University of Applied Sciences
School of Management and Law

Machine Learning in Asset Pricing

Master Thesis

Written by: Dorina Ababii
Supervisor: Dr. Patrick Hauf
Co-supervisor: Dr. Ruben Seiberlich

Zürich, Switzerland, December 2021

Dedication

It is with genuine gratitude and warm regard I dedicate this work
to my beloved family and friend. A special feeling of recognition to
my beloved grandfather, Constantin, my mother, Lucia,
my sisters, Madalina and Anisoara, and Samuel,
whose words of encouragement for tenacity
helped me continue my road.

Acknowledgments

I cannot express enough thanks to my teachers for their support and encouragement: Dr. Patrick Hauf, Senior lecturer at the Zurich University of Applied Sciences School of Management and Law, and Dr. Ruben Seiberlich, Program Director MSc Banking and Finance at the Zurich University of Applied Sciences School of Management and Law. I express my sincere appreciation for the learning opportunities that favor my development.

My thesis could not have been accomplished without the support of my employer, Founder & CEO at HCP Asset Management SA – thank you for allowing me time away to research and write the thesis.

Finally, I would like to express my most profound appreciation to my teachers from the University of Rouen, Mr. Jean-Marie Hommet, who has introduced me to macroeconomics and the portfolio theory, and Jean-Didier Zanos, who continued to nourish my interest in applied mathematics.

Table of Content

Part 1 Preliminary *7*

Chapter 1. 1 Introduction **7**
Chapter 1. 2 Goal and Objectives **8**
Chapter 1. 3 Organization of the Paper **9**

Part 2 Machine learning Methods *9*

Chapter 2. 1 Supervised learning as function Approximation **10**
2.1.1 Regression Methods 11
2.1.2 Linear methods 11
2.1.3 Trees and Random Forests 12
2.1.4 Neural Networks 14
2.1.5 Hyperparameter Tunning 17

Chapter 2. 2 ML in Asset Pricing **19**
2.2.2 Concussion 26

Part 3 Data set and Methodology *26*

Chapter 3. 1 Study of Intracontinental Equities **27**
3.1.1 Data set 27
3.1.2 Data modeling 29
3.1.3 Data Description 33
3.1.4 Treating the missing values and other transformations 36
3.1.5 Normalization 36
3.1.6 Feature correlation 37
3.1.7 Portfolio Construction Framework 39
3.1.8 Problem definition 40
3.1.9 Training and Prediction 40
3.1.10 Performance Evaluation 41

Chapter 3. 2 Prediction models **43**
3.2.1 Features Importance 43
3.2.2 Linear Regression 44
3.2.3 CatBoost Model 47
3.2.4 Long Short-Term Memory Model 50

Chapter 3. 3 Empirical Study **52**
3.3.1 Results of Long - Only portfolio 52
3.3.2 Results of Short–Only Portfolio 53
3.3.3 Results of Long-Short Portfolio 54

Chapter 3. 4 Conclusion Remarks **56**

References *58*
Appendix *60*

Data Transformation. Strategy **60**
Macro Variables 61

Score distribution **62**
CatBoost Model Parameters **63**
CatBoost Model Graphic Visualization of the Model's Prediction **64**
CatBoost Model Loss Evolution **65**
Features Importance **66**
False Positive and False Negatives **68**
Long-Short Term Memory Model Loss Evolution **69**
Cumulative Returns **70**
Long-Only Portfolios 70
Short-Only Portfolios 72
Long-Short Portfolios 74
All Portfolios Together. An Asset Pricing ML Approach 76

TABLE OF FIGURES

Figure 1: Feature Interactions 13
Figure 2: Regression Tree Example 13
Figure 3: Neural Network With Two Hidden Layers 15
Figure 4: Cross-Validation Folds: Train - Prediction Sets 19
Figure 5:Data Source Frequency Per Portfolio 27
Figure 6: Feature Frequency Per Index And Index Constituents Together 28
Figure 7: Data Merging 32
Figure 8: Spx500, Data Description 34
Figure 9: Gdaxi, Data Description. 34
Figure 10: Hsi, Data Description 35
Figure 11: Topx500, Data Description 35
Figure 12: Spx500, Feature Correlation 37
Figure 13: Prediction Mapping 40
Figure 14: Catboost, Rmse Results 47
Figure 15: Catboost Parameters 48
Figure 16: Actual Versus Prediction 49
Figure 17 (B): Lstm, Rmse Results 51
Figure 18:Performance Comparison (1) 55
Figure 19: Performance Comparison (2) 56
Figure 20: Performance Comparison (3) 56
Figure 21: Macro Variables, Data Source: Fred 61
Figure 22: Score Distribution, Gdaxi Portfolios 62
Figure 23: Score Distribution, Hsi Portfolios 62
Figure 24: Score Distribution, Spx500 Portfolios 62
Figure 25: Score Distribution, Topx500 Portfolios 62
Figure 26: Score Distribution, Catboost Predictions 63
Figure 27: Catboost Parameters 63
Figure 28: Tree Structure 64
Figure 29: Topx500 Portfolio Loss Measure While Gradually Removing 404 Features 65
Figure 30: Spx500 Portfolio Loss Measure While Gradually Removing 390 Features 65
Figure 31: Hsi Portfolio Loss Measure While Gradually Removing 106 Features 65
Figure 32: Gdaxi Portfolio Loss Measure While Gradually Removing 115 Features 65
Figure 33: Feature Importance. The 20 Most Predictive Features, Gdaxi Portfolio 66
Figure 34: Feature Contribution To Model Prediction, Gdaxi 66
Figure 35: Feature Importance. The 20 Most Predictive Features, Hsi Portfolio 66
Figure 36: Feature Contribution To Model Prediction, Hsi 67
Figure 37: : Features Importance. Top 20 Most Predictive Features, Spx500 Portfolios 67
Figure 38:Features Contribution To Model Prediction 67
Figure 39: Feature Importance. Top 20 Most Predictive Features, Topx500 Portfolios 68
Figure 40: False-Negative (Fn) And False-Positive (Fp) 68
Figure 41: Training Loss Evolution During Training An Lstm Model With 15 Variables 69
Figure 42: Training Loss Evolution During Training An Lstm Model With 150 Variables 69
Figure 43: Performance Comparison Long-Only Portfolios, Gdaxi 70
Figure 44: Performance Comparison Long-Only Portfolios, Hsi 70
Figure 45: Performance Comparison Long-Only Portfolios, Topx500 71
Figure 46: Performance Comparison Long-Only Portfolios, Spx500 71
Figure 47: Performance Comparison Short-Only Portfolios, Gdaxi 72
Figure 48: Performance Comparison Short-Only Portfolios, Hsi 72
Figure 49: Performance Comparison Short-Only Portfolios, Topx500 73
Figure 50: Performance Comparison Short-Only Portfolios, Spx500 73
Figure 51: Performance Comparison Long-Short Portfolios, Gdaxi 74
Figure 52: Performance Comparison Long-Short Portfolios, Hsi 74
Figure 53: Performance Comparison Long-Short Portfolios, Topx500 75
Figure 54: Performance Comparison Long-Short Portfolios, Spx500 75
Figure 55: Performance Comparison All Portfolios, Gdaxi 76
Figure 56: Performance Comparison All Portfolios, Hsi 76
Figure 57: Performance Comparison All Portfolios, Spx500 77
Figure 58: Performance Comparison All Portfolios, Topx500 77

Abbreviations

ML	Machine Learning
AI	Artificial Intelligence
OLS	Ordinary Least Squares
NN	Neural Networks
CatBoost	Categorical Boosting
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory
CV	Cross-Validation

ABSTRACT

Quantitative studies have gained more and more attention across multiple fields, including finance, since Daca Science has emerged. Moreover, since Gu et al. (2020) showed the superiority of algorithms such as neural networks and decision trees over fundamental predictive models, studying AI in the context of financial modeling becomes more relevant for financial institutions.

Stock price prediction is an important field of research in finance and a hot topic at a real-life application level. However, different methods have been studied until now. Since the relationship between features and stock price prediction is nonlinear, machine learning models, and more precisely, deep learning, seem attractive in asset management. However, it is unclear yet if these methods can indeed be helpful in real applications.

This paper conducts comparative experiments across a few different algorithms across indexes in four other countries: Germany, United States (US), Japan, and China. First, one presents a cross-sectional monthly stock price prediction framework in terms of machine learning. Next, one builds a portfolio using the publicly available information at the time t and predicts its return at the market opening the next month.

The paper conducts an empirical analysis of the four indexes that have been selected and confirms the superiority of nonlinear algorithms over the linear ones on a performance level and predictive accuracy. Furthermore, neural networks perform better overall in a universal portfolio. The paper demonstrates how using data and more company-specific information at the portfolio level can increase portfolio performance compared to the benchmark portfolio. The result results show that one can apply a Long-Short strategy to boost portfolio performance according to results.

Part 1 PRELIMINARY

Chapter 1. 1 INTRODUCTION

Prediction is a crucial task in asset pricing. Generally, investors predict stock prices similar to the discounted future cash flow model. In investments, predicted asset returns trigger signals. Investors use those signals to find outperforming stocks before applying a particular trading strategy. Researchers have proved that asset prices models look for the most influential variables to forecast return variations across time (J. M. Chen 2021; Fama and French 2015a).

The number of relevant predictors drastically increased over time, mainly because of the publicly available information. Nowadays, corporate financial reports represent only a tiny part of processable data to predict asset returns. Therefore, the modern database includes historical information regarding stock prices, the volume of transactions, news, sentiment analyses, and many other processable data available on social media and online reviews.

As the number of potential predictors increases, the data abundance contributes to a rise in statistical problems. For example, consider a universe of $N = 100$ stocks and K number of predictors. As the number of predictors increases, one has $\lim_{K \to \infty} K \cdot = \infty \rightarrow K > N$. At this point, OLS becomes ineffective because it overfits statistical noise. To obtain meaningful estimations, the number of predictors must be considerably low in the OLS. Therefore, a careful selection of variables is essential.

Empirical asset pricing studies have focused on low-dimensional models accordingly to a firm's characteristics. However, the new data collection entails sparsity problems in asset pricing because researchers may skip highly relevant predictive variables. Moreover, the ad-hoc sparsity is related to theoretical asset pricing returns, raising questions around investors' decision-making. Therefore, budling better forecasting models is a multi-dimensional problem that cannot be solved with conventional statistical models.

ML is a relatively young computer science field that offers tools to solve high-dimensional predictive problems related to financial time series. Broadly, ML is a set of algorithms designed to learn from the data set. First, the algorithms are trained using a subset of the data. Second, the models are be employed to forecast the signals for the

investment strategies. Generally, having different properties than technology, medicine, and other fields data sets, asset pricing models require particular adaptations. To develop adaptations, one must understand the environment that produces data.

Chapter 1. 2 GOAL AND OBJECTIVES

The stock market is a place of gains and losses. As a result, investors and portfolio managers face challenging decision-making regarding stock selection or portfolio construction. Exchangeable-traded funds (ETFs) can, in this regard, be useful investment tools for all kinds of investors. Like mutual funds and traded like stocks, ETFs became famous for portfolio diversification and stability reasons. Typically linked to specific benchmark indexes in a certain industry, it can be pictured as a safe haven for investors.

What if investors would take a more active investment perspective? What if investors would go outside of traditional portfolio construction models and use more sophisticated methods like ML to build up their portfolio to diversify their portfolio, attend a higher performance, and make more returns on their portfolio? What if ML can considerably improve decision-making processes within investments?

Various empirical studies have been conducted on ML in Asset pricing (Gu, Kelly, and Xiu 2020; Stefan Nagel, Ross, and Duffe 2021). These studies have consistently found that deep learning models attend better portfolio performance than traditional asset pricing models. Other studies have been conducted on the conventional asset pricing models (Fama and French 2015b). While efforts have been made to develop new models for portfolio construction, traditional investors are reluctant to ML developments and often prefer established financial modeling. This paper aims to study potential opportunities for portfolio construction with ML in four regions across the Globe. Long-only, short-only, long-short portfolios performance is compared to the benchmark portfolio's performance. The intuition is that deep learning models can overperform the traditional asset pricing models on a predictive accuracy level and portfolio performance assessment.

This paper (1) presents how to build 44 portfolios per region, using index constituents as the universe of stocks. The study aims to construct portfolios using three machine learning models: Categorical Boosting model (CatBoost), Long-Short Term Memory neural network model (LSTM) and, Linear Model. Next, (2) to assess the portfolio performance, this paper compares the annualized Sharpe ratios, Information Ratio, mean

return, and the mean active return of the constructed portfolios with the benchmark portfolios. In addition, the paper tries to identify if there is a difference between the portfolio returns given the four countries and if there are any comparative advantages in the portfolio construction framework using ML.

Chapter 1. 3 ORGANIZATION OF THE PAPER

The paper is organized into three parts. Part one introduces the research objective and aims to give an overview of the organizational way of the paper. Next, part two provides a literature review of the ML models, followed by part three, which details the working process, and results.

Sequentially, the parts are organized in chapters. Chapter 2.1 gives an overview of ML models that can potentially be applied in asset pricing for portfolio construction. Next, Chapter 2.2 introduces the challenges that face ML models regarding time series and financial data sets.

Once the paper introduces a few ML models and ways to cope with traditional ML to financial data, Part 3 provides details on data treatment, the model set up, portfolio construction, and performance measurements (chapter 3.1). The next part presents the three models employed (chapter 3.2) to construct the 44 portfolios. Finally, chapter 3.3 explains the results, followed by remarks and conclusions. Supplementary figures and information are provided in Appendix.

Part 2 MACHINE LEARNING METHODS[1]

In the first part, one provides a short overview of the supervised learning methods. This paper bases the research on affluent studies in asset pricing literature, focusing on Gu et al. (2020). Therefore, the outline of the models is not exhaustive. Furthermore, one only brings in the analysis of the most relevant models for asset pricing data. Thus, rather than giving a detailed account of methods in the field, one focuses on explaining the key elements of the most prominent models.

[1] In this part, formulas are general and taken from the literature in asset pricing, ML, and econometrics

Chapter 2. 1 SUPERVISED LEARNING AS FUNCTION APPROXIMATION

In this chapter, the paper presents a summary of supervised learning in computer science. The knowledge in the field is broad and fast-growing. Therefore, rather than miming each explanation of the models applied to asset pricing, the attention is focused on the elementary element of the methods that seem crucial to understanding using ML in asset pricing.

Generally, supervised learning aims to forecast the outcomes y (e.g., asset returns) based on K predictors, each represented in the x_i vector. The general model uses a training data set $\{y_i, x_1\}_1^N$ to determine the function $f \in \mathcal{F}$ that satisfy

$$y_i = f(x_i) + \varepsilon_i. \tag{2.1.1}$$

where ε_i represents the mean zero noise that cannot be expected by x, and $f \in \mathcal{F}$. The goal of the modeling in asset pricing is to train the initial data set to predict based on the same statistical model (2.1.1) so that the estimated function $\hat{f}(x_i)$ can predict correctly out-of-sample (Stefan Nagel, Ross, and Duffe 2021).

In supervised learning, one distinguishes two groups of methods. On the one hand, classification methods predict a categorical variable y. On the other hand, regression methods are destined to predict continuous dependent variables like stock returns. Regression models are most common in asset pricing modeling, while classification methods can help predict binary variables like Buy or Sell.

The class function $\mathcal{F}$ differs considerably from one regression method to another. The key attribute of all those supervised learning models is the capacity to predict high-dimensional environments where the number of variables exceeds the number of observations in the training data.

In their research, Wolpert and Wiliam G. Macready (1997), in their study, show that there are no universal learning algorithms. Their results imply no free-lunch theorem in ML, and there should be at least some kind of prior knowledge about the dependent variable. In general, Wolpert's result gets to a point where even if a universal algorithm would be existent, one cannot know beforehand which method would perform better out-of-sample. Furthermore, Wilson and Martinez (Wilson and Martinez 1997) affirm that successful results require knowledge on the choice of algorithm, how it can be implemented, and the best algorithms for the prediction problem that one wants to solve.

To successfully apply ML in asset pricing, one must better understand the asset price data one chose to predict. Not all the ML models are suited to work with asset price data,

and one first must understand which model fits best. In the next section, the paper introduces supervised learning methods shortly.

2.1.1 REGRESSION METHODS

Long tradition statistical and econometrical model regression models enjoy popularity among research thankful to the applicability of the models and the predictive performance (Hastie, Tibshirani, and Friedman 2009). It is not only well known for solving multi-dimensional environments but also can eventually apply to nonlinear problems. Still very popular among finance researchers, the linear methods have the great advantage of approximating nonlinearities through the nonlinear approximation of predictors (J. M. Chen 2021; Fama and French 2015b).

2.1.2 LINEAR METHODS

In linear models, one may assume that the function $f(x_i)$ is linear and follows an equation type:

$$y_i = x_i\, g + \varepsilon_i, \tag{2.1.3}$$

where g is a vector of the unknown regression coefficients, for the stock i, given the predictors x. While y_i Suggests linear relationship with the variable x_i, the vector x_i could include nonlinear predictors(Stefan Nagel, Ross, and Duffe 2021).

A typical linear transformation used in modeling includes variable interactions. For example, suppose one has a vector of features $b_i = \left(b_{1,i}, b_{2,i}\right)'$ type 2×1 one can compose a vector $x_i = (b_{1,i}, b_{1,i}b_{2,i}, b_{2,1})'$, where one includes$b_{1,i}b_{2,i}$ among variables to capture the features interactions.

In a sample with $N \times 1$ time periods[2], one has a vector of predictors $y = (y_1, y_2, \dots, y_N)'$, K predictors, $N \times K$ is than a matrix of variables $X = (x_1, x_2, \dots, x_N)'$. A way to estimate the function g from (2.1.3) is to take g such that it minimizes the sum of squared errors:

$$\min_g (y - Xg)'(y - Xg). \tag{2.1.2}$$

When one sets first derive to zero and solve the equation for g, one obtains the OLS estimator:

[2] Data entries, observations

$$\hat{g} = (X'X)^{-1}X'y, \tag{2.1.3}$$

and the list of in sample fitted values:

$$\hat{y} = X\hat{g}. \tag{2.1.4}$$

As the number of features K increases and exceed the number of observations, the results of the OLS regression became untrustworthy, usually with very high in-sample R^2 and low or negative R^2 for out-of-sample predictions. The low or negative R^2 for out-of-sample predictions is generally caused because the OLS model overfits the statistical noise rather than takes advantage of the signals when it estimates the $\hat{g}$. Another disadvantage of the OLS regression is that as $K > N$, the regression has multiple solutions for $\hat{g}$. With this, the model is more centered to fit the ε_i in $y_i = f(x_i) + \varepsilon_i$ and not in finding the $(x_i) + \varepsilon_i$.

In 1970, Hoerl and Kennard introduced L^2-norm penalty $g'g$ for improving predictive performance of the OLS regression. In the ridge regression, authors show that increased performance is possible if the sum-of-squared-errors is minimized, such as:

$$\min_{g} \left[\frac{1}{N} \underbrace{(y - Xg)'(y - Xg)}_{(1)\ loss} + \underbrace{\gamma g'g}_{(2)\ penalty} \right]. \tag{2.1.5}$$

The problem can be solved by minimizing the loss (first part of the goal). In the second part, the parameter γ keeps under control the degree of penalty. The problem can be solved as follow:

$$\hat{g} = (X'X + \gamma I_k)^{-1}X'y, \tag{2.1.6}$$

where I_k is a $K \times K$ identity matrix.

2.1.3 TREES AND RANDOM FORESTS

The simple linear regression does not capture the nonlinear effect of variables on the output. More complex models are trees and random forests, which on top of considering the nonlinear impact of the predictors, also iterates among predictors. This means the models can capture the effects between variables (Bielby et al. 2010)(Bensic, Sarlija, and Zekic-Susac 2005; Bielby et al. 2010).

Two variables X_1 and X_2 interact when the impact of the explanatory variable X_1 on the output Y is a function of Y and X_2. Example figure 1:

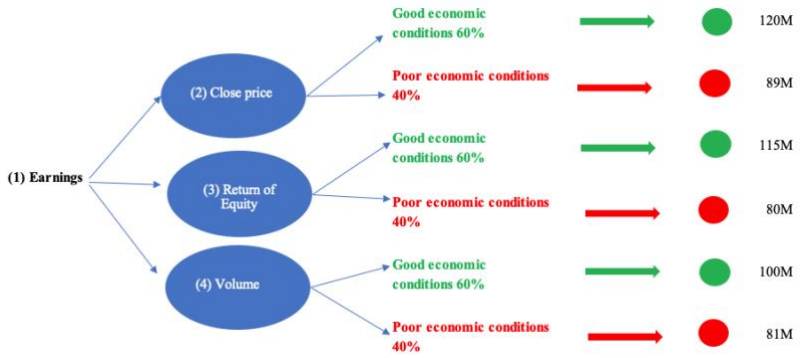


Figure 1: Feature Interactions

The figure describes how decision trees capture interactions among four variables earnings, close price, return on equity, and volume.

Decision trees are fully nonparametric and follow a logical path based on linear regression methods. The main idea is to identify groups that behave similarly. When one adds steps, one grows the tree by adding branches destined to sort the data in new categories based on a feature. Then, the sequential branching separates the data into categories. Finally, the slicing approximates the unknown conditional expected return function with the average output of each variable.

Figure 1 describes an example of three predictors: earnings, close price, and return on equity. The left figure shows how the tree assigns observations to a category accordingly to its predicting values. For example, the observations above 0.34 breakpoint are given to category 5. Those which have more information are then sorted by close price and return on equity. For example, suppose the close price goes up15% in good economic conditions. In that case, the earnings are considered to grow 40%; then the output is classed in category 1 — the same logic for all other classifications. Finally, the prediction for observations in one category equals the average of the outcome variable's forecast.

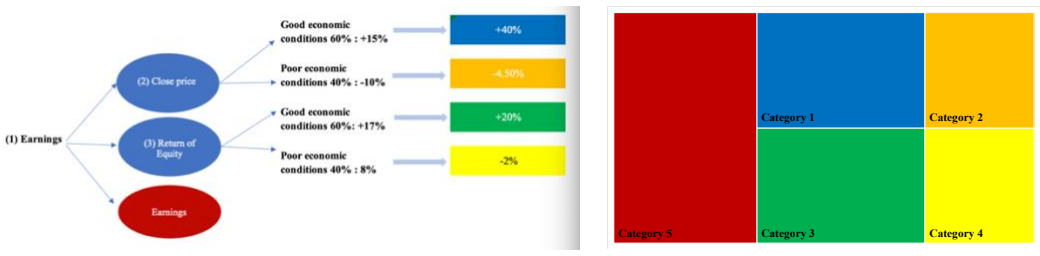


Figure 2: Regression tree example

This figure shows two diagrams of a regression three (on the left) in the scape of tree characteristics (earnings, close price, and return on equity). The terminal nodes of the tree (also called leaves) are colored in blue, orange, green, yellow, and red. Based on the values of these characteristics, the sample of earnings is divided into 5 categories.

Translated in a formula, the prediction of a tree, τ, with K 'leaves'[3], and the depth L, equals:

$$g(m_{i,t}; \theta, K,) = \sum_{k=1}^{K} \theta_k 1_{\{m_{i,t} \in C_k\ (L)\}} \quad (2.1.7)$$

where $C_k\ (L)$ is one of the K categories of data, θ_k the category's constant equals the average of the variables within the class, and $m_{i,t}$ the feature i at the time t. To grow a tree, one should find bins that best discriminate between outcomes(Gu, Kelly, and Xiu 2020; Hastie, Tibshirani, and Friedman 2009).

The main advantages of the model are that it can capture severe nonlinearities and that a tree depth L can capture $(L-1)$ types of features interactions. However, because decision trees are very flexible, they face overfitting. Therefore, high regularization is needed to overcome the overfitting effect.

One popular way to overcome overfitting is to 'boost'. This method implies an endless combination of forecasts from multiple significantly reduced trees. The method's main idea is that the small decision trees are very bad learners and make lots of mistakes, but as an ensemble, they feed the 'best learner' with more stability than each small tree taken one by one (Gu, Kelly, and Xiu 2020).

A popular regularization method is a gradient boosting regression trees technique. In the first step, the model fits a small tree. In the next step, the model uses a second simple tree with the same depth to fit the errors from the first tree. The output is then the sum of the predictions of these two trees, where the projection of the second tree is flattened by a factor $\mathcal{V} \in\ (0,1)$. The shrinkage prevents the model from overfitting the residuals.[4] At each step b, a small tree is fitted to the residuals of the previous step model. The model iterates till it reaches the B number of decision trees (Hastie, Tibshirani, and Friedman 2009).

2.1.4 NEURAL NETWORKS

Neural Networks are essential if one works with nonlinear regressions. Given the input x_i, the output y_i, one has J covariates in a NN. The inputs and the prediction are connected

[3] Also called terminal nodes

[4] Generally, overfitting the residuals refer to the event when the model, rather than working on the features' relationship, start describing the error in the data set.

via a hidden (or multiple) layer (s) of H nodes. Going back to equation (2.1.1), for NN, the function becomes:

$$f(x_i) = a_2 + w'_2 g\,(a_i + W_1 x_i)\,, \qquad (2.1.8)$$

where $a_i + W_1 x_i$ represents the vector of potential variables, a_2 the constant, w'_2 the weight and g the activation function. In general, the activation function is nonlinear and operates on each potential variable. A typical type of activation function is ReLU[5]. The number of nodes defines the flexibility of the model. If the number of nodes is high, one can consider NN as 'universal approximators' (Kur Hornik, Maxwell Stincombe 1989; Stefan Nagel, Ross, and Duffe 2021). Thankfully to its flexibility, the NN can capture nonlinear interactions between variables. At the same time, the model's complexity increases considerably and classes the model among least transparent models, least interpretable, and most parametrized ML algorithms (Molnar 2020).

The general architecture of a NN contains:

- an input layer (the variables),
- some hidden layers that interact in between each other and transform the nonlinear relationships between variables,
- and an output layer that is the prediction.

The output layer sums the output of the hidden layers and ultimately returns the prediction. Like the brain's neurons, the layers represent 'neurons' interconnected that transfer the information to other layers. When one can add layers to NN, the formula (2.1.8), in the case of a NN with two hidden layers, becomes (illustration figure 3):

$$f(x_i) = a_3 + w'_3 g\,(a_2 + w'_2 g\,(a_i + W_1 x_i))\,. \qquad (2.1.9)$$

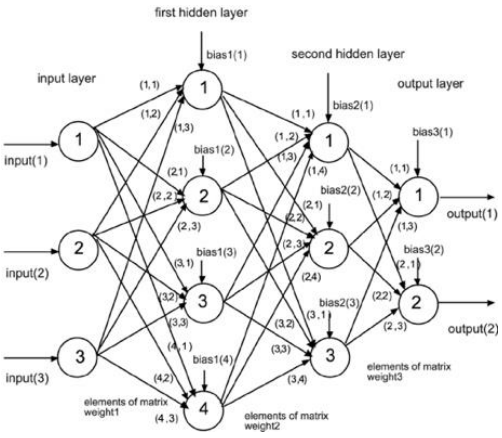


Figure 3[6]: Neural Network with two hidden layers
This figure shows a diagram of a simple neural network with two hidden layers. The first layer to the left denotes the input layer, and the last layer to the right represents the output layer. Each arrow is associated with a weight coefficient. In a neural network, the first layer transforms the input before passing it to the second layer. Likewise, the second layer transforms the input from the first layer before passing it to the output layer.

[5] Known non-linear activation function, ReLU is largely employed in deep learning because, as an overlapping activation function, it does not activate all the neurons at the same time.

[6]Source:www.researchgate.net/figure/Diagram-of-a-NN-with-two-hidden-layers_fig4_235308454

The NN is then a sequence of linear mappings that account for nonlinear interactions and add layers. The total number of parameters varies considerably from one NN to another. For a single hidden layer with H nodes, one output, and J variables, the number of parameters equals $H \times J + H + 1$. If one opts for a fully interconnected NN, one can add a hidden layer that equals the input layer. The number of parameters becomes than $H \times J + H + 1 + H \times H + H$. Adding more hidden layers to NN involves cross-features interactions.

To fit NN, one can use the sum of squared errors like for linear regression models. Considering the vector θ a vector that assembles the parameters of the NN, the goal is:

$$\min_{\theta} \sum_{i=1}^{N} [y_i - f\ (x_i, \theta)]^2. \qquad (2.1.10)$$

According to Hastie, Tibshirani, and Friedman (2009), to minimize the mean squared function, one can use numerical methods as stochastic gradient descent. Still, activation functions can be more advantageous.

In NN, it is likely to have overfitting of the training data. This would cause outstanding results in-sample and poor outcomes in out-of-sample prediction. To prevent overfitting, one can regularize the data set or add a penalty term. Scaling the inputs is primordial in that matter. One may be interested in having low magnitudes in the data set to give much more importance to the variable weights. A common approach is to scale the inputs to a 0 mean and one-unit standard deviation. This method is not necessarily the best, and researchers are encouraged to compare multiple ways (Stefan Nagel, Ross, and Duffe 2021).

The paper introduces Recurrent Neural Networks (RNN) shortly, before speaking about the LSTM model, one will apply to realize the goal. RNN is a particular type of model showing promising results. Contrary to other NN, RNN gathers information, in its internal state, from the past over a period, not fixed from the start, but is a function of the weights and the input sequence. The output is then a sequence that considers the context of the events (Bengio, Simard, and Frasconi 1994).

According to Youshua, et al. (1994), an RNN must satisfy three prerequisites. First, the model must be able to store information for each given period. Second, the model must be resistant to white noise[7]. Third, the model parameters must be trainable. That signifies that the model must employ contextual information to forecast. To this extend,

[7] This means, the model, given the data set, can make a difference in between extreme fluctuation.

contextual information must be learned. According to Hassim, et al. (2014), RNN 'contain cycles that feed the network activation from a previous time step as inputs to the network influence the prediction at the time the current time step'.

Due to its complexity, the RNN cannot learn over a period greater than 5 to 10 discrete time steps between the input and the event itself. Recent developments on a new model called Long-Short Term Memory, an extension of RNN, show improvement over classical RNN. The LSTM can learn to connect time lags in more than 1000 discrete time steps. Furthermore, it is possible to strengthen the error flow through the 'constant error carrousels' with cells (Gers, Schmidhuber, and Cummins 2000). On top of this, the LSTM can overcome the lack of RNN to work with a vast range of contextual information (Graves et al. 2009) (more details on LSTM are in the next part). Next, one introduces the notion of hyperparameter tuning.

2.1.5 HYPERPARAMETER TUNNING

Every model includes several hyperparameters that must be specified before training and testing. To make the model perform well in predictions, one may set hyperparameter values that minimize the prediction error. One thing to avoid is to use the in-sample error in the training data set because it is a biased estimate out-of-sample. For example, the in-sample error in the ridge regression is a function of estimates of $\hat{g}\ (\gamma)$ [8]and the square error:

$$mse(\gamma) = \frac{1}{N}[y - X\hat{g}(\gamma)]'[y - X\hat{g}\ (\gamma)]. \qquad (2.1.11)$$

In the training data, the model can capture the changes of the prediction y as a function of X[9] and, at the same time, it captures the statistical noise in the residual ε (Stefan Nagel, Ross, and Duffe 2021). The model complexity, therefore, is the reason how successful the model can capture statistical noise. To determine the out-of-sample estimation error, one must correct the in-sample estimation error by considering the biases introduced by the model. In linear regression models like in the OLS regression, the complexity can be the number of estimated parameters. One must apply the penalty in ridge regression because one cannot simply choose the γ to solve for $mse(\gamma)$ but also, one must consider the increase in prediction precision consisting of reduced model complexity and a higher

[8] Estimate of the penalty term where γ is the penalty term

[9] Universe of covariates

γ (Stefan Nagel, Ross, and Duffe 2021). Once one can consider the fitted valued as $\hat{y} = Hy$[10], one can then express the number of parameters as the trace of the projection matrix (Hastie, Tibshirani, and Friedman 2009):

$$d(\gamma) = tr[X\ (X'X + \gamma I_K)^{-1} X'] \tag{2.1.12}$$

with $tr[...]$ the trace operator, X the matric of covariates, X' the transpose of the matrix of covariates, and I_K the identity matrix. With $\gamma > 0$ in the ridge regression, the numbers of parameters $d < K$. In conclusion, the number of effective parameters can be considerably reduced with the shrinkage present in the model.

Once one has the effective numbers of parameters, one can then calculate the adjusted measure of fit of in-sample error. One such measure is Akaike Information Criterium (AIC). It can be helpful to get a general idea of the possible error and hyperparameters to be used to minimize the error in out-of-sample prediction. However, AIC itself required sophisticated manipulations. Given the complexity of AIC, ML practicians prefer fewer demanding methods (Hastie, Tibshirani, and Friedman 2009).

One such method, a purely data-oriented analysis, is cross-validation (CV) . The forecasts from the training data set are used to predict on a different data set, called validation data set. The errors obtained from the validation are used further as estimates for the prediction error. Stefan Nagel, Ross, and Duffe (2021) consider that the CV is a natural way to assess predictive performance. On top of this, Stone (1977) shows in his paper that both models, CV, and AIC, converge to identical results. Moreover, the error one obtains in CV can be used to minimize when using hyperparameter tuning.

In many cases, performing the simple data split in training and testing samples is not considered optimal. According to Hastie, Tibshirani, and Friedman (2009), k-fold CV improves the model's efficiency by using the complete data set for training and validation. The folds have equal sizes. One of the folds is used to validate the model and the others for training model prediction. The procedure continues till all the folds have been used (illustration figure 4).

[10] In NN, the prediction vector equals the product between the nodes matrix and the output variables

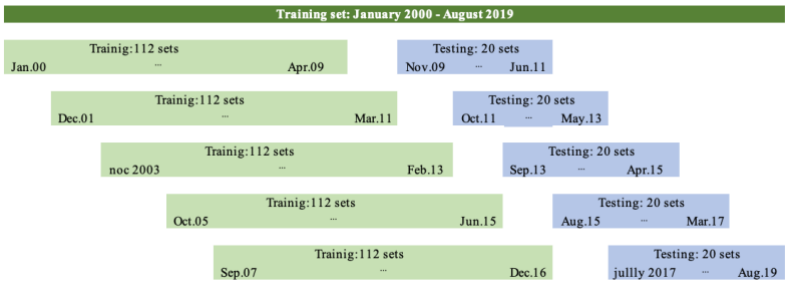


Figure 4: Cross-Validation folds: Train - Prediction sets

The figure describes the cross-validation folds one uses to train models. CV folds are subsets of the training data. One hundred twelve-monthly sets are used for training steps, 20 monthly sets are used, and three months are left out for validation.

There is not an optimal setup on what is the optimal number of folds in ML. The general trade-off is that keeping the number of folds generally small reduces the model performance. Same if one uses a large number of folds. In the case of many folds k, the estimation is overlapping at a high degree. The bias-variance trade-off consists of a large number of folds closer to the unbiased estimate but high variance. Therefore, a significant degree of uncertainty (Hastie, Tibshirani, and Friedman 2009). In ML practice, common is to use a tiny number of folds compared to the total number of observations.

Since one uses the folds to estimate the penalty $\hat{\gamma}$ to minimize the prediction error in the validation set, the penalty γ is an optimistic biased estimate of the forecast error (Stefan Nagel, Ross, and Duffe 2021).

Chapter 2. 2 ML IN ASSET PRICING

The models studied in the previous chapter are of high interest in asset pricing. In general, models in asset pricing are highly dimensional, and many of the variables can contain important predictive information. Therefore, a lot of variables can be used as predictors.

Researchers have imposed sparsity to deal with high dimensionality. Rather than studying an extensive set of predictors, the attention was focused on the marginal predictive information (Fama and French 2015a; Hou, Xue, and Zhang 2015). The big disadvantage of this method is that one obtains the same results as already studied in the literature, considering small groups of variables. Economic motivation supports approaches that can handle this high dimensionality and exploit all the potential of publicly available information (Gu, Kelly, and Xiu 2020).

The most intuitive difference between typical ML applications and asset pricing modeling is the signal-to-noise ratio in the data set. In contrast with image classification

models, in return modeling, the training data set would only capture the realized returns of the assets r_{t+1}, and not the expected return of the asset $E_t[r_{t+1}]$ because it cannot be observable. The realized return is then a noisy signal. In an asset return data set, the expected return cannot be visible through the variance of the actual asset returns. Therefore, the signal-to-noise in asset return ML models is very low (Stefan Nagel, Ross, and Duffe 2021).

In contrast to other ML models, predicting an individual outcome is not the final goal in asset pricing. Instead, investors are more interested in predicting the asset return at an aggregated level. Whether individual accurate predictions can predict, at the same time, the stock price, and the portfolio return, is still an open question in asset pricing ML (Ta, Liu, and Tadesse 2020).

Because the attention is concentrated at a portfolio level, researchers are more interested in covariances in asset pricing modeling than in any other ML. In addition, because one aggregates multiple stocks in a portfolio, its volatility can be calculated across covariances of return estimations. In ML, it is important to estimate the portfolio volatility because knowing the properties or the error covariances would define the optimal ML algorithms, the regularization process, how the performance can be evaluated and how one constructs the optimal portfolio (Stefan Nagel, Ross, and Duffe 2021).

In ML, it is essential to make a hypothesis and have information on the properties of the data set. In linear regressions, one crucial property is sparsity. While Hastie et al. (Hastie, Tibshirani, and Friedman 2009) show that sparse models perform well in algorithms for image classification and text analysis most times, there are no primary reasons (or clear explanations) to expect sparsity in asset pricing ML (Stefan Nagel, Ross, and Duffe 2021).

Another difference between asset pricing ML and fundamental ML is that in asset pricing, the training data sets are continuously supplied with outcomes of an investment decision, be that a produced outcome caused by humans or machines. Thus, the equilibrium is far from being stationary. Without an endogenous mechanism capable of considering changes in the data sets' properties, as the publicly available data changes, one must continuously readjust the weights.

This chapter treats fundamental questions in asset pricing ML such as performance measurement, data preprocessing, estimator regularization, and solutions to structural changes of the data sets.

2.2.1.1 Cross-Sectional Stock return Prediction

Return prediction is the central goal in asset pricing modeling literature. For example, researchers have been studying determinants of risk premia and market efficiency (Hodrick 1981). In quantitative investments, return estimation models help investors define investment opportunities.

In this section, one looks at the return estimation modeling as a function of its past returns. This strategy for predicting stock returns is minimal and cannot cope with the market evolutions nowadays. Therefore, there are better ways of budling a trading strategy. Strategies based on such limited information can perform, but the performance would be low, and the lessons have been learned already on the capital markets. Simple return-based prediction reveals the problems an asset manager must solve before applying ML methods in asset pricing (Stefan Nagel, Ross, and Duffe 2021).

Generally, the simple prediction of asset returns is a regression problem where the agent considers the model, takes advantage of the foster information, and adapts his expectations. This model is inspired by macroeconomics modeling[11] when first introduced by John F. Muth (Muth 1961) in 1961, later developed by Robert Lucas Jr. (Lucas and Sargent 1981) and Deidre McCloskey (McCloskey 1994). Derived from the theory of rational expectation, in asset pricing, one aims to define the expected returns by trying to approximate the function $f\ (.)$:

$$E\left[r_{i,t+1}|x_{i,t}\right] = f\ (x_{i,t}) \qquad (2.2.1.1)$$

where one maps the observable asset price variable $x_{i,t}$ into an expectation of the return and r_i is the individual stock return as a function of the market index return function of the stock i at the time t (Fama and French 2015a). Even with such a limited universe of variables, the model can be high-dimensional, and one may consider an ad-hoc sparsity to get better results.

Cross-sectional return prediction helps anticipate ups and downs in the average returns across stocks. This analysis is therefore essential in building a portfolio. In addition, firm characteristics such as size and valuation metrics were proven to help predict stock return (Fama and French 2015b).

[11] In macroeconomics, researchers have been studied how agents take decisions in uncertain environments over a period and focuses on predicting the expectations at a holistic and individual level given the evolution of economic conditions.

An analysis of cross-section expected stock return analysis was made by Lewellen J. (2013). In his paper, Lewellen studies the range of cross-section stock return variations and if the predictions are reliable. His study shows that 'expected-return-sorted portfolios are economically and statistically large among large stocks'. Also, Lewellen shows that expected returns are stable, and their predictive power appears to persist over a year (Lewellen 2013).

2.2.1.2 Predictive Performance Assessment

ML models concentrate on minimizing the sum of squared errors. Therefore, it seems that those functions of sum squared errors R^2 are suitable predictive performance measures for stock returns. However, according to Nagel, S., Ross, and Duffe (2021), R^2 being a good performance measure for stock return prediction, it is unclear if it is a suitable approach. The justification is that what works well for single stock analysis may not be appropriate at a portfolio level. Same for market performance and risk premia prediction.

Studies show that the Sharpe ratio is a function of the sum and R^2 is statistically silent about sum properties. This means that a portfolio constructed on an OLS regression has a better Sharpe ratio than other linear regressions. The discrepancy between the lack of predictive performance measures of R^2 on performance metrics is justified by the presence of statistical noise in portfolio weights[12] (Amihud and Goyenko 2012; Stefan Nagel, Ross, and Duffe 2021). In addition to the covariance matrix properties, other reasons justify the conflict between R^2 and the portfolio performance measures. The following section introduces the regularization and its implications on the portfolio performance measures.

[12] According to OLS regression properties expected portfolio return is negatively correlated with estimation error. This means that the Sharpe ratio is higher in the training set and the R^2 is lower, while out of sample the models make more mistakes, therefore the R^2 is higher and the Sharpe ratio is lower. The general awareness is that one trades on noise rather than on the signal (Hastie, Tibshirani, and Friedman 2009; Sharpe 1994; Stefan Nagel, Ross, and Duffe 2021).

2.2.1.3 Regularization and Investment Performance

Typically, portfolio managers are not directly interested in the out-of-sample R^2. This is justified by the fact that improvements in R^2 leave the weights in the portfolio unchanged. Therefore the Sharpe ratio stays the same.

Given that the variance of the portfolio is independent of the shrinkage, improvements in the expected returns will induce a higher Sharpe ratio (DeMiguel, Martin-Utrera, and Nogales 2013; Kozak, Nagel, and Santosh 2020). In other words, shrinkage can improve portfolio performance if only the covariates of expected return and estimation error are heterogeneous.

It seems that standardization or Z-score normalization is a regularization method capable of capturing the heterogeneity of the covariates and therefore entails a slighter improvement in the Sharpe ratio in the validation set. In addition, the portfolio performance in the validation data set is sensitive to the type of regularization applied to the covariates. Therefore, it is essential to have prior knowledge of the data type. In case, asset managers may consider the link between the financial time series covariates and expected returns. In the next section, one brings to attention these interconnections and the type of regularization used.

2.2.1.4 Expected Returns and Covariances

Economic reasoning discusses that covariates used to predict returns must be linked to stock return covariates. In addition, if the covariance matrix is such that a long-short portfolio shows high returns, it also must have high return volatility. Said differently, if the expected return of a small or medium company is expected to grow, and the stock is a good buy in a small company and a sell in between large stocks, the company stock should also have high volatility (Kozak, Nagel, and Santosh 2020). Various asset pricing models studied and predicted this connection between covariates and expected returns (J. M. Chen 2021; Feldman and Lepori 2016; Jack L. Treynor 1961; Sharpe 1994).

Typically, in asset pricing, covariates of forecast errors have a much more critical role than other ML applications. For example, suppose one uses returns prediction to construct a portfolio with higher returns given the same level of risk. In that case, the covariance matrix of the unpredictable component of return takes an important place in the prediction.

Theoretically, the covariance matrix is considered given. However, in practice, one must introduce an additional layer to forecast the errors. This raises a substantial issue at a portfolio level. For example, if one must construct a portfolio using thousands of stocks, finding the efficient combination implies calculating an immense covariance matrix. In addition, individual stock return data sets are imbalanced, extended over a few decades, and the covariance properties can change considerably.

In conclusion, given the above explication, it makes sense to construct a portfolio given the covariates of the return rather than the covariates of an individual stock. This process would imply a more stable portfolio covariance. At a portfolio level, the aggregation of individual stocks has a better estimate of covariance matrix as long as the number of variables K is smaller than the number of stocks N.

2.2.1.5 Nonlinearity

ML models concentrate on the nonlinearities between the covariates. NN and other methods constructed on decision trees and random forests are favorite methods because they can, in a way, successfully consider the nonlinearities between the predictors.

According to the study on ridge regression, Stefan Nagel, Ross, and Duffe (2021) show that considering second and third-order improves predictive abilities. Still, contrary decreasing the weights of these nonlinear terms helps enhance predictive power. Thus, the authors conclude that, even if their empirical study shows that downgrading the coefficients of nonlinear terms shows better results, nonlinearities are essential in asset pricing modeling (Stefan Nagel, Ross, and Duffe 2021).

It is essential to understand that, in asset pricing, the interaction between predictors is more reasonable than a simple addition of nonlinearities. For example, few return patterns can be noticed among small market capitalization stocks. Likewise, one can be more interested in variables that show stock sensitivity to investors' sentiment. In addition, studies on deep NN show that firm characteristics and momentum covariates show interactions (Gu, Kelly, and Xiu 2020).

Further studies on NN and decision trees in asset pricing show that interaction between covariates influences predictions (L. Chen, Pelger, and Zhu 2019; Gu, Kelly, and Xiu 2020). On top of this, Bryzgalova, Pelger, and Zhu (2019) show in their study how covariates interactions are essential in identifying the cross-company risk and return

(Bryzgalova, Pelger, and Zhu 2019). Other studies show one can capture the interaction effects even with a limited number of variables (Moritz and Zimmermann 2016).

2.2.1.6 Sparsity

Many models in asset pricing limit the universe of predictive variables, which is known as sparsity. Therefore, it is crucial to understand the problem of sparsity in asset pricing modeling.

Researchers have been proving acceptable results, robust predictive performance. Many studies have been made on lasso-type models, thankful to its fame in accepting sparsity (F. L. Chen and Li 2010; DeMiguel, Martin-Utrera, and Nogales 2013; Gu, Kelly, and Xiu 2020, and others). But as already said at the beginning of the paper, there is no apparent justification of sparsity in asset pricing modeling. Stefan Nagel, Ross, and Duffe 's (2021) study show that ridge regression somehow shows worse results than the lasso method or other ML models (Stefan Nagel, Ross, and Duffe 2021). This brings us to the conclusion that sparsity does not help attend a better predictive performance.

2.2.1.7 Structural change

Contrary to other fields, a data-generation process in asset pricing suffers continuous change as for any other financial applications in finance. Multiple causes are the source of those changes. First, the economy in general faces various changes from one period to another. The main drivers are the technology, legislation, institutional environment that has continuously evolved during the last few decades, including previous pandemic crises. All these changes have considerably modified the relationships between the company’s characteristics and return prediction. On top of this, investors make adaptative anticipations, as said at the start of the paper. Given adaptative anticipations, investors continuously upgrade their strategies and therefore destroy historical return patterns. All this makes that historical data becomes a liability, and investors must come with more innovative models.

In asset pricing ML research, a typical way to overcome structural change is to train parameters to adapt over time. A standard method in financial modeling is the rolling windows approach, where old data is discharged from the model analysis. Exponential weights are helpful because as one advances in the present, the confidence in the past data downgrade, and therefore, one gives less importance to the past data (Monti,

Anagnostopoulos, and Montana 2018). For illustration, figure 4 shows a five-rolling window. In this example, one tunes the hyperparameter using within windows 122 months and leaves six months out.

Even using k fold CV method, structural changes cannot be avoided or correctly implemented. Generally, timing is irrelevant in implementing cross-validation (CV) folds. In static data, it seems that one can derive validation folds for older data than the training set. With structural changes, the concept became inappropriate, and it is more logical to consider more recent data for the validation set than for the training period. In asset pricing, the direction of time is essential

2.2.2 CONCUSSION

This part introduced three typical ML methods: Linear regression, Decision Trees and Random forest, and Neural Networks. Given that financial data is more sophisticated than static data from other fields, ML application in finance demands particular adaptations. In general, ML techniques are helpful in asset pricing and investment management. One may consider models that imply a forecast of cash flow rather than return, for example.

In this part, the discussion highlights that ML methods are helpful and can produce meaningful results if one considers the specialties of the financial time series. Forecasts in asset pricing differ substantially from the initial forecasts the ML techniques were first developed. Therefore, a correctly preprocessing process is essential for attending meaningful results. On top of this, the choice of ML techniques can largely take advantage of the patterns in data, and feature sparsity becomes a liability. Given the shallow signal to noise in asset pricing data sets, one may want to let data speak and propose a fully automated and data-driven investment solution.

To conclude, it is essential to have an analytical perspective on the economic and financial data before implementing ML in asset pricing. Therefore, the next part starts with introducing the reader to the data set and data modeling before presenting the portfolio framework in the context of portfolio construction, performance measurements, and results.

Part 3 DATA SET AND METHODOLOGY

This section will describe the cross-sectional monthly stock strategy prediction framework using machine learning techniques applicable to real investment management

cases. All data manipulations are executed in Python 3, on Google Collaboratory, using GPU.

Chapter 3. 1 STUDY OF INTRACONTINENTAL EQUITIES

3.1.1 DATA SET

The paper considers the universe of the constituents of the four selected indexes: DAX Performance Index (Germany), Hang Seng Index (China), S&P 500 (US), and Tokyo Stock Price Index (Japan) using data available from Eikon Refinitiv. Four independent sets of data were constructed. Finally, the macro variables data set was collected from the Federal Reserve Economic Data (FRED). Thus, on average, one has 95% of the data collected from Eikon Refinitiv and 5% from FRED (figure 5: Data Source Frequency per portfolio). The 10-year Treasury bond is considered to be the proxy for the risk-free rate in this study.

The data starts in January 1970 and extends till October 2021. So, one gave a nominal vector (the company vector), discrete, and continuous data. So, one has roughly 65% of discrete data for the TOPX500 portfolio, 62% discrete data for the SPX500 portfolio, 43% for the HSI portfolio, and 29% for the GDAXI portfolio. The rest is continuous data.

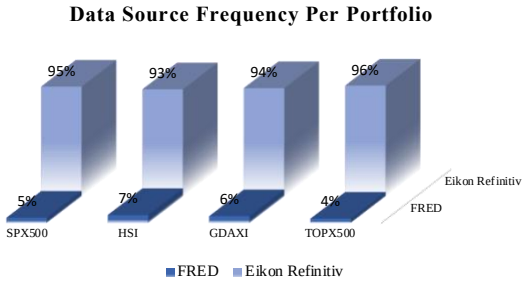


Figure 5:Data Source Frequency per Portfolio

The data is monthly, quarterly, and annually updated. However, the industry-specific variables (for example, for the industry sector: technology, aviation, etc.) are categorical data types. It is collected per company and is not updated over time. Therefore, the industry-specific information is collected once.

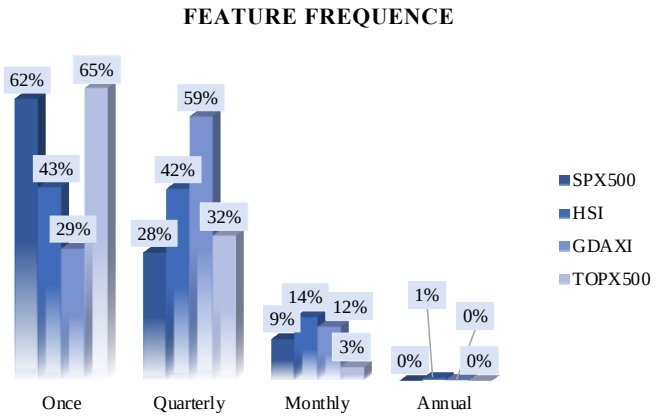


Figure 6: Feature frequency per index and index constituents together

As the indexes have preselection criteria for their members, one does not input any limit regarding the market capitalization or other liquidity metrics. Before diving into the data processing and modeling, here is a summary of the indexes, which will later be used as the benchmark portfolio.

Hang Seng Index (HSI[13])

The Hang Seng Index is one of the pioneers of the market indexes in Hong Kong. It was launched in November 1969 and counts 58 constituents. As a rule, the list of constituents strictly monitors daily price changes of the largest companies listed in the Hong Kong stock exchange. The index is calculated accordingly to the capitalization weighting method.

DAX Performance Index (GDAXI[14])

Also known as DAX, the German market index counts 40 constituents. GDAXI includes the most prominent German blue-chip companies training on the Frankfurt Stock Exchange. The market index was launched on the market in July 1988. The index is constructed based on the market capitalization weighting method.

S&P 500 (SPX500[15])

[13] Source : https://www.hsi.com.hk/eng

[14] Source : https://www.dax-indices.com/index-details?isin=DE0008469008

[15] Source : https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview

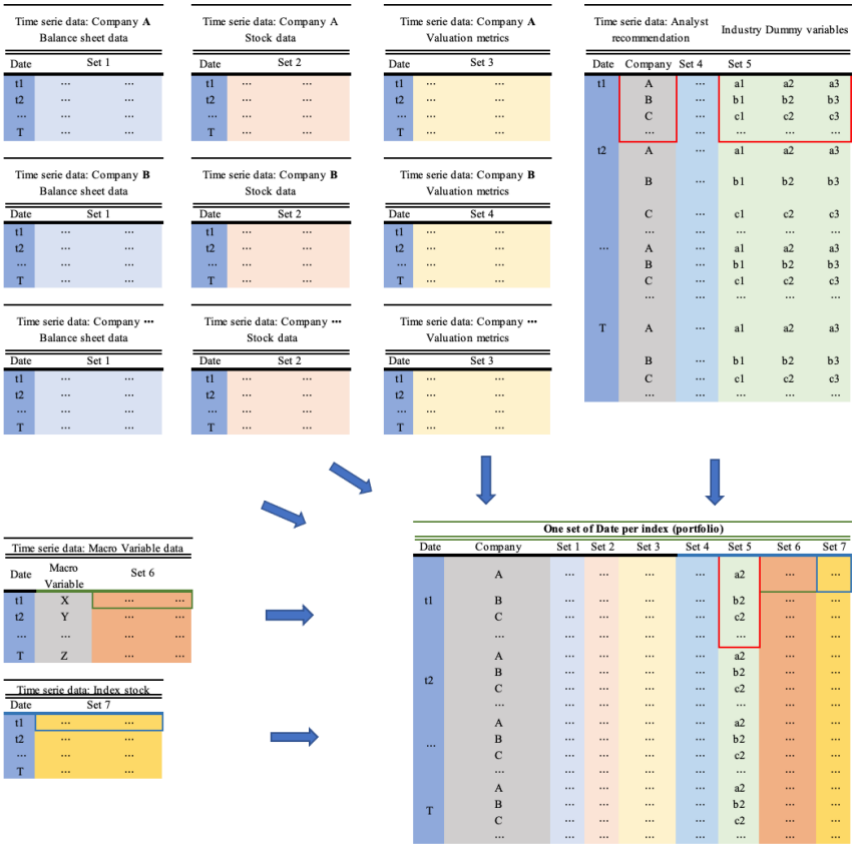
Also known as The Standard and Poor's 500, SPX500 is a market index meant to track the performance of the 500 largest companies listed in the United States. First launched in March 1957, it is composed based on the free-float capitalization weighting method.

Tokyo Stock Price Index (TOPX500[16])

Known as the Tokyo Stock Price Index, TOPX500 is a market index designed to follow up on the top 500 domestic companies listed on the Exchange's First Section. The calculus method of the index changed over time from the company weighing base to the number of shares available for training. The market index was launched in July 1969.

3.1.2 DATA MODELING

1 Organizing one distinct data set per index. The figure shows how the seven data sets are merged in one unique data set per index.



[16] Source: https://www.jpx.co.jp/english/markets/indices/topix/

To fulfill the goal of the paper, the data set was collected and organized sequentially. The data set size is particular for each index, and the variables may partially differ from one index constituent to another. The data is collected in $28 \times p$ distinct data sets (seven distinct data sets per index × four indexes), where p stands for the number of constituents of an index. Or, in another way formulated, one will have seven distinct data sets per company $\times\ p\ \times 4$ indexes. The seven distinct data sets are company data, valuation metrics data, stock market data, business, and industry data, analyst recommendation data, index data, and macro variables data. The data transformation looks like in the figure above.

First, one selects four indexes across four countries: GDAXI (Germany), HIS (China), SPX500 (US), and TOPX500 (Japan). Second, using Eikon Refinitiv monthly open price, low price, high price, close price, and volume data were collected for each index from January 1970 to September 2021, in total 621 data points per index (index date). Concomitantly, a macro variables data set is collected for each country that is represented.

On top of country-specific data, one adds one variable, which stands for the crude oil price, and four variables which stand for the exchange rate: US dollar – Japanese Yen, US dollar – Euro, US dollar – Chinese renminbi. The five variables are interdependent on the international market, and adding them to the country-specific data may favor better predictions. Each macro variables data set has a particular size because of the differences in the available data per country. US macro variable data set has 621 rows × 25 columns, the Europe macro variable data set has 621 rows × 22 columns, the Japan macro variables data counts 621 rows and 22 columns, the Chinese macro variables data set has 621 rows × 21 columns. The macro variables are collected from Federal Reserve Economic Data.

Next, company-specific data is extracted. Five data set types were identified that are collected sequentially because of limits imputed to data extraction and because collecting sequentially the data sets help build a better understanding of the data type and treatment possibilities. Among those five data sets, the data corresponds to balance sheet data, valuation metrics, financial ratio, stock price, industry-related information, and analyst recommendation[17]. Except for the industry-related information, all these vectors represent a matrix disposing of 621 rows and m columns. The m represents the number of features

[17] Check Table 1 in appendix for broader look at the feature rows.

per vector type, and the number varies from one company to another and from one index set to another. Generally, one has the following numbers of features per data set:

- HSI: $m_{Balance\ Sheet} = 130$, $m_{Valuation} = 27$, $m_{Stock\ Price} = 4$.
- GDAXI: $m_{Balance\ Sheet} = 220$, $m_{Valuation} = 27$,$m_{Stock\ Price} = 4$.
- SPX500: $m_{Balance\ Sheet}$ =180, $m_{Valuation} = 27$, $m_{Stock\ Price} = 4$.
- TOPX500: $m_{Balance\ Sheet} = 180$, $m_{Valuation} = 27$, $m_{Stock\ Price} = 4$.

The analyst report data set has $612 \times z$ rows, where z is the number of analysts who gave a recommendation at the time t, and four columns: data, broker name, 1-5 broker recommendation, and broker estimate. The column broker estimate represents the strategy (Buy, Hold, Sell, or others). Pivot tables are used for this data set were to group by the date the brokers and use the column broker estimates as values. As one obtains a considerable number of recommendations for the same date, one is only interested in recording the most common recommendation at the time t. Therefore, one creates a new column, 'Strategy' in the data set where the most frequent strategy at the time t is stored. If more than one strategy with the same frequency was identified, one selects the first with the highest frequency. This paper considers the strategy as given because, recommended by the largest brokers, it influences the overall investor sentiment.

Before merging the company-specific data sets, manipulations are needed regarding the industry-related information data. Generally, the data set has k columns and l rows, where k are the number of constituents per index. The l columns are instrument (same as index constituent), economic sector name, business sector name, industry group name, industry name, activity name, exchange name. It is not a time series data set like the other six data sets, and it is a categorical type of data. To merge it with the other data set, one first makes it a Dummy Variables data set. The final data has 124 columns for HSI constituents, 110 columns for GDAXI constituents, 372 columns for SPX500 constituents, and 363 columns for TOPX500 constituents. At this step, the data set is still not complete. Therefore, while merging the seven data sets, one copies the line corresponding to the index constituent at every 621 time points (figure 7 shows how one executes the combination of the two data sets).

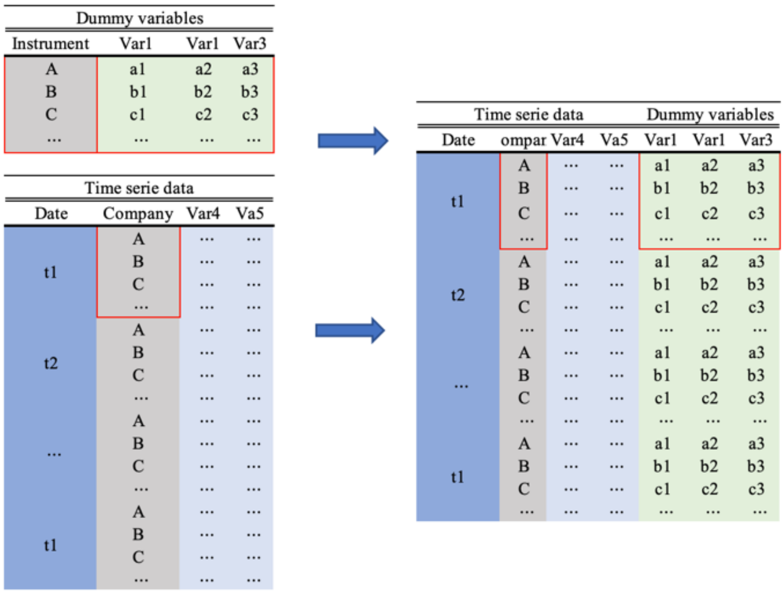


Figure 7: Data merging

The figure describes how one combines the time series data frame and the non-time series dummy variables.

Now that all these manipulations are done, one can merge the seven data sets can be merged into one data set per company. In the case of HSI, one obtains 58 data frames, which, on average, represents a matrix type $621\ rows \times 311\ columns$. As a reminder, the columns of a data set represent the number of variables. Following the same logic, one has matrix type 621×388 for GDAXI representants, $621 \times 1{,}118$ for SPX500 representants, and $621 \times$ for TOP500 representants. To generate and work with portfolios, combining the data sets in one single data frame appears necessary. This implies one combines the data frames which correspond to the constituents of the HSI, for example. Now the data collection and transformation are accomplished for further analysis. The final data set should have a shape type:

$$\begin{pmatrix} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Balance\ Sheet} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Valuation} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Stock\ Price} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Industry} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Analyst\ report} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Index} \overbrace{\begin{bmatrix} a_{111} \cdots a_{11m} \\ \ddots \\ a_{1n1} \cdots a_{1nm} \end{bmatrix}}^{Macro} \\ \vdots \\ \begin{bmatrix} a_{t11} \cdots a_{n1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \begin{bmatrix} a_{t11} \cdots a_{t1m} \\ \ddots \\ a_{tn1} \cdots a_{tnm} \end{bmatrix} \end{pmatrix}$$

where t is the time in months and $t = 621$, n is the number of companies listed in the index at the date t, and m is the number of features for each vector. Thus, on average,

one has: $n_{HSI} = 58$, $n_{GDAXI} = 40$, $n_{SPX500} = 500$, $n_{TOPX500} = 500$ companies at time t.

The working data set will be a matrix type ($621 \times number\ of\ index\ represenatnts\ at\ the\ date\ t$ rows) and the same number of columns as the individual company matrix + the company-specific data which is available for a single company in case it exists. For example, in the HSI portfolio case, one will have a matrix with $621\ months \times 58\ companies = 36{,}018$ rows and 311 columns. Same reasoning for the following three portfolios.

One more step is necessary to finalize the data set modeling. The strategy vector for the four portfolios represents a categorical feature. The analysis shows that the vector has 14 unique values for the GDAXI portfolio, 28 for HIS portfolio, 113 unique values for the SPX500 portfolio, and 94 unique values for the TOPX500 portfolio. That means one does not have only a Buy, Hold, or Sell recommendation but also an additional strong sell, strong buy, etc., analyst recommendation. A new vector, 'Strategy nb' is added, where after a numerical transformation, where '1' replaces the Buy group recommendations, '0' the Hold group recommendations, and '-1' the Sell group recommendation. Complete details of the transformation in Appendix Data Transformation: Strategy.

All this done, a universe of stocks filtered in advance is available for modeling for the purpose of this paper. Let's recapitulate where one is right now: a universe of stock was selected based on some criteria. Next, using the available information, one predicts the next strategy recommendation: Sell, Buy or Hold to build long-only, short-only, and long-short investment strategies. Once one knows what the next recommendation should be, the portfolio is constructed. Then the portfolio performance is compared with the index itself, called the benchmark portfolio.

3.1.3 DATA DESCRIPTION

Overall, the company-specific data is right-skewed distributed (generally, the mean is greater than the median (figures 8-11)). The data has a long tail to the right. Figures 8 to 11 describe the central tendency and variability of the data set of the 30 most predictive variables for each portfolio set (details on reading the macro variables acronyms are in the appendix Macro variables). Given these findings, one affirms that the financial data has a lower boundary.

	count	mean	std	min	25%	50%	75%	max
GDPCI	285'713.00	15'392.74	2'496.88	10'097.36	13'477.36	15'671.61	17'437.08	19'465.20
Close	280'016.00	60.99	130.52	0.05	18.63	35.26	63.93	5'222.60
HQMCB10YRP	285'713.00	4.98	1.64	1.93	3.59	5.00	6.35	8.64
Return On Equity - Mean	189'939.00	21.94	96.29	-2'524.08	10.59	16.82	25.45	6'616.48
IRLTLT01USM156N	285'713.00	3.67	1.70	0.62	2.26	3.56	4.91	7.96
MABMM301USM189S	285'713.00	9'114'544'532'105.79	4'379'773'645'608.88	3'456'700'000'000.00	5'494'400'000'000.00	8'427'600'000'000.00	12'192'500'000'000.00	20'797'000'000'000.00
IEABC	244'512.00	-124'641.36	38'826.99	-218'442.00	-156'229.00	-109'987.00	-97'269.00	-60'838.00
MISL	285'713.00	2'975.33	3'951.19	1'063.80	1'190.70	1'617.50	3'043.00	19'862.20
TR.F.ComShrTrezIssue	274'881.00	77'319'402.61	315'535'707.74	0.00	0.00	449'995.50	35'342'987.00	4'912'000'000.00
P/E (Daily Time Series Ratio)	202'801.00	73.18	772.71	0.32	16.27	22.33	33.78	30'031.45
EXUSEU	244'512.00	1.20	0.15	0.85	1.11	1.20	1.32	1.58
PPIACO	285'713.00	170.40	31.66	118.60	135.00	181.30	199.20	235.40
Price / EPS (Mean Estimate)	243'689.00	43.92	512.14	0.45	13.92	18.36	25.59	43'139.00
.SPX HIGH	285'713.00	1'666.95	856.74	452.79	1'130.38	1'376.38	2'111.05	4'537.36
AAA10YM	285'713.00	1.48	0.44	0.64	1.12	1.53	1.80	2.68
NA000352Q	285'713.00	426'472.99	153'220.61	168'996.00	260'704.00	442'272.00	569'081.00	645'702.00
HQMCB30YRP	285'713.00	5.65	1.48	2.79	4.41	5.79	6.84	8.88
Company Market Cap	255'262.00	29'327'880'586.40	74'975'001'074.77	25'983'482.14	4'078'980'597.62	10'530'646'840.98	25'123'948'121.62	2'185'627'522'856.67
USSTHPI	285'713.00	326.96	81.88	180.30	267.33	330.30	376.41	510.08
BAA	285'713.00	6.12	1.54	3.16	4.80	6.20	7.49	9.32
.SPX LOW	285'713.00	1'559.75	810.28	435.86	1'049.72	1'282.86	1'993.26	4'367.73
UNRATE	285'713.00	5.84	1.84	3.50	4.50	5.40	6.50	14.80
Total Return	254'910.00	1.81	9.19	-57.74	-2.85	1.78	6.31	106.90
PAYEMS	285'713.00	135'070.53	8'864.39	111'733.00	130'511.00	134'159.00	141'525.00	152'523.00
TR.F.DebtLTMatYr3	230'164.00	1'500'995'246.71	5'885'289'353.69	0.00	10'151'000.00	244'700'000.00	992'000'000.00	115'600'000'000.00
TRBC Economic Sector Name_Utilities	285'713.00	0.06	0.24	0.00	0.00	0.00	0.00	1.00
Low	280'016.00	56.85	121.75	0.04	17.12	32.82	60.00	5'055.00
Open	280'016.00	60.38	128.43	0.05	18.50	35.07	63.53	5'267.59
TR.F.DebtLTMatTot	218'546.00	0.02	0.07	0.00	0.00	0.01	0.02	1.00
IPB50001SQ	235'695.00	0.62	0.26	0.00	0.41	0.70	0.84	1.00

Figure 8: SPX500, Data description

The figure describes the statistics of the 30 most predictive features for the SPX500 data set. One sees that ½ of the top 30 predictive variables are macro-economic variables. This shows that stock return (among SPX500 constituents) is influenced mainly by economic cycles. Company data seems to have a fewer influence on the investor sentiment, and finally on the strategy.

	count	mean	std	min	25%	50%	75%	max
Volume	8894.00	63089837.14	93189641.10	4850.22	9896177.66	25837743.75	77472107.13	923635408.50
CSCICP03EZM665S	8894.00	99.79	1.65	95.82	98.73	99.85	101.07	102.83
Return On Equity - Mean	8894.00	14.39	10.12	-45.49	9.87	13.94	18.92	78.48
MABMM301EZM189S	8894.00	9504778665954.58	27039227975 88.91	4723291000000.00	7152523000000.00	9490159000000.00	11452748000000.00	15148862000000.00
Price / EPS (Mean Estimate)	8894.00	20.17	21.13	4.40	12.07	15.61	20.60	264.26
MYAGM2EZM196N	8894.00	8306367176059.14	2275096106691.73	413365100000.00	6168737000000.00	8568031090000.00	10797860000000.00	10876141000000.00
TR.F.InvstInAssocJVsUnconsolSubs	8894.00	2921439747.66	6002911392.93	0.00	29100000.00	545362838.28	3398000000.00	45309152635.97
TR.F.ComShrIssuedTot	8894.00	804375898.46	1304819078.43	36605169.88	193000000.00	455300000.00	914203421.00	10247043439.82
TR.F.ComprIncAccumTot	8894.00	-133568866.09	2581463991.41	-7514000000.00	-940000000.00	-105750000.00	134000000.00	18264000000.00
EXJPUS	8894.00	106.72	12.38	76.64	102.06	108.85	115.48	133.64
RBXMB1S	8894.00	97.84	6.49	83.45	93.17	96.40	103.20	111.58
TR.F.DebtLTMatIn1Yr	8894.00	6291851958.70	160486671 18.98	0.00	185263803.31	844000000.00	3362000000.00	187044000000.00
TR.F.OthNonCurrLiab	8894.00	5010587578.66	20360599142.50	-3538139817.88	25900000.00	234000000.00	1576000000.00	138228000000.00
POILBREUSDM	8894.00	64.13	28.53	18.60	43.06	62.16	77.79	133.90
P/E (Daily Time Series Ratio)	8894.00	28.72	54.24	1.05	12.55	18.25	25.97	887.05
TRBC Business Sector Name_Mineral Resources	8894.00	0.03	0.17	0.00	0.00	0.00	0.00	1.00
EXUSEU	8894.00	1.21	0.15	0.85	1.11	1.20	1.32	1.58
TR.F.MinIntrEq	8894.00	1296189085.14	3403624956.83	-18000000.00	31400000.00	239000000.00	1055000000.00	36628000000.00
TR.F.OthNonCurrAssetsTot	8894.00	1415495303.47	5854438927.80	-36204510000.00	107879000.00	744000000.00	1853725000.00	50486000000.00
Low	8894.00	53.19	50.07	0.38	20.47	38.10	68.87	510.40
Cash & Short Term Investments - Total	8894.00	4559285064.11	7480040026.81	6712000.00	520300000.00	1837000000.00	4853000000.00	65327888888.89
BPBLTT01EZQ636S	8894.00	17808002023.84	23629725995.99	-46351000000.00	803000000.00	20231000000.00	39871000000.00	39871000000.00
High	8894.00	60.63	56.79	0.82	23.47	43.96	77.79	578.80
TR.F.ComShrOutsTot	8894.00	671814331.63	829608348.64	1675439.83	183521257.00	434777878.00	840308716.59	5684661896.62
TR.F.ComStockIssuedPaid	8894.00	1192121878.25	1998095532.08	0.00	190000000.00	563700000.00	1340000000.00	12189000000.00
Open	8894.00	56.95	53.38	0.51	21.94	41.04	73.14	511.80
TR.F.IntangAccumAmortTot	8894.00	3301683593.49	6724924023.26	546000.00	317000000.00	1022400000.00	3393000000.00	56641000000.00
AAA10YM	8894.00	1.56	0.42	0.64	1.27	1.60	1.84	2.68
TR.F.CurrPortOfLTDebtCapLeases	8894.00	4955948238.25	12594479963.17	0.00	94693000.00	639000000.00	2793000000.00	89757000000.00
TR.F.GoodwCostInExcessOfAssetsPurchNet	8894.00	6573347012.21	7445003684.81	0.00	1031000000.00	2905000000.00	10767035000.00	40597000000.00

Figure 9: GDAXI, Data description.

The figure describes the statistics of the 30 most predictive variables for the GDAXI data set. One sees 1/3 of the predictive variables are macro-economic variables. Therefore, one could hypothesize that, given the region, economic circumstances, and development opportunities, company-specific data starts to weight more on the strategy for the GDAXI constituents.

Feature	count	mean	std	min	25%	50%	75%	max
TRBC Business Sector Name_Utilities	11875	0.10	0.30	0.00	0.00	0.00	0.00	1.00
RBCNBIS	11875	108.73	15.34	81.83	94.79	109.58	122.44	130.94
TR.F.IncTaxPbleLTST	11875	5866659347.95	15141659406.90	0.00	110353000.00	553000000.00	3095000000.00	109601000000.00
TR.F.PPEAccumDeprTot	11875	67031206844.36	228370289859.02	0.00	787000000.00	3735000000.00	17090000000.00	2220267000000.00
CSCICP03CNM665S	11875	99.84	2.82	94.93	97.55	99.56	101.26	105.51
TR.F.TotCurrAssets	11875	78657366423.11	144535952180.04	18933217.47	6062000000.00	23288043000.00	86593500000.00	1742156000000.00
TR.F.MinIntrTot	11875	12523392660.18	35149771046.94	-562080846.74	83912000.00	1042900000.00	5792000000.00	334366000000.00
TR.F.TotLiab	11875	1000470263196.26	3375831131227.87	-118754000000.00	8590854594.80	50043000000.00	294060000000.00	30435543000000.00
TR.F.FomCcyTranslAdjAccum	11875	-1649011990.96	7079317558.52	-43452000000.00	-379000000.00	-28000.00	180626000.00	172800000000.00
AAA10YM	11875	1.55	0.42	0.64	1.27	1.60	1.83	2.68
TR.F.OthAssetsTot	11875	27090553219.07	67251127403.38	-7889000000.00	368804000.00	3777578000.00	20626676000.00	937757000000.00
TR.F.ComShrOutsTot	11875	25899839018.96	61823421193.44	558866825.00	2221050000.00	4838401583.08	16677182310.30	356406257089.00
TRBC Economic Sector Name_Utilities	11875	0.10	0.30	0.00	0.00	0.00	0.00	1.00
TR.F.DefTaxInvstTaxCreditsLT	11875	3723093306.32	5730935963.98	0.00	69766535.85	997558000.00	5480000000.00	59598000000.00
LMUNRRTTCNQ156S	11875	4.12	0.07	4.00	4.10	4.10	4.10	4.30
TR.F.ComShrOutsIssue	11875	16726816286.65	32880917610.48	479051560.00	2162481000.00	4781471827.92	12609922800.50	178203128544.50
TR.F.TotLiabEq	11875	1189301516052.73	3663294965347.14	25181667.48	24093205000.00	132693100000.00	571983000000.00	33345058000000.00
TR.F.MinIntrEq	11875	12523392660.18	35149771046.94	-562080846.74	83912000.00	1042900000.00	5792000000.00	334366000000.00
TR.F.ComStockIssuedPaid	11875	36061656125.34	80405496189.09	0.00	242698000.00	2136000000.00	19285000000.00	402130000000.00
TR.F.ComShrIssuedTot	11875	25813230475.64	61845997836.85	539504316.00	2221050000.00	4838401583.08	15832284184.00	356406257089.00
TR.F.ProvTot	11875	7136421536.88	16922633655.31	0.00	106895000.00	1365000000.00	7243300000.00	159346000000.00
TR.F.CashSTInvstTot	11875	145732718729.80	509136247238.21	0.00	2491284000.00	10265815000.00	47444000000.00	3613872000000.00
TRBC Activity Name_Electric Utilities (NEC)	11875	0.05	0.22	0.00	0.00	0.00	0.00	1.00
TR.F.ComEqTot	11875	1741341131 03.65	318624450952.20	-127272361000.00	12138678000.00	56137000000.00	176714000000.00	2667683000000.00
TR.F.AssetAccruals	11875	196482124929.82	385410715750.97	-368721000000.00	7792482000.00	58781567000.00	212824000000.00	2635714000000.00
TR.F.InvstTot	11875	2968364020 08.54	1012464368528.77	0.00	1288000000.00	10089389000.00	110759000000.00	8778477000000.00
MANMM101CNM189S	11875	304702155993 89.60	172914964503 50.80	45005125866 98.68	14321331204 493.00	29465643950 441.40	513367950174 96.20	5431448057247 4.70
TR.F.CashCashEquiv	11875	19976538162.46	38406544389.76	10000000.00	1622000000.00	6393724000.00	19364000000.00	368721000000.00
TR.F.PPENetTot	11875	94011924647.43	258921635138.02	4524739.66	1794595738.39	10865000000.00	50358000000.00	2263535000000.00
TR.F.PbleAccrExpn	11875	31501913294.60	66406579627.49	0.00	1442479000.00	4003881000.00	21377000000.00	561743000000.00

Figure 10: HSI, Data description

The figure describes the statistics of the 30 most predictive variables for the constituents of the HSI portfolio. One sees that only 1/6 of the most predictive variables are macroeconomic variables. One may think that, in Asia, companies are their own blacksmith.

Features	count	mean	std	min	25%	50%	75%	max
Return On Equity - Mean	156180	0.50	0.01	0.00	0.49	0.50	0.50	1
Company Market Cap	156180	0.02	0.04	0.00	0.01	0.01	0.02	1
MYAGM1JPM189S	156180	0.64	0.28	0.00	0.53	0.59	0.90	1
QJPN628BIS	156180	0.27	0.24	0.00	0.08	0.19	0.35	1
TR.F.RetainedEarnTot	156180	0.08	0.04	0.00	0.07	0.07	0.08	1
TR.F.ComStockTrezRepurch	156180	0.02	0.03	0.00	0.02	0.02	0.02	1
TR.F.AccrualsST	156180	0.65	0.03	0.00	0.65	0.65	0.66	1
Close	156180	0.01	0.02	0.00	0.00	0.01	0.02	1
Enterprise Value To EBITDA (Daily Time Series Ratio)	156180	0.12	0.01	0.00	0.12	0.12	0.12	1
TR.F.WkgCapExclOthCurrAssetsLiab	156180	0.25	0.04	0.00	0.24	0.24	0.25	1
TR.F.LandBuildNet	156180	0.06	0.12	0.00	0.01	0.02	0.05	1
TR.F.ComStockIssuedPaid	156180	0.02	0.06	0.00	0.00	0.01	0.02	1
MYAGM2JPM189S	156180	0.53	0.34	0.00	0.24	0.47	0.91	1
P/E (Daily Time Series Ratio)	156180	0.00	0.01	0.00	0.00	0.00	0.00	1
TR.F.LandImprovNet	156180	0.07	0.16	0.00	0.00	0.01	0.04	1
Low	156180	0.02	0.02	0.00	0.01	0.01	0.02	1
EXCHUS	156180	0.49	0.35	0.00	0.19	0.35	0.91	1
TR.F.TotShHoldEq	156180	0.04	0.05	0.00	0.02	0.02	0.03	1
TR.F.STInvstTot	156180	0.01	0.03	0.00	0.00	0.00	0.00	1
TR.F.EqNonContribRsrvRetainedEarn	156180	0.07	0.04	0.00	0.06	0.06	0.07	1
TR.F.LoansRcvblNetST	156180	0.02	0.07	0.00	0.00	0.01	0.02	1
TR.F.TradeAcctTradeNotesRcvblGross	156180	0.02	0.04	0.00	0.00	0.01	0.02	1
TR.F.LoansRcvblTot	156180	0.01	0.06	0.00	0.00	0.00	0.00	1
TR.F.OthCurrLiabTot	156180	0.23	0.03	0.00	0.22	0.22	0.23	1
Open	156180	0.01	0.02	0.00	0.00	0.01	0.01	1
TR.F.TotLiabEq	156180	0.01	0.05	0.00	0.00	0.00	0.00	1
BPBLTD01JPQ637N	156180	0.45	0.40	0.00	0.00	0.49	0.85	1
TRESEGJPM052N	156180	0.66	0.28	0.00	0.50	0.73	0.87	1
TR.F.MinIntrNonEq	156180	0.03	0.08	0.00	0.01	0.01	0.02	1
TRBC Industry Group Name_Electric Utilities & IPPs	156180	0.02	0.15	0.00	0.00	0.00	0.00	1

Figure 11: TOPX500, Data description

The figure describes the statistics of the 30 most predictive variables for the constituents of the TOPX500 portfolio. As to the HSI constituents, 1/6 of most predictive variables are macro-variables. Therefore, it appears that company characteristics variables are most predictive for the TOPX500 constituents.

In general, the company itself most influences the performance of portfolios constructed on Asian companies. It seems there may be region-specific biases. Next, data treatment processes, normalization, and the portfolio construction framework are described before discussing the results.

3.1.4 TREATING THE MISSING VALUES AND OTHER TRANSFORMATIONS

Most companies register information from 1995 on. Therefore, one considers the start date of analysis January 2000. Next, one drops all the columns that are missing more than 80% of the data.

On top of this, the data set is supplemented with 12 momentum columns derived from the Total Return column. The Total return is collected from Eikon Refinitiv with the other valuation metrics. A time lag from 1 to 12 is used to obtain the past returns over a time period of 1 to 12 months. At the same time, we drop the duplicates rows. Then, one uses a simplistic method filling the missing values with the past value, and if not available, with the next value. Once the step is accomplished, one has:

HSI: $23{,}912\ rows \times 751\ columns \rightarrow 21{,}783\ rows\ \times 227\ columns$;

GDAXI: $19{,}305\ rows \times 743\ columns\ \rightarrow 8{,}894\ rows\ \times 268\ columns$;

SPX500:$285{,}713\ rows\ \times\ 1{,}130\ columns\ \rightarrow\ 129{,}192\ rows\ \times 513\ columns$;

TOPX500:$284{,}234\ rows \times 1{,}035\ columns\ \rightarrow 156{,}180\ rows \times 526\ columns$.

3.1.5 NORMALIZATION

Our variables are measured at different scales and cannot contribute equally to the model training and testing, creating biases. Some manipulations are necessary to improve numeric stability (Kuhn and Johnson 2013). The normalization technique is applied. This technique will benefit the CatBoost and the LSTM model used for predictions.

To concentrate exclusively on the cross-sectional return analysis predictability, one performs a specific normalization on the characteristics later used as predictors. For each variable x of the stock i at the time t, one applies a Min-Max scaling before the model training. One has, therefore:

$$x_{t,i,scaled} = \frac{x_i - \min\ (X_i)}{\max(X_i) - \min\ (X_i)}$$

where X_i Is the entire vector of values for the variable x_i. In the RNN model, the Min-Max scaling ensures a faster and more stable backpropagation.

It's important to understand that using monthly data directly to estimate future portfolio returns considers uncertainty in covariance estimates. Thus, to calculate portfolio return, the paper considers that the individual stocks are equally weighted. Every month Long-only, Short-only, or Long-Short portfolios are rebalanced.

3.1.6 FEATURE CORRELATION

Correlation is a statistical measure and refers to the relationship between variables. Generally, portfolio variables are positively correlated. High correlation among balance sheet type of data is recorded. Correlation decreases when one adds valuation metrics, index data, macro-variables, and industry data. A high correlation is also recorded between stock market variables (index price, stock price, etc.). However, on average, the correlation between variables is low or medium. Given this, one can use all the given variables in seeking prediction portfolio returns. Interesting would be to include negatively correlated features to see if the portfolio returns would significantly improve, but this could be another study.

A last analysis on the most predictive features is run before setting the training, validation, and testing sets. Spearman's rank correlation coefficient is used to check if the variables are monotonically correlated with the strategy Buy, Hold or Sell. Figures 15 to 18 give more insights on feature correlation.

General intuition is that for the SPX500 portfolio constituents, the most predictive variables are the macro variables that influence the SPX500 portfolio constitution. The macro variables' importance decreases among small region-specific portfolios.

Feature 1	Feature 2	corr
MABMM301USM189S	M2SL	1
TR.F.CashCashEquivTot	TR.F.CashCashEquiv	1
TR.F.ShHoldEqCom	TR.F.ComEqTot	0.999996813
TR.F.NetBookCap	TR.F.NetOpAssets	0.999986481
TR.F.TotAssets	TR.F.TotLiabEq	0.999904758
TR.F.PPEExclAssetsLeasedOutAccumDeprTot	TR.F.PPEAccumDeprTot	0.999795092
TR.F.PPEExclAssetsLeasedOutGross	TR.F.PPEGrossTot	0.999591394
Total Capital	Total Book Capital	0.99942203
TR.F.TotCap	TR.F.TotBookCap	0.999391853
TR.F.DebtInclPrefEqMinIntrTot	TR.F.DebtInclFinOpLeaseLiabSuppl	0.999127682
Open	High	0.99806095
TR.F.TotLiabEq	TR.F.TotLiab	0.99775838
TR.F.TotLiab	TR.F.TotAssets	0.997689285
TR.F.LTDebtExclCapLease	TR.F.DebtNonConvertLT	0.997619643
TR.F.PPEExclAssetsLeasedOutNetTot	TR.F.PPENetTot	0.997617153
Close	Low	0.997541945
High	Close	0.997252083
TR.F.DebtLTMatRemain	TR.F.DebtLTMatYr6Beyond	0.997219306
TR.F.DebtLTTot	TR.F.LTDebtExclCapLease	0.995877331
Low	Open	0.995651308
Low	High	0.99516324
Close	Open	0.995074546
TR.F.NetBookCap	TR.F.TotCap	0.994784492
TR.F.NetOpAssets	TR.F.TotCap	0.994745997
TR.F.DebtNonConvertLT	TR.F.DebtLTTot	0.994537226
TR.F.TotBookCap	TR.F.NetBookCap	0.994507679
TR.F.TotBookCap	TR.F.NetOpAssets	0.994438867
TR.F.ShHoldEqParentShHoldTot	TR.F.ShHoldEqCom	0.994174991
TR.F.ShHoldEqParentShHoldTot	TR.F.ComEqParentShHold	0.994172024
TR.F.ComEqTot	TR.F.ShHoldEqParentShHoldTot	0.994172024

Feature	Correlation
AAA	0.172171
HQMCB30YRP	0.166379
HQMCB10YRP	0.155068
IRLTLT01USM156N	0.146392
BAA	0.146347
BOPGSTB	0.131241
IEABC	0.130788
TRBC Economic Sector Name_Technology	0.110545
EXCHUS	0.107489
FEDFUNDS	0.096191
UMCSENT	0.087334
TRBC Industry Group Name_Software & IT Services	0.077461
TRBC Business Sector Name_Software & IT Services	0.077461
AAA10YM	0.075699
TRBC Business Sector Name_Energy - Fossil Fuels	0.074044
TRBC Economic Sector Name_Energy	0.074044
TRBC Business Sector Name_Technology Equipment	0.072937
.SPX HIGH	0.069450
.SPX OPEN	0.065134
TRBC Activity Name_Oil Related Services and Equipment (NEC)	0.064095
TRBC Industry Name_Oil Related Services and Equipment	0.064095
TRBC Industry Name_Software	0.061778
TRBC Industry Name_Semiconductors	0.059728
TRBC Industry Group Name_Semiconductors & Semiconductor Equipment	0.059156
.SPX CLOSE	0.055081
TRBC Industry Group Name_Oil & Gas	0.054736
TRBC Activity Name_Aerospace & Defense (NEC)	0.053117
TRBC Activity Name_Broadcasting (NEC)	0.051750
TRBC Activity Name_Semiconductors (NEC)	0.051445
TRBC Industry Group Name_Oil & Gas Related Equipment and Services	0.050439

Figure 12: SPX500, feature correlation

On the left, SPX500, the 30 most correlated features. The figure describes 30 most correlated features within the SPX500 portfolio. On the right: SPX500: Spearman Rank correlation is used to class 30 most correlated features with company-based investment Strategy.

Feature 1	Feature 2	Corr
TR.F.CashCashEquivTot	TR.F.CashCashEquiv	1.000000
TR.F.CashCashEquiv	TR.F.CashCashEquivTot	1.000000
TR.F.CashSTInvstTot	TR.F.CashSTInvst	1.000000
TR.F.CashSTInvst	TR.F.CashSTInvstTot	1.000000
TR.F.NetOpAssets	TR.F.NetBookCap	1.000000
TR.F.DebtNonConvertLT	TR.F.LTDebtExclCapLease	1.000000
TR.F.ComEqParentShHold	TR.F.ShHoldEqCom	1.000000
Common Shares - Outstanding - Total	TR.F.ComShrOutsTot	0.999980
TR.F.ShHoldEqCom	Shareholders Equity - Common	0.999962
Shareholders Equity - Common	TR.F.ShHoldEqCom	0.999962
TR.F.ComEqParentShHold	Shareholders Equity - Common	0.999962
Shareholders Equity - Common	TR.F.ComEqParentShHold	0.999962
TR.F.ShHoldEqParentShHoldTot	Shareholders Equity - Common	0.999961
TR.F.DebtNonConvertLT	TR.F.DebtLTTot	0.999955
TR.F.DebtLTTot	TR.F.DebtNonConvertLT	0.999955
TR.F.LTDebtExclCapLease	TR.F.DebtLTTot	0.999955
TR.F.DebtLTTot	TR.F.LTDebtExclCapLease	0.999955
TR.F.ComEqTot	Shareholders Equity - Common	0.999953
Working Capital	TR.F.WkgCap	0.999952
TR.F.TotCap	TR.F.TotBookCap	0.999843
TR.F.TotBookCap	TR.F.TotCap	0.999843
TR.F.TradeAcctPbleAccrualsST	TR.F.PbleAccrExpn	0.999784
TR.F.PbleAccrExpn	TR.F.TradeAcctPbleAccrualsST	0.999784
Intangible Assets - Total - Net	TR.F.IntangTotNet	0.999766
TR.F.DebtInclPrefEqMinIntrTot	TR.F.DebtInclFinOpLeaseLiabSuppl	0.999762
TR.F.DebtInclFinOpLeaseLiabSuppl	TR.F.DebtInclPrefEqMinIntrTot	0.999762
Total Capital	TR.F.TotCap	0.999733
TR.F.TotCap	Total Capital	0.999733
TR.F.ComShrIssuedIssue	TR.F.ComShrOutsIssue	0.999711
TR.F.PPEExclAssetsLeasedOutAccumDeprTot	TR.F.PPEAccumDeprTot	0.999652

Feature	Correlation
POILBREUSDM	0.282222
EA19XTEXVA01CXMLM	0.271447
EXUSEU	0.268958
CLVMEURSCAB1GQEA19	0.268215
QXMN628BIS	0.265359
PPIACO	0.264411
MYAGM2EZM196N	0.246075
MABMM301EZM189S	0.246054
RBXMBIS	0.225664
.GDAXI Volume	0.222219
PRINTO01EZQ661S	0.218136
PRMNVG01EZQ661S	0.205752
TR.F.ComShrIssuedTot	0.139576
TRBC Activity Name_Multiline Utilities	0.133482
TRBC Industry Group Name_Multiline Utilities	0.133482
TRBC Business Sector Name_Utilities	0.133482
TRBC Industry Name_Multiline Utilities	0.133482
TRBC Economic Sector Name_Utilities	0.133482
TR.F.RetainedEarnTot	0.107011
TR.F.CashSTInvstTot	0.103539
TRBC Economic Sector Name_Consumer Non-Cyclicals	0.100042
TRBC Industry Group Name_Consumer Goods Conglomerates	0.100042
TRBC Business Sector Name_Consumer Goods Conglomerates	0.100042
TRBC Activity Name_Consumer Goods Conglomerates	0.100042
TRBC Industry Name_Consumer Goods Conglomerates	0.100042
TR.F.TotShHoldEq	0.097738
TR.F.ComEqTot	0.097498
TR.F.ComEqParentShHold	0.097194
TR.F.ShHoldEqCom	0.097194
TR.F.ShHoldEqParentShHoldTot	0.096900

Figure 13: GDAXI, feature correlation

On the left, GDAXI, the 30 most correlated features. The figure describes the 30 most correlated features within the GDAXI investment strategy. On the right: GDAXI, the 30 most correlated features with company-based investment Strategy. Spearman Rank correlation is used to class 30 most correlated features with company-based investment Strategy.

Feature 1	Feature 2	corr
TR.F.CashCashEquivTot	TR.F.CashCashEquiv	1.000000
TR.F.CashCashEquiv	TR.F.CashCashEquivTot	1.000000
TR.F.ComShrOutsIssue	TR.F.ComShrIssuedIssue	1.000000
TR.F.ShHoldEqCom	TR.F.ComEqParentShHold	0.999994
TR.F.ComEqTot	TR.F.ShHoldEqCom	0.999994
TR.F.ShHoldEqParentShHoldTot	TR.F.ComEqParentShHold	0.999993
TR.F.ComEqParentShHold	TR.F.ShHoldEqParentShHoldTot	0.999993
TR.F.ComShrIssuedTot	TR.F.ComShrOutsTot	0.999834
MABMM301CNM189S	MYAGM2CNM189N	0.999806
TR.F.TotCap	TR.F.TotBookCap	0.999261
TR.F.TotBookCap	TR.F.TotCap	0.999261
TR.F.TotShHoldEq	TR.F.ShHoldEqParentShHoldTot	0.998662
TR.F.ShHoldEqParentShHoldTot	TR.F.TotShHoldEq	0.998662
TR.F.ShHoldEqCom	TR.F.TotShHoldEq	0.998661
TR.F.TotShHoldEq	TR.F.ComEqParentShHold	0.998656
TR.F.ComEqTot	TR.F.TotShHoldEq	0.998656
TR.F.TotShHoldEq	TR.F.TangTotEq	0.998255
TR.F.TangTotEq	TR.F.TotShHoldEq	0.998255
TR.F.ShHoldEqCom	TR.F.TangBV	0.998102
TR.F.TangBV	TR.F.ComEqParentShHold	0.998094
TR.F.ComEqTot	TR.F.TangBV	0.998094
TR.F.TangBV	TR.F.ShHoldEqParentShHoldTot	0.998092
TR.F.ShHoldEqParentShHoldTot	TR.F.TangBV	0.998092
MYAGM2CNM189N	MANMM101CNM189S	0.997309
MANMM101CNM189S	MYAGM2CNM189N	0.997309
TR.F.TangBV	TR.F.TotShHoldEq	0.997198
Close	High	0.996680
TR.F.LTDebtExclCapLease	TR.F.DebtLTTot	0.996632
TR.F.DebtNonConvertLT	TR.F.DebtLTTot	0.996561
TR.F.DebtLTTot	TR.F.DebtNonConvertLT	0.996561

Feature	Correlation
Volume	0.173890
TRBC Industry Group Name_Oil & Gas	0.158370
TRBC Business Sector Name_Energy - Fossil Fuels	0.158370
TRBC Economic Sector Name_Energy	0.158370
CSCICP03CNM665S	0.111116
.HSI OPEN	0.102232
TRBC Industry Group Name_Textiles & Apparel	0.101448
TRBC Industry Name_Apparel & Accessories	0.101448
TRBC Business Sector Name_Cyclical Consumer Products	0.100601
.HSI LOW	0.099217
.HSI HIGH	0.098074
.HSI CLOSE	0.094177
TRBC Activity Name_Oil & Gas Exploration and Production (NEC)	0.092994
TRBC Industry Name_Oil & Gas Exploration and Production	0.092994
TRBC Activity Name_Integrated Oil & Gas	0.090557
TRBC Industry Name_Integrated Oil & Gas	0.090557
TRBC Activity Name_Integrated Telecommunications Services (NEC)	0.085021
TRBC Industry Name_Integrated Telecommunications Services	0.085021
TRBC Industry Name_Oil & Gas Refining and Marketing	0.082473
TRBC Activity Name_Oil & Gas Refining and Marketing (NEC)	0.082473
TRBC Activity Name_Knitwear	0.073456
TRBC Activity Name_Investment Holding Companies (NEC)	0.069302
TRBC Industry Group Name_Investment Holding Companies	0.069302
TRBC Industry Name_Investment Holding Companies	0.069302
TRBC Business Sector Name_Investment Holding Companies	0.069302
TRBC Economic Sector Name_Consumer Non-Cyclicals	0.065611
TRBC Activity Name_Apparel & Accessories (NEC)	0.060916
TRBC Activity Name_Electronic Components	0.057337
TRBC Activity Name_Sanitary Products	0.057264
TRBC Industry Group Name_Personal & Household Products & Services	0.057264

Figure 14: HSI, features correlation

On the left, HSI the 30 most correlated features. The figure describes the 30 most correlated features within the HSI portfolio. On the right: HSI, the 30 most correlated features with company-based investment Strategy. Spearman Rank correlation is used to class 30 most correlated features with company-based investment Strategy.

Feature 1	Feature 2	corr
TR.F.TotAssets	TR.F.TotLiabEq	0.99999576
TR.F.ShHoldEqParentShHoldTot	TR.F.ComEqParentShHold	0.999905629
TR.F.ComEqTot	TR.F.ShHoldEqParentShHoldTot	0.999905629
TR.F.ComEqParentShHold	TR.F.ShHoldEqParentShHoldTot	0.999905629
TR.F.NetBookCap	TR.F.NetOpAssets	0.999807109
TR.F.ShHoldEqCom	TR.F.ComEqTot	0.999804473
TR.F.ShHoldEqParentShHoldTot	TR.F.ShHoldEqCom	0.999710427
TR.F.DebtNonConvertLT	TR.F.LTDebtExclCapLease	0.99970142
TR.F.LTDebtExclCapLease	TR.F.DebtLTTot	0.999590114
TR.F.TotCap	TR.F.TotBookCap	0.999385899
TR.F.DebtNonConvertLT	TR.F.DebtLTTot	0.999314145
Total Capital	Total Book Capital	0.998905388
TR.F.DebtInclPrefEqMinIntrTot	TR.F.DebtInclFinOpLeaseLiabSuppl	0.998903405
TR.F.TotLiabEq	TR.F.TotLiab	0.997709949
TR.F.TotLiab	TR.F.TotAssets	0.997689251
TR.F.OthNonCurrAssetsTot	TR.F.OthNonCurrAssets	0.997429076
Open	High	0.997097184
.TOPX500 HIGH	.TOPX500 OPEN	0.995002348
.TOPX500 CLOSE	.TOPX500 LOW	0.993501178
TR.F.STDebtNotesPble	TR.F.STDebtCurrPortOfLTDebt	0.991688321
TR.F.TangTotEq	TR.F.TotShHoldEq	0.991290263
TR.F.ShHoldEqParentShHoldTot	TR.F.TotShHoldEq	0.990679348
TR.F.TotShHoldEq	TR.F.ComEqParentShHold	0.990596171
TR.F.ComEqTot	TR.F.TotShHoldEq	0.990596171
TR.F.ComEqParentShHold	TR.F.TotShHoldEq	0.990596171
.TOPX500 HIGH	.TOPX500 CLOSE	0.990356322
TR.F.TotShHoldEq	TR.F.ShHoldEqCom	0.990106843
TR.F.ShHoldEqCom	TR.F.TangBV	0.988577161
TR.F.TangBV	TR.F.ComEqTot	0.988370392
TR.F.TangBV	TR.F.ShHoldEqParentShHoldTot	0.988216365

Feature 1	Correlation
Return On Equity - Mean	0.2005272
Company Market Cap	0.0599547
TRBC Business Sector Name_Real Estate	0.058431
TRBC Industry Group Name_Real Estate Operations	0.058431
TRBC Economic Sector Name_Real Estate	0.058431
CSCICP03JPM665S	0.0534212
EXCHUS	0.0523247
TRBC Activity Name_Consumer Electronics Retailers	0.0521991
BPBLTD01JPQ637N	0.0518229
TRBC Industry Name_Real Estate Rental, Development & Operations	0.0513462
TRBC Activity Name_Real Estate Rental, Development & Operations (NEC)	0.0492919
TRBC Activity Name_Footwear Retailers	0.0485194
TRBC Industry Group Name_Healthcare Equipment & Supplies	0.0475762
BPBLTD01JPQ636S	0.0472388
Volume	0.0469731
Close	0.0465576
TRBC Business Sector Name_Retailers	0.0464317
TR.F.AccrualsST	0.0458963
TR.F.WkgCapNonCash	0.0457434
TRBC Industry Name_Medical Equipment, Supplies & Distribution	0.0454265
QJPN628BIS	0.0451726
TRBC Activity Name_Entertainment Production Equipment & Services	0.0449368
Low	0.0439801
High	0.0437184
TRBC Business Sector Name_Healthcare Services & Equipment	0.0434697
Open	0.0430384
TR.F.IncTaxPbleLTST	0.0425952
TRBC Activity Name_Home Furnishings Retailers (NEC)	0.0424311
TRBC Industry Name_Home Furnishings Retailers	0.0424311
TRBC Activity Name_Heating, Ventilation & Air Conditioning Systems	0.042421

Figure 15: TOPX500, features correlation

On the left, TOPX500, The 30 most correlated features. The figure describes the 30 most correlated features within the TOPX500 investment strategy. On the right: TOPX500, the 30 most correlated features with company-based investment Strategy. One use Spearman Rank correlation to class top 30 most correlated features with company-based investment Strategy.

3.1.7 PORTFOLIO CONSTRUCTION FRAMEWORK

This paper considers three investment strategies widely used in literature: long-only, short-only, and long-short strategy. One takes equally weighted portfolios, which are simple yet mark good performance compared to the benchmark portfolio. For the long-only portfolio strategy, the 25 best companies are considered for the GDAXI and the HSI portfolios and the 100 best companies for the SPX500 and TOPX500. For the short-only portfolio strategy, the 25 worst companies are considered for the GDAXI and the HSI portfolios and the 100 worst companies for the SPX500 and TOPX500. Finally, two portfolios Long-short are considered. One Long-Short portfolio takes the mean of the sum between the 25 best-ranked companies and top 25 worst-ranked companies for the GDAXI and the HSI portfolios for the top 100 best and top 100 worst companies for the long-short portfolio strategy. Another portfolio takes the mean of the difference between long and short portfolios. The second long-short portfolio is a portfolio type that buys winners and holds on to losers.

3.1.8 PROBLEM DEFINITION

The problem is defined as a regression exercise. A universe of variables $x_{t,i} = U^k$ for the stock i at the time t is taken, where i is the portfolio's constituent at the end of the month $t-1$, k is the total number of variables[18], and the input of the variable is defined as $v_{t,i} = \{x_{t-1,i}\, x_{t-3,i}, x_{t-6,i}, x_{t-9,i}, x_{t-12,i}\} \epsilon U^{670}$ using the past five data for the three months interval[19] for the k variables. The output variable is defined as the position one may take regarding his investments the next month: Buy (1), Sell (-1), or Hold (0). Figure 16 describes the relationship between the prediction and the input variables.

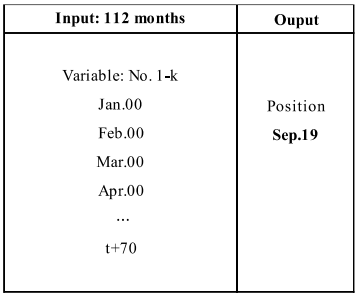


Figure 13: Prediction mapping

One considers the square root of the mean squared error ($RMSE$) as the loss function for the paper's goal. One defines the $RMSE$ as follow:

$$RMSE = \sqrt{MSE}\ where:$$

$$MSE_t = \frac{1}{n}\left\{\sum_{t=T-N-1}^{T} \sum_{i \epsilon P_{t-1}} \left(s_{t,i} - f(v_{t-1,i}; \theta_{t+1})\right)^2\right\},$$

where n is the number of companies in the portfolio P at the time $t-1$. θ_{t+1} is a parameter calculated by solving the MSE equation and taking a function type $f(.)$.

3.1.9 TRAINING AND PREDICTION

Before training the models, one needs to define the cross-validation data sets. Cross-validation is a procedure broadly applied in ML on time series analysis. Generally, setting cross-validation sets help the model adapt and correct itself. First, one defines the training set. Next, one divides the set into five folds, each equal to 112 dates. Next, one gradually takes each unique group as a hold, fits the model on the four-remaining group, and then tests the model using the fold one set as a hold. A train-test procedure is realized first. Then, the next fold follows. The process is repeated until the model is trained over the entire training set (figure 4).

[18] Details of the universe of variables per stock can be found following the link to the internet resources provided in appendix.

[19] As a reminder, when one set the CV folds, one train model on 112 months, one let out 2 months, and one test the model on the next 20 months (illustration of the process in figure 4) with a total length of the training set of 670 months.

We train the model by using 112 training steps. One calculates the prediction by adding the latest input into the model after the training has been realized. The cross-sectional predictive strategy (score) of stock i at the time t is calculated from the inputs at the time $t-1$ by substituting the input at the time $t-1$ into the function $f(.)$. The score equals than:

$$Score_{t+1,i} = f(v_{t,i}; \hat{\theta}_{t+1}),$$

where $\hat{\theta}_{t+1}$ is calculated from the MSE function with $N = 112$.

For example, to calculate the prediction score in September 2019 (t) from August 2019 ($t-1$), one must take the input value as follow: August 2019 ($t-1$), May 2019 ($t-3$), March ($t-5$), December 2018 ($t-8$), and September 2018 ($t-11$). Thus, the training period is extended over 19 years (figure 2). Figures 25 to 29 in the appendix Score distribution describe the score distribution among the portfolios.

We notice more defined distributions among the Linear regression model and CatBoost model. Typically, the distribution is left-skewed which suggests a high boundary. This is in line with the goal of this paper goal. One only predicts three types of positions an investor may take. The LSTM models show no precise distribution, and if there is a distribution, the score is centered on specific scores. This may be the cause of the poor results, or NN is more complicated to understand. The distribution of the Linear regression is unsteady in strategy prediction. This is caused by high multicollinearity in the data set. CatBoost has a steadier score distribution than both LSTM and Linear regression models in terms of score assessment. The regularization method favors tree ensemble prediction and disadvantages the LSTM modeling this in asset pricing study.

LSTM model score distribution suggests that the model doesn't run at all after a certain number of variables (differs per portfolio category, for example, for HSI: 100, and for GDAXI: 80). The same conclusions can be made by analyzing the portfolio performance assessment (figures 21-23).

3.1.10 PERFORMANCE EVALUATION

In evaluating the portfolio, this paper uses four widely used measures: average return per portfolio, average active return, information ratio, and Sharpe ratio.

First, let the average return of the long-only portfolio at the time t be R_t^L, the average return of the short-only portfolio at the time t be R_t^S, and the average return of the long-short portfolio at the time beingR_t^{LS}, $\bar{R}^L$ the average return per portfolio for the testing

period, $\bar{R}^S$ the average return per portfolio for the testing period, $\bar{R}^{LS}$ the average return per portfolio for the testing period, $|t|$ the length of the testing period, $|L_t|$ the length of the universe of stocks and $r_{i,t}$ the return of the portfolio constituent at the time t. The average return for the long portfolio is:

$$\bar{R}_t^L = \frac{1}{|L_t|}\sum_{i\,\epsilon L_t} r_{i,t},$$

$$\bar{R}^L = \frac{1}{|t|}\sum \bar{R}_t^L$$

Same reasoning for calculating the average of the other two portfolios. As already said, $|L_t| = 25$ for the Long-only, Short-only, and Long-Short portfolios for both GDAXI and HSI. Likewise, $|L_t| = 100$ for the Long-only, Short-only, and Long-Short portfolios for both SPX500 and TOPX500.

Second, the mean active return is computed. The active return is the excess return against the benchmark portfolio. Let $\overline{Alpha}$ be the mean active return, and R_t^I the benchmark portfolio return. The mean active return is:

$$Alfa_t^L = \bar{R}_t^L - \bar{R}_t^I$$

$$\overline{Alpha} = \frac{1}{|t|}Alfa_t^L$$

Same reasoning for the rest of the portfolios.

Likewise, one computes the annualized Sharpe ratio (SR). SR is a financial metric most used in finance to assess portfolio performance per unit of portfolio volatility. Let SR^L be the Sharpe ratio for the Long-only portfolio, and S_L the standard deviation of the long portfolio. The Sharpe ratio is:

$$SR^L = \sqrt[12]{\frac{\overline{Alpha}}{S_L}}$$

Same reasoning for the other portfolios.

On top of those three measures, it helps measure the portfolio return beyond the return of the benchmark portfolio compared to the return of its volatility. Next, the information ratio is computed. The information ratio, also known as appraisal rating, is a measure used to compare the portfolio return relative to the benchmark portfolio. Finally, the paper compares the IR to assess if a constructed portfolio can outperform the index. The IR and SR are similar, the only difference is that the SR uses the free risk rate as the benchmark,

such as the 10 Year Treasury bond yield, while the IR uses a riskier index as the benchmark, such as the S&P500. Let IR^L be the information ratio of the long portfolio. The formula is then:

$$IR^L = \sqrt[12]{\frac{\overline{\sum (R^L_{t,\iota} - R^I_{t,\iota})}}{\sigma_{R^L - R^I}}}$$

where $\overline{\sum (R^L_{t,\iota} - R^I_{t,\iota})}$ is the mean active return and $\sigma_{R^L - R^I}$ is the standard deviation of the difference between the portfolio and benchmark portfolio.

Next, one may note that the mean active return and the information ratio for the benchmark portfolio are equal to 0.

$$Alfa^I_t = \bar{R}^I_t - \bar{R}^I_t = 0 \rightarrow \overline{Alpha} = 0 \rightarrow \text{ and } SR^I = \sqrt[12]{\frac{\overline{Alpha}}{S_L}} = 0$$

Chapter 3. 2 PREDICTION MODELS

Two deep learning models and one linear Regression model have been employed to predict stock recommendations to fulfill the paper's goal. In addition, the scoring metrics are used to calculate the prediction accuracy for the recommendation at each point in time to construct portfolios. The models were run in Google Collaboratory using GPU[20] .

3.2.1 FEATURES IMPORTANCE

Before introducing and presenting the results in asset pricing modeling using ML, the paper may first investigate the importance of the features in ML predictions.

As one may see so far, deep learning algorithms are heavy, constructed on many functions, and cannot be easily interpreted. An asset manager or portfolio manager would probably be interested in knowing which variables contribute most to the prediction. This information can help than preselect the universe of stock in a much sustainable way. Few ML algorithms try to approximate the black-box model to understand how the black box functions (Molnar 2020).

Before passing on the portfolio overview, the paper first introduces SHAP value. SHAP values are an abbreviation of the Shapley Additive Explanations algorithm developed by Lundberg and Lee (2016). The goal of the local explanation method is to

[20] The CatBoost model is run on GPU; therefore, one use few predefined parameters.

interpret how a prediction was made. This is caused by computing the contribution of each feature to the forecast (Lundberg and Lee 2017).

To identify the feature importance, the SHAP values method was employed in the trained CatBoost model. First, one may observe that the company itself very much influences the portfolio strategy. For the SPX500 and TOPX500 portfolios, the company affects most of the prediction. For GDAXI, and HSI, one may observe more distributed weights of feature importance. Among the top 20, the most influential variables in the prediction-making process are, after the company variable, stock-related covariates, valuation metrics, and macro-variables.

The paper also summarizes the individual effect of the variable on the prediction. As shown in figures 35-41 in appendix Features importance, the company variables, for example, have a +13% positive effect on the investment position for the SPX500 portfolio, -147% negative effect on the investment strategy for the TOPX500, and -37% effect on the HSI portfolio. Similar analysis can be made for volume data, which negatively influences the GDAXI portfolio (-30%) and positively impacts the HSI portfolio investment strategy (+8%). The same reasoning may be made to compare the individual effect of the other variables.

In the case of small portfolios (less than 100 stock in the universe), valuation metrics seems to carry out the most predictive signals for the investment position within a portfolio. However, for a more significant portfolio (more than 100 stocks in the universe), on top of valuation metrics, macroeconomic variables seem to significantly influence the prediction at a portfolio level.

3.2.2 LINEAR REGRESSION

After the Linear regression models were introduced in part two, the paper focuses now on comparing the performance of the Linear Regression model in the portfolio's construction prediction.

Five types of tests per universe were run. The paper makes a comparative analysis between the four types of portfolios, first between portfolios considering the same numbers of variables, then compares the intra portfolio analysis with a different number of variables. Finally, the study assesses the portfolio performance in between regions. The general goal of the study is to see if adding more variables helps construct portfolios. Five random numbers n were taken $n =$: 15, 50, 80, 100, 150. Next, a Linear Regression

model training is run. As discussed before, Linear Regression models perform weakly in the training sample with very low R^2 (Stefan Nagel, Ross, and Duffe 2021). The results suggest that the model performs predictions that are very volatile (figure 17).

Before testing the model, the SHAP values method was employed to identify the feature importance. First, SHAP values are sorted in descending. Next, the variables matrix is reorganized into a smaller one taking n number of variables gradually used to realize five tests per index type. Third, the model proceeds to test, and a comparative analysis is made on the in-sample and out-of-sample R^2. On the one hand, one may notice a very negative out-of-sampleR^2 (figure 17 (a)). The tests show similar results as the literature study on Linear Regression in asset pricing regarding the R^2 (already discussed in part 2). On the other hand, one may observe that Linear Regression models perform extremely poorly at composing short portfolios.

Nb. Of Features	TOPX500		SPX500		HSI		GDAXI	
	In-sample	Out-of-sample	In-sample	Out-of-sample	In-sample	Out-of-sample	In-sample	Out-of-sample
15	3.10%	-2682.83%	2.93%	-31.00%	11.01%	-9.81E+25	7.97%	-41.99%
50	4.64%	-4331.37%	4.88%	-2.88E+22	14.08%	-944622.70%	14.06%	-280.95%
80	5.75%	-8955.79%	5.67%	-1.38E+20	14.35%	-7.55E+27	20.11%	-5.18E+25
100	6.22%	-6970.57%	6.41%	-2.38E+23	17.78%	-8514146.39%	23.32%	-56141.89%
150	7.50%	-11043.63%	8.32E-02	-15409.05%	24.08%	-1.72E+26	27.37%	-4.18E+25

Figure 17 (a): In-sample and out-of-sample R^2 using Linear Regression

The figure describes the in-sample and out-of-sample R^2. The Linear regression model is good at predicting in-sample strategies, but the R^2goes very fast in negative territories in out-of-sample.

In addition to very negative R^2, the linear regression model shows better RMSE results in in-sample predictions, while the spread between the residuals and the line of the best fit[21] is very high. The results suggest that the out-of-sample predictions are far away from the best fit line (for details, check figure 17 (b)).

Nb. Of Features	TOPX500		SPX500		HSI		GDAXI	
	In-sample	Out-of-san	In-sample	Out-of-sample	In-sample	Out-of-sample	In-sample	Out-of-sample
15.0000	0.0918	0.7449	0.0669	0.0875	0.2855	0.2048	0.2817	0.2689
50.0000	0.0904	1.0818	0.0655	0.8288	0.2757	4867.5572	0.2630	6.5941
80.0000	0.0893	22.3174	0.0650	4.88E+21	0.2697	23803.5956	0.2445	43.0129
100.0000	0.0889	38.7800	0.0644	1.49E+22	0.2638	600840.5698	0.2347	4415.4748
150.0000	0.0877	30.7512	0.0632	4.75E+22	0.2431	1.16E+25	0.2223	3.69E+26

Figure 17 (b): In-sample and out-of-sample $RMSE$ using Linear Regression model The figure describes the in-sample and out-of-sample $RMSE$. The linear regression model shows that, on top of a very negative R^2 for in-sample predictions, the volatility is very high out-of-sample. The results are under the theoretical facts discussed in part two.

[21] The best fit line is a graphical representation of a line that is considered to be the central trend of the data set.

Next, the paper compares the true positives (TP[22]) and the true negatives (TN[23]) of the prediction using Linear regression. Overall, for both investment strategies predictions, the out-of-sample TN and TP are lower for TOPX500, HSI, and GDAXI portfolios than the in-sample TN. This implies that the model makes more minor mistakes in out-of-sample predictions, but the ability to predict the right investment strategy decreases. One may notice that for the SPX500 portfolios, the out-of-sample TN is greater than the in-sample TN for both investment position predictions. Still, the model is performing better in terms of TP in out-of-sample compared to the other portfolios. Adding supplementary layers of variables generally helps achieve improved results for the four portfolios. Still, the model shows poor results in terms of short-only portfolios.

Linear Regression		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
Nb. Of Features	Actual	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
15	-1	4.14%	13.28%	0.92%	3.65%	1.05%	2.40%	4.81%	3.11%	4.64%	6.40%	0.92%	3.38%	2.86%	4.46%	0.00%	2.77%
	0	33.36%	57.56%	45.08%	63.38%	23.26%	45.17%	27.76%	37.25%	16.76%	28.18%	4.77%	10.46%	24.92%	31.19%	21.54%	31.54%
	1	62.49%	29.16%	54.00%	32.96%	75.69%	52.43%	67.44%	59.63%	78.60%	65.42%	94.31%	86.15%	72.22%	64.35%	78.46%	65.69%
50	-1	4.31%	12.83%	0.62%	3.46%	0.82%	2.48%	5.73%	2.58%	3.49%	7.64%	2.62%	2.77%	2.66%	4.68%	1.85%	2.46%
	0	31.12%	59.22%	45.19%	61.81%	23.45%	47.80%	32.87%	28.68%	16.52%	28.47%	12.00%	8.15%	23.55%	32.77%	31.54%	23.08%
	1	64.57%	27.94%	54.19%	34.73%	75.73%	49.72%	61.40%	68.74%	79.99%	63.89%	85.38%	89.08%	73.79%	62.55%	66.62%	74.46%
80	-1	4.02%	14.17%	1.00%	2.92%	0.92%	2.64%	2.81%	2.58%	3.65%	7.17%	2.92%	2.46%	1.63%	5.00%	1.08%	1.69%
	0	31.14%	58.56%	51.38%	61.65%	21.86%	48.57%	37.64%	24.88%	15.32%	29.72%	6.92%	10.62%	22.92%	32.90%	25.08%	30.31%
	1	64.84%	27.28%	47.62%	35.42%	77.23%	48.78%	59.55%	72.55%	81.03%	63.11%	90.15%	86.92%	75.45%	62.10%	73.85%	68.00%
100	-1	3.78%	13.97%	1.38%	3.23%	0.99%	2.69%	2.58%	6.04%	3.17%	7.69%	2.00%	0.62%	1.14%	5.05%	1.69%	1.08%
	0	30.64%	58.36%	50.46%	60.69%	21.09%	48.97%	27.99%	35.02%	15.37%	29.40%	6.31%	13.85%	22.73%	33.50%	30.92%	25.08%
	1	65.58%	27.67%	48.15%	36.08%	77.92%	48.35%	69.43%	58.94%	81.46%	62.92%	91.69%	85.54%	76.13%	61.45%	67.38%	73.85%
150	-1	3.42%	15.35%	2.08%	2.81%	0.90%	2.74%	3.84%	3.69%	2.62%	8.35%	2.62%	2.77%	1.31%	5.03%	2.62%	1.38%
	0	29.81%	56.81%	52.04%	51.00%	19.75%	52.33%	25.03%	33.49%	13.80%	31.11%	12.77%	7.69%	21.80%	33.70%	31.69%	30.31%
	1	66.77%	27.85%	45.88%	46.19%	79.35%	44.92%	71.13%	62.82%	83.58%	60.54%	84.62%	89.54%	76.89%	61.26%	65.69%	68.31%

True negative

True positive

Figure 18 (a): Actual versus Prediction

The figure describes the Actual versus predictive performance for in-sample and out-of-sample prediction. The Linear Regression model is horrible at constructing a short portfolio. On top of this, results show that the True-negative proportion is higher in the out-of-sample forecast. One may read the figure as follow: HSI Long-only in-sample portfolio constructed using a Linear regression model with 15 variables has 4.64% of the portfolio's components that are losing money (money losers). Therefore, the total number of money losers in the HSI portfolio equals $4.64\% * 24$ [24] ≈ 2 money loser stocks in the portfolio componence.

A CV was run to look for the optimal number of variables in the Linear Regression. First, five CVs were set, and a random step to 100 was chosen. Second, the hyperparameter range to tune the model was set equal to the range of numbers from 1 to 150. Third, a grid search was performed. RFE [25](Recursive Feature Elimination) method

[22] True positives (TP) is a measure that indicates the rate of correctly indicated predictions.

[23] False negative (FN) is a measure that indicates the correctly wrong predictions.

[24] The length of the portfolio which equals 100 stocks in the SPX500 and TOPX500 portfolios, and 25 stocks in the HSI and GDAXI portfolios.

[25] RFE is a variable selection technique that fit's a model and exclude gradually the variables till the wanted number of variables is reached.

was used to perform the grid search. As the number of features increases for all four portfolios, one may observe that the better the R^2 (figure 15b).

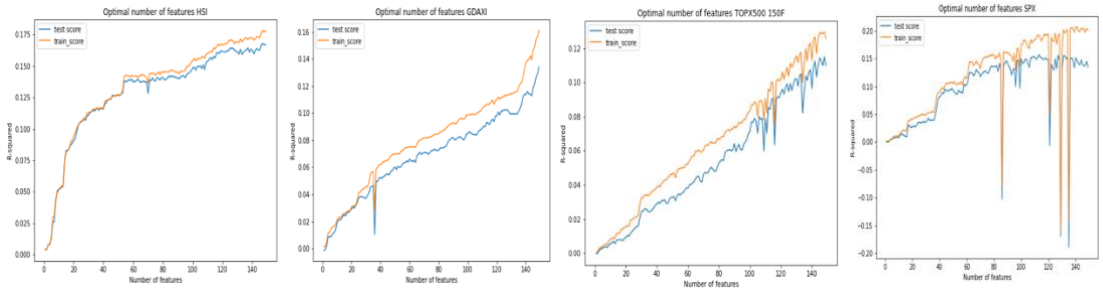


Figure 18 (b): Number optimal of features, Linear Regression model

The figure describes the number optimal of features per portfolio (from left to right: HSI, GDAXI, TOPX500, and SPX500) for the Linear Regression model. Results show that the R^2 improves as the number of variables increases to 150 for in-sample predictions. It is important to keep in mind that the SR and the R^2 are negatively correlated. This means that improved R^2 does not improve the SR (explication in part two of this paper).

3.2.3 CatBoost Model[26]

After the reader was introduced to general results on the Linear Regression model, the paper now describes the next model. CatBoost[27] is an open-source machine learning algorithm from Yandex that uses gradient boosting on decision trees. For the loss function, the paper considers the QueryRMSE[28] defined by Yandex as follow:

$$\sqrt{\frac{\sum_{Group \in Goups} \sum_{i \in Groups} w_i \left(tar_i - a_i - \frac{\sum_{j \in Groups} w_j \, (tar_j - a_j)}{\sum_{\in Groups} w_j} \right)}{\sum_{Group \in Goups} \sum_{i \in Groups} w_i}}$$

where tar is the target w are the weights, i are the companies, j are the features, a is the actual value. After running the model, the results are the following ones (figure 19q):

	SPX			TOPX500			HSI			GDAXI		
	Learn	Validation	Test	Learn	Validation	Test	Learn	Validation	Test	Learn	Validation	Test
RMSE	0.3722	0.3349	0.1664	0.4646	0.4070	0.2058	0.3139	0.2119	0.2119	0.3199	0.2950	0.2950

Figure 14: CatBoost, RMSE results

The figure describes the RMSE evaluation in-sample and out-of-sample for the four portfolios

According to the results, the validation RMSE shows improved results compared to the training set. This conclusion is in accordance with the theoretical results. Furthermore, one may observe that, in the case of the CatBoost model, the RMSE in out-of-sample

[26] Source: https://catboost.ai/en/docs/concepts/python-reference_catboost

[27] www.catboost.ai

[28] As given by Yandex

prediction show improvements compared to the in-sample and validation sample predictions.

This paper considers the number of trees equal to 1000 pairwise nodes (equals the number of iterations) to realize the purpose. The training stops when the overfitting bias recurs. The random seed used for training the model is set at 0, the learning rate used to reduce the gradient step is set at 0.03, the depth of one tree equals six nodes. At each iteration, the model randomizes the variables, and the number of total permutations equals 64. The model adds the parameter's value to a leaf denominator equal to 3 (l2_leaf_reg) at each step. The bootstrap method used for the regularization is type Bayesian. This means that the weight is readjusted at each step based on the parameter defined by the bootstrap procedure (Rubin 1981). The best model parameter is not set; therefore, the number of threes is not saved in the resulting model. Instead, the minimal number of trees that the models should have been set to one. The maximum number of features that can be combined is set at four combinations (more details of the universe of parameters can be found in figure 30 in appendix CatBoost model parameters). In addition, in Appendix CatBoost model graphical visualization of the model's predictive performance, figure 31, one may see the tree structure in investment strategy prediction

Parameters	SPX50	TOPX500	HSI	GDAXI
iterations (no. Of trees)	1000	1000	1000	1000
fold_permutation_block	64	64	64	64
l2_leaf_reg	3	3	3	3
random_strength	1	1	1	1
max_ctr_complexity	4	4	4	4
use_best_model	FALSE	FALSE	FALSE	FALSE
random_seed	0	0	0	0
depth	6	6	6	6
best_model_min_trees	1	1	1	1
loss_function	QueryRMSE	QueryRMSE	QueryRMSE	QueryRMSE
learning_rate	0.029999999	0.029999999	0.029999999	0.029999999
bootstrap_type	Bayesian	Bayesian	Bayesian	Bayesian
max_leaves	64	64	64	64
permutation_count	4	4	4	4

Figure 15: CatBoost Parameters

The figure describes the essential parameters for running the model on the data set.

CatBoost		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
	Actual	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
	-1	0.15%	28.46%	0.01615	8.85%	0.03%	6.45%	0.00615	8.15%	0.74%	9.06%	0.00%	5.23%	0.31%	5.10%	0.77%	2.31%
	0	5.31%	67.60%	22.08%	0.814615	1.23%	84.37%	10.65%	64.01%	7.74%	37.96%	1.85%	20.15%	16.02%	37.26%	16.15%	40.62%
	1	94.55%	3.94%	0.76307	9.69%	98.75%	9.18%	88.74%	27.84%	91.53%	52.99%	98.15%	74.62%	83.68%	57.64%	83.08%	57.08%

True negative
True positive

Figure 16: Actual versus Prediction

The figure describes the Actual versus predictive performance for in-sample and out-of-sample prediction. CatBoost model is better at constructing a short portfolio compared to the Linear Regression model. On top of this, results show that True-negative proportions are lower in the out-of-sample prediction long-only and more significant for out-of-sample short-only predictions. One may read the figure as follow: SPX500 Long-only in-sample portfolio constructed using a CatBoost model has 1.6% of the portfolio's components that are losing money (money losers). Therefore, the total number of money losers in the SPX500 portfolio equals $1.60\% * 100$ [29] $\approx$ 2 money loser stocks in the portfolio componence.

Next, the paper compares the TP and the TN of the prediction using the CatBoost model. Overall, for both investment strategies predictions, the out-of-sample TN and TP are stable for HSI and GDAXI portfolios. Given this, one may conclude that the model makes similar mistakes in out-of-sample predictions as for in-sample predictions. This shows that the CatBoost performs well on average for a small universe of stocks. However, one may notice that for the SPX500 and TOPX500 portfolios, the out-of-sample TN is higher than the in-sample TN for both investment position predictions. The model is also performing worse in terms of TP in out-of-sample compared to the other two portfolios. Overall, one may conclude that as the information increases and the universe of stocks increases, the model entails lower performance.

Furthermore, Figures 32 to 35, in appendix CatBoost model Loss Evaluation, show that keeping 150 of variables, the SPX500, HSI, and GDAXI portfolios performance may increase, one may conclude that removing variables from till a total of 150 variables are left seems to continue to diminish the total loss in the in-sample prediction. Contrary to TOPX500, keeping 150 variables out of the entire set of variables increases the in-sample loss. Compared to the Linear regression model, the CatBoost can capture the interactions between the variables and, using a features selection method, it can use the best predictive variable combination. Therefore, it is a reliable model to be used in asset pricing modeling.

[29] The length of the portfolio which equals 100 stocks in the SPX500 and TOPX500 portfolios, and 25 stocks in the HSI and GDAXI portfolios.

3.2.4 LONG SHORT-TERM MEMORY MODEL

As for the Linear Regression, this paper aims to use random numbers of variables n and see how well the model performs when one adds supplementary layers of information. This paper uses the LSMT model implemented by an open-source machine learning library called PyTorch[30].

The paper experimented using a recurrent neural network architecture called Long Short-Term Memory (LSTM). It is a well-suited model for the classification and prediction of time series data. In contrast to feedforward neural networks, it takes into account the feedback connection(Gers, Schmidhuber, and Cummins 2000; Letham et al. 2015). A typical LSTM architecture comprises a cell, an input layer, an output layer, and a hidden layer (Hochreiter and Schmidhuber 1997). One sets the starting learning rate at 0.1, a traditionally standard default value, and uses the Adam optimizer to update the attributes of the neural network (NN), such as weights and learning rate over the learning time.

The Adam (Kingma and Ba 2015) (adaptive moment estimation) optimization algorithm continues the stochastic gradient descendent algorithm. In addition, it is designed to update the weights of the NN accordingly to the training data.

The input dimension equals the number of features. The hidden dimension equals 15, 50, 80, 100, and 150 for each simulation. The batch size equals the number of constituents at the time t, a total number of recurrent layers equal two, and the output equals one predicted value. The number of recurrent layers stacks the LSTMs on top of each other. The first LSTM layer will provide a range for a sequence of output. The second LSTM layer takes the output of the first layer and predicts the final result[31]. The model is trained over 100 epochs.

According to the results presented in Figure 20c, one may confirm that LSMT shows more significant results in out-of-sample prediction than CatBoost and Linear Regression model for all portfolios. The TN of the LSTM prediction does not exceed the TN in the in-sample forecast for long investment strategies. In addition, the model shows fewer TP in out-of-sample prediction than in the in-sample. According to figures 20a and 19b, the model better predicts long investments in out-of-sample. However, similar to the other

[30] For more detailed Informations about the function and implementaion of an LSTM network check out: https://cnvrg.io/pytorch-lstm/.

[31] Source: www.pytorch.org

two models, LSTM is bad at constructing a short-only portfolio. The TP for short investment positions are considerably greater than for the long investment position. In addition, it seems that adding variables into the model can improve many of the long-only portfolio predictions. Still, it adds biases that, in a way, negatively affect the short-only portfolio.

LSTM		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
Nb. Of Features	Actual	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
15	-1	12.60%	2.43%	9.49%	1.28%	2.06%	4.97%	0.74%	6.42%	2.77%	9.60%	0.00%	7.95%	3.19%	5.06%	0.92%	2.77%
	0	55.31%	33.56%	67.44%	59.49%	33.90%	37.57%	27.16%	32.35%	13.50%	37.09%	3.85%	19.74%	27.51%	28.02%	30.00%	29.08%
	1	32.09%	64.01%	23.08%	39.23%	64.04%	57.46%	68.64%	57.78%	83.73%	53.31%	96.15%	72.31%	69.29%	66.92%	69.08%	68.15%
50	-1	2.94%	6.24%	1.03%	1.28%	2.54%	0.56%	0.51%	0.51%	5.82%	3.53%	2.05%	1.79%	2.51%	5.40%	0.62%	2.15%
	0	35.28%	35.93%	55.13%	36.67%	40.03%	33.70%	22.31%	34.87%	18.14%	20.85%	6.67%	10.51%	26.05%	30.25%	20.31%	34.00%
	1	61.78%	57.82%	43.85%	62.05%	57.43%	65.73%	77.18%	64.62%	76.05%	75.62%	91.28%	87.69%	71.44%	64.35%	79.08%	63.85%
80	-1	6.02%	7.97%	3.08%	5.64%	2.54%	0.56%	3.46%	0.99%	4.44%	6.41%	2.05%	1.28%	3.33%	4.58%	2.31%	2.77%
	0	43.64%	40.73%	50.00%	42.31%	40.03%	33.70%	15.80%	22.96%	21.30%	17.43%	10.77%	8.21%	25.65%	30.51%	27.54%	26.77%
	1	50.34%	51.30%	46.92%	52.05%	57.43%	65.73%	77.28%	72.59%	74.27%	76.16%	87.18%	90.51%	71.02%	64.92%	70.15%	70.46%
100	-1	6.10%	7.85%	3.33%	5.38%	1.13%	1.95%	0.00%	3.95%	5.88%	3.98%	0.00%	1.54%	3.33%	4.58%	0.00%	2.82%
	0	43.42%	40.82%	47.95%	44.36%	42.74%	26.33%	27.65%	12.59%	23.95%	15.37%	4.36%	10.00%	25.65%	30.51%	26.15%	27.69%
	1	50.48%	51.33%	48.72%	50.26%	56.13%	71.72%	68.89%	80.00%	70.17%	80.65%	95.64%	88.46%	71.02%	64.92%	73.85%	69.49%
150	-1	6.05%	7.51%	3.59%	5.13%	1.33%	1.89%	0.25%	2.96%	5.82%	4.01%	0.00%	1.54%	3.33%	4.58%	0.00%	2.82%
	0	43.50%	39.80%	50.26%	38.97%	40.06%	38.36%	22.47%	29.63%	24.44%	15.25%	4.36%	10.00%	25.65%	30.51%	26.15%	27.69%
	1	50.45%	52.68%	46.15%	55.90%	58.62%	59.75%	73.83%	63.95%	69.75%	80.73%	95.64%	88.46%	71.02%	64.92%	73.85%	69.49%

True negative
True positive

Figure 20 (a) **Actual versus Prediction**

The figure describes the Actual versus predictive performance for in-sample and out-of-sample prediction. The LSTM model is terrible at constructing a short portfolio. On top of this, results show that the True-negative proportion is higher in the out-of-sample forecast. One may read the figure as follow: TOPX500 Long-only in-sample portfolio constructed using an LSTM model with 15 variables has 12.60% of the portfolio's components that are losing money (money losers). Therefore, the total number of money losers in the TOPX500 portfolio equals $12.60\% * 100$ [32] ≈ 13 money loser stocks in the portfolio componence.

Next, the paper investigates the $RMSE$. One may see looking at figure 20b that the model is less volatile than the Linear regression model's output but higher volatility than the CatBoost model's RMSE.

Nb. Of Features	TOPX500			SPX500			HSI			GDAXI		
	In-sample	Validation	Out-of-sample	In-sample	Validation	Out-of-sample	In-sample	Validation	Out-of-sample	In-sample	Validation	Out-of-sample
15	0.6157	0.5712	0.5418	0.5271	0.5234	0.5519	0.5554	0.5053	0.4944	0.5655	0.4887	0.4507
50	0.6172	0.5724	0.5445	0.5551	0.5460	0.5765	0.5562	0.5055	0.4945	0.5724	0.4670	0.4196
80	0.6198	0.5764	0.5444	0.5670	0.5736	0.5942	0.6620	0.5974	0.5818	0.5785	0.4632	0.4128
100	0.6193	0.5713	0.5475	0.5983	0.6077	0.6256	0.5590	0.5204	0.5130	0.6202	0.6127	0.5948
150	0.6157	0.5713	0.5416	0.6814	0.6946	0.7076	0.7371	0.7336	0.7351	0.7130	0.7438	0.7366

Figure 20 (b): LSTM, RMSE results

The figure describes the RMSE evaluation in-sample and out-of-sample for the four portfolios considering a different number of variables.

In addition, according to the in-sample predictions, loss over the training data set (figures 41-42 in appendix Long-Short Term Memory Model Loss Evolution) shows that an LSTM model's accuracy, regardless of the number of variables considered to interact in between, increases as the number of epochs increases.

[32] The length of the portfolio which equals 100 stocks in the SPX500 and TOPX500 portfolios, and 25 stocks in the HSI and GDAXI portfolios.

Chapter 3. 3 EMPIRICAL STUDY

As already discussed at the start of this paper, individual asset pricing is not the goal of the investors. Diversification lowers the risk, and therefore, investors typically aim to predict asset returns in a portfolio framework.

Once the paper has introduced the effect of individual variables on the investment strategy at a portfolio level, together with the employed models, one may proceed to the portfolio framework and performance assessment.

To remind, the out-of-sample test evaluates the models' performance on behalf of a sample outside of the training set. The out-of-sample data set starts from September 2019 and continues till October 2021. The paper compares the SR, IR, mean returns, and mean active returns of the 44 portfolios against the benchmark portfolio (Francisco Barillas And Shanken 2018).

3.3.1 RESULTS OF LONG - ONLY PORTFOLIO

Overall, all three models, Linear Regression, CatBoost, and LSTM, successfully construct Long-only portfolios. According to the results described in figures 23 to 25, LSTM is a great model to use a portfolio out of a small universe similar to the GDAXI index (40 stocks). The model not only helps achieve a more $\times 2$ SR of the benchmark portfolio (1.96 over 0.85), but it also outperforms the benchmark portfolio $+79\%$ with only 50 variables. Great results are achieved as well using a Linear regression model. The long-only portfolio constructed, using Linear regression with 100 variables, reaches more $\times 2$ SR of the benchmark portfolio (2.07 over 0.85) and overperforms the benchmark portfolio by $+81\%$. The results are justified by very negative R^2 and high portfolio volatility. Using the CatBoost model, the constructed portfolio achieves a bit less than $\times 2$ SR of the benchmark portfolio (1.47 over 0.85) and overperforms by 54%.

Similar to the HSI index, the Linear regression with 150 variables model seems to lead in terms of performance with 1.60 SR (over -0.25 for the benchmark portfolio) for a bigger universe of stock. Moreover, the models help construct a portfolio that overperforms the benchmark by $more\ than\ 262\%$. Next, using LSTM with only 100 variables, the long-only portfolio for a +50 stocks universe, the constructed long-only portfolio overperforms the benchmark portfolio by $+218\%$ and achieves a 1.15 SR. Finally, the CatBoost model performs less in such a type of portfolio with a 0.98 SR but still overperforms the benchmark portfolio by $+170\%$.

With a bigger universe of stocks (more than 500) similar to the TOPX500 index, the Linear Regression model results with 15 variables and CatBoost model show similar results with 1.71 SR and 1.68. However, the portfolio underperforms the benchmark by 58% and 66%, respectively. This suggests that the TOPX500 is more volatile than the constructed long-only portfolio. Therefore it has, on average, better returns (check figures 43-46 in the appendix Cumulative Returns Long-only Portfolios for comparing cumulative returns). On the other hand, LSTM with 150 variables shows outstanding SR of the long-only portfolio (1.36), but still underperforms the benchmark by 73%.

SPX500 long-only type of portfolio shows that it is harder to overperform a well-established benchmark as SPX500. Best SR is achieved with the CatBoost model (1.68), followed by the LSTM with 100 variables (1.63) and the Linear regression model with 150 variables (1.10). The constructed long-only portfolio type SPX500 underperforms the index by 72%, 76%, and 170%.

Taken these results, one may ask why the first two portfolios can overperform the market and the last two no. Results may change if the portfolios are proportional-sized. In addition, economic events in the last two years regarding the evolution of the pandemic may considerably influence the return expectations.

3.3.2 Results of Short–Only Portfolio

Compared to the long-only portfolio, the short-only portfolio is quite alarming. Overall, the short-only portfolios underperform the index, and the SR is lower, entering in negative territory (figures 23-25).

For small portfolios with a small universe of stocks (<40), the best SR is achieved with a Linear regression model with 100 variables (-1.26), followed by CatBoost with -1.33 SR and LSTM with -1.36 SR. For small portfolios with a universe of stock (>50), the best SR is achieved with a Linear Regression model with 15 variables (0.70). Thus, the Short-only portfolio not only has a positive SR but also overperforms the index by 40%. Again, the results are justified by the very high volatility of the portfolios constructed with the linear regression models. Next, the portfolio achieves a -0.03 SR using LSTM with 50 variables, followed by an SR equal to -0.94 SR.

For more significant portfolios with a greater universe of stocks (>500), the best SR is achieved with the Linear Regression model with 15 variables for TOPX500 (-0.40) and with the LSTM model with 50 variables for SPX500 (-0.63). Next, a -0.46 SR is achieved

with CatBoost for TOPX500, followed by LSTM with 50 variables (-0.70). Finally, a second-class prediction model for SPX500 is the Linear Regression model with an SR equal to -1.02, followed by CatBoost (-1.14 SR).

In general, there is a tendency for the linear regression model to be preferred for short-only portfolios in terms of portfolio performance, but the last choice in terms of volatility. Typically, a low number of variables are favored (15-50). Graphical representations of long-only portfolio cumulative returns compared against the index are explained in the appendix Cumulative Returns Short-only Portfolios for comparing cumulative returns, figures 47-50.

3.3.3 Results of Long-Short Portfolio

The paper conducts a comparative analysis of two types of long-short portfolios. On the one hand, the paper considers the long-short portfolio 1 the sum of the long-only and the short-only portfolios. On the other hand, the long-short portfolio 2 considers the difference between the long-only and short-only portfolios.

According to the results (figures 23-25), one may conclude that the long-short portfolio 1 can achieve the best results with a linear regression model with 15 variables for the TOPX500 and HSI500 portfolios where the SR of the portfolio equals 1.38 and 3.24, respectively. Not only does the SR of the HSI long-short portfolio 1 shows great SR, but also the portfolio overperforms the market by $+124\%$. The second-best performance can be attended using the LSTM model with 50 variables for the long-short portfolio 1 for SPX500 with an SR 1.07. Overall, the LSTM model is preferred in terms of model accuracy and portfolio volatility.

One may see that long-short portfolio two can achieve best-in-class results with LSTM with 150 variables for all constructed portfolios. One reaches an SR of 1.98, 1.21, 1.34, and 1.47 for GDAXI, HSI, TOPX500, and SPX500. Furthermore, one may see that, for small universe stocks, the model generates a portfolio that outperforms the index by $+62\%$ and $+235\%$, respectively. For a greater universe of stocks, the model can create a portfolio with similar performance as the index (TOPX500) or underperforms the index by 103% (SPX500).

For a small universe of stocks, results show that the LSMT with a number of variables between 80 and 150 can be used to construct a long-short portfolio that outperforms the index and yet have an incredible outperformance per unit of portfolio volatility. On the

other hand, for a universe of stock more significant than 500, it seems that the more variables one adds, the better there the performance of the portfolio is both on a portfolio performance against the index and portfolio performance per unit of portfolio volatility. In addition, the $RMSE$ (figure 20b) shows that the model can, on average, predict the investment position except for SPX500. Therefore, it may appear necessary to add more variables into the model or change the size of the SPX500 portfolio to attend better performance results.

CatBoost model, can in general, be used to construct a long-short portfolio. The SR of the generated portfolio is close to one or greater than one. The newly created portfolios can overperform the benchmark if the benchmark is considerably small. As the index increases, one may choose proportional sizes for the new portfolios to obtain better results. Also, the $RMSE$ table (figure 19a) shows that the model can be reliable because it accurately predicts the investment strategy.

Linear regression models show promising results in terms of performance out of sample. As one adds more variables, the performance measures improve. However, given the very negative out-of-sample R^2, linear regression models are very volatile. Graphical representation of long-short portfolio cumulative returns compared against the index are explained in appendix Cumulative returns, Long-short portfolios, figures 51-54.

Nb. Of Features		TOPX500								SPX500							
		Linear Regression				LSTM				Linear Regression				LSTM			
		Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio
15	**Long**	0.01	-0.01	-0.58	1.71	0.00	-0.02	-1.31	0.28	0.01	-0.01	-1.49	1.08	0.02	-0.01	-0.98	1.44
	Short	0.00	-0.02	-0.82	-0.40	-0.01	-0.02	-1.04	-1.07	-0.02	-0.04	-1.61	-1.66	-0.02	-0.04	-1.47	-1.45
	Long/Short Portfolio 1	0.00	-0.01	-0.74	1.38	0.00	-0.02	-1.14	-1.29	0.00	-0.02	-1.71	-1.29	0.00	-0.02	-1.48	-0.01
	Long/Short Portfolio 2	0.01	-0.01	-0.94	1.02	0.01	-0.01	-1.08	0.70	0.01	-0.01	-1.30	1.41	0.02	-0.01	-1.07	1.39
	Index	0.01	0.00	0.00	0.99	-0.01	0.00	0.00	0.99	0.01	0.00	0.00	1.55	-0.02	0.00	0.00	1.55
50	**Long**	0.01	-0.01	-0.74	1.40	0.01	-0.01	-0.93	0.99	0.02	-0.01	-0.97	1.74	0.02	-0.01	-1.28	1.40
	Short	0.00	-0.02	-0.89	-0.62	-0.01	-0.02	-0.93	-0.70	-0.01	-0.03	-1.32	-1.02	-0.01	-0.03	-1.14	-0.63
	Long/Short Portfolio 1	0.00	-0.02	-0.85	0.70	0.00	-0.02	-0.94	0.61	0.00	-0.02	-1.29	0.69	0.00	-0.02	-1.23	1.96
	Long/Short Portfolio 2	0.01	-0.01	-0.94	1.00	0.01	-0.01	-1.00	0.87	0.01	-0.01	-1.60	1.35	0.01	-0.01	-2.01	0.98
	Index	0.01	0.00	0.00	0.99	0.00	0.00	0.00	0.99	0.02	0.00	0.00	1.55	-0.01	0.00	0.00	1.55
80	**Long**	0.01	-0.01	-0.97	0.92	0.01	-0.01	-0.87	1.10	0.02	-0.01	-0.73	0.95	0.02	-0.01	-1.04	1.47
	Short	-0.01	-0.02	-0.98	-0.90	-0.01	-0.03	-1.07	-1.22	-0.02	-0.04	-1.70	-1.92	-0.02	-0.04	-1.55	-1.50
	Long/Short Portfolio 1	0.00	-0.02	-0.98	-0.19	0.00	-0.02	-1.01	-0.98	0.00	-0.02	-1.88	-0.21	0.00	-0.02	-1.56	-0.81
	Long/Short Portfolio 2	0.01	-0.01	-0.97	0.94	0.01	-0.01	-0.84	1.17	0.02	0.00	-1.06	1.34	0.02	-0.01	-1.03	1.47
	Index	0.01	0.00	0.00	0.99	-0.01	0.00	0.00	0.99	0.02	0.00	0.00	1.55	-0.02	0.00	0.00	1.55
100	**Long**	0.01	-0.01	-0.87	1.12	0.01	-0.01	-0.76	1.33	0.01	-0.01	-1.81	1.06	0.02	0.00	-0.76	1.63
	Short	-0.01	-0.03	-0.98	-0.89	-0.01	-0.03	-1.09	-1.33	-0.02	-0.04	-1.61	-1.65	-0.02	-0.04	-1.46	-1.42
	Long/Short Portfolio 1	0.00	-0.02	-0.95	-0.14	0.00	-0.02	-0.99	-0.11	0.00	-0.02	-1.79	-0.55	0.00	-0.02	-1.43	0.69
	Long/Short Portfolio 2	0.01	-0.01	-0.92	1.02	0.01	-0.01	-0.75	1.33	0.01	-0.01	-1.59	1.34	0.02	-0.01	-0.93	1.45
	Index	0.01	0.00	0.00	0.99	-0.01	0.00	0.00	0.99	0.01	0.00	0.00	1.55	-0.01	0.00	0.00	1.55
150	**Long**	0.01	-0.01	-0.79	1.30	0.01	-0.01	-0.73	1.36	0.01	-0.01	-1.70	1.10	0.02	-0.01	-1.04	1.47
	Short	-0.01	-0.03	-0.99	-0.93	-0.01	-0.03	-1.09	-1.31	-0.02	-0.04	-1.62	-1.65	-0.02	-0.04	-1.55	-1.50
	Long/Short Portfolio 1	0.00	-0.02	-0.93	0.01	0.00	-0.02	-0.98	0.70	-0.01	-0.03	-1.71	-1.94	0.00	-0.02	-1.56	-0.81
	Long/Short Portfolio 2	0.01	-0.01	-0.86	1.12	0.01	-0.01	-0.75	1.34	0.02	0.00	-1.07	1.45	0.02	-0.01	-1.03	1.47
	Index	0.01	0.00	0.00	0.99	-0.01	0.00	0.00	0.99	0.01	0.00	0.00	1.55	-0.02	0.00	0.00	1.55

Figure 17:Performance comparison (1)

The figure describes the performance of TOPX500 and SPX500 Portfolio for portfolio constructed with Linear regression model and LSTM model with 15, 50, 80, 100, and 150 variables.

Nb. Of Features		HSI								GDAXI							
		Linear Regression				LSTM				Linear Regression				LSTM			
		Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio
15	Long	0.02	0.03	2.97	2.14	0.00	0.00	0.26	-0.17	0.02	0.00	0.62	1.69	0.02	0.01	0.73	1.75
	Short	0.01	0.01	0.41	0.70	-0.02	-0.01	-0.57	-1.92	-0.02	-0.03	-1.11	-1.42	-0.01	-0.03	-1.12	-1.37
	Long/Short Portfolio 1	0.01	0.02	1.24	3.24	-0.01	-0.01	-0.36	-2.37	0.00	-0.01	-0.79	0.39	0.00	-0.01	-0.71	1.11
	Long/Short Portfolio 2	0.01	0.01	1.61	1.08	0.01	0.01	1.52	0.93	0.02	0.00	0.49	1.58	0.02	0.00	0.50	1.61
	Index	0.02	0.00	0.00	-0.25	-0.02	0.00	0.00	-0.25	0.02	0.00	0.00	0.85	-0.02	0.00	0.00	0.85
50	Long	0.01	0.01	1.14	0.68	0.02	0.02	2.56	1.83	0.02	0.01	0.77	1.87	0.02	0.01	0.79	1.97
	Short	-0.01	-0.01	-0.31	-1.29	0.00	0.00	0.15	-0.03	-0.02	-0.03	-1.12	-1.50	-0.02	-0.03	-1.26	-1.80
	Long/Short Portfolio 1	0.00	0.00	0.08	-1.07	0.01	0.01	0.86	2.41	0.00	-0.01	-0.73	1.07	0.00	-0.01	-0.78	0.40
	Long/Short Portfolio 2	0.01	0.01	1.47	1.03	0.01	0.01	1.68	1.10	0.02	0.00	0.60	1.69	0.02	0.01	0.72	1.89
	Index	0.01	0.00	0.00	-0.25	0.00	0.00	0.00	-0.25	0.02	0.00	0.00	0.85	-0.02	0.00	0.00	0.85
80	Long	0.01	0.02	1.82	1.29	0.00	0.00	0.43	-0.05	0.01	0.00	0.18	1.31	0.02	0.01	0.61	1.96
	Short	-0.01	0.00	-0.09	-0.76	-0.02	-0.01	-0.65	-2.39	-0.02	-0.03	-1.27	-1.80	-0.02	-0.03	-1.33	-1.98
	Long/Short Portfolio 1	0.00	0.01	0.47	1.17	-0.01	-0.01	-0.39	-3.07	0.00	-0.02	-1.02	-1.03	0.00	-0.01	-0.86	-0.24
	Long/Short Portfolio 2	0.01	0.01	1.65	1.11	0.01	0.01	1.81	1.12	0.02	0.00	0.57	1.59	0.02	0.01	0.62	1.98
	Index	0.01	0.00	0.00	-0.25	-0.02	0.00	0.00	-0.25	0.01	0.00	0.00	0.85	-0.02	0.00	0.00	0.85
100	Long	0.00	0.00	0.47	0.01	0.01	0.02	2.18	1.15	0.02	0.01	0.81	2.07	0.02	0.01	0.61	1.96
	Short	-0.01	-0.01	-0.36	-1.24	-0.01	-0.01	-0.41	-1.28	-0.02	-0.03	-1.05	-1.26	-0.02	-0.03	-1.33	-1.98
	Long/Short Portfolio 1	-0.01	0.00	-0.15	-1.42	0.00	0.00	0.20	-1.13	0.00	-0.01	-0.65	0.71	0.00	-0.01	-0.86	-0.24
	Long/Short Portfolio 2	0.01	0.01	1.47	0.73	0.01	0.02	2.35	1.21	0.02	0.00	0.74	1.64	0.02	0.01	0.62	1.98
	Index	0.00	0.00	0.00	-0.25	-0.01	0.00	0.00	-0.25	0.02	0.00	0.00	0.85	-0.02	0.00	0.00	0.85
150	Long	0.01	0.02	2.62	1.60	0.01	0.02	2.18	1.15	0.02	0.00	0.40	1.59	0.02	0.01	0.61	1.96
	Short	0.00	0.00	0.19	0.07	-0.01	-0.01	-0.41	-1.28	-0.02	-0.03	-1.12	-1.46	-0.02	-0.03	-1.33	-1.98
	Long/Short Portfolio 1	0.01	0.01	0.83	2.28	0.00	0.00	0.20	-1.13	0.00	-0.01	-0.82	-0.24	0.00	-0.01	-0.86	-0.24
	Long/Short Portfolio 2	0.01	0.01	1.52	0.91	0.01	0.02	2.35	1.21	0.02	0.00	0.55	1.54	0.02	0.01	0.62	1.98
	Index	0.01	0.00	0.00	-0.25	-0.01	0.00	0.00	-0.25	0.02	0.00	0.00	0.85	-0.02	0.00	0.00	0.85

Figure 18: Performance comparison (2)

The figure describes the performance of HSI and GDAXI Portfolio for portfolio constructed with Linear regression model and LSTM model with 15, 50, 80, 100, and 150 variables.

Portfolio	Position	Mean Return	Mean Active Return	Information Ratio	Sharpe Ratio
GDAXI	Long	0.02	0.00	0.54	1.47
	Short	-0.02	-0.03	-1.08	-1.33
	Long/Short Portfolio 1	0.00	-0.01	-0.81	0.36
	Long/Short Portfolio 2	0.02	0.00	0.41	1.42
	Index	0.02	0.00	0.00	0.85
HSI	Long	0.01	0.01	1.70	0.98
	Short	-0.01	0.00	-0.14	-0.94
	Long/Short Portfolio 1	0.00	0.01	0.38	0.48
	Long/Short Portfolio 2	0.01	0.01	1.54	1.03
	Index	0.01	0.00	0.00	-0.25
SPX500	Long	0.02	0.00	-0.72	1.68
	Short	-0.01	-0.03	-1.40	-1.15
	Long/Short Portfolio 1	0.00	-0.02	-1.34	1.39
	Long/Short Portfolio 2	0.02	-0.01	-1.42	1.45
	Index	0.02	0.00	0.00	1.55
TOPX500	Long	0.01	-0.01	-0.66	1.68
	Short	0.00	-0.02	-0.83	-0.46
	Long/Short Portfolio 1	0.00	-0.01	-0.78	1.09
	Long/Short Portfolio 2	0.01	-0.01	-0.95	0.95
	Index	0.01	0.00	0.00	0.99

Figure 19: Performance comparison (3)

The figure describes the performance of the four portfolios for portfolios constructed with the CatBoost model.

Chapter 3. 4 CONCLUSION REMARKS

This paper aimed to identify effective ML models in asset pricing. Three ML techniques were implemented to predict one-month-ahead investment position in the European, American, Japanese, and Chinese stock markets. Based on a quantitative analysis of portfolio performance compared to region-specific benchmark portfolios, the superiority of deep learning techniques over the linear regression model can be confirmed on a predictive accuracy level and portfolio performance. Tree ensembles prove to be outperformed by NN models in a region-specific universe of stocks. The study shows that restricting the number of variables can limit the portfolio performance in the absence of diversifiable information. The linear regression model can, in a way, compete with

deep learning methods only if we preselect the most predictive variables given a tree ensemble model. Still, the predictive accuracy is very low.

However, outstanding results in terms of portfolio performance can be achieved using the long-short strategy predicted by an LSTM model with two hidden layers with more than 80 variables. The tests show that efficient scoring can enhance the portfolio performance with a higher SR than the index SR. In addition, the paper concludes that a portfolio constructed with NN can outperform an index with up to 50 constituents. Therefore, a long-strategy strategy is preferred to a long-only strategy. Also, the paper concludes that deep learning models are terrible at constructing short portfolios. Finally, the paper shows the potential of explainable algorithms as SHAP values in deep learning portfolio modeling. The analysis concludes that interpretable methods can offer valuable insights regarding the behavior of the deep learning models.

Further research is needed to determine if NN can construct outperforming portfolios given a benchmark portfolio like S&P500. Other studies are required to identify the relationship between market performance and the size of the portfolio. In addition, it would be interesting to determine if there might be a better portfolio construction framework rather than the expected stock return and if a strategy type sells winners and holds on losers can be a reliable strategy.

REFERENCES

Amihud, Yakov, and Ruslan Goyenko. 2012. "Mutual Fund's R^2 as Predictor of Performance." *Review of Financial Studies* 26.

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. "Learning Long-Term Dependencies with Gradient Descent Is Difficult." *IEEE Transactions on Neural Networks* 5(2): 157–66.

Bensic, Mirta, Natasa Sarlija, and Marijana Zekic-Susac. 2005. "Modelling Small-Business Credit Scoring by Using Logistic Regression, Neural Networks and Decision Trees." *Intelligent Systems in Accounting, Finance and Management* 13(3): 133–50.

Bielby, J., M. Cardillo, N. Cooper, and A. Purvis. 2010. "Modelling Extinction Risk in Multispecies Data Sets: Phylogenetically Independent Contrasts versus Decision Trees." *Biodiversity and Conservation* 19(1): 113–27.

Bryzgalova, Svetlana, Markus Pelger, and Jason Zhu. 2019. "Forest Through the Trees: Building Cross-Sections of Stock Returns." *SSRN Electronic Journal*.

Chen, Fei Long, and Feng Chia Li. 2010. "Combination of Feature Selection Approaches with SVM in Credit Scoring." *Expert Systems with Applications* 37(7): 4902–9. http://dx.doi.org/10.1016/j.eswa.2009.12.025.

Chen, James Ming. 2021. "The Capital Asset Pricing Model." *SSRN Electronic Journal*: 915–33.

Chen, Luyang, Markus Pelger, and Jason Zhu. 2019. "Deep Learning in Asset Pricing." *SSRN Electronic Journal*.

DeMiguel, Victor, Alberto Martin-Utrera, and Francisco J. Nogales. 2013. "Size Matters: Optimal Calibration of Shrinkage Estimators for Portfolio Selection." *Journal of Banking and Finance* 37(8): 3018–34.

Fama, Eugene F., and Kenneth R. French. 2015a. "A Five-Factor Asset Pricing Model." *Journal of Financial Economics* 116(1): 1–22. http://dx.doi.org/10.1016/j.jfineco.2014.10.010.

———. 2015b. "A Five-Factor Asset Pricing Model." *Journal of Financial Economics* 116(4): 1–22.

Feldman, Todd, and Gabriele Lepori. 2016. "Asset Price Formation and Behavioral Biases." *Review of Behavioral Finance* 8(2): 137–55.

FRANCISCO BARILLAS, and AY SHANKEN. 2018. "Comparing Asset Pricing Models." *THE JOURNAL OF FINANCE* LXXIII(2): 715–54.

Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. 2000. "Learning to Forget: Continual Prediction with LSTM." *Neural Computation* 12(10): 2451–71.

Graves, Alex et al. 2009. "A Novel Connectionist System for Unconstrained Handwriting Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5): 855–68.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2020. "Empirical Asset Pricing via Machine Learning." *Review of Financial Studies* 33(5): 2223–73.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. Springer *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9(8): 1735–80.

Hodrick, Robert J. 1981. "International Asset Pricing with Time-Varying Risk Premia." *Journal of International Economics* 11(4): 573–87. https://www.sciencedirect.com/science/article/pii/0022199681900350.

Hou, Kewei, Chen Xue, and Lu Zhang. 2015. "Digesting Anomalies: An Investment Approach." *Review of Financial Studies* 28(3): 650–705.

Jack L. Treynor. 1961. "MARKET VALUE, TIME, AND RISK." *Kaos GL Dergisi* 2(75): 147–73.

Kingma, Diederik P., and Jimmy Lei Ba. 2015. "Adam: A Method for Stochastic Optimization." *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*: 1–15.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh. 2020. "Shrinking the Cross-Section." *Journal*

of Financial Economics 135(2): 271–92. https://doi.org/10.1016/j.jfineco.2019.06.008.
Kuhn, Max, and Kjell Johnson. 2013. Applied Predictive Modeling. *Applied Predictive Modeling*.
Kur Hornik, Maxwell Stincombe, Halber White. 1989. "Multilayer Feedforward Networks Are Universal Approximators." *Practical Neural Network Recipies in C++* 2: 359–66.
Letham, Benjamin, Cynthia Rudin, Tyler H. McCormick, and David Madigan. 2015. "Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model." *Annals of Applied Statistics* 9(3): 1350–71.
Lewellen, Jonathan. 2013. 2 "The Cross Section of Expected Returns."
Lucas, Robert E, and Thomas J Sargent, eds. 1981. *Rational Expectations and Econometric Practice*. NED-New. University of Minnesota Press. http://www.jstor.org/stable/10.5749/j.ctttssh5.
Lundberg, Scott M., and Su In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems* 2017-Decem(Section 2): 4766–75.
McCloskey. 1994. 61 Economica *If You're So Smart: The Narrative of Economic Expertise.*
Molnar, C. 2020. *Interpretable Machine Learning*.
Monti, Ricardo Pio, Christoforos Anagnostopoulos, and G Montana. 2018. "Adaptive Regularization for Lasso Models in the Context of Nonstationary Data Streams." *Stat. Anal. Data Min.* 11: 237–47.
Moritz, Benjamin, and Tom Zimmermann. 2016. "Tree-Based Conditional Portfolio Sorts: The Relation between Past and Future Stock Returns." *SSRN Electronic Journal*.
Muth, John F. 1961. "Rational Expectations and the Theory of Price Movements." *Econometrica* 29(3): 315.
Rubin, Donald B. 1981. "The Bayesian Bootstrap." *The Annals of StatisticsAnnals of Statistics* 9(1): 130–34.
Sharpe, William F. 1994. "The Sharpe Ratio." *The Journal of Portfolio Management* 21: 49–58.
Stefan Nagel, Stephen A. Ross, and William F.Sharpe Darrell Duffe. 2021. *MLinAssetPricing.Pdf*. ed. Princeton University Press.
Stone, M. 1977. "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion." *Journal of the Royal Statistical Society. Series B (Methodological)* 39(1): 44–47.
Ta, Van Dai, Chuan Ming Liu, and Direselign Addis Tadesse. 2020. "Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading." *Applied Sciences (Switzerland)* 10(2): 437–57.
Wilson, D. Randall, and Tony R. Martinez. 1997. "Improved Heterogeneous Distance Functions." *Journal of Artificial Intelligence Research* 6(June 2000): 1–34.
Wolpert, David H., and Wiliam G. Macready. 1997. "No Free Lunch Theorems." *Natural Computing Series* 1(1): 287–322.

APPENDIX[33]

DATA TRANSFORMATION. STRATEGY

As described in section 3.1.2, the strategy vector counts different strategies. Before being able to work with predictions of investment positions, one must first apply a numerical transformation. The changes are the following ones:

GDAXI	{'Hold':0, 'Buy':1, 'Neutral':0, 'Accumulate':0, 'Outperform':1, 'Reduce':-1,'Overweight':-1, 'Sell':-1, 'Market Perfermor':0, 'Underperform':-1,'Strong Buy':1, 'Add':1, '2/Outperform':1, '4/Sell':-1}
HSI	{'Buy':1, 'Hold':0, 'Sell':-1, 'Accumulate':0, 'Outperform':1, 'Underperform':-1,'Market Perform':0, 'Neutral':0, 'Reduce':-1, 'Subscribe':1,'In-Line/Neutral':0, 'Underperform/Underwt':-1, 'Underperform/Mktwt':-1,'Overwt/Cautious':-1, 'Fully Valued':0, 'Outperform/Mktwt':1,'Peer Perform/Mktwt':0, 'In-Line':0, 'Trading Sell':-1,'1-Overwght/Positive':-1, 'Overweight':-1, 'Buy (1)':1, 'Overwt/In-Line':-1,'Overwt/Attractive':0, 'Underweight':1, 'Add':1, 'Equalwt/Attractive':1,'Equalwt/In-Line':0, 'Marketperform':0}
SPX500	,'Mkt Performance':0,'Marketperform':0,'Market Average':0, 'Accumulate':1, 'Market Performer':0,'Strong Buy/Buy':1, 'Marketperform':0, 'Market Perform':0,'Market Average':0, 'Perform In Line':0, 'Mkt Performer':0,'Long Term Buy':1, 'Sell':-1, 'Lt Buy/Accum':1, 'Maintain Position':0,'Market Outperform':1, 'Attractive':1, 'Market Performance':0, 'Add':1,'Underperform':-1, 'Underwt/In-Line':1, 'Sector Perform':0, 'Reduce/Sell':-1,'1-Overwght/Positive':-1, 'Recommended List':1, 'Outperform/Mktwt':1,'Sector Perf/Overwt':-1, '3-Underwght/Positive':1, 'Outperform/Overwt':-1,'Outperf/Mktwt':0, 'Sector Perf/Underwt':1, 'Outperf/Overwt':-1,'Underperform/Overwt':0, 'Neutral/Mktwt':0, '2-Equalwght/Positive':0,'Overweight':-1, 'Peer Perform/Mktwt':0, '3 Underperform':0, 'In-Line':0,'Mp/Avoid':-1, '2 In-Line':0, '2-Equalwght/Neutral':0, 'Underperf/Mktwt':0,'Outperf/Underwt':0, 'Avoid/Mp':-1, 'Neutral/Overwt':-1,'Peer Perform/Underwt':0, 'Buy/Mp':1, '3-Underwght/Neutral':1,'Underperform/Mktwt':0, 'Underperf/Overwt':-1, 'Outperform/Underwt':1,'Underweight':1, 'Underwt/Cautious':-1, 'Peer Perform/Overwt':-1,'Neutral/Underwt':0, 'Sector Perf/Mktwt':0, 'Buy-On-Weakness':1,'In-Line/Neutral':0, '2-Equalwght/Negative':0, 'Grad. Accumulate':1,'Equalweight':0, 'Underperf/Neutral':0, 'Outperf/Neutral':-1,'Neutral Weight':0, 'Buy (1)':1, 'Equalwt/In-Line':0,'1-Overwght/Neutral':-1, 'Overwt/In-Line':-1, 'Equal-Weight':0, 'Average':0,'Neutral/Neutral':0, 'Perform':0, '3-Underwght/Negative':1,'1-Overwght/Negative':-1, 'Neutral/Hold':0, 'Equalweight/Neutral':0,'Fairly Valued':-1, 'Equal Weight':0, 'Equalweight/Positive':0,'Undervalued':1, '3-Stars (Hold)':0, 'Fair Value':-1, 'Over Weight':-1,'Sell (3)':-1, 'Sector Weight':0, 'Gradually Accumulate':1,'Equalwt/Attractive':1, 'Long':1, 'Neutral (2)':0, 'In Line':0,'Perform In Line':0,'-1':-1,'Add':1,'Reduce/Sell':-1,'Recommended List':1,'Sector Perf/Overwt':-1,'3-Underwght/Positive':0,'Sector Perf/Underwt':0,'3 Underperform':1,'Mp/Avoid':-1,'2 In-Line':0,'Underperf/Mktwt':1,'Outperf/Underwt':-1,'Avoid/Mp':-1,'Peer Perform/Underwt':0,'Buy/Mp':1,'3-Underwght/Neutral':0,'Underperf/Overwt':-1,'Outperform/Underwt':-1,'Neutral/Underwt':0,'Buy-On-Weakness':1,'In-Line/Neutral':0,'Grad. Accumulate':1,'Outperf/Neutral':-1,'Average':0,'Perform':0}
TOPX500	{'Neutral':0, 'Sell':-1, 'Underperform':-1, 'Buy':1, 'Outperform':1, 'Hold':0,'Add':1, 'Accumulate':0, 'Strong Buy':1, 'Attractive':1, 'Reduce':-1,'Market Perform':0, '1':0, 'Buy On Weakness':1, 'Strong Sell':-1, 2:0, '3':1,'2':0, 'Over 5-15%':-1, '+-5%':0, 'Over 15%':-1, 'Market Performer':0, 'Underperf/Mktwt':-1, 'Neutral/Mktwt':0, 'Performance':0, 3:1,'Under 5-15%':1, '5':0, 'Equalwt/In-Line':0, 'In-Line':0, 'Outperf/Mktwt':0, '1-Overwght/Positive':-1, 'Underweight':1, '4':0, 'Outperf/Overwt':1,'Underperf/Underwt':-1, 'Outperf/Attractive':1, 'Buy (1)':1, 'Neutral/Overwt':0, 'A+ Outperform':1, 'Overweight':-1, 'In-Line/Attractive':1, 'Hold (2)':0, 'A Outperform':1, 'Underperf/Overwt':-1, '3-Underwght/Positive':1, 'Neutral/Underwt':0,'Outperf/Cautious':0, '3-Underwght/Neutral':0, 'B Out/Underperform':-1,'2-Equalwght/Neutral':0, '1-Overwght/Neutral':-1, 'Outperf/Neutral':0, 'B+':1, 4:0, 'A':1, 'Outperf/Underwt':1, 'Underperf/Attractive':0, 1:1,'2-Equalwght/Positive':0, 'B':1, 'In-Line/Neutral':0, 'Buy/Neutral':1,'Equalwt/Attractive':1, 'Outperform 5-15%':1, 'Buy/Attractive':1, 'Overwt/Attractive':0, 'Sell/Neutral':-1, 'Neutral/Neutral':0,'2-Equalwght/Negative':-1, '3-Underwght/Negative':1, '1-Overwght/Negative':-1, 'B-':-1, 'Sell (3)':-1, 'Moderately Strong':0,'Equalweight/Neutral':0, 'Outperform By >15%':1, 'Marketperform':0,'Outperform +15%':1, 'Neutral Plus':0, 'Neutral/Hold':0,'Overweight/Positive':-1, 'Overwt/In-Line':-1, 'Positive':0,'Overweight/Neutral':0, 'Neutral/Attractive':0, 'Neutral (2)':0,'Strong':1, 'Equalweight':0, 'Underwt/Attractive':1, 'Underwt/In-Line':0, 'Mixed':0, 'Strong Outperform':0}

[33] You can find the data set, python notebooks and result files following the link: https://drive.google.com/drive/folders/1GDsd6MkpzIc4gSosvJHzsUxRsFKmU1VT?usp=sharing

MACRO VARIABLES[34]

Figure 27 presents the macro variables used as covariates in forecasting individual stock investment strategies with ML.

Country	Firm Characteristic	Type	Data Source	Frequency	Description
US	CPALTT01USQ657N	Macro variables	FRED	Quarterly	Consumer Price Index
	GDPC1	Macro variables	FRED	Quarterly	Real Gross Domestic Product
	IPB50001SQ	Macro variables	FRED	Quarterly	Industrial Production
	EXUSEU	Macro variables	FRED	Monthly	U.S. Dollars to Euro Spot Exchange Rate
	EXCHUS	Macro variables	FRED	Monthly	Chinese Yuan Renminbi to U.S. Dollar Spot Exchange Rate
	EXJPUS	Macro variables	FRED	Monthly	Japanese Yen to U.S. Dollar Spot Exchange Rate
	UNRATE	Macro variables	FRED	Monthly	Unemployment Rate
	PAYEMS	Macro variables	FRED	Monthly	All Employees, Total Nonfarm
	AAA10YM	Macro variables	FRED	Monthly	Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity
	AAA	Macro variables	FRED	Monthly	Moody's Seasoned Aaa Corporate Bond Yield
	BAA	Macro variables	FRED	Monthly	Moody's Seasoned Baa Corporate Bond Yield
	HQMCB10YRP	Macro variables	FRED	Monthly	10-Year High Quality Market (HQM) Corporate Bond Par Yield
	IRLTLT01USM156N	Macro variables	FRED	Monthly	Long-Term Government Bond Yields: 10-year: Main (Including Benchmark) for the United States
	HQMCB30YRP	Macro variables	FRED	Monthly	30-Year High Quality Market (HQM) Corporate Bond Par Yield
	BOPGSTB	Macro variables	FRED	Monthly	Trade Balance: Goods and Services, Balance of Payments Basis
	NA000352Q	Macro variables	FRED	Quarterly	Exports of Goods and Services
	POILBREUSDM	Macro variables	FRED	Monthly	Global price of Brent Crude
	PPIACO	Macro variables	FRED	Monthly	Producer Price Index by Commodity: All Commodities
	UMCSENT	Macro variables	FRED	Monthly	University of Michigan: Consumer Sentiment
	IEABC	Macro variables	FRED	Quarterly	Balance on current account
	USSTHPI	Macro variables	FRED	Quarterly	All-Transactions House Price Index for the United States
	TRESEGUSM052N	Macro variables	FRED	Monthly	Total Reserves excluding Gold for United States
	FEDFUNDS	Macro variables	FRED	Monthly	Federal Funds Effective Rate
	M1SL	Macro variables	FRED	Monthly	M1
	M2SL	Macro variables	FRED	Monthly	M2
	MABMM301USM189S	Macro variables	FRED	Monthly	M3
	GPDI	Macro variables	FRED	Quarterly	Gross Private Domestic Investment
China	CHNCPIALLQINMEI	Macro variables	FRED	Quarterly	Consumer Price Index: All Items for China
	RGDPNACNA666NRUG	Macro variables	FRED	Annual	Real GDP at Constant National Prices for China
	PRINTO01CNQ663N	Macro variables	FRED	Quarterly	Total Industry Production Excluding Construction for China
	EXUSEU	Macro variables	FRED	Monthly	U.S. Dollars to Euro Spot Exchange Rate
	EXCHUS	Macro variables	FRED	Monthly	Chinese Yuan Renminbi to U.S. Dollar Spot Exchange Rate
	EXJPUS	Macro variables	FRED	Monthly	Japanese Yen to U.S. Dollar Spot Exchange Rate
	LMUNRRTTCNQ156S	Macro variables	FRED	Quarterly	Registered Unemployment Rate for China
	AAA10YM	Macro variables	FRED	Monthly	Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity
	BPBLTD01CNQ637N	Macro variables	FRED	Quarterly	Current Account Balance: Total Trade of Goods for China (DISCONTINUED)
	CHNXTEXVA01NCMLM	Macro variables	FRED	Monthly	International Trade: Exports: Value (goods): Total for China
	POILBREUSDM	Macro variables	FRED	Monthly	Global price of Brent Crude
	PPIACO	Macro variables	FRED	Monthly	Producer Price Index by Commodity: All Commodities
	CSCICP03CNM665S	Macro variables	FRED	Monthly	Consumer Opinion Surveys: Confidence Indicators: Composite Indicators: OECD Indicator for China
	BPBLTT01CNQ637N	Macro variables	FRED	Quarterly	Total Current Account Balance for China (DISCONTINUED)
	QCNR628BIS	Macro variables	FRED	Quarterly	Real Residential Property Prices for China
	TRESEGCNM052N	Macro variables	FRED	Monthly	Total Reserves excluding Gold for China
	RBCNBIS	Macro variables	FRED	Monthly	Real Broad Effective Exchange Rate for China
	MANMM101CNM189S	Macro variables	FRED	Monthly	M1 for China
	MYAGM2CNM189N	Macro variables	FRED	Monthly	M2 for China
	MABMM301CNM189S	Macro variables	FRED	Monthly	M3 for China
	PLOINVCNA624NUPN	Macro variables	FRED	Annual	Price Level of Investment for China

Country	Firm Characteristic	Type	Data Source	Frequency	Description
EU28	CPALTT01GPQ	Macro variables	FRED	Quarterly	Consumer Price Index: All items: Total: Total for the European Union
	CLVMEURSCAB1GQEA19	Macro variables	FRED	Quarterly	Real Gross Domestic Product (Euro/ECU series) for Euro area (19 countries)
	PRINTO01EZQ661S	Macro variables	FRED	Quarterly	Total Industry Production Excluding Construction for the Euro Area
	EXUSEU	Macro variables	FRED	Monthly	U.S. Dollars to Euro Spot Exchange Rate
	EXJPUS	Macro variables	FRED	Monthly	Japanese Yen to U.S. Dollar Spot Exchange Rate
	LRHUTTTTEUM156S	Macro variables	FRED	Monthly	Harmonized Unemployment Rate: Total: All Persons for the European Union
	AAA10YM	Macro variables	FRED	Monthly	Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity
	IRLTLT01EZM156N	Macro variables	FRED	Monthly	Long-Term Government Bond Yields: 10-year: Main (Including Benchmark) for the Euro Area
	BPBLTD01EZQ636N	Macro variables	FRED	Quarterly	Current Account Balance: Total Trade of Goods for the Euro Area (DISCONTINUED)
	EA19XTEXVA01CXMLM	Macro variables	FRED	Monthly	International Trade: Exports: Value (goods): Total for the Euro Area
	POILBREUSDM	Macro variables	FRED	Monthly	Global price of Brent Crude
	PPIACO	Macro variables	FRED	Monthly	Producer Price Index by Commodity: All Commodities
	CSCICP03EZM665S	Macro variables	FRED	Monthly	Consumer Opinion Surveys: Confidence Indicators: Composite Indicators: OECD Indicator for the Euro Area
	BPBLTT01EZQ636S	Macro variables	FRED	Quarterly	Total Current Account Balance for the Euro Area
	QXMN628BIS	Macro variables	FRED	Quarterly	Residential Property Prices for Euro area
	TRESEGEZM052N	Macro variables	FRED	Monthly	Total Reserves excluding Gold for Euro Area
	RBXMBIS	Macro variables	FRED	Monthly	Real Broad Effective Exchange Rate for Euro Area
	MYAGM1EZM159N	Macro variables	FRED	Monthly	M1 for EU
	MYAGM2EZM196N	Macro variables	FRED	Monthly	M2 for EU
	MABMM301EZM189S	Macro variables	FRED	Monthly	M3 froe EU
	PRMNVG01EZQ661S	Macro variables	FRED	Quarterly	Total Production of Investment Goods for Manufacturing for the Euro Area
	BPFAOI03EZQ636N	Macro variables	FRED	Annual	Total Production of Investment Goods for Manufacturing for the Euro Area
Japan	JPNCPIALLQINMEI	Macro variables	FRED	Quarterly	Consumer Price Index of All Items in Japan
	JPNRGDPEXP	Macro variables	FRED	Quarterly	Real Gross Domestic Product for Japan
	PRINTO01JPQ661N	Macro variables	FRED	Quarterly	Total Industry Production Excluding Construction for Japan
	EXUSEU	Macro variables	FRED	Monthly	U.S. Dollars to Euro Spot Exchange Rate
	EXCHUS	Macro variables	FRED	Monthly	Chinese Yuan Renminbi to U.S. Dollar Spot Exchange Rate
	EXJPUS	Macro variables	FRED	Monthly	Japanese Yen to U.S. Dollar Spot Exchange Rate
	LRHUTTTTJPM156S	Macro variables	FRED	Monthly	Harmonized Unemployment Rate: Total: All Persons for Japan
	AAA10YM	Macro variables	FRED	Monthly	Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity
	IRLTLT01JPM156N	Macro variables	FRED	Monthly	Long-Term Government Bond Yields: 10-year: Main (Including Benchmark) for Japan
	BPBLTD01JPQ637N	Macro variables	FRED	Quarterly	Current Account Balance: Total Trade of Goods for Japan (DISCONTINUED)
	JPNXTEXVA01CXMLM	Macro variables	FRED	Monthly	International Trade: Exports: Value (goods): Total for Japan
	POILBREUSDM	Macro variables	FRED	Monthly	Global price of Brent Crude
	PPIACO	Macro variables	FRED	Monthly	Producer Price Index by Commodity: All Commodities
	CSCICP03JPM665S	Macro variables	FRED	Monthly	Consumer Opinion Surveys: Confidence Indicators: Composite Indicators: OECD Indicator for Japan
	BPBLTD01JPQ636S	Macro variables	FRED	Monthly	Total Current Account Balance for the United States (DISCONTINUED)
	QJPN628BIS	Macro variables	FRED	Monthly	Residential Property Prices for Japan
	TRESEGJPM052N	Macro variables	FRED	Monthly	Total Reserves excluding Gold for Japan
	RBJPBIS	Macro variables	FRED	Monthly	Real Broad Effective Exchange Rate for Japan
	MYAGM1JPM189S	Macro variables	FRED	Monthly	M1 for Japan
	MYAGM2JPM189S	Macro variables	FRED	Monthly	M2 for Japan
	MABMM301JPM189S	Macro variables	FRED	Monthly	M3 for Japan
	JPNRGDPPNI	Macro variables	FRED	Quarterly	Real Private Nonresidential Investment for Japan

Figure 20: Macro variables, Data Source: FRED

[34] Exhaustive list of all the variables can be found in the shared drive folder.

SCORE DISTRIBUTION

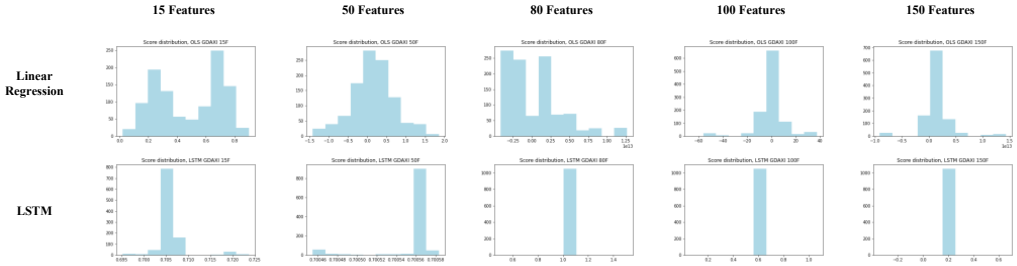


Figure 21: Score distribution, GDAXI portfolios

This figure describes how the Linear Regression, and LSTM ranks the position of the 40 stocks within GDAXI portfolios.

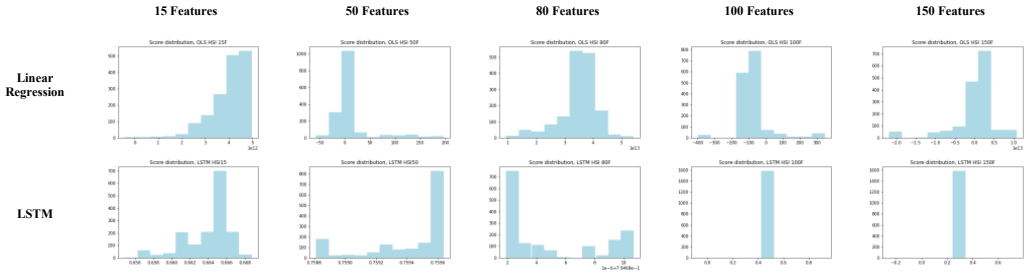


Figure 22: Score distribution, HSI portfolios

This figure describes how the Linear Regression, and LSTM ranks the position of the 58 stocks within HSI portfolios.

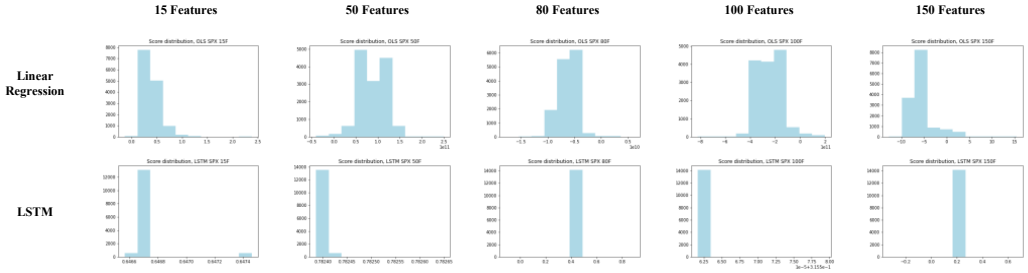


Figure 23: Score distribution, SPX500 portfolios

This figure describes how the Linear Regression, and LSTM ranks the position of the 58 stocks within SPX500 portfolios.

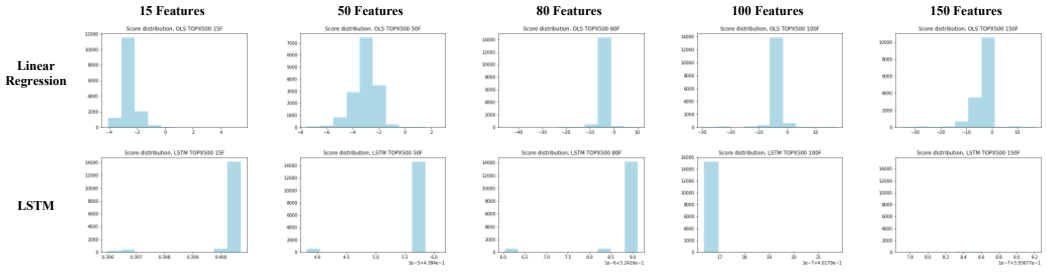


Figure 24: Score distribution, TOPX500 portfolios

This figure describes how the Linear Regression, and LSTM ranks the position of the 58 stocks within SPX500 portfolios.

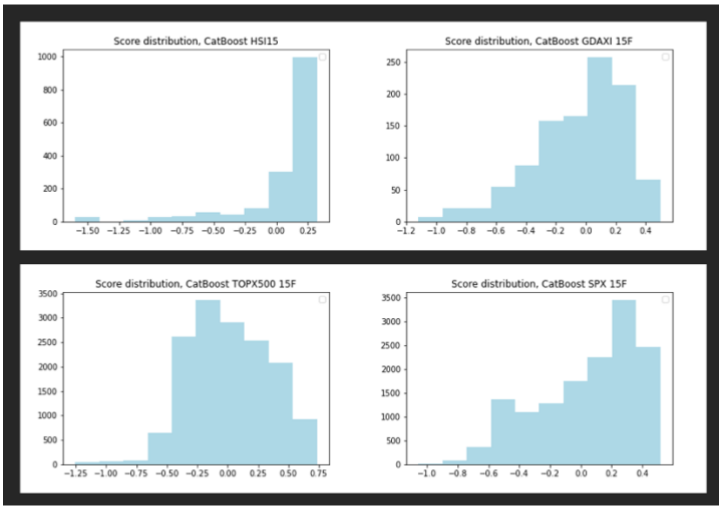


Figure 25: Score Distribution, CatBoost predictions

CATBOOST MODEL PARAMETERS

Parameters	SPX50	TOPX500	HSI	GDAXI
nan_mode	Min	Min	Min	Min
gpu_ram_part	0.95	0.95	0.95	0.95
eval_metric	QueryRMSE	QueryRMSE	QueryRMSE	QueryRMSE
combinations_ctr	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=Median:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=Median:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=Median:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=Median:Prior=0/1']
iterations	1000	1000	1000	1000
fold_permutation_block	64	64	64	64
leaf_estimation_method	Newton	Newton	Newton	Newton
observations_to_bootstrap	TestOnly	TestOnly	TestOnly	TestOnly
counter_calc_method	SkipTest	SkipTest	SkipTest	SkipTest
grow_policy	SymmetricTree	SymmetricTree	SymmetricTree	SymmetricTree
boosting_type	Plain	Plain	Ordered	Ordered
ctr_history_unit	Group	Group	Group	Group
feature_border_type	GreedyLogSum	GreedyLogSum	GreedyLogSum	GreedyLogSum
bayesian_matrix_reg	0.100000001	0.100000001	0.100000001	0.100000001
one_hot_max_size	10	10	10	10
devices	-1	-1	-1	-1
pinned_memory_bytes	104857600	104857600	104857600	104857600
force_unit_auto_pair_weights	FALSE	FALSE	FALSE	FALSE
l2_leaf_reg	3	3	3	3
random_strength	1	1	1	1
rsm	1	1	1	1
boost_from_average	FALSE	FALSE	FALSE	FALSE
gpu_cat_features_storage	GpuRam	GpuRam	GpuRam	GpuRam
fold_size_loss_normalization	FALSE	FALSE	FALSE	FALSE
max_ctr_complexity	4	4	4	4
model_size_reg	0.5	0.5	0.5	0.5
simple_ctr	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=MinEntropy:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=MinEntropy:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=MinEntropy:Prior=0/1']	['Borders:CtrBorderCount=15:CtrBorderType=Uniform:TargetBorderCount=1:TargetBorderType=MinEntropy:Prior=0/1:Prior=0.5/1:Prior=1/1', 'FeatureFreq:CtrBorderCount=15:CtrBorderType=MinEntropy:Prior=0/1']
pool_metainfo_options	{'tags': {}}	{'tags': {}}	{'tags': {}}	{'tags': {}}
use_best_model	FALSE	FALSE	FALSE	FALSE
meta_l2_frequency	0	0	0	0
random_seed	0	0	0	0
depth	6	6	6	6
ctr_target_border_count	1	1	1	1
has_time	FALSE	FALSE	FALSE	FALSE
border_count	128	128	128	128
min_fold_size	100	100	100	100
data_partition	FeatureParallel	FeatureParallel	FeatureParallel	FeatureParallel
bagging_temperature	1	1	1	1
classes_count	0	0	0	0
auto_class_weights	None	None	None	None
leaf_estimation_backtracking	AnyImprovement	AnyImprovement	AnyImprovement	AnyImprovement
best_model_min_trees	1	1	1	1
min_data_in_leaf	1	1	1	1
add_ridge_penalty_to_loss_function	FALSE	FALSE	FALSE	FALSE
loss_function	QueryRMSE	QueryRMSE	QueryRMSE	QueryRMSE
learning_rate	0.029999999	0.029999999	0.029999999	0.029999999
meta_l2_exponent	1	1	1	1
score_function	Cosine	Cosine	Cosine	Cosine
task_type	GPU	GPU	GPU	GPU
leaf_estimation_iterations	1	1	1	1
bootstrap_type	Bayesian	Bayesian	Bayesian	Bayesian
max_leaves	64	64	64	64
permutation_count	4	4	4	4

Figure 26: CatBoost Parameters

CATBOOST MODEL GRAPHIC VISUALIZATION OF THE MODEL'S PREDICTION

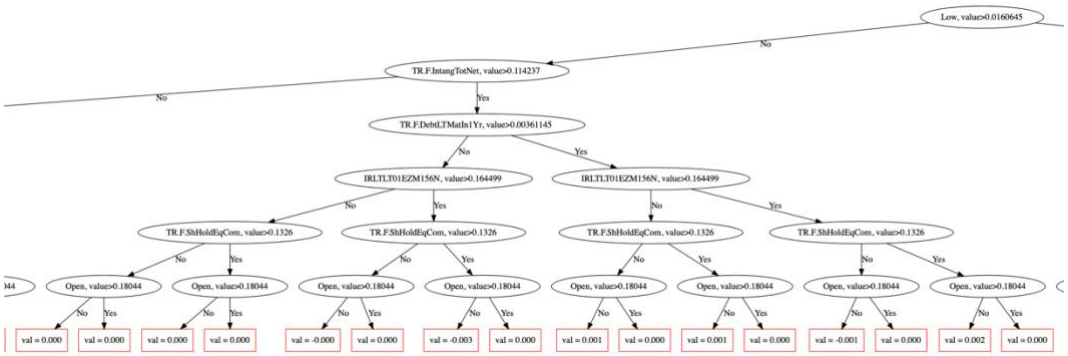


Figure 27: Tree structure

The figure describes the out-of-sample prediction mapping using the CatBoost model for the GDAXI portfolio. As seen in the figure, the tree depth equals 6 with $2^6 = 64$ terminal leaves. The figure can be read as follow: If the low value of the stock i at the time t is lower than 0.016045, then check if the total net intangible assets of the stock i at the time t is greater than 0.114237. If the total net intangible assets of the stock i at the time t is superior to 0.1142237, and if the long-term debt is superior to 0.00361145, the models go to the 4th level of the tree. If the long-term Government Bond-Yield: 10-year for the Euro Area is greater than 0.164499 the model goes to the fifth level. At the fifth level, if the common shareholders' equity has a value greater than 0,1326 (level 5th) and if the Open value of the stock i at the time t is greater than 0.18044 in the 6th level than the score for the stock i at the time t for the investment strategy is 0. As a reminder, the values of the variables are transformed values using Min-Max scaling procedure.

CatBoost Model Loss Evolution

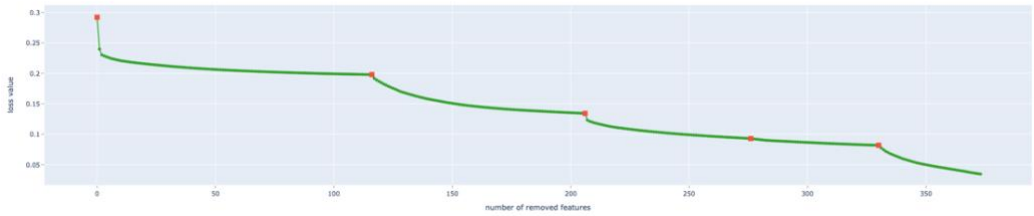


Figure 28: TOPX500 Portfolio Loss measure while gradually removing 404 features

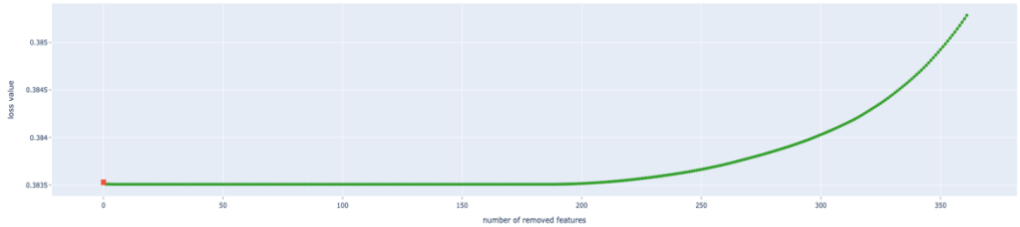


Figure 29: SPX500 Portfolio Loss measure while gradually removing 390 features

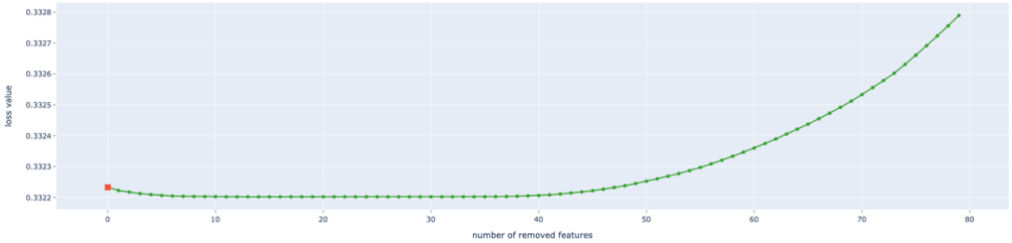


Figure 30: HSI Portfolio Loss measure while gradually removing 106 features

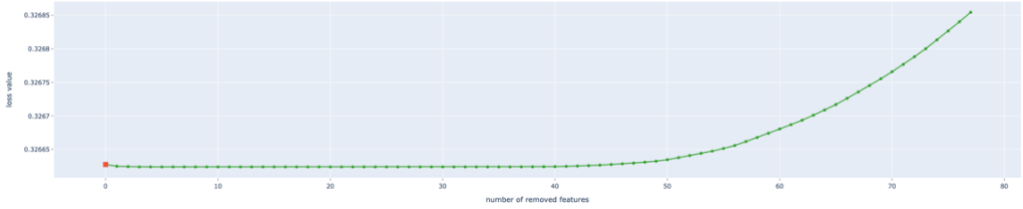


Figure 31: GDAXI Portfolio Loss measure while gradually removing 115 features

FEATURES IMPORTANCE

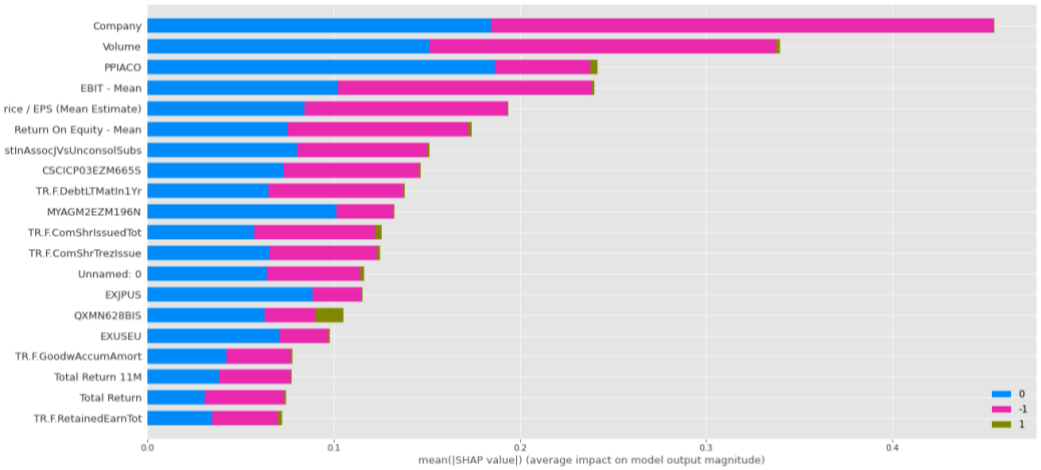


Figure 32: Feature Importance. The 20 most predictive features, GDAXI portfolio

The figure describes the effect of the most predictive features on the prediction for the GDAXI portfolio, using SHAP values on a trained CatBoost model. For example, one may understand the following figure: the EBIT - mean variable have a 20% apport on predicting the investment strategy.

Figure 33: Feature contribution to model prediction, GDAXI

The figure describes the variables' contribution on the 'Sell' forecast for the GDAXI portfolio, using SHAP values on a trained CatBoost model. One sees that the four months momentum return influences the prediction negatively by 8%.

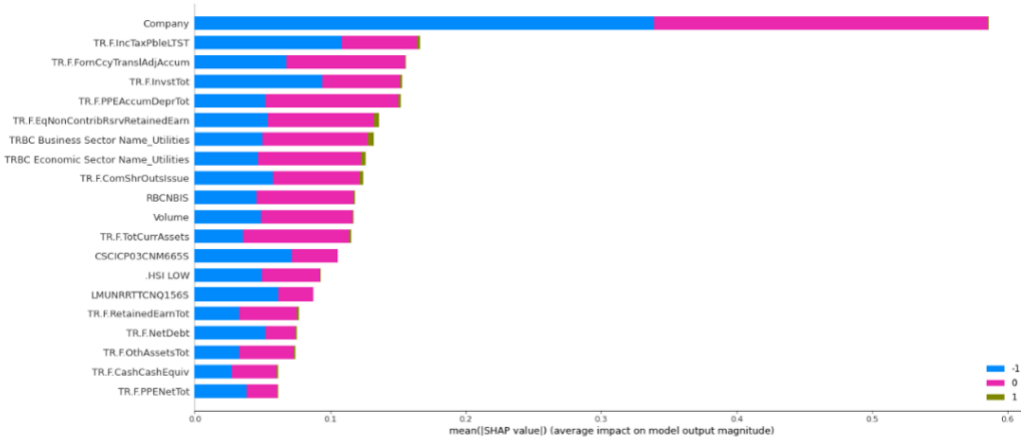


Figure 34: Feature Importance. The 20 most predictive features, HSI portfolio

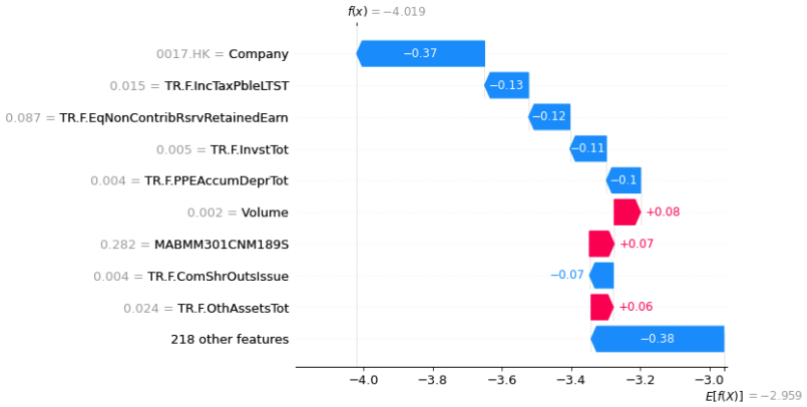


Figure 35: Feature contribution to model prediction, HSI

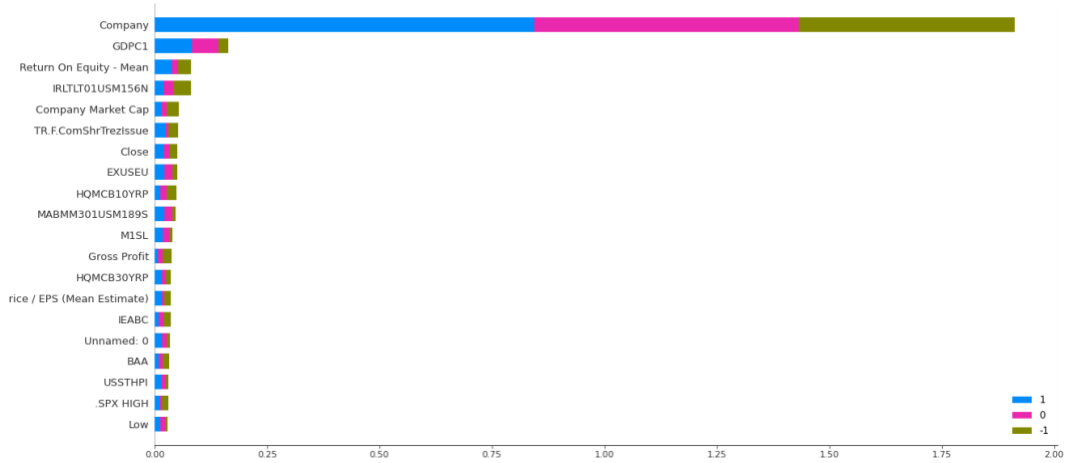


Figure 36: : Features Importance. Top 20 most predictive features, SPX500 portfolios

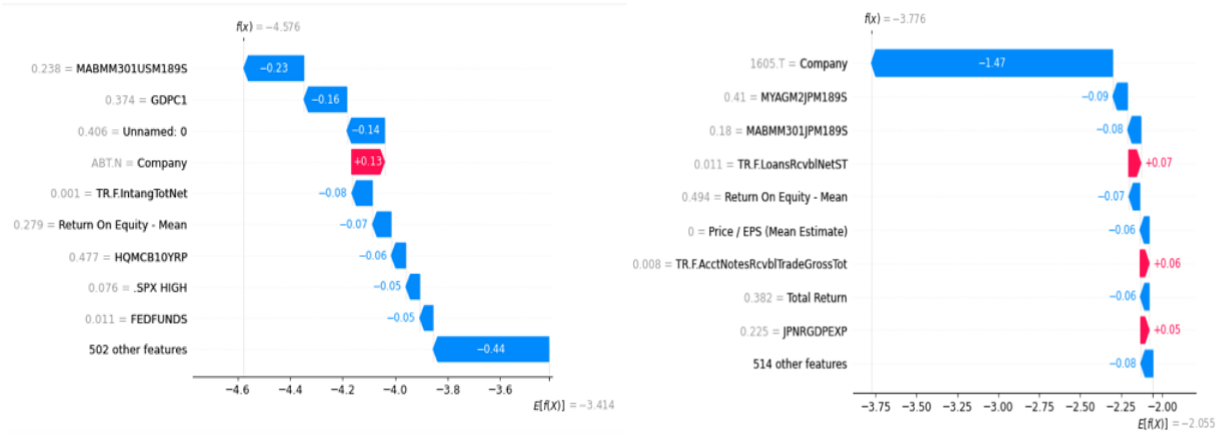


Figure 37:Features contribution to model prediction

On the left-hand side one has the features contribution for the SPX500 Portfolio, and on the right hand-side, the features importance for the TOPX500 portfolio.

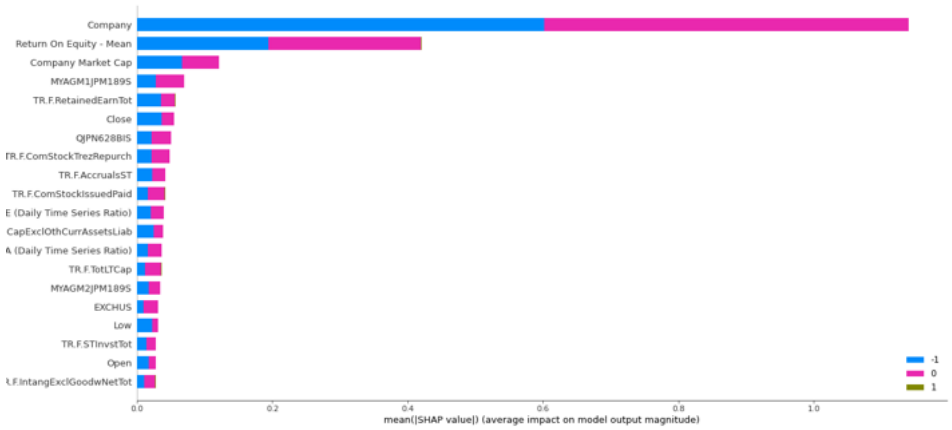


Figure 38: Feature Importance. Top 20 most predictive features, TOPX500 portfolios

FALSE POSITIVE AND FALSE NEGATIVES

Linear Regression		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
Nb. Of Features	FP/FN	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
15	FP	93.78%	31.30%	98.32%	9.98%	98.64%	4.38%	93.35%	4.96%	94.43%	8.92%	99.03%	3.78%	96.18%	6.48%	100.00%	4.04%
	FN	6.22%	68.70%	1.68%	90.02%	1.36%	95.62%	6.65%	95.04%	5.57%	91.08%	0.97%	96.22%	3.82%	93.52%	0.00%	95.96%
50	FP	93.74%	31.47%	98.88%	9.06%	98.93%	4.76%	91.47%	3.61%	95.82%	10.67%	97.03%	3.02%	96.52%	6.96%	97.30%	3.20%
	FN	6.26%	68.53%	1.12%	90.94%	1.07%	95.24%	8.53%	96.39%	4.18%	89.33%	2.97%	96.98%	3.48%	93.04%	2.70%	96.80%
80	FP	94.17%	34.19%	97.94%	7.62%	98.83%	5.14%	95.50%	3.43%	95.69%	10.21%	96.86%	2.75%	97.89%	7.45%	98.56%	2.43%
	FN	5.83%	65.81%	2.06%	92.38%	1.17%	94.86%	4.50%	96.57%	4.31%	89.79%	3.14%	97.25%	2.11%	92.55%	1.44%	97.57%
100	FP	94.54%	33.55%	97.20%	8.22%	98.75%	5.26%	96.42%	9.29%	96.26%	10.89%	97.87%	0.71%	98.53%	7.60%	97.55%	1.44%
	FN	5.46%	66.45%	2.80%	91.78%	1.25%	94.74%	3.58%	90.71%	3.74%	89.11%	2.13%	99.29%	1.47%	92.40%	2.45%	98.56%
150	FP	95.12%	35.53%	95.67%	5.73%	98.88%	5.75%	94.87%	5.55%	96.96%	12.13%	97.00%	3.00%	98.33%	7.59%	96.17%	1.99%
	FN	4.88%	64.47%	4.33%	94.27%	1.12%	94.25%	5.13%	94.45%	3.04%	87.87%	3.00%	97.00%	1.67%	92.41%	3.83%	98.01%

LSTM		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
Nb. Of Features	FP/FN	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
15	FP	71.81%	3.66%	70.87%	3.16%	96.88%	7.96%	98.93%	10.00%	96.80%	15.27%	100.00%	9.90%	95.60%	7.03%	98.68%	3.90%
	FN	28.19%	96.34%	29.13%	96.84%	3.12%	92.04%	1.07%	90.00%	3.20%	84.73%	0.00%	90.10%	4.40%	92.97%	1.32%	96.10%
50	FP	95.46%	9.74%	97.71%	2.02%	95.76%	0.85%	99.34%	0.79%	92.89%	4.46%	97.80%	2.01%	96.60%	7.74%	99.23%	3.26%
	FN	4.54%	90.26%	2.29%	97.98%	4.24%	99.15%	0.66%	99.21%	7.11%	95.54%	2.20%	97.99%	3.40%	92.26%	0.77%	96.74%
80	FP	89.32%	13.44%	93.85%	9.78%	95.76%	0.85%	95.72%	1.34%	94.36%	7.77%	97.70%	1.40%	95.52%	6.59%	96.82%	3.78%
	FN	10.68%	86.56%	6.15%	90.22%	4.24%	99.15%	4.28%	98.66%	5.64%	92.23%	2.30%	98.60%	4.48%	93.41%	3.18%	96.22%
100	FP	89.22%	13.27%	93.60%	9.68%	98.03%	2.65%	100.00%	4.71%	92.27%	4.71%	100.00%	1.71%	95.52%	6.59%	100.00%	3.90%
	FN	10.78%	86.73%	6.40%	90.32%	1.97%	97.35%	0.00%	95.29%	7.73%	95.29%	0.00%	98.29%	4.48%	93.41%	0.00%	96.10%
150	FP	89.30%	12.48%	92.78%	8.40%	97.79%	3.07%	99.67%	4.43%	92.30%	4.73%	100.00%	1.71%	95.52%	6.59%	100.00%	3.90%
	FN	10.70%	87.52%	7.22%	91.60%	2.21%	96.93%	0.33%	95.57%	7.70%	95.27%	0.00%	98.29%	4.48%	93.41%	0.00%	96.10%

CatBoost		TOPX500				SPX500				HSI				GDAXI			
		Train		Test		Train		Test		Train		Test		GDAXI		Test	
	FP/FN	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short	Long	Short
	FP	99.84%	87.84%	97.93%	47.72%	99.97%	41.26%	99.31%	22.65%	99.20%	14.60%	100.00%	6.55%	99.64%	8.13%	99.08%	3.89%
	FN	0.16%	12.16%	2.07%	52.28%	0.03%	58.74%	0.69%	77.35%	0.80%	85.40%	0.00%	93.45%	0.36%	91.87%	0.92%	96.11%

Figure 39: False-negative (FN[35]) and False-positive (FP)

The figure describes the wrongly predicted strategies (FN) and the positive class (FP). For example, one has a 6.22% chance to predict a short strategy when in fact, one should take a long position.

[35] False negative (FN) error, or simply false negative, is a statistical term to show a test result which wrongly predict a condition that does not hold. In our case, false-negative refers to money-loser given the two types of strategies Buy/Sell. $FN = \frac{TN}{TN}$, where TP are the true positive (correctly indicated conditions), and P_0 is the probability that the investment position is 'Hold'. Following same procedure, the $FP = \frac{TP}{TN}$. One may one to minimize the FN. Both FP and FN are money loser in a portfolio.

LONG-SHORT TERM MEMORY MODEL LOSS EVOLUTION

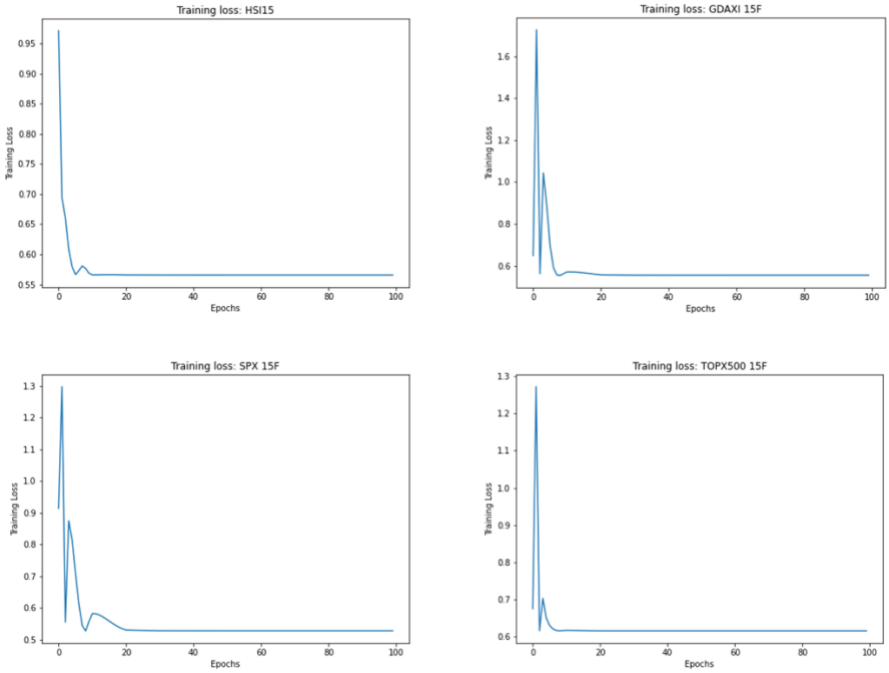


Figure 40: Training loss evolution during training an LSTM model with 15 variables

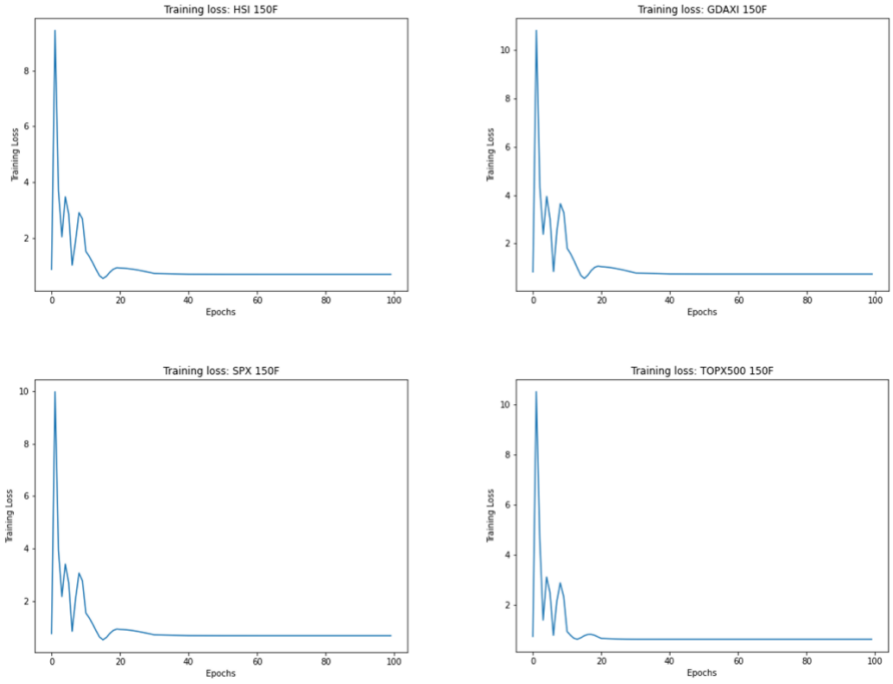


Figure 41: Training loss evolution during training an LSTM model with 150 variables

CUMULATIVE RETURNS[36]

LONG-ONLY PORTFOLIOS

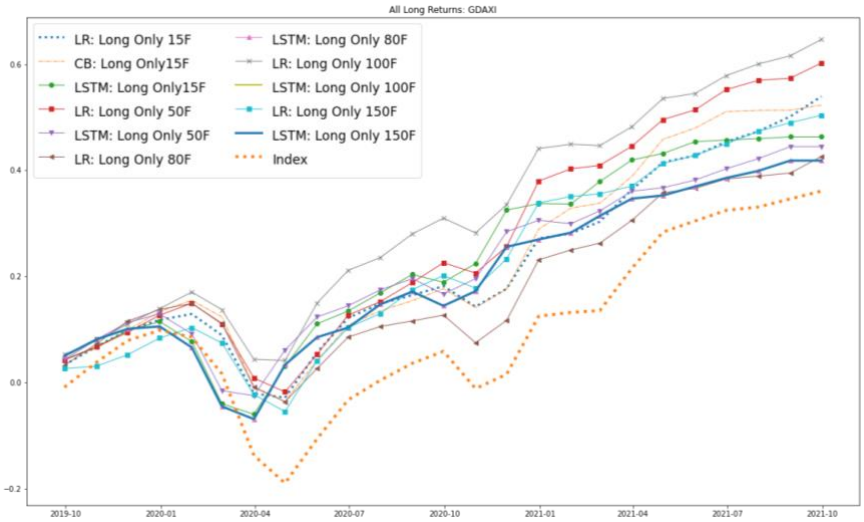


Figure 42: Performance comparison Long-only portfolios, GDAXI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecasts for selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0, one has a long position in percentage, one for values lower than 0 has short positions in percentage. Thus, all portfolios are equally weighted.

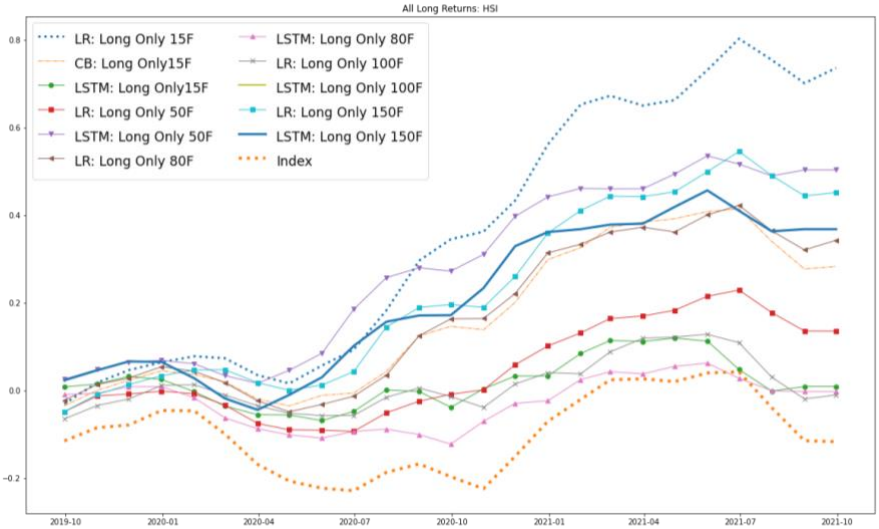


Figure 43: Performance comparison Long-only portfolios, HSI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0, one has a long position in percentage, one for values lower than 0 has short positions in percentage. Thus, all portfolios are equally weighted.

[36] In this section one may make a graphical comparison in terms of portfolio total return of a portfolio from October 2019 till October 2021 using ML models. The index cumulative return is calculated by subtracting the 10-year Treasury bond yield of the country where the index is based.

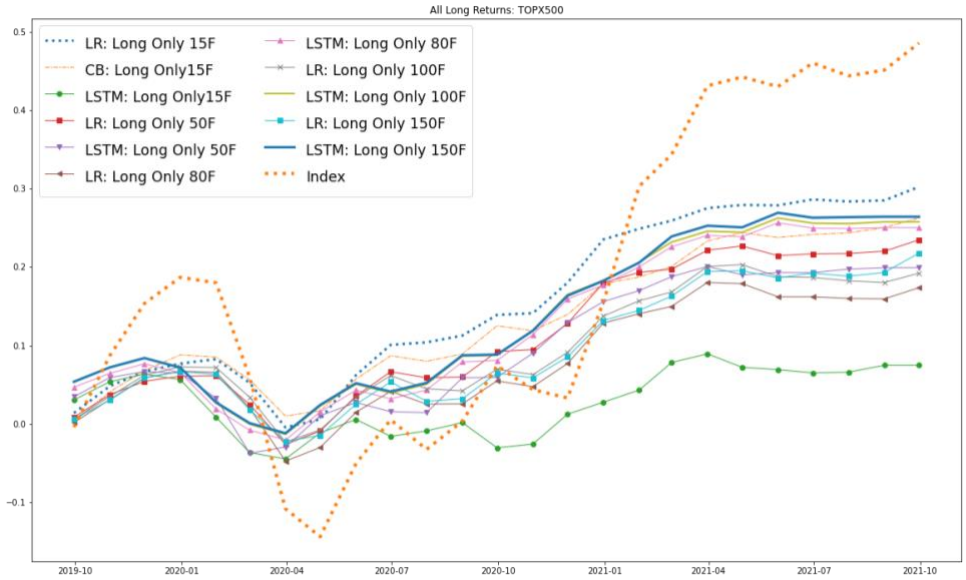


Figure 44: Performance comparison Long-only portfolios, TOPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, on for values lower than 0, one has short positions in percentage. Thus, all portfolios are equally weighted.

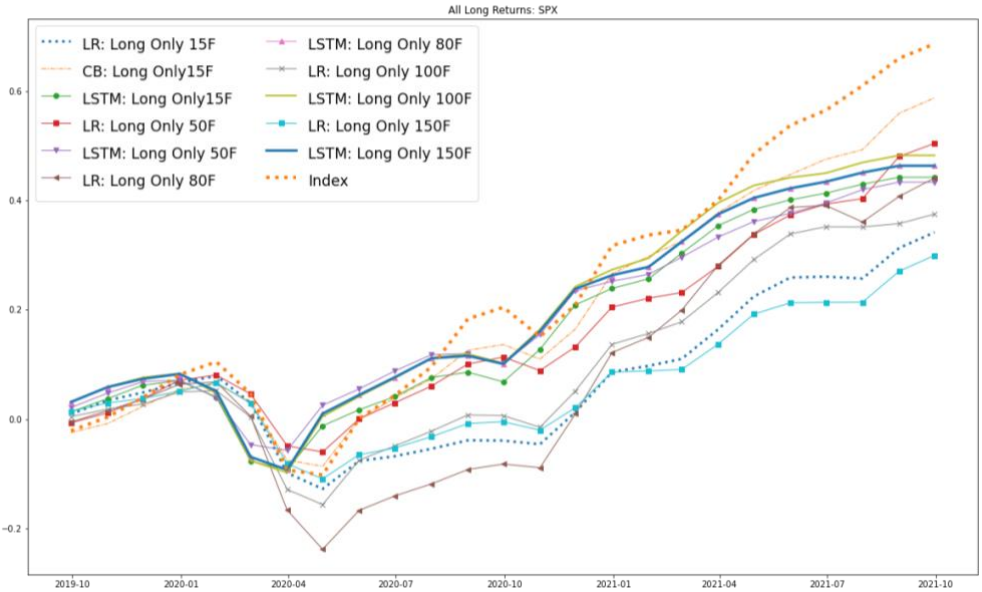


Figure 45: Performance comparison Long-only portfolios, SPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

SHORT-ONLY PORTFOLIOS

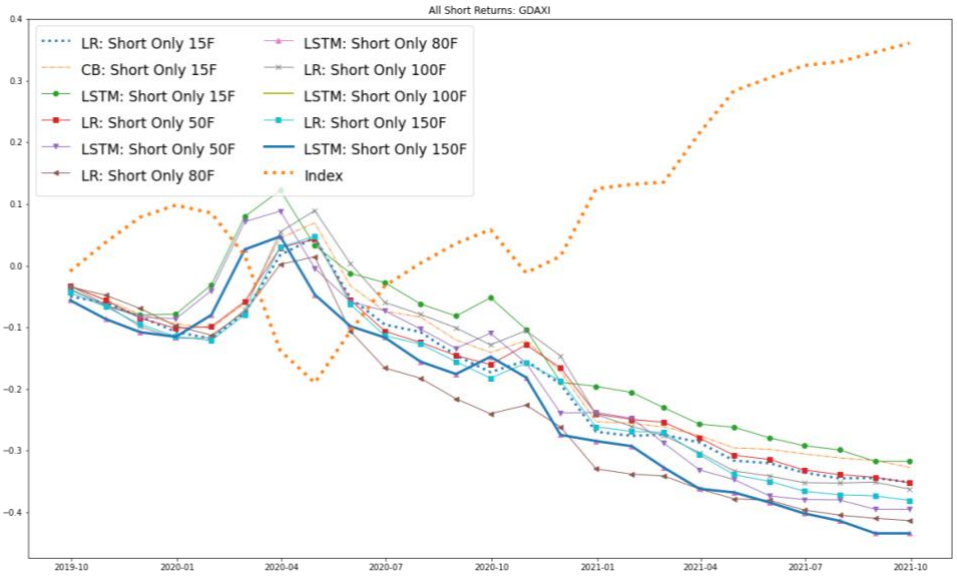


Figure 46: Performance comparison short-only portfolios, GDAXI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

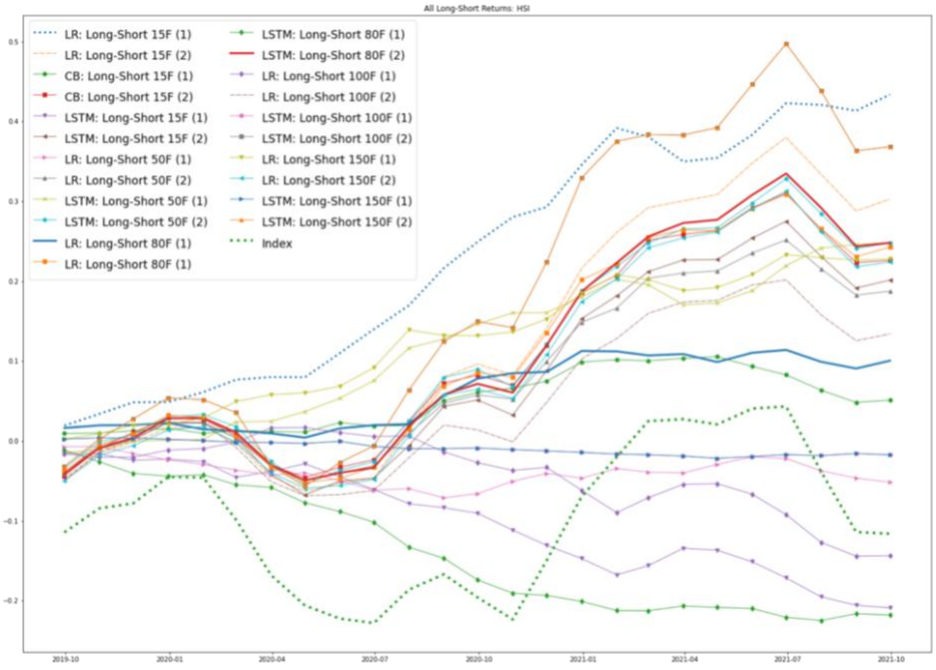


Figure 47: Performance comparison short-only portfolios, HSI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

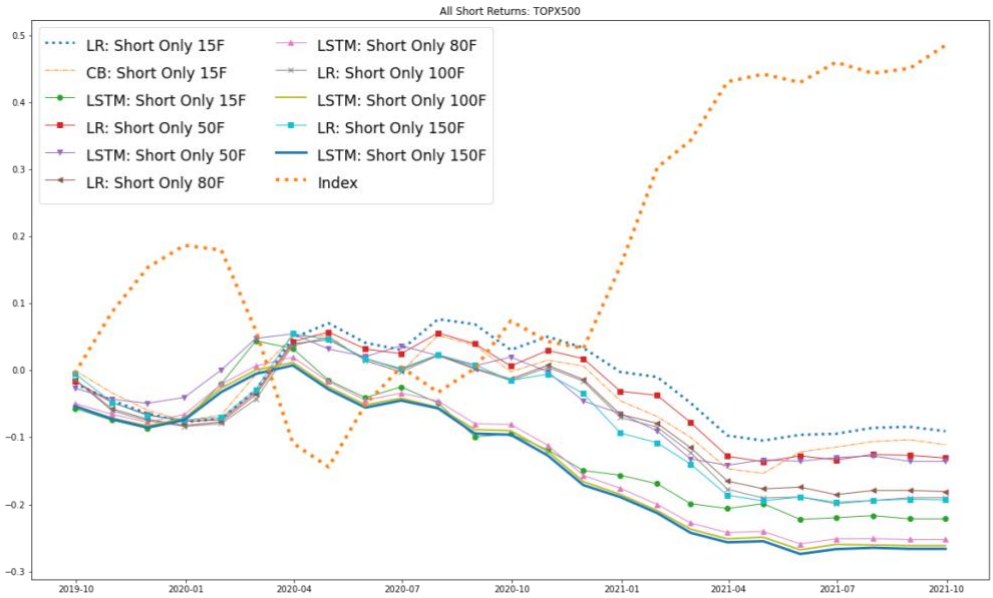


Figure 48: Performance comparison Short-only portfolios, TOPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

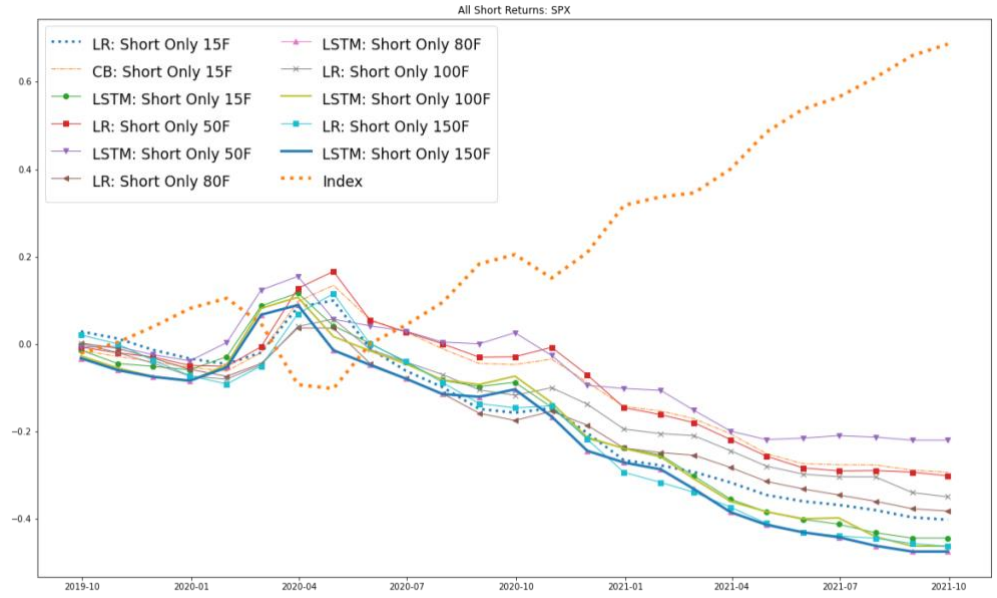


Figure 49: Performance comparison Short-only portfolios, SPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

LONG-SHORT PORTFOLIOS

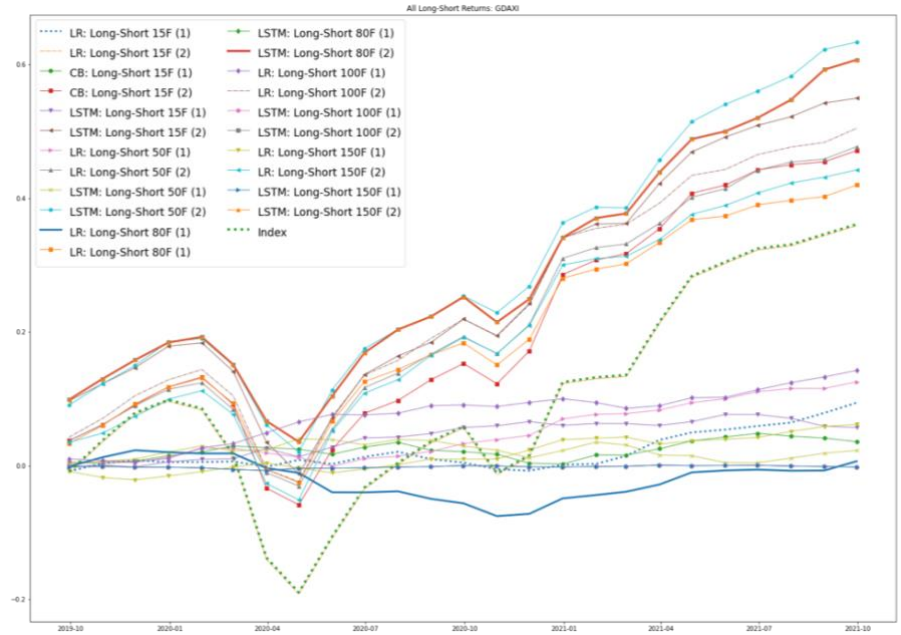


Figure 50: Performance comparison Long-short portfolios, GDAXI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

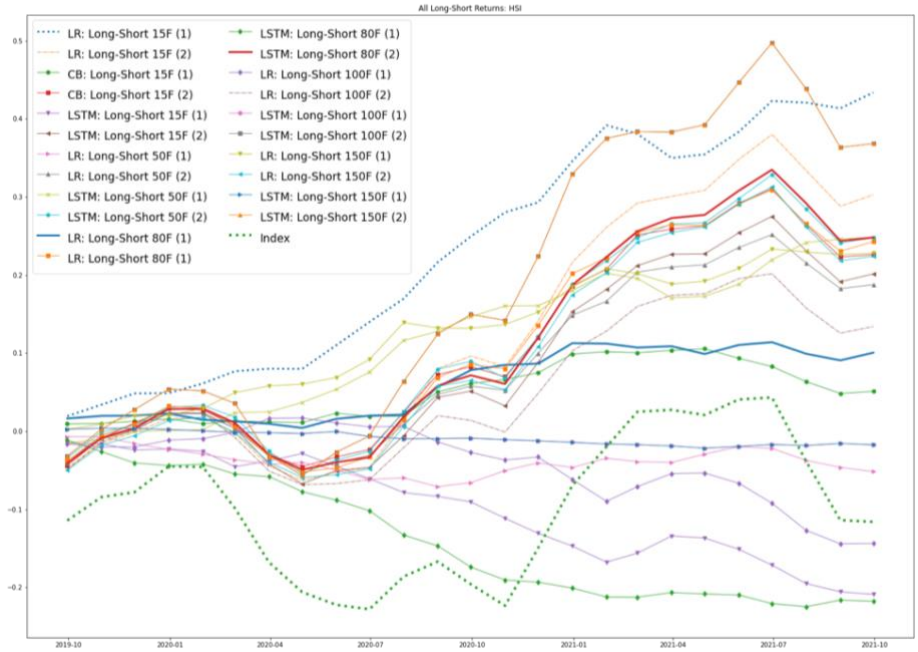


Figure 51: Performance comparison Long-short portfolios, HSI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

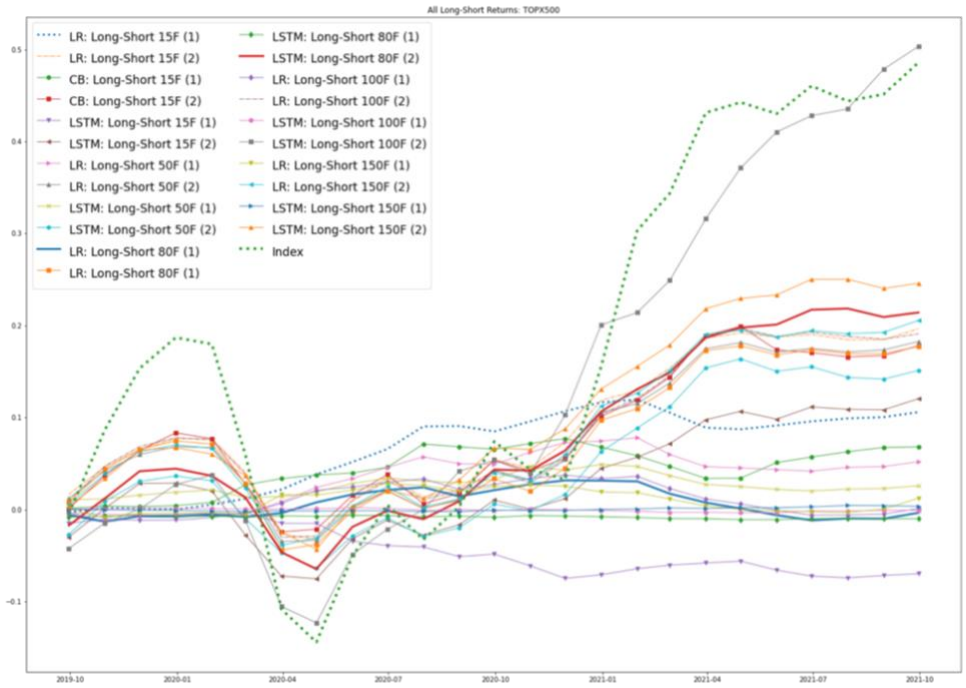


Figure 52: Performance comparison Long-short portfolios, TOPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

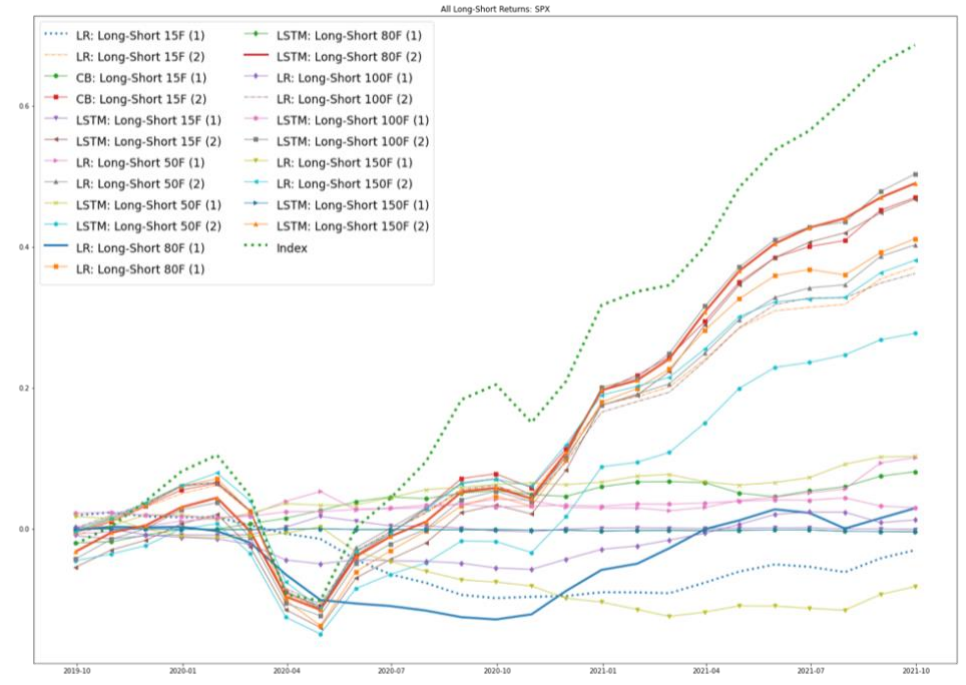


Figure 53: Performance comparison Long-short portfolios, SPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

ALL PORTFOLIOS TOGETHER. AN ASSET PRICING ML APPROACH

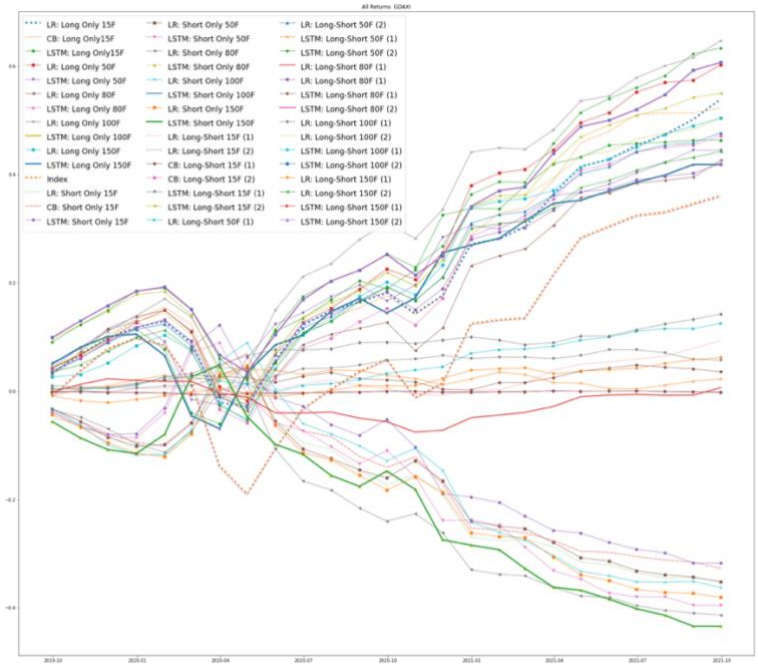


Figure 54: **Performance comparison of all portfolios, GDAXI**

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

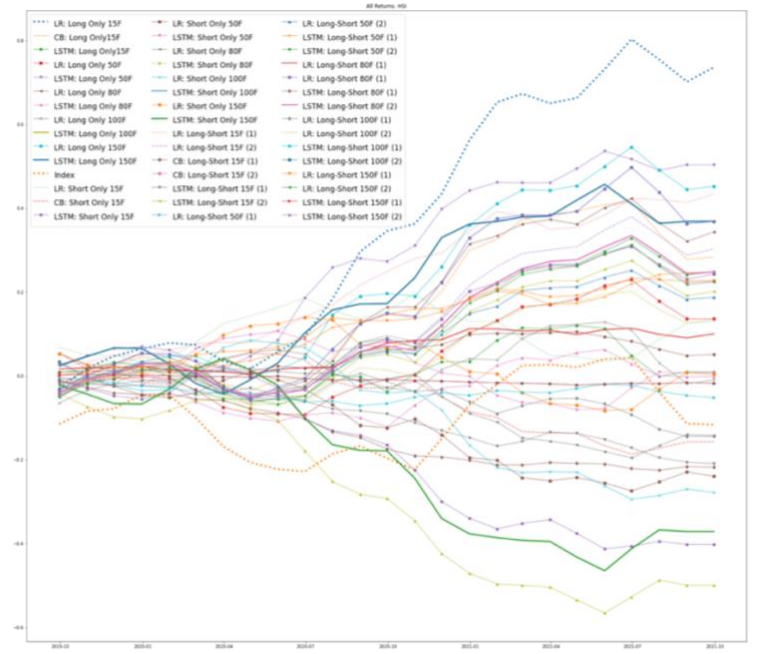


Figure 55: Performance comparison of all portfolios, HSI

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

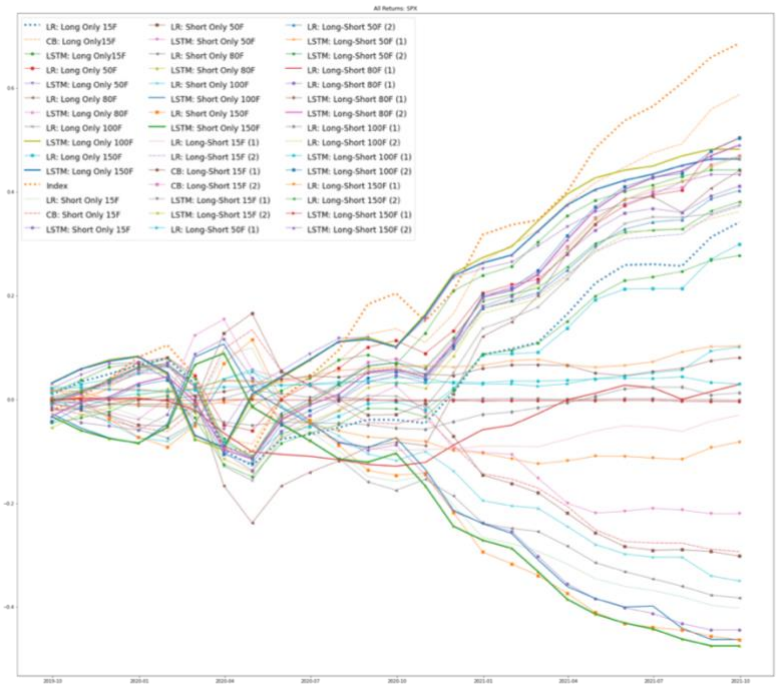


Figure 56: Performance comparison of all portfolios, SPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.

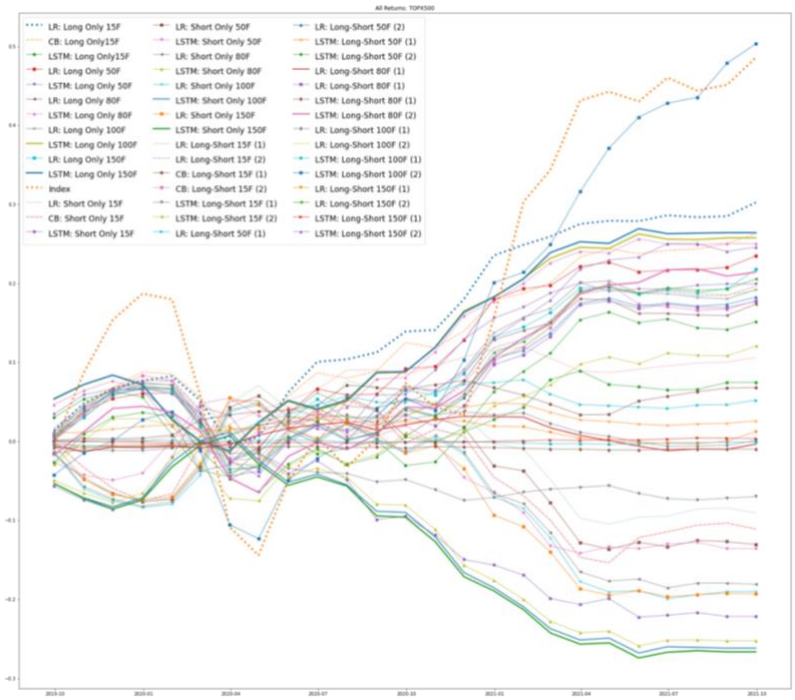


Figure 57: Performance comparison of all portfolios, TOPX500

The figure describes cumulative returns of Long-Only portfolios on out-of-sample machine learning forecast selecting 15, 50, 80, 100, 150 variables for Linear Regression, CatBoost, and LSTM Model. On the vertical lines, for values greater than 0 one has a long position in percentage, one for values lower than 0 one has short positions in percentage. Thus, all portfolios are equally weighted.