**Department of Business and Management**

**Bachelor Degree in Management and Computer Science**

# Asset Pricing via Machine Learning

**Relatore:**
**Prof. Michela Altieri**

**Candidato:**
**Matteo Antonazzo**

**Academic Year 2021/2022**

**SUMMARY**

# Introduction

This paper's aim is to apply machine learning methods to predict stocks' risk premium. The content of this paper is one of the most debated topics in finance: assets pricing. The question is how to predict effectively and accurately stocks' excess returns. The idea behind this paper is that stock price can be forecasted by proven economically significant features. These features are both price indicators and firm-level characteristics. This paper uses machine learning methods to investigate which characteristics have more predictive power for stock returns. Asset pricing theories are the product of decades of studies, consequently the candidate economic variables are very large in number. Dimensionality reduction is one of the machine learning applications. In that way, less weight is given to less significant predictors, or they can be even deleted and not taken into consideration. Until now, the relationship between selected variables and risk premium was assumed to be linear. Machine learning methods not only consider non-linear patterns but can also implement interaction between predictors. All these reasons are behind the choice of combining finance and machine learning. The benchmark for this thesis is Gu et al. (2020) paper called "Empirical Asset Pricing via Machine Learning".

The paper structure provides first a literature review, which will resume the pillars in asset pricing field. The second chapter will be a broad introduction of the machine learning techniques used in this paper. The analysis includes the following methods: CAPM, FamaFrench three-factor model, simple linear regression, penalized approach, Principal Component Regression (PCR) and Gradient Boosted Regression Trees (GBRT). The third part will be the empirical analysis where all those models would be applied to a dataset of US market listed firms, from 2006 to 2020. The last part consists in a final resume of the empirical results and consequent conclusions.

The results of our analysis show that among all implemented models, Gradient Boosted Trees is by far the best model. This conclusion reflects Gu er al. (2020) results. In fact, without considering Neural Networks, their best performing model was GBRT, outperforming Ordinary Least Squares, penalized approaches, PCR and Bagging Trees. For what concerns variable importance, the results reflect once again Gu et al. findings. In fact, in our analysis, size factors (mvel1 and mve0) result to be the most influential variables. The set of other relevant variables, indeed, as in Gu's paper, include momentum variables, illiquidity, bid-ask spread and return volatility.

# 1 Literature Review

The first pillar of asset pricing literature is the Capital Assets Pricing Model (CAPM). It was elaborated by William Sharpe and John Lintner (1965), and it is still widely used. The main advantage of CAPM is that it expresses an immediate relation between risk and return, and an easy way to forecast the

return. Its simplicity represents at the same time its greatest drawback. In fact, CAPM performs relatively poorly empirically, and it is the consequence of theoretical flaws and difficulty in implementing a valid set. The main assumptions of CAPM, added by Sharp and Lintner to Markowitz model, are 'complete agreement' (i.e., investors agree on asset return distribution) and 'unlimited borrowing and lending at risk-free rate'.

According to Fama and French (2004), 'unrestricted risk-free borrowing and lending is an unrealistic assumption'. In 1996 they proposed an alternative to the Capital Asset Pricing Model. More precisely, Fama-French three factor model can be considered an integration of CAPM. Fama and French starting point were previous studies that showed a connection between return and firm-level characteristics. Patterns that were not explained by CAPM were called 'anomalies'. So, another model called 'Fama-French three factor model' was elaborated. Practically, they added two new factors to better picture the market behaviour. In fact, the first factor (the market one), is the same factor that is used in CAPM. The other two factors are the size factor, also called Small Minus Big (SMB), and the value factor, also called High Minus Low (HML). The first one reflects the ability of some small company to outperform larger ones. The second factor expresses the tendency of value stock to perform better than growth stock. This model showed better performance than the CAPM and the Arbitrage pricing theory. The literature after this publication debated a lot about the economic meaning of the Fama-French factors (for example, Chung, Y. Peter et al. 2006). While it was clear that market factor was a proxy for risk, the interpretation of the other factors was not clear. Fama and French claimed that those factors represent firm distress. Rolph (2003) claimed that the factors are a proxy for leverage, while several other papers, like for example Berk (1995), suggested that the predictive power of size and value factors is spurious. An alternative explanation is proposed by Chung, Johnson and Still (2006), who claimed that SMB and HML are proxy for the part of risk not captured by the market factor.

The following literature was about finding other anomalies and other factors that can explain them. The cross-sectional studies focused on proving financial indicators as valid predictors for risk premium. An example could be Cooper et al. (2008) paper that proves the relation between asset growth and expected return. Another mention must be made to Jagadeesh and Titman (1993) because, for the first time, they introduced price momentum for US stock prices. This step is very important because price momentum is proven to be one of the most significant predictors for excess return. There are many other works in the literature that increased the predictors set. In that sense, Lewellen (2011) took a good picture of the cross-sectional analysis situation, providing answers to the question on 'whether the characteristics can actually be used, either individually or in combination, to estimate expected stock returns in real time'. He also proves the validity of cross-sectional stock returns prediction.

For what concerns the financial applications of machine learning technique the literature is not as extensive as one could imagine. The machine learning application is often limited to just one method, and it is difficult to find a paper that can draw the big picture. On the contrary, the shrinkage methods are the most investigated in the literature because they offer the immediate answer to the dimensionality reduction problem. For example, Freyberger (2020) uses adaptive group lasso to investigate the additional contribution of each variable for the prediction of expected return. Even though his model performs better than the traditional one (i.e., Fama-MacBeth regression), the resulting implementation is still sensible to outliers. Also, Kozak (2020) and Rapach et al. (2013) apply Lasso on the financial framework. The resulting model is not more complex of the traditional ones (four or five factors), but it performs better out of sample testing. Another perspective to the dimensionality reduction problem consists in the Principal Component Analysis. PCA consists basically in rotating the data space in order to reduce the dimension. This possibility is investigated by Giglio and Xiu (2021): their approach 'approach uses principal components of test asset returns to recover the factor space and additional regressions to obtain the risk premium of the observed factor'. Their model is considered a two-factors cross-sectional model, where the factors are the two Principal Components. In that way, all the selected predictors are 'weighted' and included in the prediction, overcoming the 'omitted predictors' problem. Coqueret and Guida (2018), instead, used regression trees to 'determine which firm characteristics are most likely to drive future returns.' Their predictor set was made by 30 attributes and one related to momentum seemed to be 'by far' the most impactful. Hastie er al. (2009) proved that ensemble methods tend to perform better than individual algorithms (i.e., random forest outperform regression tree). According to general machine literature 'random forest is one of the most automatic algorithms'. In this paper, random forest would be deeply investigated.

The deep learning applicability was discussed, among the others, by Heaton et al. (2016). The paper explores the applicability of deep learning hierarchical models to financial problems, highlighting the point that deep learning can 'detect and exploit interactions in the data that are, at least currently, invisible to any existing financial economic theory.' The newest neural network application (i.e., Autoencoder) is proposed by Gu et al. (2021). Even though neural networks perfectly fit some kind of financial application, they are not widely used in this field. This is because finance require a deep understanding of the decision-making process, and neural networks lack of interpretability. Hidden layers weight inputs in a not human understandable form, and this is a great limitation.

Finally, Gu et al. (2020) is the paper which compares all the above cited techniques and offers a complete picture of the link between asset pricing and machine learning. The common takeaways of almost all those articles are:

- The perfect match between machine learning and stock return prediction, in terms of dimensionality reduction, variable selection, variable importance and avoiding overfitting.

- The importance in stock return prediction of variables such as price momentum, volatility and liquidity.

# 2 Methodology

This section describes the collection of methods used in our analysis. Each subsection would be dedicated to a specific method, and each method would be characterized firstly by the description of the statistical model, secondly by the objective function for parameter estimation, and thirdly by the computational algorithm. The statistical model part would consist in the delineation of a general function for risk premium prediction. The second part, i.e., the one about parametrization, is essential to deal with the risk of overfitting the model, in order to improve out of sample performance. In fact, every objective function shares the same base goal: minimizing mean square prediction errors (MSE). The Mean Squared Errors formula is the following:

$$MSE = \frac{\sum(y_i - \hat{y_i})^2}{n}, \tag{1}$$

where $n$ is the number of observations, $y_i$ is the observed value and $\hat{y_i}$ is the corresponding predicted value. Modification about the objective function can be made, for example, with respect to robustification against outliers or parametrization penalties. The rationale for the third part, i.e., the part about the specific algorithm, lies in the fact that there are many variants of every machine learning technique, and its aim is to let the reader know which one would be used in the analysis. The asset's excess return (or risk premium) is described as an additive prediction error model. This means that every prediction is intended as the expected return plus the error:

$$r_{i,t+1} = E_t(r_{i,t+1}) + \epsilon_{i,t+1}\,, \tag{2}$$

where,

$$E_r(r_{i,t+1}) = g * (z_{i,t}). \tag{3}$$

This notation means that the excess return prediction is isolated as a function of a set of predictors, i.e., $z$. In this formula $i$ stands for the stock index and $t$ for the year. A crucial assumption is that the return depends on this function $g$ that does not change over time or across different stocks. In other words, the prediction is independent from time and stocks, but it is made just according to the stock predictors. This is a significative difference with respect to cross-sectional model that re-estimates the model over a time period or estimates a model for each stock.

## 2.1 Sample splitting and tuning

The common sample splitting procedure consists in splitting the whole data space into two subsets: training set and test set. Machine learning algorithms, indeed, require a further step. In fact, when a great number of both observations and predictors occurs, it is necessary to create a new subset, the validation one. In order to fully understand this step, we have to introduce the concept of 'hyperparameter' (or, equivalently 'tuning parameters'). Hyperparameters are the specific parameters of each machine learning algorithm and control their complexity. They can be for example the penalization parameters in lasso, the number of layers in a neural network, the number of leaves in a tree or the number of trees in a random forest. Their 'tuning' is essentially the biggest defence against noise (i.e., random or systematic error) and overfitting. So, as it was said before, the standard machine learning procedure consists in dividing the whole sample into three disjoint subsets: training, validation and test set. The first one is used to estimate the model and the parameters. The second one is used to 'validate' those parameters which means that predictions are made on the validation set using the model built in the training set. Then the objective function is evaluated on forecast errors and iteratively re-estimated changing hyperparameters in order to optimize the function. The parameters are chosen based on the performance on the validation set but are estimated just by the training set. The idea is to simulate an out of sample test with the aim of making the parameters tuning reliable and robust. Clearly, the validation set does not represent anymore an out of sample test since it is used to tune the parameters. Since that, the third subset, the 'test set', is used to test the predictive performance of the model.

The forecast evaluation literature provides three different splitting methods: the 'fixed' scheme, the 'rolling' scheme and the 'recursive' scheme. The 'fixed' scheme simply consists in splitting, generally randomly, into the three different subsets. It maintains each set composition constant. The 'rolling' approach consists in iteratively shifting the set compositions in order to include always more recent data. The 'recursive' performance evaluation scheme gradually increases training and validation sets. Practically, it works just as the 'rolling' scheme, without keeping the number of observations in the sets constant but increasing it gradually. Other papers about this topic use the last two schemes (i.e., 'rolling' and recursive') or sometimes even a hybrid of them (Gu et al. 2020). The great drawback is that they are computationally very expensive. Since pros does not overcome this great disadvantage, the chosen performance evaluation scheme would be the 'fixed' one.

## 2.2 Capital Asset Pricing Model

The first model would be the first pillar of asset pricing theory. In the literature the rationale behind this model is exhaustively explained. So, some practical annotations will follow. The CAPM formula is the following:

$$R_i = R_f + \beta_i * (R_m - R_f) \quad (4)$$

where $R_i$ is the risk premium, $R_f$ the risk-free rate and $R_m$ the market return. $\beta_i$ is the coefficient regression obtained, and represents, economically, stock sensitivity to market change. A $\beta_i$ greater than 0 means that stock return follows market behaviour, while a negative $\beta_i$ means that the risk premium has an opposite behaviour with respect to the market. Standard & Poor index (S&P 500 index) is almost always used as a proxy for market behaviour.

## 2.3 Simple Linear Model

The second model analysed is the "simple linear regression" one. This is the simplest among all the models presented in this paper. Given that we are able to represent all the data points as a scatterplot, the underlying idea of the model is to 'build a line' that reflects the tendency of all those points. More mathematically, the idea is to find the line that minimizes the distance with respect to all the points in the dataspace. The linear model states that the conditional expectation can be approximated as a linear function of the predictor vector,

$$g(z_{i.t}; \theta) = z'_{i,t}\theta, \quad (5)$$

where $g$ is the function, $z$ is the stock and $\theta$ is the parameter vector. As the model's name suggests, this method considers only linear effects and interactions between variables.

The objective function would be the Standard Least Squares function (or '$l_2$'). This function is used to approximate the solution of overdetermined systems (i.e., systems with more equations than variables) and consists in minimizing the residual sum of squares (considering the residuals as the difference between the prediction and the observed value),

$$L(\theta) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{i=1}^{T}(r_{i,t+1} - g(z_{i,t+1}; \theta))^2, \quad (6)$$

it is predictable that the performance of this model would be poor. This is due to the number of dimensions (i.e., predictors) that the problem involves. Given that, the simple linear regression is a reference point for highlighting the progress made with other models. Same clever implementations of this model can be made, such for example modifying the objective functions (giving weights to variables in order to give more weight to the ones more statistically or economically valid) or penalizing the presence of many predictors. The last case is the so called 'penalized approach'.

## 2.4 Penalized approach

The dimensionality problem can be solved by reducing the number of predictors. This is the underlying idea behind penalized approaches. As aforementioned, the signal to noise ratio is relatively low in this kind of problem, which means that, at a certain level, the model tends to add noisy information rather than significative one. An optimal starting point could be reducing the number of predictors. The literature in that sense offers an immediate solution: the penalized linear approach. The base step consists once again in estimating a straight line which minimizes the RSS. The statistical model would be the same one of the previous paragraphs (i.e., the baseline considering only linear interactions). The great difference is represented by penalty appended to the standard loss function:

$$\mathcal{L}(\theta;\cdot) = \mathcal{L}(\theta) + \phi(\theta;\cdot), \tag{7}$$

where $\varphi(\theta;\cdot)$, is the penalty function. The most common shrinkage methods are three: Ridge regression, LASSO (Least Absolute Shrinkage and Selection Operator) regression and Elastic Net. Ridge regression shrinks the regression coefficients so predictors with less significant contributions will have a coefficient close to zero. The penalty term of Ridge regression is called '$l_2$ *norm*', and basically is the sum of squared coefficients,

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y_i} - y_i)^2 + \lambda\sum_{i=1}^{N}|\theta_i|, \tag{8}$$

The critical part is 'tuning' $\lambda$. with $\lambda$ equal to zero the model would be the OLS but as $\lambda$ increases the effect of penalty grows until the coefficients get close to zero. Ridge selection would include all predictors into the final model. Lasso regression, indeed, would shrink some coefficients to exactly zero. Its penalty term is called '$l_1$ norm' and practically is the sum of the absolute value of the coefficients,

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y_i} - y_i)^2 + \lambda\sum_{i=1}^{N}\theta_i^2. \tag{9}$$

The essential part is again 'tuning' the $\lambda$ coefficient. The greatest advantage of Lasso regression is that the result is easier to interpret but it usually works better when there is a great difference among the regression coefficients. On the other hand, Ridge works better when the coefficients are almost of the same size. The third method is the so-called 'Elastic Net'. Elastic Net is a model which is penalized by both $l_1$ and $l_2$ norm. So, it can be considered a hybrid model with respect to the other two. The product of this penalization is that some coefficients are shrunken close to zero (Ridge part) and other exactly to zero (Lasso part). The penalty term in this case would be:

$$\phi(\theta, \lambda, \rho) = \lambda(1-\rho)\sum_{j=1}^{P}|\theta_j| + \frac{1}{2}\lambda\rho\sum_{j=1}^{P}\theta_j^2. \quad (10)$$

If $\rho$=0 the penalty term would be the $l_1$ norm (i.e., Lasso regression), if, instead, $\rho$=1 we are in the case of Ridge regression. For every value different from 1 and 0 the model would be the hybrid one. In that case, two hyperparameter should be tuned: $\rho$ and $\lambda$.

## 2.5 Principal Component Regression

Choosing a subset of predictors set is just a way to reduce dimensionality problems. Another way to do it is creating new predictors as linear combinations of the given ones. This is the essence of Principal Component Regressions. The rationale for PCR is that penalized approaches are prone to find local optima rather than global ones. What is more, they are not able to detect systematic noise, which is quite frequent in economic data. PCR is a two-step method. The first step is the so-called Principal Component Analysis (PCA) which explores statistical correlations among elements of a given dataset. It aims to find data representation that retains the maximum non redundant and uncorrelated information. This process is basically a rotation of the dataspace that allows us to delete redundant information.

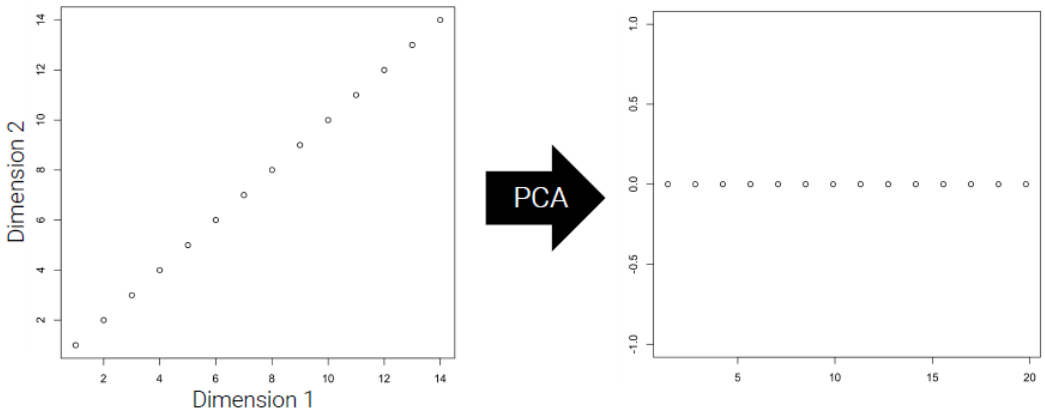



Figure 1

**Principal Component Analysis Example**

The figure above represents an example of PCA functioning. The initial dataset has two dimensions while the final one has just one dimension. It is easy to notice that PCA just rotates the axes.

In the figure above it is possible to observe a simple example of PCA. The two dimensions in the first graph become one dimension (i.e., the x-axes) in the second plot. This dimension is the principal component that explains the maximum variance (in this case the whole variance) of the dataset. So, PCA outputs are those linear combinations of predictors called Principal Components. It is important to underline that PCA does not apply to dimension reduction itself, but just recombine variables. The second step, the Principal

Component Regression (PCR), reduces the dimensionality. PCR regularizes the problem by eliminating low variance components. So, the regression is re-arranged as follows:

$$R = (Z\Omega_K)\theta_K + \tilde{E}, \tag{11}$$

$Z$ is the predictors' vector, while $\Omega_K$ is a matrix containing weights combinations, so $Z\,\Omega_K$ is the dimension reduced version of the original dataset. PCR chooses the matrix's weights recursively in order to solve the following equation:

$$w_j = \arg\max Var(Zw)\ \ s.t. w'w = 1, \quad Cov(Zw, Zw_l),\ l = 1,2 \dots, j-1, \tag{12}$$

Clearly, PCR search for linear combinations of $Z$ which most explains predictor set variance. The objective function clearly shows that weights selection is not bounded by forecasting at all. The emphasis is just on selecting components with higher variance. The solution for the above written equation is obtained via singular value decomposition, making the algorithm extremely computationally efficient.

## 2.6 Decision Trees and Random Forest

Until now, all the models taken into consideration do not account for interactions among predictors. A feasible alternative could be creating a new predictor matrix by multiplying the predictor set time itself. Clearly, it would be extremely expensive from the computational perspective. Random trees are machine learning algorithms that incorporate multiway predictor interactions. Trees are nonparametric models and the idea behind them is completely different from the regression logic. Trees aim is to divide the dataspace into 'areas' which contain observations that have a similar behaviour. A tree is formed by repeating a series of steps. At each step is performed one 'split'. According to the attribute selection method ('splitting criterion') observations are divided into two 'bins' with respect to one attribute. The most common splitting methods are three: Information Gain, Information Ratio and Gini Gain. The Information Gain is computed as follows:

- Firstly, for each observation is evaluated the expected information needed to classify it, as the sum of the probability to belong to a certain class times the log probability

$$Info(D) = -\sum_{i=1}^{m} p_i log_2\,(p_i). \tag{13}$$

- Secondly, for each class of attributes (i.e., each variable), called $A$, is calculated the expected information gained from portioning for that specific class

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} * Info(D_j). \quad (14)$$

- The last step consists in subtracting the total information needed, by the information given by any class. The result is the so-called Gain. The attribute that grants the highest Gain is the one selected for the split

$$Gain(A) = info(D) - info_A(D). \quad (15)$$

Gain ratio approach is a normalization of the Information Gain method. In fact, it takes into consideration the total number of observations. A 'split info' factor is introduced, and each gain is divided by this term

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} * log_2\left(\frac{|D_j|}{|D|}\right). \quad (16)$$

So, the algorithm does not consider anymore just the absolute gain but the relative one. The Gain Ratio is given by

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}. \quad (17)$$

The last method is the Gini Gain. It evaluates the 'impurity' of each class as

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2, \quad (18)$$

where $p_i$ is the probability of each observation to belong to each class. A class gain then is given by the weighted sum of the impurity of each of the two subset created, since Gini Gain works just for binary splits. Then, as in the previous models, it is selected the attribute as the one which maximizes the difference between total gain and attribute specific gain. Given that in the literature Gini impurity index method is considered the best, it will be the one used in our analysis.

After choosing the splitting criteria, the algorithm keeps splitting the observations until one of this three criteria is met: all the remaining obervations belong to the same class, there are no attributes remaining or there are no observations remaining.

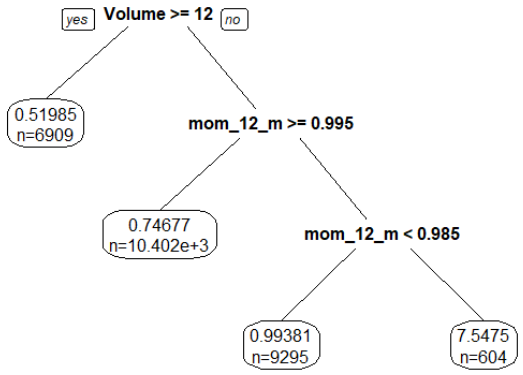



Figure 2

**Regression Tree Example**

The example shows a tree that takes into consideration only two variables and according to them splits the dataset.

The figure above is an example of a regression tree. The two variables taken into account in the example are Volume and Momentum. The first varibale which is considered is the Volume (i.e., the one which maximize Gini impurity index). The space is divided into two parts: observations whose volume is greater or equal than 12 (6909 observations) and the ones with volume smaller than 12. The two subsequent splits are made by Momentum. The second subgroup is so divided into three other parts: observations with price momentum equal to or greater than 0.995 (10402 observations), observations with momentum smaller than 0.985 (9295 observations) and the one with momentum in between 0.985 and 0.995 (604 observations). The whole data space is so divided in four parts. Firms with greater return, in this specific example, are the ones with a price momentum between 0.985 and 0.995 (i.e., a return of 7.5475). This is a good example to understand how regression trees work practically.

The main advantages of decision trees are : the interpretability of the model, the possibility to include both numerical and categorical variables, the robustness and the low computational effort required. The main great disadvantage is that decisions trees are prone to overfitting. An idea could be to post-prune the tree. In order to do that, a cost function is build to find the sub-tree that minimize it.

The recent literature in that field shows clearly that the best methods are the ones that involves ‘ensemble’ regularizations. It means putting toghether forecasts from many differet trees. Most common ‘ensemble’ methods are Gradient Boosted Trees and ‘bagging’ trees. GBRT are a set of shallow trees whose forecasts are combined together. The idea is that combining ‘weak learners’ (shallow trees) is more robust and less complex than reling just on a single big tree. To be more detalied, GBRT fits a shallow tree (e.g.,depth L=1) , and then use the residuals to fit an equally shallow tree. This time the forecast component is shruken by a factor $\nu$ (between 0 and 1), in order to avoid residuals overfitting. At each step this process is repeated until the number of total trees is B. The output is a set of shallow tree with three hyperparameters (L, B, $\nu$) , chosen thorugh the validation set. Also ‘bagging’ trees aggregates different trees predictions.

While GBRT are based on a set of 'weak learners', random forests build B different trees, which are deep and overfitted. The difference between those trees is the dataset. In fact, at the beginning of the process, B bootstrapped sample of data are built, and each one is used by a different tree. Each tree contributes to the final output, since the predicted value would be the average of all the results. Since there is the concrete risk of creating highly correlated tree, Random Forest algorithm decorrelates it using the 'dropout' method. It consists in considering only a random subset of variables at each split. In this way, few trees will not split on 'dominant' variables, granting a more widespread range of solutions. Also for Random Forest the hyperparameters would be three: trees depth (L), number of predictors for each split (n) and number of bootstrap sample (B).

## 2.7 Performance Evaluation

Assessing predictive performance of each model would be the last part of our analysis. The two perfomance indicator would be out of sample $R^2$ and out of sample Root Mean Square Error (RMSE). R-squared is a statistical measure of dependent variable variance proportion explained by the independent variables. It explains to what extent a predictive model is able to explain the objective variable (i.e., risk premium). Practically, $R^2$ is calculated as 1 minus the ratio between unexplained variation and total variation. The unxeplained variation corresponds to the sum of the residual. So, the formula would be the following:

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \widehat{r_{i,t+1}})^2}{\sum_{(i,t)\in\mathcal{T}_3} r^2_{i,t+1}}, \tag{19}$$

This is a standard R-squared. The only key notation that has to be made regards $\mathcal{T}_3$ . It stands for test set and it has the role of highlighting the fact that this statistic is computed out of sample, given that the model are built with training and validation set. Clearly, the higher $R^2$ the better is the model. Root Mean Square Error, indeed, is a simple sum of errors. They are squared in order to avoid that a negative error can reduce the index. The formula is the following:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y} - y_i)^2}{n}}. \tag{20}$$

In addition to measures that evaluate each model performance, a pairwise comparison method would be useful. In that sense Diebold Mariano test perfectly fits the problem. This test (Diebold-Mariano 1995) was introduced to compare models forecast performance. Several years later, Diebold himself, provided a meaningful description of his model: 'The need for formal tests for comparing predictive accuracy is surely

obvious. We've all seen hundreds of predictive horse races, with one or the other declared the "winner" (usually the new horse in the stable), but with no consideration given to the statistical significance of the victory. Such predictive comparisons are incomplete and hence unsatisfying. That is, in any particular realization, one or the other horse must emerge victorious, but one wants to know whether the victory is statistically significant. That is, one wants to know whether a victory "in sample" was merely good luck, or truly indicative of a difference in population' (Diebold 2015). Practically speaking, it tests the equal accuracy hypothesis:

$$H_0: E[\mathcal{L}(e_1)] = E[\mathcal{L}(e_2)], \tag{21}$$

where $\mathcal{L}$ is the loss function[1], $e_1$is the first model error and $e_2$is the second model error. The alternative hypothesis, indeed, is:

$$H_1: E[\mathcal{L}(e_1)] \neq E[\mathcal{L}(e_2)]. \tag{22}$$

Given that the test is built over loss differential (i.e., difference between the two loss functions), called $d$ ,the null hypothesis can be rewritten as:

$$H_0: E[d] = 0, \tag{23}$$

It follows that Diebold Mariano statistic is:

$$DM = \frac{d}{\sqrt{\frac{2\pi \widehat{f_d}(0)}{T}}}, \tag{24}$$

where $2\pi \widehat{f_d}(0)$ is an estimator of the asymptotic variance $\sqrt{Td}$. Diebold Mariano statistic converges to a normal distribution, so the threshold to reject null hypothesis, with 95% of confidence, is $|DM|$> 1.96. In the other case, $|DM|\leq$ 1.96, $H_0$ cannot be rejected.

[1] Most common loss functions are two: squared-error loss function and absolute-error loss function. Both functions are symmetric to the origin point. Squared-error loss function penalizes more larger errors.

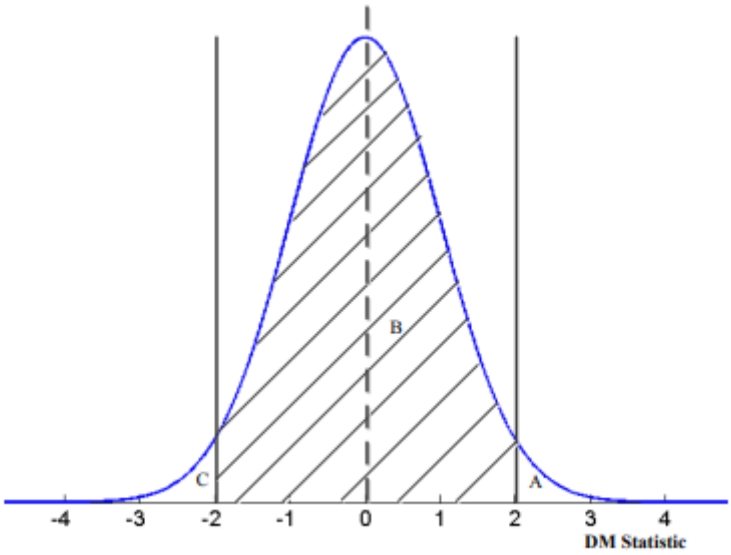



Figure 3

**Normal Distribution**

The figure above describes the normal distribution to which the Diebold Mariano statistic converge. The distribution is divided into 3 different areas: A,B,C. A and C areas correspond to $|DM|>$ 1.96, so rejection of null hypothesis (two models have different accuracy). The other case is represented by B area[2].

## 2.8 Variables Importance

Our analysis goal is not only finding the best model for excess return prediction, but also find which covariates contribute the most to the model. This concept is called variable importance. There are several measures of variable importance, but the one that is going to be evaluated is associated with the 'varImp' function in R. This function uses two different methods for linear regression (i.e., Simple OLS, LASSO, Elastic Net, PCR) and ensemble methods (i.e., Gradient Boosted Trees and Bagging Trees). Variable importance in linear regression is given by the absolute value of parameter specific 't-statistic'. It is the result of the so called 't-test'. It is a significance test performed to check the relationship between each predictor and the response variable. Firstly, the aforementioned 't-statistic' is computed as follows:

$$t = \frac{\beta}{SE_\beta} \tag{25}$$

In this case $\beta$ is the predictor coefficient and $SE$ the associated standard error. Then, if the t-statistic is smaller than some threshold, the null hypothesis (i.e., the coefficient slope is equal to zero) is rejected. So, variable importance in regression models is determined in this way. For what concerns ensemble method,

[2] Image source Chen, Hao, Qiulan Wan, and Yurong Wang. 2014. "Refined Diebold-Mariano Test Methods for the Evaluation of Wind Power Forecasting Models" *Energies* 7, no. 7: 4185-4198. https://doi.org/10.3390/en7074185

the situation is slightly different. For each tree, accuracy is recorded in terms of mean square errors, on out of sample date (i.e., out-of-bag), and then the same process is repeated after deleting one variable from the tree. After that the differences between the MSE obtained from each tree are averaged and normalized by the standard error (as the coefficient in t-statistic). The higher is the average normalized difference, the higher is the variable importance.

# 3 Empirical Analysis

As aforementioned, the empirical analysis consists in applying all the descripted machine learning techniques to a dataset containing US equity listed firms. The paper outcome would be twofold. At first, looking at the predictive performance of each model, it is evaluated the model out of sample $R^2$, the Root Mean Square Error (RMSE) and the Diebold-Mariano pairwise test statistic. Afterwards, variable importance will be taken into consideration. This is because we do not only want to seek for the best model, but we also want to understand it. Variables importance would give us the measure of economic significance of each predictor.

## 3.1 Data Description

The dataset contains American listed firms, from 2006 to 2020, with an average of almost 6,000 observations per month, and 70,000 per year. The total observations are 1,000,000. No restriction on stock price was imposed, in order to obtain a heterogeneous sample with both high-priced and low-priced stocks. In fact, the minimum price is 1.28$ while the maximum value is 83.02$.

The dataset contains also large predictors set. Chosen predictors are all stock-level characteristics. In order to choose stock-level features, the reference point is the cross-section of stock return literature. A group of variables can be identified as the one concerning momentum. Momentum was firstly theorized by Jegadeesh and Titman (1993). Their paper was the answer to some studies (i.e., De Bondt and Thaler) which suggested that poorly performing stocks in a previous period of 3-5 years achieve higher return of stocks that performed well in the same period. Jegadeesh and Titman's claim, indeed, is that stocks with an above average return in the previous year, tend to outperform stock whose performances were poorer in the same period. So, they defined the so called 'price momentum', which consists in averaging the weekly stock price over a year and dividing it by the last ten weeks average. Obviously, other momentum measures can be obtained by changing the time horizon. In fact, in the dataset are present four momentum variables: 12-month momentum, 6-month momentum, 1-month momentum and 36-month momentum. Also changes in 6-month momentum are considered relevant in explaining stock return. Furthermore, Moskowitz and Grinblatt (1999) documented that investing strategies driven by industry momentum performed better in terms of profit than momentum driven strategies. Fama and McBeth's beta and beta squared are part of the dataset too. Other variables can be divided into two subgroups: features related to the stock itself, and

features related to the firm. Stock specific predictors are for example the number of share outstanding, maximum daily return dividend to price or bid-ask spread. On the other hand, variables such as leverage, operating profitability, capital expenditures, R&D or depreciation reflect the firm current state and potential. The last relevant variables group is the liquidity one. Chordia, Subrahmanyam and Anshuman (2001) find out a surprisingly negative 'strong cross-sectional relationship between stock returns and the variability of dollar trading volume and share turnover'. Among the predictors, share turnover and dollar trading are proxies for liquidity, and their volatility (i.e., standard deviation) are used as variables too. The last feature of this group is illiquidity. According to Amihud (2002), 'illiquidity measure here is the average across stocks of the daily ratio of absolute stock return to dollar volume', and he suggests that 'positively affects ex ante stock excess return, suggesting that expected stock excess return partly represents an illiquidity premium'.

Each of the mentioned model, as it was said before, are designed to estimate the equation $E_r(r_{i,t+1}) = g * (z_{i,t})$ (3). In some model, *g*(.)* is not forced to be linear and takes into consideration the nonlinear interactions. It expands the feature set with some transformations of $z_{i,t}$.

The one million observations are divided into three datasets. The largest one is the training set, which count for six tenth of the total data (i.e., 600,000 observations). Validation set and test set are equally sized (i.e., 200,000 rows each). The splitting criteria are entirely random.

## 3.2 Cross Section of individual stock

### 3.2.1 $R^2$ evaluation

Table 1 shows machine learning techniques performance with respect to the out of sample $R^2$. Final models are five: Ordinary Least Squares with all the covariates in the dataset, Least Absolute Shrinkage and Selection Operator (LASSO), Elastic Net, Principal Component Regression (PCR) and Gradient Boosted Regression Trees (GBRT). Details about models-specific hyperparameter are provided in the appendix.

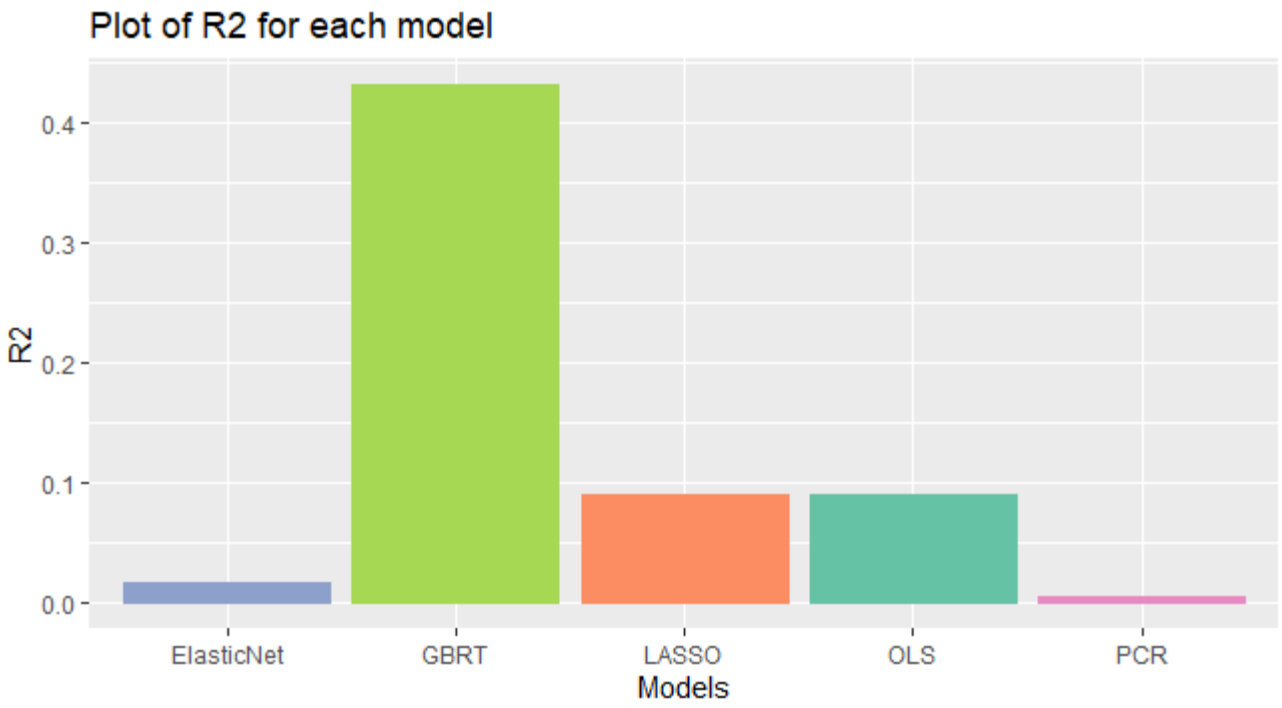
The first row of Table 1 reports $R^2_{OOS}$ for the entire dataset. Ordinary Least Square model has $R^2_{OOS}$ of about 0.09. This is quite surprising because using all predictors belonging to the predictor set is not so rare to obtain a totally meaningless model. In fact, restricting the OLS model, to a three-factor model containing just size, value and momentum, performances fall drastically both in terms of $R^2_{OOS}$ and RMSE. Shrinking coefficients with LASSO and Elastic Net has very different results. LASSO performs slightly better than the simple linear regression model (0.09025 versus 0.9019), but with a very singular coefficient selection as we will see in the variable importance section. Elastic Net, once validated, tends more to a

Ridge model than to a LASSO one, given that the resulting α is 0.2[3]. The two λs, indeed, are very different. LASSO lambda is very small ($1x10^{-6}$) while Elastic Net one is 0.01. Considering all these differences, Elastic Net model's performances are poorer than LASSO ones, with an out of sample $R^2$ of about 0.016.

Table 1

**Monthly out-of-sample stock-level prediction performance ($R^2_{OOS}$)**

| | Elastic Net | GBRT | LASSO | OLS | PCR |
|---|---|---|---|---|---|
| R2 | 0.0162506 | 0.4308502 | 0.0902529 | 0.0901991 | 0.0056860 |




PCR model performs even worse. The $R^2_{OOS}$ is less than the half of Elastic Net performance. Dimension reduction algorithms perform better in a redundant environment, with highly correlated covariates. This result suggests that predictors are quite uncorrelated. In order to confirm that, variable pairwise correlation is further investigated. In fact, Figure 3 represents the dataset correlation plot. Exception made for those variables correlated by construction (for example beta and beta squared); the graph shows that most of the correlation coefficients are close to zero. This can explain, for example, satisfying OLS performance. This is because no multicollinearity (independence of predictors) is one of the assumptions of multiple linear regression.

[3] LASSO α is always equal to 1, while Ridge's α is always equal to 0.

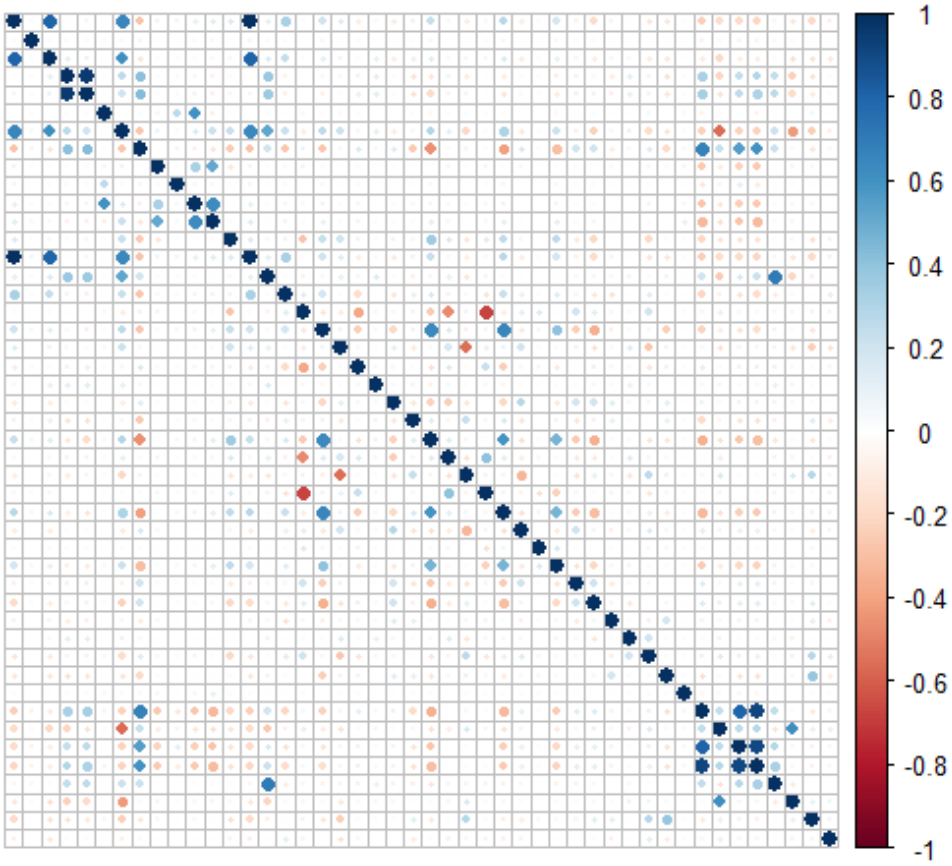



Figure 3

**Correlation Plot**

The figure above shows the pairwise correlation between predictors. A colour close to blue indicates a correlation close to 1, while a colour close to red indicates a correlation close to -1. If there is neither a positive nor a negative correlation, the colour would be close to white. Obviously, the correlation matrix is symmetric.

In conclusion, the last model to be evaluated is Gradient Boosted Regression Trees. It is clear that GBRT outperforms all the other techniques. This algorithm tends to outperform random forests (see Gu et al. 2020), and it is far better than the other models taken into consideration. Combining a large number of weak learners (i.e., ntree=1000) allows the model to explain most of variance, avoiding the overfitting. The conclusions highlighted by Table 1 are that:

- GBRT outperforms significantly all the other models
- LASSO and OLS are competitive
- Elastic Net and PCR performance are poor

These results are quite different with respect to the ones obtained by Gu et al. (2020). The main differences concern OLS and PCR results. In their paper, the OLS out-of-sample R-squared is negative. This is due to the fact that their set of predictors was very large. The predictor set used in this paper is the result of Gu et al. findings in terms of variable importance and, since that, the environment is more ready to feed a simple linear regression. The same reasoning can be made for PCR. A less noisy predictor set makes the benefit of PCR way less impactful. LASSO and Elastic Net performances are more aligned

with Gu et al. (2020). LASSO is more competitive in both papers, while Elastic Net seems to have poorer performance. In conclusion, GBRT outperforms other models by far in both papers.
In order to validate $R^2_{OOS}$ results, it is necessary to investigate other performance measures.

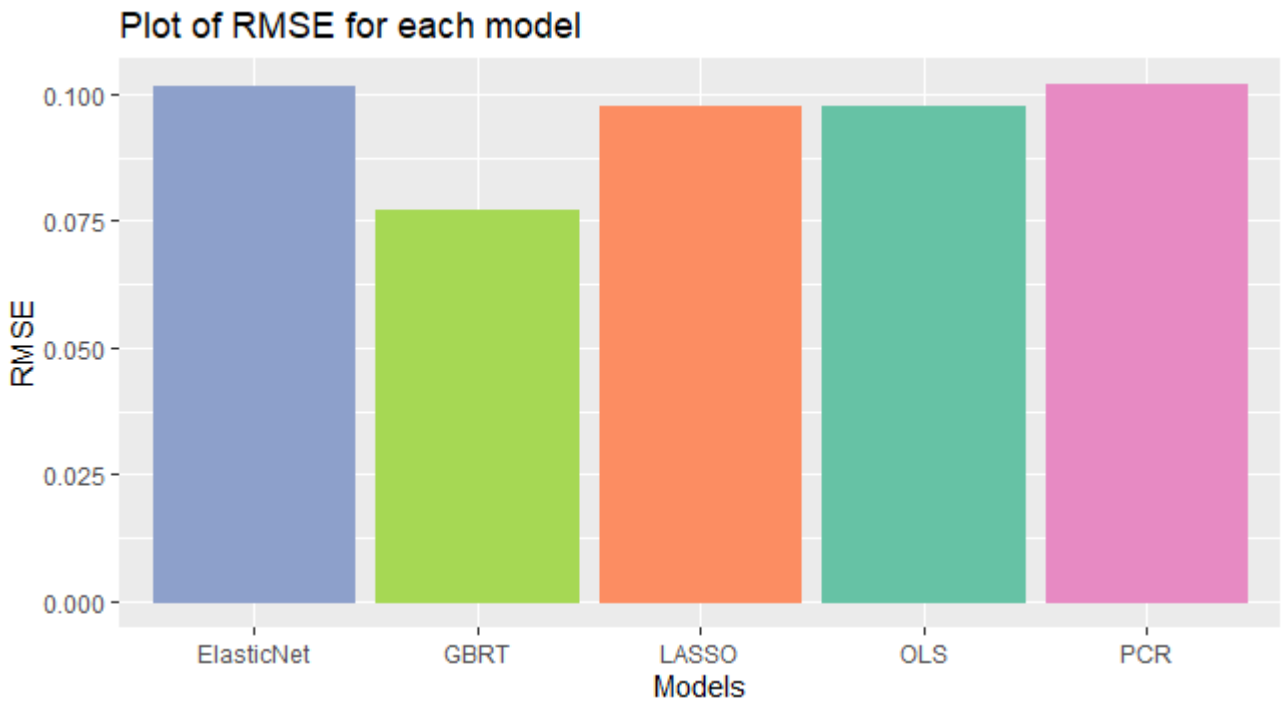
## 3.2.2 Root Mean Square Error Evaluation

Table 2 shows machine learning techniques performance with respect to the out of sample Root Mean Square Error. The machine learning approaches are the same of the previous section.

Table 2

**Monthly out-of-sample stock-level prediction performance ($RMSE_{OOS}$)**

| | ElasticNet | GBRT | LASSO | OLS | PCR |
|---|---|---|---|---|---|
| RMSE | 0.1013274 | 0.0770723 | 0.0974418 | 0.0974447 | 0.1018701 |




RMSE trend reflects the same behaviour of out of sample R squared. Simple linear regression and LASSO errors are almost the same, such as Elastic Net and PCR. For the last pair of models, the difference in $R^2_{OOS}$ is bigger and more significative than the difference in Root Mean Square Error. As said before, GBRT performances are drastically better.

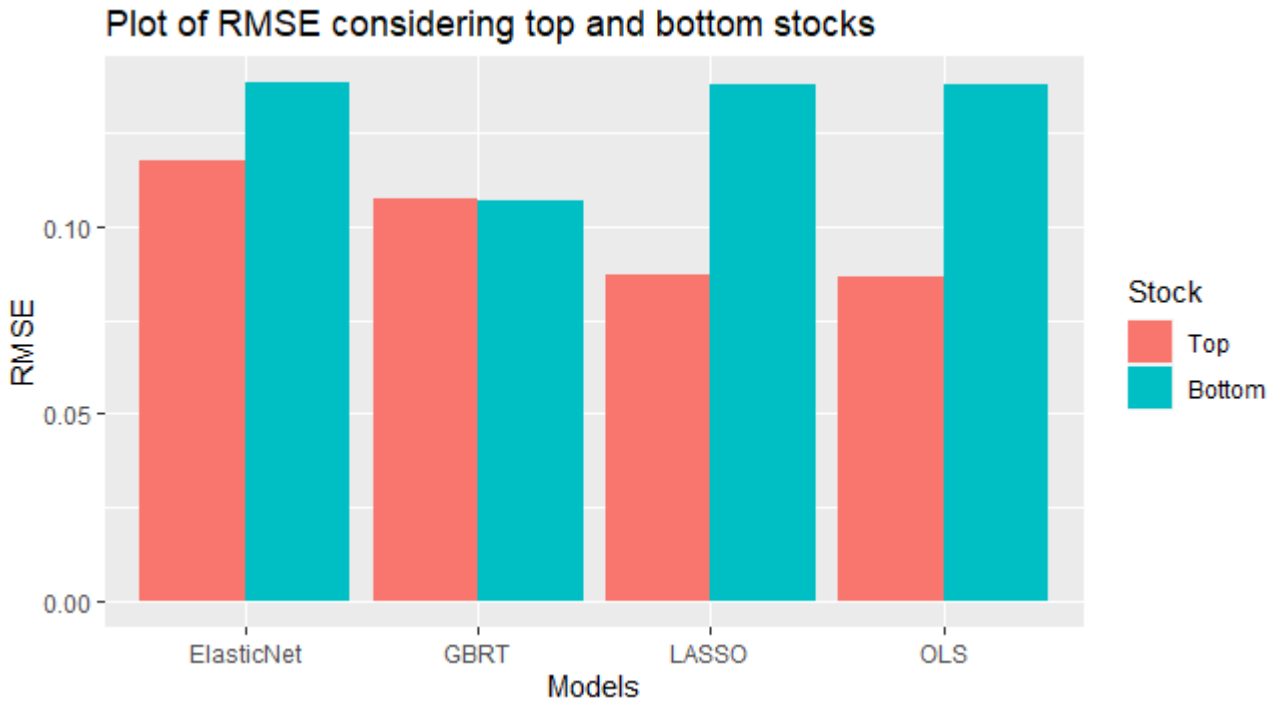
Another perspective to address the problem can be considering models performance at the extreme of the dataset. Practically speaking, it would be useful to evaluate models' performance on the top

hundred more expensive stocks and on the hundred least expensive. In fact, next table and plot show accuracy for large stocks (the top-100 stocks by price) and small stocks (the bottom-100 stocks by price). This is based on model trained with the full dataset (using all stocks), but it is tested among the two subsamples. Table 3 shows models RMSE on 'top' and 'bottom' datasets[4].

Table 3

**Monthly out-of-sample stock-level prediction performance on top 100 priced stocks and bottom 100 priced stocks ($RMSE_{OOS}$)**

| | ElasticNet | GBRT | LASSO | OLS |
|---|---|---|---|---|
| Top | 0.1174750 | 0.1073919 | 0.0869042 | 0.0867748 |
| Bottom | 0.1384307 | 0.1068528 | 0.1379131 | 0.1379151 |




As we can see, models are better at forecasting top shares with respect to bottom ones. Particularly, Ordinary Least Square and LASSO perform better on the top subsample than on the whole test set. Elastic Net performs worse than on the complete dataset, and the same results hold for GBRT. Furthermore, GBRT is the only model which break out greater accuracy on the bottom subset rather than the top one. In this case, the difference is not so significant. Given that Gradient Boosted Regression Trees is the best method, it will be further investigated with the following tables. Table 4 and 5 contain all values of RMSE and $R^2_{OOS}$ for all the analysed samples (i.e., all, top, bottom).

[4] PCR results are not included in the table since the values are not relevant.

Table 4

**GBRT performance on all subsamples (RMSE)**

| | GBRT |
|---|---|
| Bottom | 0.1068528 |
| All | 0.07707226 |
| Top | 0.1073919 |

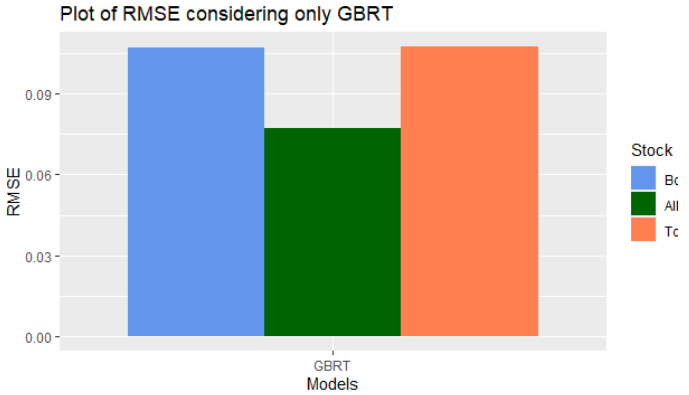
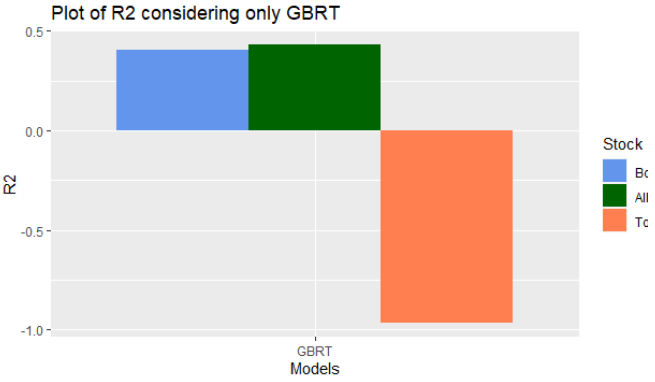



Table 5

**GBRT performance on all subsamples ($R^2_{OOS}$)**

| | GBRT |
|---|---|
| Bottom | 0.4040442 |
| All | 0.4308502 |
| Top | -0.9659346 |




While RMSE table and plot (Table 4) results are not surprising, the out of sample R squared plot (Table 5) is more interesting. First of all, it is meaningful that $R^2_{OOS}$ on the bottom dataset is very close to the original model value. Secondly, the anomalous value of top $R^2_{OOS}$ needs additional explanations. In fact, according to Table 3, models tend to have greater accuracy on top subsample. Values of out of sample R squared on the top dataset, indeed, looking at GBRT but also at the other models, are very low. Low RMSE and low $R^2_{OOS}$ suggested that data are not skewed. Consequently, models' predictive ability is not to be addressed to the predictors ability to explain observations' variance.

## 3.2.3 Diebold-Mariano Test

Last comparison method is the Diebold-Mariano test. Previous accuracy measures addressed quantitative techniques' performances, while, indeed, Diebold-Mariano test addresses statistical significance of those results. The null hypothesis chosen is the 'two-sided' one, that means that we are testing equal accuracy for the two models. Table 6 reports the results of this pairwise comparisons. The interpretation is that a positive value means that the column model dominates the row model. As we can see, among those models which have close performance on quantitative methods, LASSO outperforms simple linear model, while Elastic Net dominates PCR. As we expected Gradient Boost Regression Trees outperforms all the other models.

Table 6

**Comparison of out-of-sample prediction using Diebold-Mariano test statistic**

| | Lasso | Elastic Net | PCR | GBRT |
|---|---|---|---|---|
| Linear model | **0.837187** | -28.220382 | -28.447007 | 99.379606 |
| Lasso | | -28.83532 | -29.06501 | 100.87528 |
| Elastic Net | | | -5.035476 | 225.395290 |
| PCR | | | | 225.3765 |

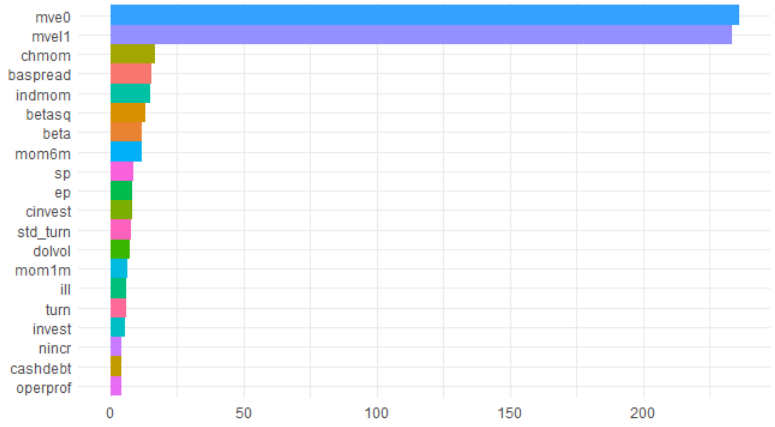
This table shows Diebold-Mariano test statistics comparing the out-of-sample prediction performance among five models. Positive values indicate the column model outperforms the row model. Bold font stands for a difference significant at 5% level or better for individual tests.

## 3.3 Variable Importance

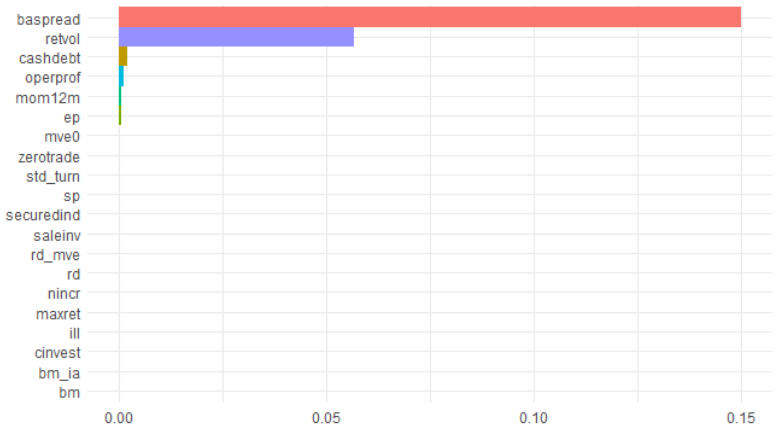
This section consists in analysing individual predictor's importance as described in section 2.8. Figure 4 reports variable importance for the first twenty features of each model.[5]

Figure 4

**Variable importance for the top-20 most important variables in each model**

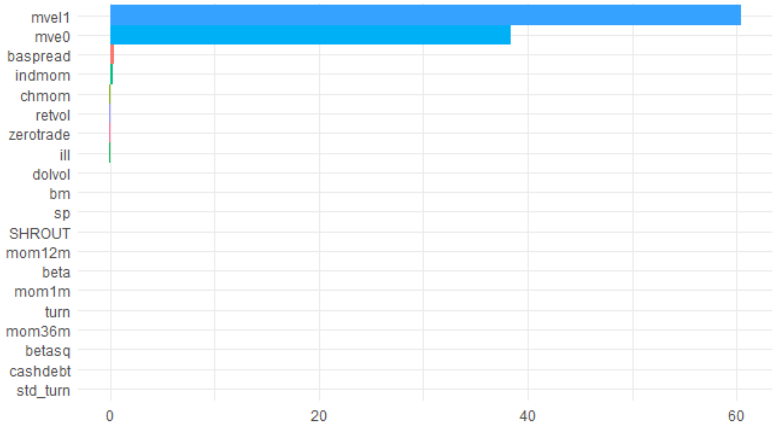



Linear model variable importance

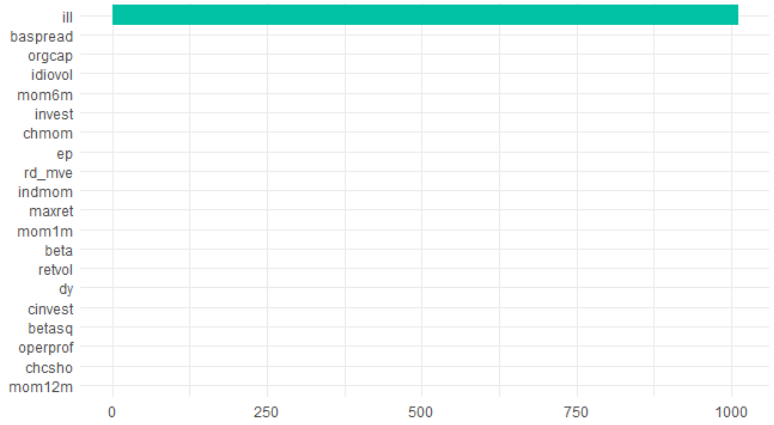



Elastic Net variable importance

[5] Since PCR predictions are based on Principal Components, variable importance is not relevant considering the scope of the analysis




Gbrt variable importance




Lasso variable importance

The first thing to notice concerns LASSO model. It is very singular that being so skewed to one variable (i.e., illiquidity) leads to better performances with respect to OLS, according to both quantitative and statistical significative methods. For what concerns the other models, they are not so 'democratic'. Linear regression and GBRT are skewed to size and factors (i.e., mvel1 and mve0). On the other hand, Elastic Net is skewed to bid-ask spread variable. With regards to the 'secondary variables', if we can call them like that, the ones to be mentioned are illiquidity, momentum variables and bid-ask spread. Illiquidity is significant in LASSO, as said before, OLS and GBRT. Momentum variables (i.e., change in momentum, 12-month momentum, 6-month momentum, industry momentum) are important in OLS, Elastic Net and GBRT. Bid-ask spread, in addition to the importance in Elastic Net, is relevant in OLS and GBRT too. A mention must be made also for return volatility: it is the second most important variable in Elastic Net and it is relevant in GBRT too.

Comparing variable importance outcomes with the ones obtained by Gu et al. (2020) highlights some common aspects and some differences. First of all, models' variable importance is skewed to two or three features in both papers. The variables that result to be the most relevant in this paper are mvel1 (Log market equity) and mve0 (and market equity at the beginning of the period), while momentum variables are the most important in Gu et al.'s paper. Another great difference concerns bid-ask spread, which

seems not to be so relevant in Gu et al.'s models. In conclusion, the two overall sets of important variables are very similar. In fact, both papers select momentum variables, liquidity variables and size variables as the most relevant ones.

# Conclusions

This paper studies the forecasting application of some of the most common Machine Learning techniques on the US market, trying to give a contribution to the 'Empirical Asset Pricing' debate. The thesis scope, as aforementioned, is to find out the best model, amongst the ones investigated and implemented, and to analyse the variable importance of each model, in order to highlight a potentially meaningful set of predictors. Models' performances and accuracy are evaluated in terms of $R^2_{OOS}$ and RMSE, while variable importance is assessed in terms of statistical relevance and marginal accuracy (section 2.8).

For what concerns models performance, the model which achieves the best results is Gradient Boosted Regression Trees. In fact, both in terms of $R^2_{OOS}$ and RMSE, it outscored all the other models. The statistical relevance of this result is certificated by the Diebold Mariano test. The performance is close to the one obtained by Gu et al (2020), with an of $R^2_{OOS}$ of 0.34 (Gu's et al. result) versus an of $R^2_{OOS}$ of 0.43. Also considering the other techniques, Ordinary Least Square performs better than expected, LASSO outperforms Elastic Net as penalized approach, and PCR performs poorly, given that the environment is not so noisy and redundant.

Variable importance results are more interesting and surprising. LASSO, that outscored simple linear regression with statistical relevance (Table 6), basically forecasts considering only illiquidity. The other models reflect Gu et al (2020) findings. Log market equity (mvel1) and market equity at the beginning of the period (mve0) dominates variable importance of both OLS and GBRT. The set of overall influential predictors is composed by momentum variables, illiquidity, return volatility and bid-ask spread. This last variable represents the real difference with respect to Gu et al (2020). In fact, looking at the model implemented by both paper, bid-ask spread is not between the twenty most relevant variables for any of them, exception made for GBRT. It is, indeed, relevant for Neural Networks implementation, that is the best performing one in Gu et al (2020) paper.

This paper, such as the other from this topic literature, shows the strong potential connection between Machine Learning and Empirical Asset Pricing. This link and its development could be the starting point for many interesting research and studies.

# References


Akyildirim, Erdinc & Nguyen, Khuong & Sensoy, Ahmet & Šikić, Mario. (2021). Forecasting high-frequency excess stock returns via data analytics and machine learning. European Financial Management. 10.1111/eufm.12345.

Amihud, Yakov, Illiquidity and Stock Returns: Cross-Section and Time-Series Effects (2000). NYU Working Paper No. FIN-00-041, Available at SSRN: https://ssrn.com/abstract=1295244

Campbell, John Y. "Asset Pricing at the Millennium." *The Journal of Finance* 55, no. 4 (2000): 1515–67. http://www.jstor.org/stable/222372.

Chordia, Tarun and Subrahmanyam, Avanidhar and Anshuman, V. Ravi, Trading Activity and Expected Stock Returns (Undated). Available at SSRN: https://ssrn.com/abstract=204488 or http://dx.doi.org/10.2139/ssrn.204488

Chen, Luyang and Pelger, Markus and Zhu, Jason, Deep Learning in Asset Pricing (April 4, 2019). Available at SSRN: https://ssrn.com/abstract=3350138 or http://dx.doi.org/10.2139/ssrn.3350138

Chung, Y. Peter, et al. "Asset Pricing When Returns Are Nonnormal: Fama-French Factors versus Higher-Order Systematic Comoments." The Journal of Business, vol. 79, no. 2, 2006, pp. 923–40. JSTOR, https://doi.org/10.1086/499143. Accessed 27 May 2022.

Cooper, Michael J. and Gulen, Huseyin and Schill, Michael J., Asset Growth and the Cross-Section of Stock Returns (July 10, 2007). AFA 2007 Chicago Meetings Paper, Available at SSRN: https://ssrn.com/abstract=760967 or http://dx.doi.org/10.2139/ssrn.760967

Dong, Xi and Li, Yan and Rapach, David and Rapach, David and Zhou, Guofu, Anomalies and the Expected Market Return (November 15, 2021). Journal of Finance, Forthcoming, Available at SSRN: https://ssrn.com/abstract=3562774 or http://dx.doi.org/10.2139/ssrn.3562774

Drobetz, Wolfgang and Haller, Rebekka and Jasperneite, Christian and Otto, Tizian, Predictability and the Cross-Section of Expected Returns: Evidence from the European Stock Market (August 10, 2019). Available at SSRN: https://ssrn.com/abstract=3436051 or http://dx.doi.org/10.2139/ssrn.3436051

Drobetz, W., Otto, T. Empirical asset pricing via machine learning: evidence from the European stock market. *J Asset Manag* **22,** 507–538 (2021). https://doi.org/10.1057/s41260-021-00237-x

Fama, Eugene F. and French, Kenneth R., Multifactor Explanations of Asset Pricing Anomalies. J. OF FINANCE, Vol. 51 No. 1, March 1996, Available at SSRN: https://ssrn.com/abstract=7365

Fama, Eugene F. French, Kenneth R. The Capital Asset Pricing Model: Theory and Evidence Journal of Economic Perspectives 18 3 25-46 2004 10.1257/0895330042162430 https://www.aeaweb.org/articles?id=10.1257/0895330042162430

Francis X Diebold & Robert S Mariano (2002) Comparing Predictive Accuracy, Journal of Business & Economic Statistics, 20:1, 134-144, DOI: 10.1198/073500102753410444

Francis X. Diebold (2015) Comparing Predictive Accuracy, Twenty Years Later: A Personal Perspective on the Use and Abuse of Diebold–Mariano Tests, Journal of Business & Economic Statistics, 33:1, 1, DOI: 10.1080/07350015.2014.983236

Frank, Murray Z. and Yang, Keer, Predicting Firm Profits: From Fama-MacBeth to Gradient Boosting (September 7, 2021). Available at SSRN: https://ssrn.com/abstract=3919194 or http://dx.doi.org/10.2139/ssrn.3919194

Giglio, Stefano and Xiu, Dacheng, Asset Pricing with Omitted Factors (September 14, 2019). Chicago Booth Research Paper No. 16-21, Available at SSRN: https://ssrn.com/abstract=2865922 or http://dx.doi.org/10.2139/ssrn.2865922

Jegadeesh, Narasimhan, and Sheridan Titman. "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency." *The Journal of Finance* 48, no. 1 (1993): 65–91. https://doi.org/10.2307/2328882.

Lewellen, Jonathan. (2011). Institutional Investors and the Limits of Arbitrage. Journal of Financial Economics. 102. 62-80. 10.1016/j.jfineco.2011.05.012.

Lewellen, Jonathan. (2011). The Cross-section of Expected Stock Returns. Critical Finance Review. 4. 10.1561/104.00000024.

Linnainmaa, Juhani T. and Roberts, Michael R., The History of the Cross Section of Stock Returns (November 24, 2016). Marshall School of Business Working Paper No. 17-17, Available at SSRN: https://ssrn.com/abstract=2897719 or http://dx.doi.org/10.2139/ssrn.2897719

Moskowitz, Tobias J., and Mark Grinblatt. "Do Industries Explain Momentum?" *The Journal of Finance* 54, no. 4 (1999): 1249–90. http://www.jstor.org/stable/798005.

Robert A. Haugen, Nardin L. Baker, Commonality in the determinants of expected stock returns, Journal of Financial Economics, Volume 41, Issue 3,1996, Pages 401-439, ISSN 0304-405X, https://doi.org/10.1016/0304-405X(95)00868-F.

Shihao Gu, Bryan Kelly, Dacheng Xiu, Empirical Asset Pricing via Machine Learning, *The Review of Financial Studies*, Volume 33, Issue 5, May 2020, Pages 2223–2273, https://doi.org/10.1093/rfs/hhaa009

# APPENDIX A

Table A.1 Summary Statistics

| Acronym | Variable | Min | Mean | Max | Std |
|---|---|---|---|---|---|
| mvel1 | Size | 15999.1907 | 2118123.1436 | 15851782.1603 | 4008342.2572 |
| SHROUT | Share Outstanding | 2762 | 72000.2726 | 392976.25 | 99299.1011 |
| beta | Beta | 0.1155 | 1.0427 | 2.1457 | 0.5362 |
| betasq | Beta Squared | 0.0192 | 1.4063 | 4.6242 | 1.2393 |
| chmom | Change in 6-month momentum | -0.6839 | -0.008 | 0.7161 | 0.339 |
| dolvol | Dollar trading volume | 8.3118 | 12.9522 | 17.2168 | 2.5579 |
| idiovol | Idiosyncratic return volatility | 0.0141 | 0.051 | 0.1181 | 0.0283 |
| indmom | Industry momentum | -0.3259 | 0.0669 | 0.4726 | 0.2022 |
| mom1m | 1-month momentum | -0.2069 | 0.001 | 0.2092 | 0.1008 |
| mom6m | 6-month momentum | -0.4424 | 0.0176 | 0.5056 | 0.2316 |
| mom12m | 12-month momentum | -0.6005 | 0.0464 | 0.8014 | 0.3459 |
| mom36m | 36-month momentum | -0.6802 | 0.1755 | 1.3504 | 0.4943 |
| mve0 | Market value at the start of the period | 16054.5 | 2136057.0726 | 15900930.3254 | 4018953.1698 |
| turn | Share turnover | 0.1279 | 1.5739 | 5.5226 | 1.4357 |
| age | # Years since first coverage | 2 | 16.6828 | 44 | 11.3683 |
| agr | Asset growth | -0.5879 | -0.0958 | 0.2173 | 0.1799 |
| cashdebt | Cash flow to debt | -1.1144 | -0.0159 | 0.6927 | 0.385 |
| cashpr | Cash productivity | -33.4053 | 0.572 | 33.3886 | 13.7839 |
| chcsho | Change in shares outstanding | -0.0575 | 0.0537 | 0.378 | 0.0994 |

Table A.1 Summary Statistics

| Acronym | Variable | Min | Mean | Max | Std |
|---|---|---|---|---|---|
| convind | Convertible debt indicator | 0 | 0.1023 | 1 | 0.2719 |
| depr | Depreciation/PP&E | 0.048 | 0.3217 | 1.0208 | 0.2413 |
| dy | Dividend to price | 0 | 0.016 | 0.0725 | 0.0198 |
| ep | Earnings to price | -0.4181 | -0.0237 | 0.1223 | 0.1277 |
| invest | Capital expenditures and inventory | -0.0558 | 0.0411 | 0.2061 | 0.0589 |
| lev | Leverage | 0.0498 | 1.7818 | 8.506 | 2.192 |
| lgr | Growth in long term debt | -0.2575 | 0.1466 | 0.9118 | 0.2652 |
| operprof | Opertating profitability | -0.4547 | 0.1342 | 0.5329 | 0.2114 |
| orgcap | Organizational capital | 4e-04 | 0.0056 | 0.0145 | 0.0034 |
| pchsale_pchrect | % change in sale - % change in A/R | -0.5845 | -0.0285 | 0.4516 | 0.2235 |
| ps | Financial statement scores | 2 | 4.7289 | 7 | 1.3353 |
| rd | R&D increase | 0 | 0.1471 | 1 | 0.3102 |
| rd_mve | R&D to market capitalization | 0 | 0.0586 | 0.1463 | 0.0342 |
| saleinv | Sales to inventory | 2.9484 | 31.5801 | 97.2655 | 23.5805 |
| securedind | Secure debt indicator | 0 | 0.483 | 1 | 0.4483 |
| sp | Sales to price | 0.0418 | 1.0644 | 3.992 | 0.988 |
| cinvest | Corporate investment | -0.2776 | 0.1174 | 0.4258 | 0.2214 |
| nincr | Number of earnings increases | 0 | 0.8454 | 3 | 0.8487 |
| baspread | Bid-ask spread | 0.0079 | 0.0365 | 0.0957 | 0.0239 |
| ill | Illiquidity | 0 | 0 | 0 | 0 |
| maxret | Maximum daily return | 0.0106 | 0.0557 | 0.176 | 0.0441 |

Table A.1 Summary Statistics

| Acronym | Variable | Min | Mean | Max | Std |
|---|---|---|---|---|---|
| retvol | Return volatility | 0.0056 | 0.0257 | 0.0705 | 0.0174 |
| std_turn | Volatility of liquidity | 0.4449 | 4.888 | 22.9716 | 5.765 |
| zerotrade | Zero trading days | 0 | 0.1217 | 1.9091 | 0.4451 |
| bm | Book to market | 0.0755 | 1.0006 | 2.3573 | 0.6943 |
| bm_ia | Industry-adjusted book to market | -3.3817 | -0.3992 | 0.9636 | 0.9043 |

# APPENDIX B

| | OLS | LASSO | Elastic Net | PCR | GBRT |
|---|---|---|---|---|---|
| Hyperparameters | | $\lambda=1x10^{-6}$ | $\lambda=1x10^{-2}$<br>$\alpha=0.2$ | # Components<br>n=1 | Depth=6<br># Trees=1000<br>Shrinkage=0.01 |