Which predictors are relevant for return forecasting ?

Arnaud Felber[1], Mina Attia[1], Rami Atassi[2], and Paulo Ribeiro[2]

[1] Master Student in Financial Engineering, EPFL**, Switzerland
arnaud.felber@epfl.ch [302283]
mina.attia@epfl.ch [309974]
[2] Master Student in Data Science, EPFL, Switzerland
rami.atassi@epfl.ch [296346]
paulo.ribeirodecarvalho@epfl.ch [314985]

Group 17

Abstract. This report investigates the application of machine learning techniques to predict stock returns. The study is divided into three phases: data extraction and visualization, data processing, and predictor selection. We trained and evaluated several models, including Ordinary Least Squares (OLS), Ridge, Lasso Regression, Elastic Net, and Random Forest. OLS and Random Forest models performed exceptionally well on the training set, suggesting a tendency to overfit. In contrast, the Ridge, Lasso, and Elastic Net models exhibited better generalization capabilities, with Elastic Net achieving the best results on the test set. These outcomes highlight significant considerations from both machine learning and financial perspectives. From a machine learning standpoint, the findings emphasize the trade-off between model complexity and generalization performance. Financially, identifying key predictors such as Market Capitalization and Price-to-Earnings Ratio offers valuable insights for investment strategies.

Keywords: Finance · Machine Learning · Regression · Data Analysis · Data Imputation.

1 Data Extraction and Visualisation

1.1 Data Extraction

The dataset is sourced from Open Asset Pricing (OAP) and provides predictor values based on different times and assets. Consequently, the primary portion of the dataset could be directly obtained. However, due to restrictions on the distribution of asset prices, the price data were not included in the provided dataset. Instead, these prices data were extracted from the WRDS (Wharton Research Data Services) database using OAP open-source R script.

** École Polythéchnique Fédérale de Lausanne

1.2 Data visualisation

Once the dataset is extracted, it results in a matrix comprising 5 million rows of records, covering 37,774 assets over the period from 1925 to 2023. The dataset includes 209 columns of predictors, each with different meanings, with the log price being the final predictor, as illustrated in the figure 1.

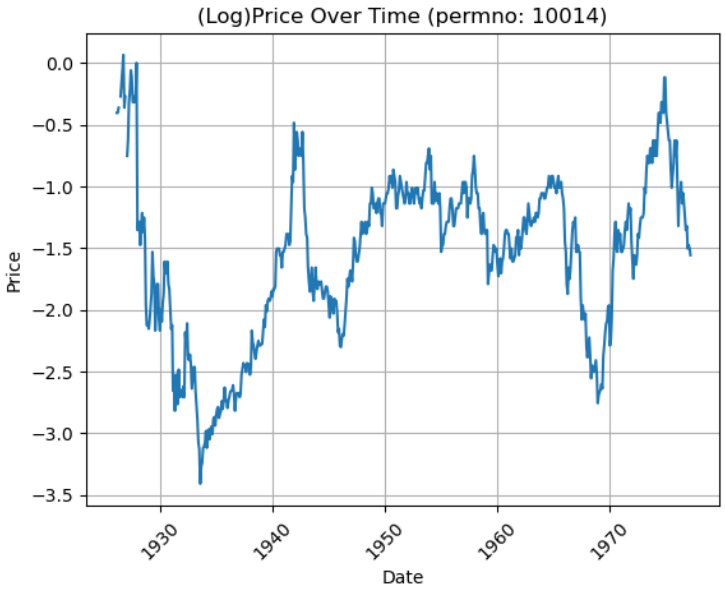

Fig. 1: Logarithmic Price Over Time for Asset PERMNO 10014

The sheer volume of data presents significant computational challenges, making it necessary to use efficient data processing techniques. Additionally, there are many NaN values in this dataset, which complicates the analysis. These missing values must be dealt with to ensure the accuracy and reliability of the statistical and machine learning models used.

2 Dataset reduction

As demonstrated in the previous section, the dataset is large, necessitating a reduction to decrease computational time and minimize the number of NaN values. To achieve this, we employ three primary methods. Firstly, we apply a time range selection to limit the period under consideration. Secondly, we exclude predictors based on their NaN ratio, ensuring only predictors with an acceptable level of missing values are retained. Finally, we impose a NaN value limit on records, removing those that exceed this threshold.

2.1 Time range selection

To determine the appropriate time range for our analysis, we utilized four distinct graphs (see Figure 2). These graphs each plot different metrics over time, with the x-axis representing years and the y-axis representing the values for each specific metric. The four metrics analyzed over time are: the asset number, data quantity, ratio of missing price data, and the overall missing data ratio.

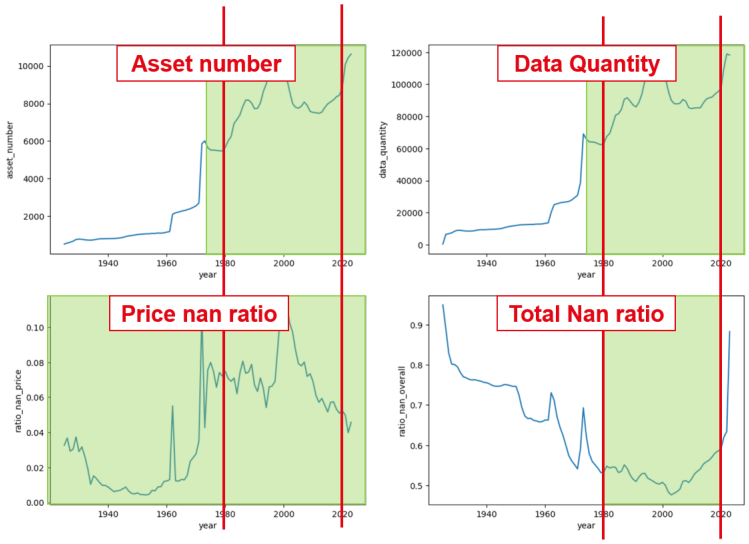

Fig. 2: Evolution of Dataset Metrics Over Time

- **Asset Number Evolution** The top left graph displays the evolution of the number of assets in the dataset from 1940 to 2020. This graph shows a significant increase in assets, particularly between the 1970s and 1980s, reflecting the expansion of the dataset and possibly the broader financial market's growth
- **Data Quantity Evolution** The top right graph illustrates the quantity of data points available in the dataset over the same period. Similar to the asset number, there is a marked increase in data quantity in the same period than previous graph. This increase indicates enhanced data collection efforts and the advent of more sophisticated data recording technologies.
- **Ratio of Missing Price Data Over Time** The bottom left graph presents the ratio of missing price data over the years. Notably, there are peaks in missing data around the 1960s and 1970s, which may suggest periods of market instability or changes in data collection methods. The overall trend of decreasing missing data from the 2000s onward suggests improvements in

data completeness and reliability. However, values of Nan ratio are satisfying on the whole dataset, because they are below 12%.

– **Overall Missing Data Ratio Over Time** The bottom right graph shows the overall ratio of missing data across all metrics in the dataset. This graph reveals a general decline in missing data ratios since the 1980s until 2020s.

According to Figure 2, we have identified a range of years where the data quality is satisfactory, represented in green on the graphs. From these satisfactory periods, we selected a 40-year time range between 1980 and 2020 for our detailed analysis. This period was chosen based on the consistency and completeness of the data, ensuring robust and reliable results for our study.

2.2 Predictors exclusion

Upon reviewing the dataset's attributes, a detailed classification into continuous and discrete types was performed, providing insights essential for the appropriate handling and analysis of the data. The dataset contains 179 continuous attributes, including variables such as rates, and quantitative measurements that vary incrementally across a range and are integral for regression analyses and trend modeling. Additionally, there are 33 discrete attributes, typically representing categorical data, such as binary outcomes, counts, or membership in predefined groups, which are crucial for classification tasks and frequency analysis.

Once the time range was selected, we chose to retain only those discrete and continuous predictors with a missing nan ratio below 0.6 (see Figure 3), as represented by the green area. After this filtering process, 131 predictors remained.

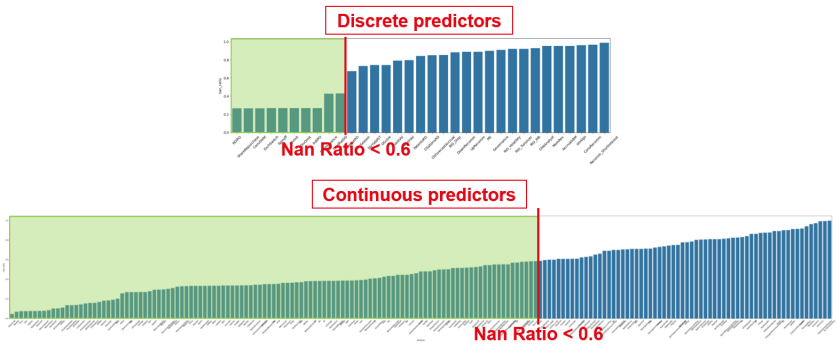

Fig. 3: Predictors exclusion based on Nan Ratio

2.3 Nan value limit

Finally, we have our dataset refined to a 40-year time range between 1980 and 2020, comprising 131 predictors and approximately 3.7 million records. However, considering the high volume of records, we opted to further reduce the dataset from 3.7 million to 1.7 million records by excluding all records with more than 15% missing values. This threshold is supported by the methodology of Stanford university implemented in the data preprocessing part. [1]

3 Data preprocessing

In the last chapter, we found that our financial data misses a large amount of data. To proceed with our analysis, it is crucial to handle these missing values appropriately. Simple, naive methods for filling in missing data often fall short in maintaining the integrity and reliability of the dataset. Therefore, we explore more sophisticated techniques to handle missing data, ensuring that our subsequent analyses are robust and meaningful.

3.1 Naive methods for missing data

Naive methods for handling missing data exhibit several significant drawbacks that undermine their effectiveness and reliability. There is two main naive method for missing data: complete case analysis and unconditional mean imputation. However, both methods have several issues and are not good estimator for financial data.

Complete case analysis involves using only those observations that have no missing values and discarding any data entries with one or more missing variables. While this method is straightforward and easy to implement, it suffers from several critical issues. One significant problem is data loss: by excluding incomplete observations, a substantial amount of data is lost. This reduction in sample size not only diminishes the statistical power of the analysis but also potentially excludes valuable information contained in the incomplete records. For example, in our financial datasets who is composed of several features there is no line where no missing features appears. Thereby method is really badly suited for our purpose. Another issue is the introduction of bias. The remaining complete cases may not be representative of the entire dataset, introducing bias into the analysis. This bias occurs because the missing data may not be randomly distributed but might depend on specific characteristics of the data, meaning the analysis could systematically exclude certain patterns or trends present in the full dataset [2].

Unconditional mean imputation replaces missing values with the mean (or median) of the observed values for that variable. Despite its simplicity, this method introduces several issues. One major issue is that it often leads to inconsistent estimators. This bias arises because the method does not account for the relationships between variables. For instance, if a variable's missing values are

replaced by its mean, the natural variability and correlations with other variables are distorted. This distortion results in estimated coefficients that can be either too large or too small compared to their true values, as seen in the case of operating profitability where the coefficient significantly differed when using unconditional mean imputation compared to conditional mean imputation or complete cases. Another problem is the underestimation of variance. This method tends to underestimate the variance and covariance between variables, leading to artificially small standard errors. Such underestimation can make statistical tests appear more significant than they actually are, thus providing misleading inference about the relationships in the data. Furthermore, this method ignores imputation uncertainty. When missing values are imputed without considering the uncertainty of these imputations, the resulting analysis does not accurately reflect the true variability in the data. This issue is critical because the imputed values are treated as if they were observed values, ignoring the fact that they are merely estimates [2].

Overall, naive methods for handling missing data fail to leverage the full information available in the dataset and often lead to biased and inefficient estimations. More sophisticated techniques, such as those based on machine learning models, provide better alternatives by iteratively predicting missing values using relationships within the data, thus offering more accurate and reliable imputations. [3].

3.2 Random Forest Imputation (MissForest)

According to a previous Stanford study [1], a few method could provide better result than the naive method. There is for example, Generative Adversarial Imputation Nets (GAIN), Variational Autoencoder (VAE), or KNN impute algorithm. However, the best of them is MissForest. This method not only achieved the best performance in filling missing financial data but also is the less expensive method in computational ressources. Therefore, we decided to implement it in our project.

Algorithm MissForest is an iterative imputation method that leverages the power of Random Forests to handle missing data. The process begins by making an initial guess for the missing values using simple methods like mean imputation. This serves as a starting point for the algorithm. Next, the features are organized based on the number of missing values, starting with the feature that has the least amount of missing data. This sorting is crucial as it allows the algorithm to progressively build more accurate models using the most complete data available. For each feature, the following steps are performed iteratively:

1. **Initial Guess**: The missing values in all features except the current one are filled with mean values.
2. **Model Fitting**: A Random Forest model is trained where the current feature with missing values is the response variable, and all other features are the predictors.

3. **Prediction**: The trained Random Forest model is used to predict the missing values for the current feature.
4. **Update**: The predicted values are used to update the dataset.

The algorithm then moves to the next feature with the second least number of missing values and repeats the process until all columns are done.Then the algorithm is repeated on all colums of the matrix according to the previous method. However this time the initial guess is not the mean but values found with the random forest. Finally, this iterative procedure continues until the difference between the imputed data matrices from consecutive iterations falls below a predefined threshold, indicating convergence. [1]

With this approach, MissForest ensures that the imputation progressively improves as more accurate estimates of the missing values are integrated into the dataset, leveraging the correlations between features captured by the Random Forest models. Then, MissForest is particularly advantageous because it can handle both continuous and categorical data simultaneously, a capability that many traditional methods lack. In various studies, MissForest has been shown to outperform other imputation methods, especially in datasets with complex interactions and non-linear relationships. Therefore, this algorithm fit really well to our Financial data.[3].

4 Machine Learning Algorithm

Because our goal is not only to forecast returns but to be able to interpret the impact of the predictors, we couldn't use advanced ML such as neural networks which act like black box. Thus we focused on linear models, as it is usually done in finance for their interpretability.

4.1 Data

In addition to the previous data preprocessing, some more choices about the data should be clarified. First, for the sake of focusing only on the provided financial predictors, we haven't added any additional -homemade- predictor, while also dropping the price, date and stock identifier -permno- from the feature set. Thus, our features -X matrix- is fully covered by the given financial predictors. Regarding the prediction target, we have opted for return rather than log-returns, as it had better experimental overview results, moreover to being easier to interpret.

In later stages of our work, we also choose to normalize our features using MinMaxScaler:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

It was preferred to standardization or other normalization methods by experimentally comparing their performance on all of our ML models. This change

resulted in a high improvement of accuracy scores. Normalizing over Standardizing also makes sense since our predictors aren't expected to follow a Gaussian distribution, which is suited for standardization. Finally, we create an in- and out- of sample set, by chronologically sorting our data before splitting them into a train and test set. This results into an in-sample train set containing 1'396'218 rows (80%) covering the period from Jannuary 1980 to December 2011, and an out of sample test set of 349'055 rows (20%) from December 2011 to November 2020.

To resume, our models will learn to forecast the next month return of a stock using only its' normalized predictors on data from 1980 to 2011. It will then be tested on the subsequent period up until November 2020.

4.2 Correlation Analysis

To get a quick insight of our predictors, we perform an analysis of their correlations. We start by computing the absolute pairwise correlation between all predictors using `PearsonCorrelation`. This helps in identifying which predictors act as duplicates by moving very similarly or oppositely. Thus, we directly find the pairs that represents the same underlying information, without having to dig into all predictors description.

Table 1: Top 10 Most Absolutely Correlated Predictor Pairs

Index	Pair	Correlation
0	(EBM, BPEBM)	0.999808
1	(AM, Leverage)	0.986174
2	(RealizedVol, IdioVol3F)	0.982106
3	(zerotradeAlt12, zerotrade)	0.973945
4	(Herf, HerfAsset)	0.945221
5	(dNoa, NOA)	0.944448
6	(zerotrade, zerotradeAlt1)	0.942450
7	(dNoa, InvestPPEInv)	0.923709
8	(Size, DolVol)	0.922138
9	(dNoa, AssetGrowth)	0.918073

As multi-collinearity can lead to several issues:

- **Inflated Variance**: It increases the variance of the coefficient estimates, making the model sensitive to changes in the model.
- **Interpretation Difficulty**: It becomes challenging to determine the effect of each predictor on the target variable.
- **Model Overfitting**: The model may overfit the training data and perform poorly on unseen data.

Still, we have chosen not to remove highly correlated predictors, since the issue is simply addressed with regularization methods. We then plot the mean correlation between predictors, which provides a comprehensive view of how each predictor correlates, in average, with the others. It is for example interesting to see that both Beta coefficient have low negative mean correlation, highlighting the importance to take into account the other predictors rather than focusing only on the market.

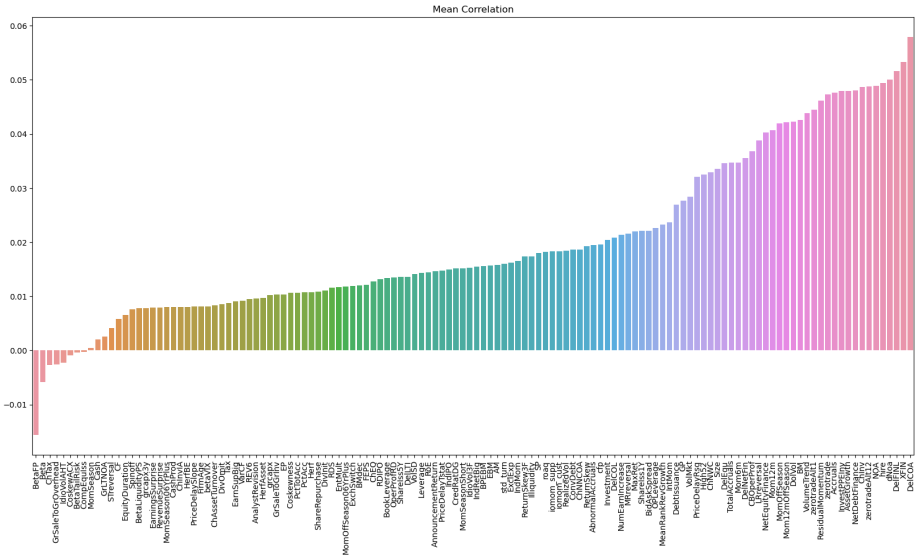

Fig. 4: Mean Correlation between Predictors

4.3 Ordinary Least Squares (OLS)

Method description In this section, we employ a simple OLS model to predict stock returns. The OLS is a fundamental linear regression technique that estimates the relationship between the predictors and the target variable by minimizing the sum of squared residuals (eq. **??**).

$$\min_{\beta} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 \right\} \tag{1}$$

***R*² and predictors** We start by training the OLS model using the training dataset. For our OLS model, the training R^2 value is 0.016 indicating that the model explains only 1.6% of the variance in the target variable. This suggests that the linear relationship between the predictors and the target variable is

weak, generally speaking it seems extremely low, but relative to the financial context of return forecasting, it is quite a good value. However, by looking at the top 20 predictors based on their absolute weights in the OLS model (Figure 5), we observe several issues such as huge coefficient weights despite the features normalization and highly correlated predictors (BPEBM-EBM, AM-Leverage) with opposite sign, nullifying each others. These are clear sign of overfitting and it is clearly reflected on the Test adjusted R^2 score with a notably low -49.4%.

A possible interpretation is that by overfitting the training data which were covering the 2008 crisis, the model learned to invest highly against the earnings forecast revision (REV6), leading to poor performance in the afterward period of prosperous growth.

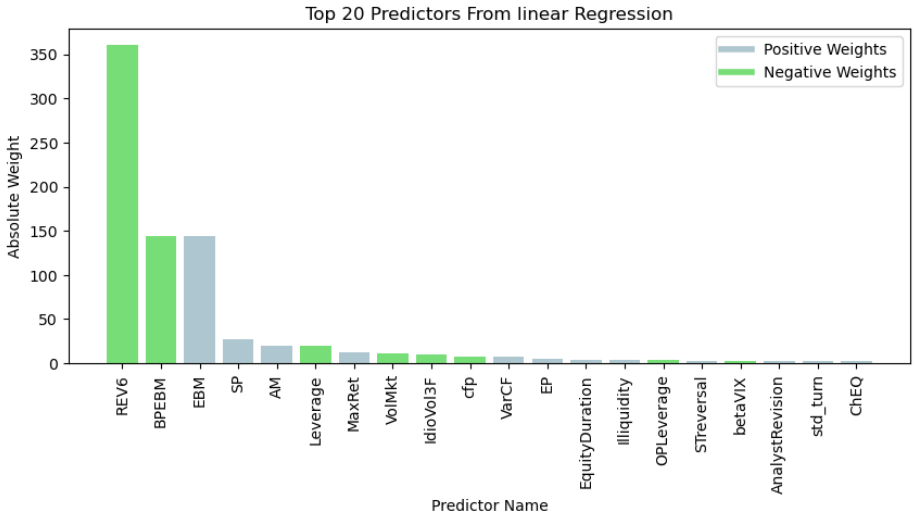

Fig. 5: Top 20 Predictors from OLS

4.4 Ridge

Method description To address this overfitting issue, we naturally look into adding regularization method to our regression. The most common one, Ridge, relies on including a L2 regularization to penalize high coefficient weights.

$$\min_{\beta} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \alpha \frac{1}{2} \|\beta\|_2^2 \right\} \tag{2}$$

To select the most appropriate alpha regularization coefficient, we perform a classical GridSearch hyperparameter tunning on a range of alpha values, evaluating each of them using 5-fold crossvalidation. After a few try of different alpha ranges, we find a clear minima from a convex curve.

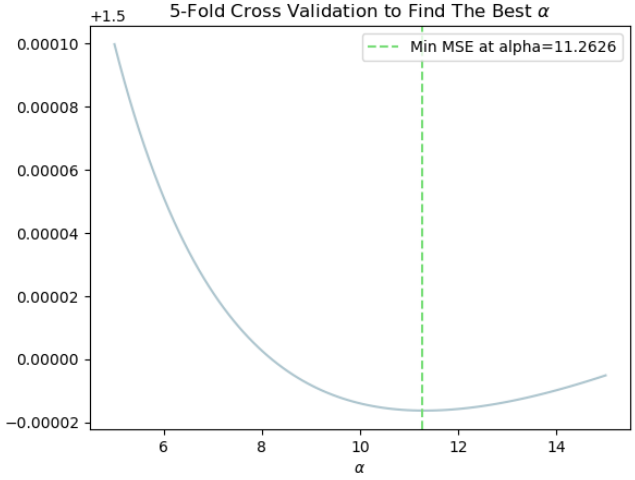

Fig. 6: 5-Fold Cross Validation to find the best α for Ridge

Optimal α is found to be 11.2626, as indicated by the vertical dashed line. This value minimizes the mean MSE over the folds.

R^2 and predictors Looking at the results, we can directly notice that we have indeed solved the overfitting issue. In fact, as we have lost 1% score compared to OLS on our train sample, resulting in a R^2 of 0.6%, it was truly for the best since our predictions are way more robust and now achieve a similar score on out-of-sample data with 0.65% test R^2. This robustness also shows up on the predictors weights that are way smaller and nicely distributed.
Sales-to-Prices shows up as the most important predictor to consider, with a positive weight indicating good returns.

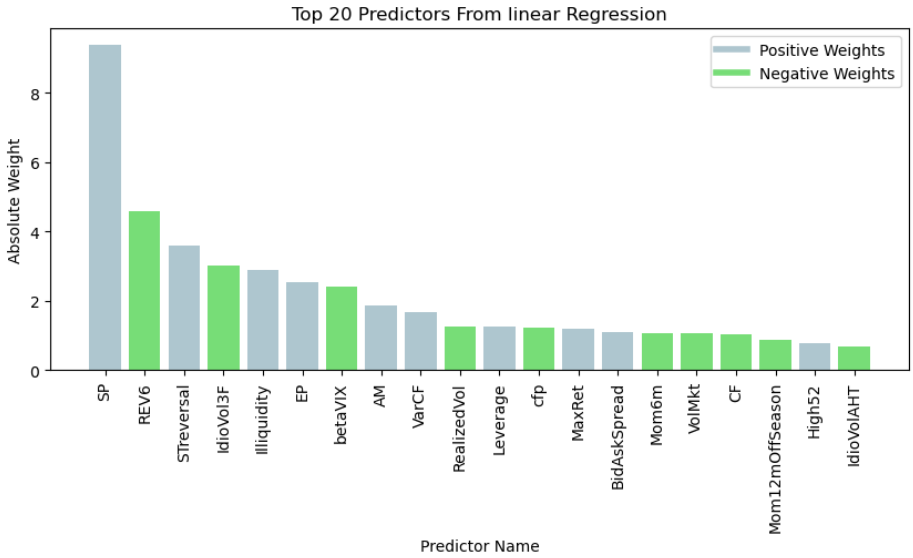

Fig. 7: Top 20 Predictors from Ridge

4.5 Lasso Regression Model

Method description The second must-see regularization method is the Least Absolute Shrinkage and Selection Operator - Lasso. Lasso regression is a linear regression technique that includes L1 regularization, which helps in feature selection by shrinking the coefficients of less important predictors to zero. This can be particularly beneficial in high-dimensional datasets where many predictors may be irrelevant or redundant. Indeed, it is very appropriate in our situation where we have more than a hundred predictors, with multi-collinearity and overfitting issues. It also provides the advantage to only select a tiny set of predictors to work with, concentrating the information.

$$\min_{\beta} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \alpha \|\beta\|_1 \right\} \tag{3}$$

As in Ridge, we select the best α value using GridSearch crossvalidation. This time we find an optimal α at 0.000008. (Figure 8).

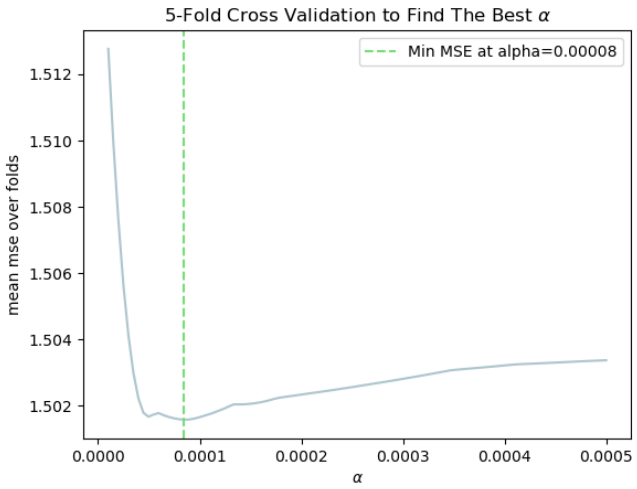

Fig. 8: 5-Fold Cross Validation to Find the Best α for Lasso

R^2 and predictors First regarding the scores, we obtain 0.4% train R^2 and 0.455% test R^2, achieving the wanted robustness. Out of the 130 predictors, only 29 remains, which is a nice advantage to precise our focus. It however comes with slightly lower scores compared to Ridge, and a unbalanced weight distribution among the selected predictors. It fact, Sales-to-Price (SP) monopolize most of the prediction power, which can be problematic as it suggests that the model

relies heavily on this predictor, thus is not robust if it happens that SP goes wrong.

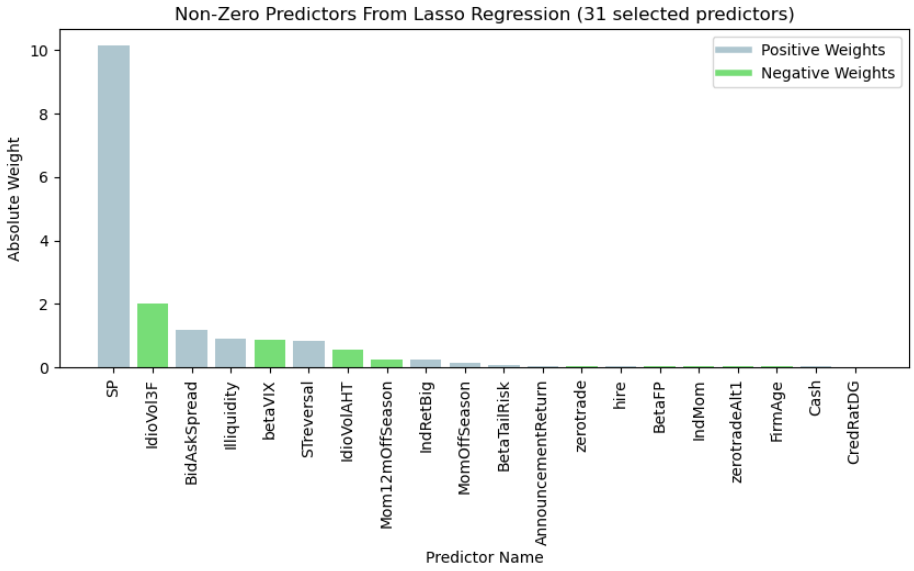

Fig. 9: Top 20 Non-Zero Predictors from Lasso (31 selected predictors)

4.6 Elastic Net Regression Model

Method description With the ambition to get the best of both Ridge and Lasso, and because it is commonly used in the industry, we implement the Elastic Net regression model. Elastic Net is a linear regression model that combines both L1 (Lasso) and L2 (Ridge) regularization techniques.

$$\min_{\beta} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \alpha \left[(1 - \text{L1 ratio}) \frac{1}{2}\|\beta\|_2^2 + \text{L1 ratio}\|\beta\|_1 \right] \right\} \tag{4}$$

The difference now is that we have 2 hyperparameters. α serves as regularization coefficient as before but it is now distributed to both L1 and L2 regularization. The L1 ratio then balance the 2 regularization types. We proceed a 2 dimensional GridSearch cross-validation to determine the best combination of α and L1 ratio. The MSE is calculated for each pair of values (Figure 10).

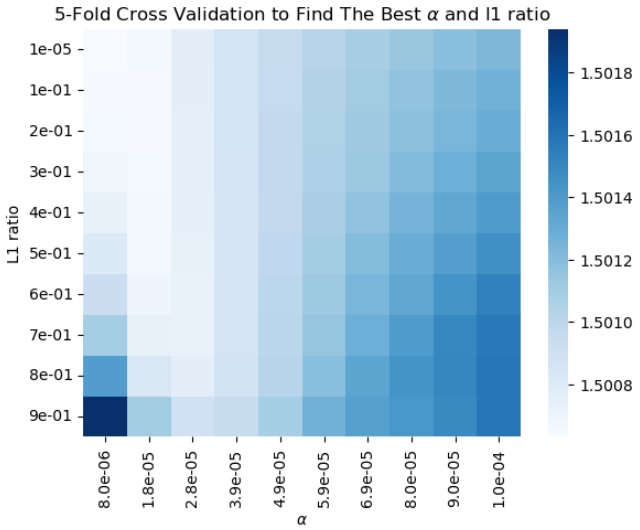

Fig. 10: 5-Fold Cross Validation to Find the Best α and L1 ratio for Elastic Net

The optimal values of α and L1 ratio are found to be $8e^{-06}$ and $1e^{-05}$ respectively, by minimizing the mean MSE over the folds. We understand from the L1 Ratio that the Elastic Net actually tends very close to the Ridge regression.

R^2 and Predictors Since the L1 Ratio is close to 0, the ElasticNet has quite exactly the same results as Ridge Regression. With train R^2 of 0.6% and test R^2 of 0.65%, the only notable difference is the drop of 2 predictors from the 130. (Figure 11). If wanted, we could choose to higher the L1 Ratio to get rid of more predictors, without losing much accuracy. This could be a good compromise

between the different techniques, thus achieving close to the best R^2 score while still reducing the parameters.

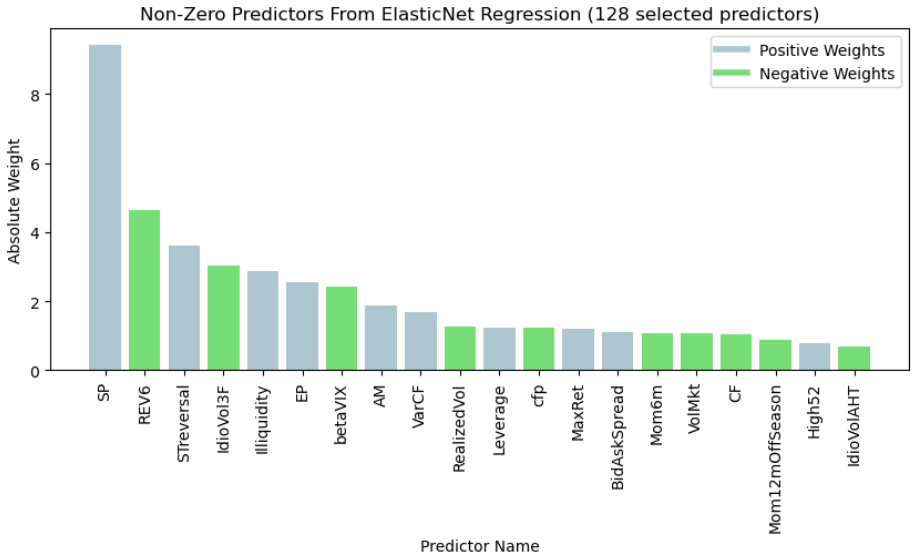

Fig. 11: Top 20 Predictors from Elastic Net (128 selected predictors)

4.7 Random Forest Regression Model

Method description To diverge from all the previous linear regression methods, we tried the Random Forest regression model, know to be very robust and efficient through all kind of ML tasks. Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of the individual trees. This approach helps in capturing complex, non-linear relationships in the data. As always, we perform GridSearch hyperparameter tuning using cross-validation on the train set to find the best model parameters. Specifically, we tune the following parameters:

- Number of estimators (trees) (`n_estimators`)
- Maximum depth of the trees (`max_depth`)
- Maximum features considered for each split (`max_features`)

Because we are working with a relatively big dataset and we are trying to select the best combination of values from 3 variables, it was very computationally expensive to run the hyperparameter tuning. So with our limited time and resources, we enter only 4 value combinations, as a gesture of goodwill. It resulted in selecting 100 estimators, a maximum depth of 3, and the use of `log2` for the maximum features.

R^2 and Predictors Random Forest obtain a train R^2 of 31.2% which isn't relevant since this algorithm can overfit any train set with enough decision leafs. We should only evaluate the out-of-sample $R^2 = -3.2\%$, revealing poor performances. It can be explained by the insufficient hyperparameter tunning, which wouldn't cover the models' tendency to overfit the train set, resulting in the same issues as OLS. We can also make the hypothesis that the predictors have indeed linear relationship to the returns, such that this models isn't very appropriate in the situation.
Still, the Random Forest model assigns importance to each feature based on how much they improve the split criterion across all trees in the forest, Figure 12 showing the top 20. Despite the poor prediction power of this model, we find the same predictors with high importance as in the other models, such as SP, and beta_VX and Illiquidity.

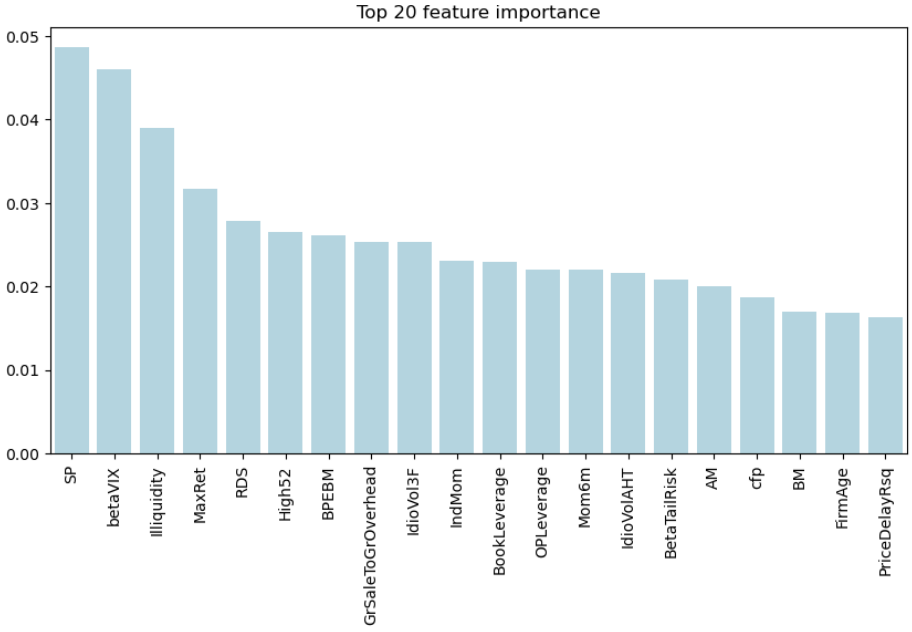

Fig. 12: Top 20 Features importance

5 Predictor Selection

5.1 Results summary

Based on the results presented in Table 2 ,OLS and Random Forest models overfitted the training set, leading to poor out-of-sample results. Conversely,

Ridge, Lasso, and Elastic Net models exhibited better generalization capabilities, with Ridge and Elastic Net achieving the best test set results. Elastic Net has in addition the advantage of performing feature selection.

Method	**Train** R^2	**Test** R^2
Simple OLS	1.6%	-48.4%
Lasso	0.40%	0.455%
Ridge	0.60%	0.65%
Elastic Net	0.60%	0.65%
Random Forest	31.2%	-3.2%

Table 2: Comparison of Machine Learning Methods for Train and Test R^2

Consequently, we will adopt the predictors identified by the Elastic Net model for our final analysis. This choice ensures that we leverage the most relevant and influential predictors, thereby improving the robustness and reliability of our return predictions.

5.2 Predictor Selection

Accordingly to Figure 11, the 20 best predictors with Elastic Net classed in a decreasing order, are the following:

- **SP**: Sales to price
- **REV6**: Earnings forecast revisions
- **STreversal**: Short-term reversal
- **Idiovol3F**: Idiosyncratic risk three factor
- **Illiquidity**: Amihud's illiquidity
- **EP**: Earnings to price ratio
- **betaVIX**: Systematic risk
- **AM**: Total asset to market
- **VarCF**: Cash flow to price variance
- **VolMKT**: Volume to market equity
- **Leverage**: Market leverage
- **RealizedVol**: Realized total volatility
- **BidAskSpread**: Bid Ask Spread
- **Mom6M**: Momentum (6 month)
- **CFP**: Operating cash flow to price
- **Mom12mOffSeason**: Momentum without the seasonal part
- **CF**: Cash flow to market
- **IntMom**: Intermediate momentum
- **MaxRet**: Maximum return over month
- **AnnouncementReturn**: Earning announcement returns

5.3 Discussion

We have identified these predictors using machine learning algorithms and will now evaluate their financial significance to support their selection. Each predictor's relevance is assessed within the context of asset pricing and return prediction. We identified different categories of predictors with financial meaning.

Firstly, ratios involving market capitalization, such as Sales to Price (SP) and Volume to Market Equity (VolMKT), provide insights into a company's valuation and market activity relative to its size. These ratios are crucial for understanding the market's perception of a company's growth prospects and liquidity.

Price-to-earnings ratios, represented by Earnings to Price (EP) and Earnings Forecast Revisions (REV6), relate a company's current share price to its per-share earnings. These ratios allow us to determine whether a stock is overvalued or undervalued compared to its historical performance or industry benchmarks.

Momentum indicators, including Short-term Reversal (STreversal) and Intermediate Momentum (IntMom), are based on the observation that stocks with positive past performance tend to continue performing well in the short to medium term. These indicators capture trends and price movements driven by investor sentiment and market dynamics.

Volatility measures, such as Systematic Risk (betaVIX) and Idiosyncratic Risk (Idiovol3F), quantify the degree of variation in a stock's price. High volatility is associated with higher risk and potential return, making these measures vital for risk assessment and portfolio management.

Liquidity measures, such as Amihud's Illiquidity (Illiquidity) and Bid-Ask Spread (BidAskSpread), provide insights into the ease with which assets can be traded without affecting their prices. These measures are important for understanding market depth and the potential impact on trading strategies.

Cash flow and earnings measures, such as Cash Flow to Market (CF), Cash Flow to Price Variance (VarCF), and Operating Cash Flow to Price (CFP), highlight the financial health and profitability of companies. These metrics are essential for understanding the sustainability of earnings and the efficiency of cash flow management.

By categorizing these predictors into ratios with market capitalization, price-to-earnings ratios, momentum indicators, volatility measures, liquidity measures, and cash flow and earnings measures, we ensure a comprehensive approach to capturing various dimensions of market behavior. Each group contributes uniquely to our understanding of the factors influencing stock returns, thereby enhancing the robustness and predictive power of our financial models.

6 Conclusion

To conclude, we employed multiple models, including Ordinary Least Squares (OLS), Ridge, Lasso Regression, Elastic Net, and Random Forest, to identify significant predictors of stock returns. OLS and Random Forest models were overfitting the sample data, which limits their predictive power on new data. Conversely, Ridge, Lasso, and Elastic Net models exhibited better generalization capabilities, with Elastic Net achieving the best test set results by balancing the trade-off between bias and variance.

These outcomes highlight key considerations from both machine learning and financial perspectives. From a machine learning standpoint, the findings underscore the importance of model complexity, where regularization techniques in Ridge, Lasso, and Elastic Net models help achieve a balance that maximizes predictive performance. Financially, our study provides valuable insights by identifying the significance of predictors such as market capitalization, price-to-earnings ratios, momentum indicators, volatility measures, liquidity measures, and cash flow and earnings measures. Furthermore, the study underscores the trade-off between model complexity and generalization performance while highlighting the financial significance of key predictors. The findings demonstrate the potential of machine learning techniques to select predictors, offering valuable tools for investors and analysts.

Looking ahead, future work will explore several promising areas. Identifying the sector for each asset and conducting a sector-based analysis could reveal differences in predictors' effectiveness across various industries. Additionally, analyzing predictors during major economic events could provide valuable insights into their robustness and adaptability in different market conditions. Due to time constraints, we concentrated on the overall picture in this study, but these future directions hold the potential to deepen our understanding and refine our predictive models further.

References

1. Ping, X., Vittayaareekul, S. & Li, Z. Imputation of Missing Financial Data. https://satitavitt.github.io/assets/img/Imputation_of_Missing_Financial_Data.pdf (2024).
2. Freyberger, J., Höppner, B., Neuhierl, M. A. & Weber, M. *Missing Data in Asset Pricing Panels* Chicago Booth Paper No. 22-19. Electronic copy available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3932438 (Fama-Miller Center for Research in Finance, The University of Chicago, Booth School of Business, 2024).
3. Stekhoven, D. J. & Bühlmann, P. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28,** 112–118. https://academic.oup.com/bioinformatics/article/28/1/112/219101 (2012).