* **71(1)**: pp. 135–171.

Baker, M. & J. Wurgler (2006): "Investor Sentiment and the Cross-Section of Stock Returns." *The Journal of Finance* **61(4)**: pp. 1645–1680.

Ban, G.-Y., N. El Karoui, & A. E. Lim (2016): "Machine Learning and Portfolio Optimization." *Management Science* **64(3)**: pp. 1136–1154.

Banz, R. W. (1981): "The Relationship between Return and Market Value of Common Stocks." *Journal of Financial Economics* **9(1)**: pp. 3–18.

Barberis, N., R. Greenwood, L. Jin, & A. Shleifer (2015): "X-CAPM: An Extrapolative Capital Asset Pricing Model." *Journal of Financial Economics* **115(1)**: pp. 1–24.

Bergmeir, C. & J. M. Benítez (2012): "On the Use of Cross-Validation for Timeseries Predictor Evaluation." *Information Sciences* **191**: pp. 192–213.

Bergstra, J. & Y. Bengio (2012): "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research* **13(Feb)**: pp. 281–305.

Black, F., M. C. Jensen, M. Scholes *et al.* (1972): "The Capital Asset Pricing Model: Some Empirical Tests." *Studies in the Theory of Capital Markets* **81(3)**: pp. 79–121.

Black, F. & M. Scholes (1973): "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy* **81(3)**: pp. 637–654.

Breiman, L. (2017): *Classification and Regression Trees*. Routledge.

Brennan, M. J., T. Chordia, & A. Subrahmanyam (1998): "Alternative Factor Specifications, Security Characteristics, and the Cross-Section of Expected Stock Returns." *Journal of Financial Economics* **49(3)**: pp. 345–373.

Campbell, J. Y. (1996): "Understanding Risk and Return." *The Journal of Political Economy* **104(2)**: pp. 298–345.

Campbell, J. Y. & T. Vuolteenaho (2004): "Bad Beta, Good Beta." *American Economic Review* **94(5)**: pp. 1249–1275.

Chan, L. K., J. Lakonishok, & T. Sougiannis (2001): "The Stock Market Valuation of Research and Development Expenditures." *The Journal of Finance* **56(6)**: pp. 2431–2456.

Chan, L. Đ. Đ., N. Jegadeesh, & J. Lakonishok (1996): "Momentum Strategies." *The Journal of Finance* **51(5)**: pp. 1681–1713.

Cox, J. C., S. A. Ross, & M. Rubinstein (1979): "Option Pricing: A Simplified Approach." *Journal of Financial Economics* **7(3)**: pp. 229–263.

Diebold, F. X. & R. S. Mariano (2002): "Comparing Predictive Accuracy." *Journal of Business & Economic Statistics* **20(1)**: pp. 134–134.

Diether, K. B., C. J. Malloy, & A. Scherbina (2002): "Differences of Opinion and the Cross Section of Stock Returns." *The Journal of Finance* **57(5)**: pp. 2113–2141.

Fama, E. F. (1968): "Risk, Return and Equilibrium: Some Clarifying Comments." *The Journal of Finance* **23(1)**: pp. 29–40.

Fama, E. F. (1971): "Risk, Return, and Equilibrium." *Journal of Political Economy* **79(1)**: pp. 30–55.

Fama, E. F., L. Fisher, M. C. Jensen, & R. Roll (1969): "The Adjustment of Stock Prices to New Information." *International Economic Review* **10(1)**: pp. 1–21.

Fama, E. F. & K. R. French (1992): "The Cross-Section of Expected Stock Returns." *The Journal of Finance* **47(2)**: pp. 427–465.

Fama, E. F. & K. R. French (1993): "Common Risk Factors in the Returns on Stocks and Bonds." *Journal of Financial Economics* **33(1)**: pp. 3–56.

Fama, E. F. & K. R. French (1996): "Multifactor Explanations of Asset Pricing Anomalies." *The Journal of Finance* **51(1)**: pp. 55–84.

Fama, E. F. & K. R. French (1998): "Value versus Growth: The International Evidence." *The Journal of Finance* **53(6)**: pp. 1975–1999.

Fama, E. F. & K. R. French (2004): "The Capital Asset Pricing Model: Theory and Evidence." *The Journal of Economic Perspectives* **18(3)**: pp. 25–46.

Fama, E. F. & K. R. French (2008): "Dissecting Anomalies." *The Journal of Finance* **63(4)**: pp. 1653–1678.

Fama, E. F. & K. R. French (2012): "Size, Value, and Momentum in International Stock Returns." *Journal of Financial Economics* **105(3)**: pp. 457–472.

Fama, E. F. & K. R. French (2015): "A Five-Factor Asset Pricing Model." *Journal of Financial Economics* **116(1)**: pp. 1–22.

Fama, E. F. & J. D. MacBeth (1973): "Risk, Return, and Equilibrium: Empirical Tests." *Journal of Political Economy* **81(3)**: pp. 607–636.

Fama, E. F. & B. G. Malkiel (1970): "Efficient Capital Markets: A Review of Theory and Empirical Work." *The Journal of Finance* **25(2)**: pp. 383–417.

Fama, E. F. & G. W. Schwert (1977): "Asset Returns and Inflation." *Journal of Financial Economics* **5(2)**: pp. 115–146.

Fang, L. & J. Peress (2009): "Media Coverage and the Cross-Section of Stock Returns." *The Journal of Finance* **64(5)**: pp. 2023–2052.

Feng, G., S. Giglio, & D. Xiu (2019): "Taming the Factor Zoo: A Test of New Factors." *Technical report*, National Bureau of Economic Research.

Feng, G., J. He, & N. G. Polson (2018): "Deep Learning for Predicting Asset Returns." *arXiv preprint arXiv:1804.09314* .

Ferson, W. E. & C. R. Harvey (1991): "The Variation of Economic Risk Premiums." *Journal of Political Economy* **99(2)**: pp. 385–415.

Ferson, W. E. & C. R. Harvey (1993): "The risk and Predictability of International Equity Returns." *Review of Financial Studies* **6(3)**: pp. 527–566.

Francis, J., R. LaFond, P. Olsson, & K. Schipper (2005): "The Market Pricing of Accruals Quality." *Journal of Accounting and Economics* **39(2)**: pp. 295–327.

French, K. R. (2019): "Fama/French 5 Factors (2x3)." Http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html#Research (online, accessed July 25, 2019).

Friend, I. & M. Blume (1970): "Measurement of Portfolio Performance under Uncertainty." *The American Economic Review* **60(4)**: pp. 561–575.

Galai, D. & R. W. Masulis (1976): "The Option Pricing Model and the Risk Factor of Stock." *Journal of Financial Economics* **3(1-2)**: pp. 53–81.

Gompers, P., J. Ishii, & A. Metrick (2003): "Corporate Governance and Equity Prices." *The Quarterly Journal of Economics* **118(1)**: pp. 107–156.

Gonedes, N. J. (1973): "Evidence on the Information Content of Accounting Numbers: Accounting-Based and Market-Based Estimates of Systematic Risk." *Journal of Financial and Quantitative Analysis* **8(3)**: pp. 407–443.

Goyal, A. (2019): "A Comprehensive Look at the Empirical Performance of Equity Premium Prediction - Updated Data (up to 2018)." Http://www.hec.unil.ch/agoyal/ (online, accessed July 28, 2019).

Green, J., J. R. Hand, & X. F. Zhang (2013): "The Supraview of Return Predictive Signals." *Review of Accounting Studies* **18(3)**: pp. 692–730.

Gu, S., B. Kelly, & D. Xiu (2018): "Empirical Asset Pricing via Machine Learning." *Technical report*, National Bureau of Economic Research.

Hamada, R. S. (1969): "Portfolio Analysis, Market Equilibrium and Corporation Finance." *The Journal of Finance* **24(1)**: pp. 13–31.

Hamada, R. S. (1972): "The Effect of the Firm's Capital Structure on the Systematic Risk of Common Stocks." *The Journal of Finance* **27(2)**: pp. 435–452.

Hastie, T., R. Tibshirani, & J. Friedman (2009): *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Hirshleifer, J. (1964): "Efficient Allocation of Capital in an Uncertain World." *The American Economic Review* **54(3)**: pp. 77–85.

Hirshleifer, J. (1965): "Investment Decision under Uncertainty: Choice-Theoretic Approaches." *The Quarterly Journal of Economics* **79(4)**: pp. 509–536.

Hou, K., C. Xue, & L. Zhang (2015): "Digesting Anomalies: An Investment Approach." *The Review of Financial Studies* **28(3)**: pp. 650–705.

Ioffe, S. & C. Szegedy (2015): "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *Technical report*.

Jagannathan, R. & Z. Wang (1996): "The Conditional CAPM and the Cross-Section of Expected Returns." *The Journal of Finance* **51(1)**: pp. 3–53.

Jegadeesh, N. & S. Titman (1993): "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency." *The Journal of Finance* **48(1)**: pp. 65–91.

Jegadeesh, N. & S. Titman (2001): "Profitability of Momentum Strategies: An Evaluation of Alternative Explanations." *The Journal of Finance* **56(2)**: pp. 699–720.

Jensen, M. C. (1968): "The Performance of Mutual Funds in the Period 1945–1964." *The Journal of Finance* **23(2)**: pp. 389–416.

Jensen, M. C. (1969): "Risk, the Pricing of Capital Assets, and the Evaluation of Investment Portfolios." *The Journal of Business* **42(2)**: pp. 167–247.

Keim, D. B. & R. F. Stambaugh (1986): "Predicting Returns in the Stock and Bond Markets." *Journal of Financial Economics* **17(2)**: pp. 357–390.

Kingma, D. P. & J. Ba (2014): "Adam: A method for stochastic optimization." *Technical report*.

Kothari, S. P., J. Shanken, & R. G. Sloan (1995): "Another Look at the Cross-Section of Expected Stock Returns." *The Journal of Finance* **50(1)**: pp. 185–224.

Krauss, C., X. A. Do, & N. Huck (2017): "Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500." *European Journal of Operational Research* **259(2)**: pp. 689–702.

Lakonishok, J., A. Shleifer, & R. W. Vishny (1994): "Contrarian Investment, Extrapolation, and Risk." *The Journal of Finance* **49(5)**: pp. 1541–1578.

Lee, C. M. & B. Swaminathan (2000): "Price Momentum and Trading Volume." *The Journal of Finance* **55(5)**: pp. 2017–2069.

Lettau, M. & S. Ludvigson (2001): "Resurrecting the (C) CAPM: A Cross-Sectional Test when Risk Premia Are Time-Varying." *Journal of Political Economy* **109(6)**: pp. 1238–1287.

Levy, H. & M. Sarnat (1970): "International Diversification of Investment Portfolios." *The American Economic Review* **60(4)**: pp. 668–675.

Lewellen, J., S. Nagel, & J. Shanken (2010): "A Skeptical Appraisal of Asset Pricing Tests." *Journal of Financial Economics* **96(2)**: pp. 175–194.

Lewellen, J. *et al.* (2015): "The Cross-Section of Expected Stock Returns." *Critical Finance Review* **4(1)**: pp. 1–44.

Lintner, J. (1965a): "Security Prices, Risk, and Maximal Gains from Diversification." *The Journal of Finance* **20(4)**: pp. 587–615.

Lintner, J. (1965b): "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets." *The Review of Economics and Statistics* **47(1)**: pp. 13–37.

Litzenberger, R. H. & K. Ramaswamy (1979): "The Effect of Personal Taxes and Dividends on Capital Asset Prices: Theory and Empirical Evidence." *Journal of Financial Economics* **7(2)**: pp. 163–195.

MacKinlay, A. C. (1995): "Multifactor Models Do Not Explain Deviations from the CAPM." *Journal of Financial Economics* **38(1)**: pp. 3–28.

Merton, R. C. (1973a): "An Intertemporal Capital Asset Pricing Model." *Econometrica* **41(5)**: pp. 867–887.

Merton, R. C. (1973b): "Theory of Rational Option Pricing." *The Bell Journal of Economics and Management Science* **4(1)**: pp. 141–183.

Merton, R. C. (1974): "On the Pricing of Corporate Debt: The Risk Structure of Interest Rates." *The Journal of Finance* **29(2)**: pp. 449–470.

Merton, R. C. (1976): "Option Pricing when Underlying Stock Returns are Discontinuous." *Journal of Financial Economics* **3(1-2)**: pp. 125–144.

Merton, R. C. (1980): "On Estimating the Expected Return on the Market: An Exploratory Investigation." *Journal of Financial Economics* **8(4)**: pp. 323–361.

Merton, R. C. (1987): "A Simple Model of Capital Market Equilibrium with Incomplete Information." *The Journal of Finance* **42(3)**: pp. 483–510.

Miller, E. M. (1977): "Risk, Uncertainty, and Divergence of Opinion." *The Journal of Finance* **32(4)**: pp. 1151–1168.

Modigliani, F. & M. H. Miller (1958): "The Cost of Capital, Corporation Finance and the Theory of Investment." *The American Economic Review* **48(3)**: pp. 261–297.

Mossin, J. (1966): "Equilibrium in a Capital Asset Market." *Econometrica* **34(4)**: pp. 768–783.

Mossin, J. (1968): "Optimal Multiperiod Portfolio Policies." *The Journal of Business* **41(2)**: pp. 215–229.

Pástor, L. & R. F. Stambaugh (2003): "Liquidity Risk and Expected Stock Returns." *Journal of Political Economy* **111(3)**: pp. 642–685.

Reinganum, M. R. (1981): "Misspecification of Capital Asset Pricing: Empirical Anomalies Based on Earnings' Yields and Market Values." *Journal of Financial Economics* **9(1)**: pp. 19–46.

Rosenblatt, F. (1958): "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological review* **65(6)**: p. 386.

Schwert, G. W. (2003): "Anomalies and Market Efficiency." In G. M. Constantinides, M. Harris, & R. M. Stulz (editors), "Handbook of the Economics of Finance," Elsevier.

Sharpe, W. F. (1964): "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk." *The Journal of Finance* **19(3)**: pp. 425–442.

Sharpe, W. F. (1965): "Risk-Aversion in the Stock Market: Some Empirical Evidence." *The Journal of Finance* **20(3)**: pp. 416–422.

Shleifer, A. & R. W. Vishny (1997): "The Limits of Arbitrage." *The Journal of Finance* **52(1)**: pp. 35–55.

Solnik, B. H. (1974): "An Equilibrium Model of the International Capital Market." *Journal of Economic Theory* **8(4)**: pp. 500–524.

Tobin, J. (1965): "The Theory of Portfolio Selection." In F. Hahn & F. Brechling (editors), "The Theory of Interest Rates," Macmillan London.

Treynor, J. L. (1961): "Market Value, Time, and Risk." *Technical report*, Unpublished manuscript.

Vassalou, M. & Y. Xing (2004): "Default Risk in Equity Returns." *The Journal of Finance* **59(2)**: pp. 831–868.

Welch, I. & A. Goyal (2007): "A Comprehensive Look at the Empirical Performance of Equity Premium Prediction." *The Review of Financial Studies* **21(4)**: pp. 1455–1508.