technische universität
dortmund

Case Studies

Winter 2023

Project III
Forecasting the equity premium: Comparisons and conclusions

Ly Le Thi
January 28, 2024

in collaboration with Alexander Langnau, Sudhir Pratap Singh Rathore, and
Jonathan Albert Karras

Lecturers: Prof. Matei Demetrescu
Dr. Paul Navas Alban

Contents

1   Introduction

Excess returns are defined as the return obtained from the return on stock (or portfolio of stocks) above the risk-free rate. The prediction of excess stock returns has been a central theme in financial economics. Accurate forecasts of excess stock returns are important for investors, policymakers, and financial analysts as they impact investment decisions and portfolio management strategies. In this paper, we go deeper into stock excess return forecasting by looking at different forecasting models, including Ordinary least squares (OLS) regressions (Autoregressive regression and Forward stepwise) and two machine learning methods (Regression Tree and Random Forest).

The motivation for this research stems from Goyal and Welch's project (Welch and Goyal, 2008), where they contend that the historical average excess stock return outperforms the combination of other predictor variables. In this study, we explore the predictive power of various models relative to the historical data of series return ((historical model) compared to the general model which combines the historical data and all other covariates as predictors.

In building the historical model, we use lagged predictor variables derived from excess returns, incorporating up to ten days of lag. The model is trained on a designated set using the Autoregressive model of order p ($AR(p)$) for OLS regression and two machine learning methods. For the general model, we integrate historical data of excess returns with all covariates in the dataset. Using Forward stepwise selection for OLS regression and two machine learning methods. All models will be trained in the training set, tuning parameters and making predictions on the validation set to obtain the model with the smallest mean square forecast error (MSFE) using information criteria (AIC) to guide the determination of the appropriate lag (p) for the $AR(p)$ model and the best subset of predictors for Forward stepwise model. In addition, the complexity parameter (cp) for the regression tree, and number of decision trees ($ntree$), and the number of randomly chosen variables for splitting ($mtry$) for Random Forest are identified. All these hyperparameters will then be utilized for the model for forecasting in the testing set. The quality of these forecasts is assessed using the MSFEs. In addition, in order to compare the model performance, we employ the Diebold-Mariano test, which is a statistical test used to compare the forecasting accuracy of two models.

The MSFE results consistently show the Random Forest historical model excelling on the validation set across frequencies. However, on the testing set, the Regression Tree

general model, with all covariates, outperforms, especially in monthly and quarterly frequencies, emphasizing its practical effectiveness. Despite the Forward Stepwise model's strong in-sample forecasting, it performs poorly on the testing set, indicating overfitting. The Diebold-Mariano Test highlights that, overall, general machine learning models surpass OLS and historical models. No clear evidence supports one general model over the other or one historical model over the other between regression tree and random forest. In forecasting, each model has distinct strengths and limitations, emphasizing the need for a nuanced understanding of their dynamics.

In addition to this introduction, the report comprises four additional sections. The second section of this report provides a description of the dataset and the preprocessing data before coming to the model structure. The following section describes the different methods and software used in this report. The fourth section of this report states all the results that are obtained using OLS and two machine learning methods mentioned in the third section of the report and discusses the limitations of the findings. Finally, the fifth section summarizes all the results.

2 Data

2.1 All predictors

The dataset used in this project is taken from Welch Goyal's project which is provided on the website $https://sites.google.com/view/agoyal145$. The original dataset includes twenty-two time series variables calculated at various frequencies (monthly, quarterly, and annually). To streamline the analysis, we exclude seven variables. First are two series $CRSP\_SPvw$ and $CRSP\_SPvwx$ as they are stock return series from Welch Goyal's project and won't be used as predictors. Additionally, other series, including Cross-Sectional Premium, Investment Capital Ratio, Dividend 3-year Price Ratio, Earnings 3-year Price Ratio, Consumption, wealth, income ratio, and Investment Capital Ratio, are excluded due to over half of their data being labeled as "NA," rendering them unsuitable for our purposes of out-of-sample forecasting for the last 10 years, dropping all rows with "NA" values, including those of these predictors, would result in insufficient training data for building a robust model. Consequently, the refined dataset utilized in this research comprises fifteen variables. Each of these variables is graphically represented in the accompanying Figure. 1.
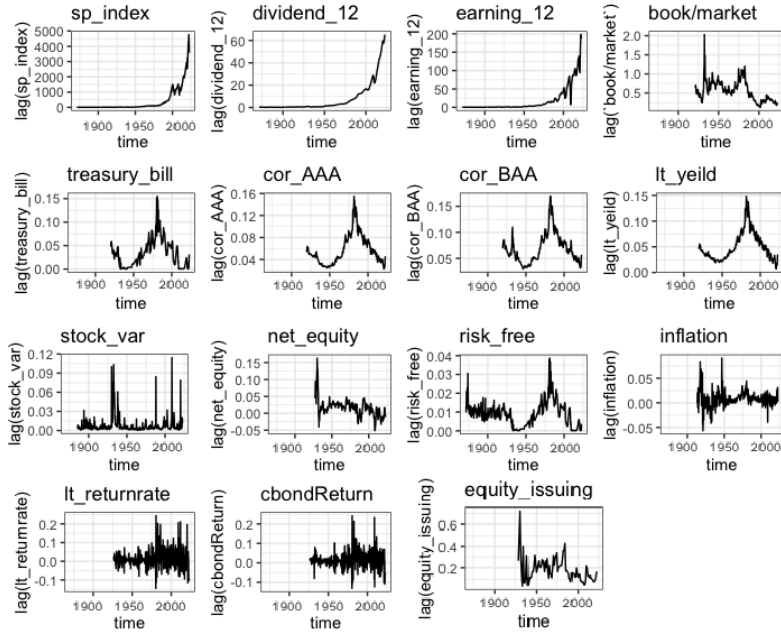
Figure 1: Plot of all fifteen variables over time.

From the plot, we can see that the compounded returns on the S&P 500 index, Dividend 12, Earning 12, have an increasing trend since they are the cumulative sum of the dividends paid on the S&P 500 index (or earnings from the S&P 500 index) over the measures period. The time series Treasure-Bills (the short-term U.S. government bonds), Corporate Bond Returns (including AAA and BAA) which indicates the performance of corporate debt markets, Long-Term Yield (the yield on long-term government bonds), and Risk-free rate (the interest rate derived from Treasury-bill rates) all have increasing trends from 1950 and reach the peak in the early 1980s then decrease until the end of the research period (2022). The other variables such as Net Equity Expansion (measures the net issuance of equity by NYSE-listed stocks over 12 months), Inflation (the Consumer Price Index of all urban consumers), Stock Variance (the sum of squared daily returns on the S&P 500 index) have large volatility in the research period.

2.2   Excess returns

Excess return is the surplus return on the investment above the risk-free rate, providing insights into the additional return earned for taking on additional risk. By subtracting the risk-free rate from the asset's return, it's possible to measure the additional return generated by the asset in comparison to a virtually risk-free investment.

Based on the definition of stock excess returns, the time series excess returns are generated for all three frequencies - monthly, quarterly, and annual- by first calculating

3

the percentage change between consecutive values in the S&P 500 index data, effectively determining the daily returns of the index with the help of *Delt* function in package *quantmod* in R, which taking the percentage change of stock price to have the stock returns $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ with P_t is the S&P 500 stock price at time t.

Following this, the function subtracts the risk-free rate from the calculated returns to derive the excess returns $ER_t = R_t - RF_t$ with ER_t as the excess return at time t, R_t as the stock return which has been calculated above and RF_t is the risk-free rate at time t. This is a common approach in finance known as the "excess return formula." The time series excess returns are plotted in Figure 2 for monthly data, and for quarterly data and annual data, plots will be placed in Figure 11 and Figure 10 in the Appendix
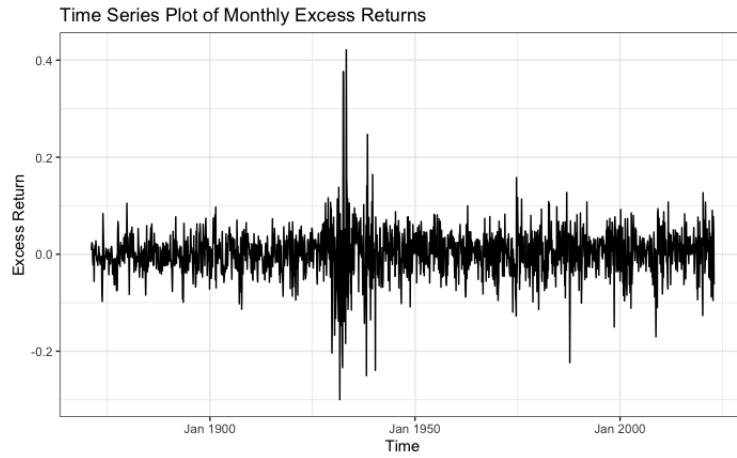

Figure 2: Monthly excess returns time series.

All three-time series excess returns for three frequencies show a constant or stationary trend. The time series excess returns vary around the mean that close to zero and do not show any upward or downward trend along the research times. This can be because the excess returns are generated by taking first differences in short-term future prices that will remove the trend of the S&P 500 and then subtract the risk-free rate. In table 1 the minimum observation, the first quantile (25% quantile), the median, the mean, variance and the third quantile (75% quantile), and the maximum observation for the excess returns time series for different frequencies (monthly, quarterly, and annually time series data).

Analyzing the monthly excess returns series, the data reveals a balanced trend with a mean value equal to -0.3, and close to zero variance (0.002) indicating overall stability. The range extends from a minimum of -0.02 to a maximum of 0.42, suggesting potential fluctuations in first differences of future S&P 500 prices. The same fluctuation happens

Table 1: Summary statistics of excess returns series.

Excess return	Min	1st Quantile	Median	Mean	Variance	3rd Quantile	Max
Monthly	-0.30	-0.02	0.003	0.002	0.002	0.03	0.42
Quarterly	-0.40	-0.04	0.01	0.006	0.009	0.06	0.86
Annual	-0.48	-0.09	0.02	0.03	0.035	0.15	0.46

with quarterly and annual data around the mean -0.4, -0.48, and variance 0.009, and 0.035 respectively. The results also show that the excess returns data less fluctuate with a higher frequency and that monthly data has the smallest variance, then quarterly and annual data have higher fluctuation.

The autocorrelations function (ACF) and the partial-autocorrelations (PACF) of the three-time series excess returns at monthly, quarterly, and annually for the training set are plotted in Figure 3. From ACF and PACF plots, an idea of which past observations are related to their own past value can be obtained. They also will be considered as the benchmark for fitting $AR(p)$ models on the training set in this project.
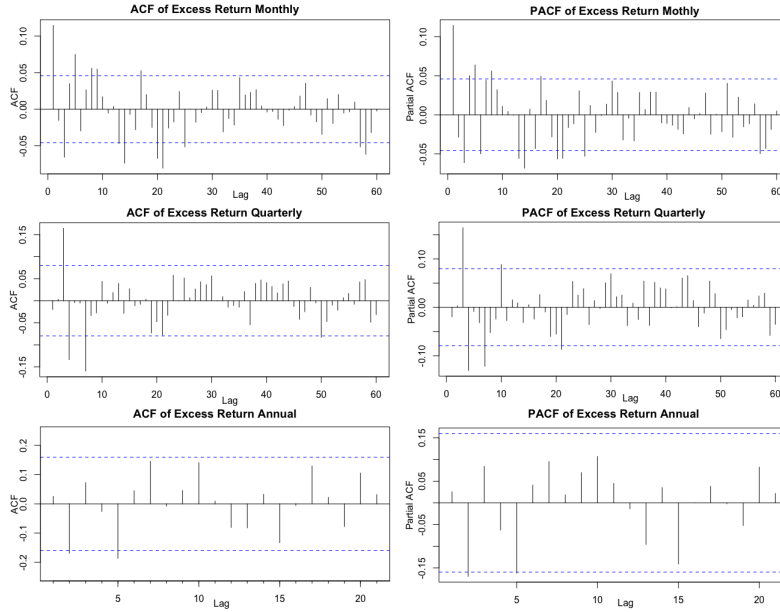
Figure 3: Plot of the autocorrelations and the partial-autocorrelations of excess return series for three frequencies on the training and validation set.

In each plot, the blue lines depict the 95% confidence interval and are indicators of the significance threshold. The lags which are beyond the blue lines are considered important. For all three frequencies, the ACF plots exhibit a gradually decaying pattern as we look at lags further into the past, especially after a lag of 21 (for monthly data), 7 (for quarterly data), and 5 (for annual data), there is no correlation exceed the confidence interval line. Looking at PACF plots for three models, we can see the significant cut-off after lag 3

for monthly data, after lag 3 the correlations are very close to the confidence interval line. For quarterly excess return data, after lagging 3 and 4, the correlation seems not so significant, and for annual data the correlation is quite weak, and only at lag 2 does the correlation level exceed the confidence interval line and lie on the confidence interval line at lag 5. Accordingly, the AR(p) model will be fitted with the range of lag p based on observations on ACF and PACF plots.

2.3 Data Preprocessing

2.3.1 Creating historical data

For the entire dataset, we generate one-month lags for all monthly covariates, analogously one-quarter and one-year lag for quarterly and yearly respectively, which will become instruments for subsequent calculations, model fitting, and forecasting. This involves creating a prediction $\hat{y}_{t+1} = \hat{f}(y_t, \ldots, y_{t-n+1}, X)$, where $\hat{f}(y_t)$ represents the estimated function. In the context of time series excess returns, historical data is constructed by incorporating one to ten lags, utilizing these historical lags as predictors for subsequent historical modeling. The historical dataset incorporates a target variable, time series excess returns, along with ten predictors representing lags of the excess returns series. The resulting historical dataset then be combined with 13 covariates (excluding the time series S&P 500 and time series risk-free rate which were used in excess returns calculation) creating the general dataset that includes one target variable, time series excess returns, and a total of 23 predictors.

2.3.2 Splitting time series

After cleaning the missing value, taking lagged for excess return series, and aligning with the primary dataset encompassing all covariates, the range of the research period is set from January 1927 to December 2022. The final dataset for monthly then contains 23 predictor variables with 1152 data points starting from 1927-01 and ending in 2022-12, 384 data points for the quarterly dataset, from the first quarter of 1927 to the last quarter of 2022. Analogously, in the annual dataset, we will have a total of 95 data points from 1927-2022. The last 10 years of data then will be taken for out-of-sample prediction. The remaining dataset will be split into training and validation sets following the timeline order with 80% proportion of the data allocated to the training set (from 1927-01 to 2003-10) and the 20% left allocated to the validation set (from 2003-10 to 2012-12). This

split enables the model to be trained on one subset of the data and evaluated on another, and using the trained model for out-of-sample forecasting.

3 Methods

In this section, we provide an overview and description of the various methods used in this project, as well as the software utilized for their implementation.

3.1 Ordinary Least Square

Ordinary Least Squares regression, referred to as OLS, is one of the most common techniques for estimating the parameters in a linear regression model, which describes the relationship between one or more independent quantitative variables and a dependent variable (simple or multiple linear regression). This model is described mathematically as equation (1) below.

$$y_t = x_t^\mathsf{T}\beta + \epsilon_t = \beta_0 + \beta_1 x_{t,1} + \beta_2 x_{t,2} + \cdots + \beta_k x_{t,k} + \epsilon_t, \tag{1}$$

with $t = 1, 2, \ldots, T$, T is the last time point or the total number of observations of regression model. The target of regression model is the N time points of vector $Y = (y_1, y_2, \ldots, y_N)^\mathsf{T}$. The part of the right-hand side is called the regression or the regression function which involves the k-dimensional (column) vector $(x_{t1}, x_{t2}, \ldots, x_{tK})^\mathsf{T}$, k-dimensional vector of the random component $(\epsilon_1, \epsilon_2, \ldots, \epsilon_T)^\mathsf{T}$ and the coefficients $(\beta_0, \beta_1, \beta_2, \ldots, \beta_k)^\mathsf{T}$ are called the regression coefficients, which represents the change in the dependent variable when the kth regressor increases by one unit while other regressors are held constant. In the language of calculus, this can be expressed as $\frac{\partial y_t}{\partial x_{tk}} = \beta_k$

For the matrix notation, define the $T \times k$ matrix of regressors X

$$\underset{T\times k}{\mathbf{X}} = \begin{bmatrix} x_1^\mathsf{T} \\ x_2^\mathsf{T} \\ \vdots \\ x_n^\mathsf{T} \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & \ldots & x_{k,1} \\ 1 & x_{1,2} & \ldots & x_{k,2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,T} & \ldots & x_{k,T} \end{bmatrix}$$

we can rewrite the linear regression equation as $\underset{T\times 1}{\mathbf{Y}} = \underset{T\times k}{X}\underset{k\times 1}{\beta} + \underset{T\times 1}{\varepsilon}$

In the context of linear regression, OLS aims to determine the set coefficients $\beta_0, \beta_1, \ldots, \beta_k$ that minimize the sum of the squared differences between the observed and predicted val-

ues, which is known as the sum of square errors (SSR) with $SSR = \sum_{t=1}^{T}(y_t - \hat{y}_t)^2$. In that way, the vector β of the coefficients can be estimated by the following formula:

$$\beta = (X^T X)^{-1} X^T Y$$

3.2 Autoregressive Process

An autoregressive (AR) process is a representation of a type of random process where the current variable is modeled as a linear combination of its past values and a stochastic component. An autoregressive process of order p (AR(p)) satisfies the equation

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t, \qquad (2)$$

where Y_t is the value of the time series variable at time t, $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients and $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$ are the lag (or past) values of the time series variable

An AR(p) process is called stationary if all the roots of its characteristic polynomial which is given by

$$1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_p z^p = 0,$$

lie outside the unit circle. In this project, we assume that the time series variables follow a Gaussian AR(p) process of the form in equation (1) where $\epsilon_t \sim i.i.d\ \mathcal{N}(0, \sigma^2)$. The vector of parameters that needs to be estimated is $\boldsymbol{\theta} = (c, \phi_1, \phi_2, \ldots, \phi_p, \sigma^2)$. (Hamilton, 1994).

3.3 Forward stepwise model

For achieving optimal subset selection, we employ the forward stepwise model selection method. This approach entails fitting a null model along with $K - k$ models during the kth iteration, where k ranges from 0 to $K - 1$. The process begins with a model devoid of predictors and incrementally adds predictors one at a time until all predictors are included. Specifically, at each step, the variable that contributes the most improvement to the model fit is added. The formal description of the forward stepwise selection procedure is outlined in Algorithm below:

(1) Let M_0 represent the null model, devoid of any predictors, predicting the sample mean for each observation. For each $k = 0, 1, \ldots, K - 1$:

(a) Evaluate all $K - k$ by adding one additional predictor to the predictors in M_k .

(2) Identify the best model among these $K - k$ models, referred to as M_{k+1}. The criteria for determining the best model include the smallest Residual Sum of Squares (RSS) or the highest R^2.

(3) Employ a selection criterion, such as the C_p (AIC) BIC or adjusted R^2 for choosing a singular best model from the set $M_0, M_1, ..., M_K$.

The computational efficiency of forward stepwise selection makes it a viable alternative to general best subset selection and often performs effectively in real-world scenarios (Hamilton, 2013).

3.4   Akaike Information Criterion

To detect the optimal number of orders (p) in an AR(p) model, and mesure the best subset in forward stepwise selections, the Akaike information criterion (AIC) has been used. AIC is a mathematical method for evaluating how well a model fits the data it was generated from. In this project, AIC is used to compare different possible models and determine which one is the best fit for the data. AIC has the general form:

$$AIC(m) = 2K - 2\log(\hat{\mathcal{L}}) \tag{3}$$

Here m denotes the model, K denotes the number of parameters in the model and $\hat{\mathcal{L}}$ is the associated likelihood of the fitted model given parameters K. The optimum lag order is chosen such that AIC is minimum over all possible lag orders (Aho et al., 2014).

3.5   Regression Tree

A regression tree, a variant of decision trees in regression analysis, predicts continuous numerical outputs. Decision trees, a popular class of machine learning algorithms, can handle both classification and regression tasks. Unlike classification trees that partition feature space into distinct classes, regression trees aim to predict a continuous numeric target value (Y) based on predictor variables $(X_1, \ldots, X_p)$ (Loh, 2011).

The algorithm constructs the tree by selecting the best feature at each node using criteria like mean squared error for regression trees. Nodes split the data into subsets, forming branches and child nodes. This recursive process continues until a stopping criterion is met.

Algorithm 1 outlines the pseudocode for building a regression tree:

(1) Start at the root node.

(2) Identify the variable X and set S that minimizes impurities in child nodes. Choose the split $(X^* \in S^*)$ with minimum overall impurity.

(3) If the stopping criterion is met, terminate. Otherwise, repeat step 2 for each child node (Loh, 2011).

The accompanying Figure 4 illustrates the hierarchical structure of a regression tree. Beginning at the root node, the algorithm iterative splits nodes based on variable values to create homogeneous subsets, resulting in decision and terminal nodes. Pruning, the removal of sub-nodes, balances complexity, and precision. A regression tree provides a structured model for clear decision paths, facilitating relationships within the dataset through piecewise constant models. Though simple to interpret, their predictive accuracy may be lower compared to smoother models.
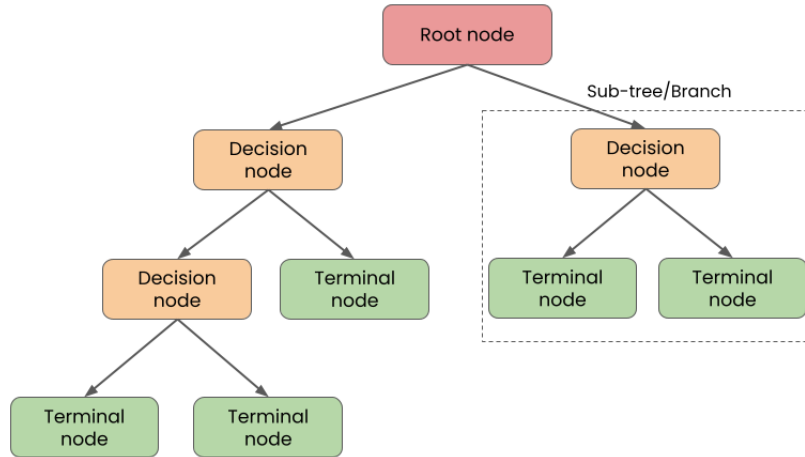

Figure 4: Architecture of Regression Tree ((Chauhan, 2022))

3.6 Random Forest

Random Forest, a tree-based machine learning algorithm, leverages the collective power of multiple decision trees for robust decision-making. Unlike a single decision tree, where nodes are determined through a specific process, Random Forest builds individual trees using a randomized approach, employing a subset of randomly chosen features for each tree. This randomness enhances diversity, reducing overfitting risks. Nodes in these trees operate on random feature subsets, and the final result is obtained by consolidating outputs from all trees.

Integral to Random Forest is bootstrapping, a stochastic method involving random

item selection from the training dataset. This diversity in decisions is amalgamated through majority voting, a process known as Ensemble Learning.

Random Forest follows the depicted structure in Figure 5, with the following key steps:

(1) Each tree is cultivated using a random, replacement-drawn subset from the training set with N cases.

(2) If there are M input variables, a constant value m (where $m < M$) is chosen. At each node, m variables are randomly selected from the M, and the node is split using the best among these m variables.

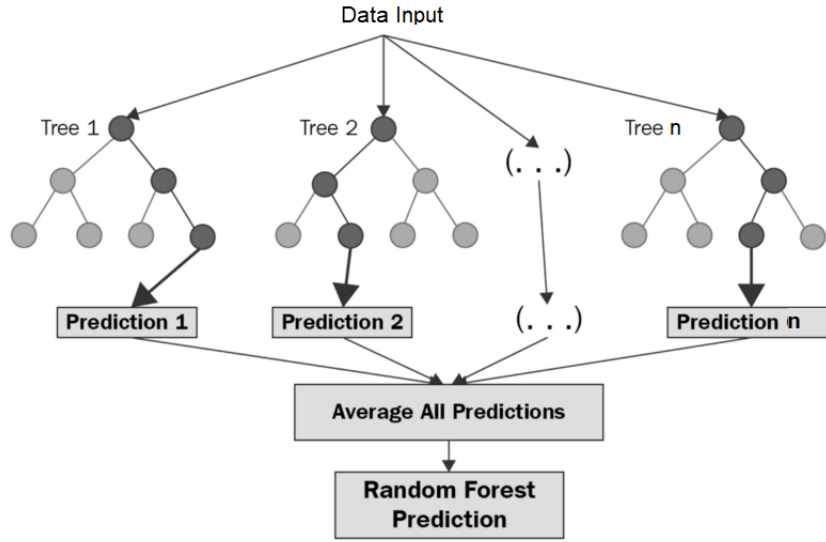(3) Trees grow to their maximum extent without pruning, and final predictions for new data are aggregated from all trees.

Figure 5: Architecture of Random Forest ((Raj, 2020))

3.7 Mean Square Forecast Error

To evaluate the accuracy of forecasts in this project, we employed Mean Square Forecast Error (MSFE). MSFE is used to measure the accuracy of forecasts in time series analysis. It's computed by taking the root of average of the squared differences between forecasted values and actual values over a specified period as formula

$$MSFE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{4}$$

.

MSFE is the key metric for evaluating the performance of forecasting methods or

models and helps in comparing different forecasting techniques. A high MSFE implies that the fitted values are further away from the actual data, the lower MSFE indicates the better forecasting accuracy.

3.8   Diebold-Mariano test

While MSFE allows for model comparison, it has limitations when assessing the statistical significance of observed differences. Merely computing confidence intervals may overlook the intricate dependency structure between forecast errors. The Diebold-Mariano test (DM-test) addresses this issue, providing a robust framework to test whether differences in forecast accuracy are statistically significant. Unlike MSFE alone, the DM test enhances testing power by considering temporal dependencies and serial correlation in forecast errors. This makes it a valuable tool for making more informed and reliable model selection decisions in forecasting applications.

We consider two forecast $\{\hat{y}_{it}\}_{t=1}^T$ and $\{\hat{y}_{jt}\}_{t=1}^T$ of the time series $\{\hat{y}_t\}_{t=1}^T$ with the associated forecast error $\{e_{it}\}_{t=1}^T = \{\hat{y}_{it}\}_{t=1}^T - \{y_{it}\}_{t=1}^T$ and $\{e_{it}\}_{t=1}^T = \{\hat{y}_{it}\}_{t=1}^T - \{y_{it}\}_{t=1}^T$. The key objective is to compare a loss differential of two losses $L_{i,t}$, measures the loss of model I at time t, and $L_{j,t}$, measures the loss of model J at a certain time t, with $L_{i,t} = \|eit\|$ for the absolute error loss or (or ℓ_1 loss) or $L_{i,t} = \|eit\|^2$ for the squared-error loss (or ℓ_2 loss). The series $\Delta_{i;j;t} = L_{i,t} - L_{j,t}$ measures the difference between both losses. If this difference is not significantly different from zero, then from the statistical point of view no forecast is better than the other one. If $\Delta_{i;j;t}$ is significantly smaller than 0, then forecast I is significantly better than forecast J, with respect to the chosen significance level. And vice versa: if $\Delta_{i;j;t}$ is significantly greater than zero, then I is significantly better than J. The test works in analogy to a t-test or z-test, where we can compare the differences between two means.

We would like to create the null hypothesis with $H_0 : E(\Delta_{i;j;t}) = 0$ with all t, which means two forecasts I and J have equal forecast accuracy, versus the alternative hypothesis $H_1 : E(\Delta_{i;j;t}) \neq 0$, which means two forecasts I and J have different levels of accuracy. Consider a path of samples denoted as $\Delta_{t=1}^T$ within the series of loss differences. If this series, reflecting variations in losses, exhibits covariance stationarity and has a short memory, standard statistical outcomes can be employed to determine the asymptotic distribution of the average loss differential within the sample.

$$\sqrt{T}(\bar{\Delta} - \mu) \xrightarrow{d} \mathcal{N}(0, 2\pi f_\Delta(0))$$

where $\bar{\Delta} = \frac{1}{T}\sum_{t=1}^{T}(L_{i,t}-L_{j,t})$ is the sample mean loss differential $f_\Delta(0) = \frac{1}{2\pi}\sum_{\tau=-\infty}^{\infty}\gamma_\Delta(\tau)$ is the spectral density of the loss differential at frequency 0, $\gamma_\Delta(\tau)$ is the autocovariance of the loss differential at lag τ and μ is the mean loss differential. It is possible to show that if the loss differential series $\{\Delta_{i;j;t} with t = 1,...,T\}$ is covariance stationary and short memory, then

$$\sqrt{T}(\bar{\Delta} - \mu) \xrightarrow{d} \mathcal{N}(0, 2\pi f_\Delta(0)) => \frac{\bar{\Delta} - \mu}{\sqrt{\frac{2\pi f_\Delta(0)}{T}}} \xrightarrow{d} \mathcal{N}(0, 1)$$

Because in the large samples the sample mean loss differential $\Delta_{i;j;t}$ is approximately normally distributed with mean μ and variance $\frac{2\pi f_\Delta(0)}{T}$ and under H_0:

$$\frac{\bar{\Delta}}{\sqrt{\frac{2\pi f_\Delta(0)}{T}}} \xrightarrow{d} \mathcal{N}(0, 1)$$

.

Suppose that the forecasts are h-step-ahead ($h \geq 1$). In order to test the null hypothesis that the two forecasts have the same accuracy, Diebold-Mariano utilizes the following statistic:

$$DM = \frac{\bar{\Delta}_{i;j;t}}{\sqrt{\frac{2\pi \hat{f}_\Delta(0)}{T}}}$$

where $\hat{f}_\Delta(0)$ is a consistent estimate of $f_\Delta(0)$ In our case consider one step ahead forecast then $h = 1$ then $\hat{f}_\Delta(0) = \frac{1}{2\pi}\hat{\gamma}_\Delta(0)$ with $\hat{\gamma}_\Delta(k) = \frac{1}{T}\sum_{t=|k|+1}^{T}(\Delta_t - \bar{\Delta})(\Delta_{t-|k|} - \bar{\Delta})$

Under the null hypothesis, the test statistics DM is asymptotically $N(0,1)$ distributed. The null hypothesis of no difference will be rejected if the computed DM statistic falls outside the range of -$z_{\frac{\alpha}{2}}$ to $z_{\frac{\alpha}{2}}$, that is if $|DM| > z_{\frac{\alpha}{2}}$ where $z_{\frac{\alpha}{2}}$ is the upper (or positive) z-value from the standard normal table corresponding to half of the desired α level of the test (Raj, 2020).

The Diebold-Mariano test should not be applied to situations where competing forecasts are obtained using two nested models. The root of the problem is that, at the population level, if the null hypothesis of equal predictive accuracy is true, the forecast errors from the competing models are the same and perfectly correlated, which means that the numerator and denominator of a Diebold-Mariano test are each limiting to zero

as the estimation sample and prediction sample grows. However, when the size of the estimation sample remains finite as the size of the prediction sample grows, parameter estimates are prevented from reaching their probability limits and the Diebold-Mariano test remains asymptotically valid even for nested models, under some regularity assumptions. Essentially, this means that model parameters are estimated using a rolling window of data, rather than an expanding one.

3.9   Rolling window forecast

Rolling forecasts are a common technique in time series modeling. The simplest approach is to estimate the model on a single set of training data, and then compute one-step forecasts on the remaining test data. This involves creating a prediction $\hat{y}_{t+1} = \hat{f}(y_t, \ldots, y_{t-n+1}, X)$, where $\hat{f}(y_t)$ represents the estimated function. This can be handled by applying the fitted model to the whole data set and then extracting the "fitted values" $\hat{y}_t$ which are simply one-step forecasts. We apply this technique in the previous two projects which focus more on the OLS and Machine learning methods, not on the forecast technique requirement.

For this 3rd project, the training data and re-estimating the model at each iteration before making forecasts, adopt a more dynamic strategy for this project. We use the last 10 years (2012-2022) as the out-of-sample testing set and the preceding 85 years (1927-2011). In the training set, we employ an 80% training split to train the model using a rolling window approach with window $D = 68$. Subsequently, N_eval windows, each representing 20% of the remaining training set are designated for evaluation ($N\_eval = 17$). The procedure for the rolling forecast will be described below:

(1)Training Phase: The model is initially trained on a fixed window of data, $D = 68$. After training, the model is used to forecast values for the evaluation set ($D + 1, \ldots, D + N_{eval} - 1$) for each iteration.

(2) Model Selection and evaluation: The goal is to identify the best model with the smallest MSFE across different models and different frequencies. The selected model, along with its hyperparameters, which minimizes the MSFE, is then used to forecast values on an out-of-sample testing set ($D + N\_eval, D + N\_eval + 1, \ldots, D + N\_eval + 10$). For machine learning models, the set of best hyperparameters will be identified and recorded for the model utilizing the testing set. For traditional OLS models like AR(p) and Forward Stepwise models in-sample forecasting is done for the entire period ($D, \ldots, D + N\_eval - 1$),

then using information criteria (AIC) to identify the best lag p of AR(p) model that resulting the smallest MSFE. For Forward Stepwise models, the best selection subset will be identified recorded, and used for forecasting model on the testing set.
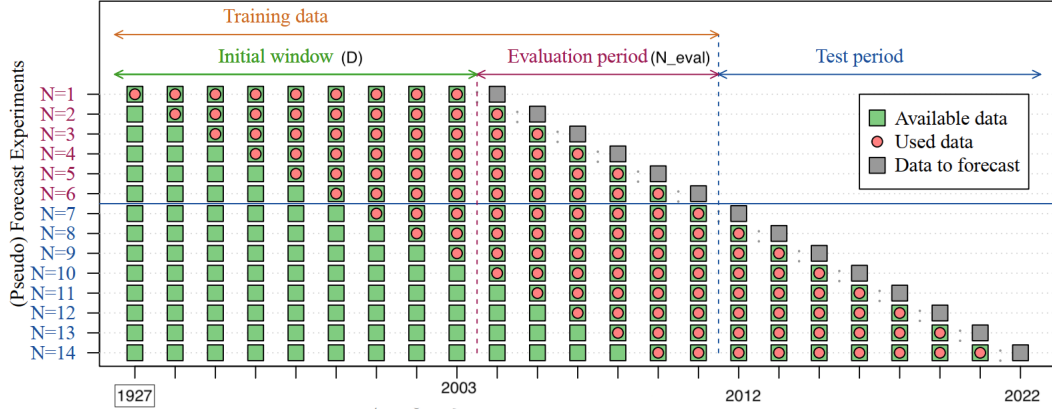

Figure 6: Rolling window forecast (Hansika Hewamalage, 2023)

(3) Rolling Window Iteration:

The whole testing period will range from $(D + N\_eval, D + N\_eval + 1, \ldots, D + N\_eval + 10)$ In each iteration, the fixed training window D is rolled forward by one step ahead for the forecast. For example, in the first iteration, the model is trained on data from $D + N\_eval$ back to the time with the length window D of training and is used to forecast at $D + 1$ Figure 6. In the second iteration, the window is rolled forward again, training on data from $D + 1$ to $D + N\_eval - 1$, reestimating the model and forecasting at $D + 2$. At each step (e.g., D+1, D+2), the real observed value is used to retrain the model for forecasting the subsequent step. This process continues iteratively, with the training window advancing one step at a time. While the training window D is rolled forward iteratively, it remains fixed during the retraining process. This ensures that the model is continually updated with new data, incorporating the most recent observations for forecasting.

3.10 Statistical Software

The programming language R (R Core Team, 2021) is used for analyzing and processing the dataset along with the package dplyr (Wickham et al., 2022). The *ggplot*2 package (Wickham, 2016) is also used in this project for data visualization. For calculation of the excess return series we employed the *Delt* function in package *quantmod*, for subset

selection package *lmSubsets* (Hofmann, 2022) has been used.

4 Results

This section is dedicated to analyzing the data using the methods defined in the preceding section. It is divided into the following subsections:

4.1 Hyperparameter

4.1.1 AR(p) Processes

This section represents the results after fitting the AR(p) process of the form (2) to the training dataset of series excess returns with 1152 observations (for monthly data), 384 observations (for quarterly data), and 95 observations (for annual data). Corporate with the plot for ACF and PACF of the excess returns time series, the lag orders are allowed to vary from 1 to 5, and the corresponding AIC is noted. The optimum orders are then determined such that the corresponding model has the lowest AIC. The formula for AIC used is the same as in equation (3). The particular combination of lag orders that have the lowest AIC is noted in Table 2.

Table 2: Optimum models with the lowest AIC for excess return series.

Excess returns	AR(p) Process	AIC
Monthly	AR(3)	-3052.96
Quarterly	AR(4)	-535.50
Annual	AR(1)	-29.45

The AR model then can be identified as $p = 3$ for the monthly excess returns model, $p = 4$ for the quarterly dataset, and $p = 1$ for the annual dataset

4.1.2 Forward stepwise subset selection

The outcomes obtained from the forward stepwise subset selection process are documented for excess returns prediction models. The model incorporates the best subsets, determined through the forward stepwise selection method incorporated with selection criteria AIC. The stepwise forecasting model for monthly dataset is given as:

$$\hat{ER_t} = \beta_0 + \beta_1 er\_lag\_3 + \beta_2 er\_lag\_1 + \beta_3 lag\_bookmarketRatio + \beta_4 er\_lag\_9 + \beta_5 er\_lag\_5$$
$$+ \beta_6 er\_lag\_8 + \beta_7 lag\_cbond\_return + \beta_8 lag\_treasury\_bill + \beta_9 lag\_dividend\_12$$
$$+ \beta_{10} er\_lag\_6 + \epsilon_t$$

(4.1)

The regression equation obtained for the quarterly dataset is given as:

$$\hat{ER_t} = \beta_0 + \beta_1 er\_lag\_7 + \beta_2 lag\_bookmarketRatio + \beta_3 \beta_4 er\_lag\_3 + \beta_5 er\_lag\_10 +$$
$$\beta_6 lag\_net\_equity + \beta_7 lag\_treasury\_bill + \beta_8 er\_lag\_4 + \beta_9 lag\_cbond\_return + \epsilon_t$$

(4.2)

The regression equation obtained for annual dataset is given as:

$$\hat{ER_t} = \beta_0 lag\_equity\_issuing + \beta_1 lag\_bookmarketRatio + \beta_2 lag\_long\_returnrate + \beta_3 er\_lag\_9, + \epsilon_t$$

(4.3)

These models are formed by combining a selected set of variables identified through forward stepwise selection. This implies that these variables significantly impact the model's performance based on the chosen selection criteria (AIC) in the stepwise procedure. Notably, in most models, the lag of series excess returns emerges as a key influence, playing a significant role in enhancing model performance. For the monthly model, out of the best subset of 10 variables, six are lag variables for excess returns (lags 1, 3, 5, 6, 8, 9). Similarly, for the quarterly model, lags 3, 4, 7, and 10 are included among the eight selected variables, and for the annual model, lag 9 is utilized. Besides lagged returns, other impact covariate variables include Book-to-Market Ratio, Corporate Bond Returns, Treasury Bills, Dividends over a 12-month period, Net Equity Expansion, Percentage Equity Issuing, and Long-Term Rate of Return. However, it's crucial to note that the selection of these variables doesn't imply causation or absolute importance in a real-world context. While forward stepwise selection identifies variables that enhance model fit based on specific criteria, their relevance should be interpreted cautiously.

4.1.3 Hyperparameter for regression tree and random forest

In machine learning, hyperparameters are values that control the learning process and influence the values of model parameters learned by a learning algorithm. They are the external configuration settings that must be specified before the training process and play a crucial role in optimizing the overall performance (MSFEs) of the model. In this project, we explore various combinations and evaluate the sets of hyperparameters that result in the smallest MSFEs on the validation set, optimizing our models for better predictive accuracy.

For the regression tree, the complexity parameter (cp) is a tuning parameter that controls the trade-off between the complexity of the tree and its goodness of fit to the training data. A larger cp leads to simpler trees, helping to prevent overfitting. We set the range of cp from 0.0001 to 0.1 with steps is 0.001 and the results are recorded for the best model that leads to the smallest MSFEs for both the historical model and for the general model and all frequencies in Table 3.

For the random forest, the hyperparameters are the number of decision trees ($ntree$) in the ensemble and the number of randomly chosen variables to consider for splitting ($mtry$) in a decision tree within the random forest is considering for the model. Increasing the number of trees and the number of splits generally improves model performance (MSFEs) but also increases computational complexity. We set the range for the number of trees from 50 to 500 with steps 1 and the split in each tree from 1 to 10, the results for both the historical and the general model and for all frequencies are recorded in Table 3.

Table 3: Hyperparameters for all machine learning models

	Historical model			General model		
Frequency	monthly	quarterly	annual	monthly	quarterly	annually
Regression Tree	cp=0.009	cp=0.1	cp=0.065	cp=0.009	cp=0.022	cp=0.01
Random Forest	ntree=162, mtry=3	ntree=110, mtry=2	ntree=60, mtry=3	ntree=320, mtry=2	ntree=120, mtry=2	ntree=65, mtry=3

For the purpose of minimizing Mean Squared Forecast Errors (MSFEs) on the validation set, optimal hyperparameters were configured and the outcomes are presented in Table 3. Notably, the monthly models, both historical and general, exhibit the smallest complexity parameter ($cp = 0.009$ compared to $cp = 0.1$ and $cp = 0.065$ for quarterly and monthly), indicating a higher level of model complexity capable of accommodating a larger number of variables and more extensive datasets. Conversely, quarterly and annual

models demonstrate higher *cp* values, suggesting reduced model complexity. Similarly, in the general model, where a larger set of predictors is considered, the complexity parameter is consistently smaller.

The random forest models follow a parallel trend, wherein more complex models, such as those handling monthly datasets or general models with a greater predictor set, tend to utilize a larger number of trees (*ntree*) and involve more extensive tree splitting (*mtry*) (with 162 tree for monthly dataset, 110 for quarterly and 60 for annual dataset, the larger number of tree also recorded for general model with the highest number of tree if 320 trees for the monthly general model). This adaptation allows the model to effectively learn from the training set, resulting in improved MSFEs performance on the validation set.

All the results for hyperparameter for all models on the training and validation set will be used for forecasting in the out-of-sample testing set of the last 10 years.

4.2   Model Performance

MSFEs were computed for each model across all frequencies for both the validation and testing sets, as detailed in Table 4 below.

Table 4: MSFEs results for all models at all frequencies

Dataset	Model	Type of Model	MSFEs		
			Monthly	**Quarterly**	**Annual**
Training Set	OLS	AR(p)	0.0549	0.1092	0.1964
		Stepwise	0.0539	0.1043	0.1784
	RT	Historical	0.0457	0.0917	0.1790
		General	0.0519	0.121	0.2138
	RF	Historical	**0.0456**	**0.0822**	**0.1671**
		General	0.0492	0.0903	0.2038
Testing Set	OLS	AR(p)	0.0441	0.0836	0.1753
		Stepwise	0.04749	0.07455	**0.3017**
	RT	Historical	0.04696	0.083646	0.17756
		General	**0.0377**	**0.0698**	0.1775
	RF	Historical	0.0466	0.08513	0.1753
		General	0.0407	0.0759	**0.1643**

4.2.1 Model performance on the validation set

Examining the MSFEs for the validation set, it is noteworthy that the historical model utilizing Random Forest consistently outperforms all other models at all frequencies with the smallest MSFE of 0.0456 in the monthly dataset (0.0822 for the quarterly and 0.1671 for the annual dataset). Surprisingly, it even outshines the more complex general model, which includes all predictors. Additionally, for the validation set, the general Random Forest model, containing all predictors, consistently outperforms all other models. This superior performance may be attributed to the robustness and adaptability of the Random Forest algorithm in capturing complex patterns within the data. In contrast, the AR(3) model for the monthly dataset performs the worst in the validation set (with MSFE of 0.0549), and for quarterly and annual frequencies, the regression tree model with all covariates has the worst performance in the validation set compared to other models with MSFEs of 0.121 and 0.2138 respectively. The poor performance of the AR(3) model might be due to its limited ability to capture the complexity of the large dataset of monthly.

4.2.2 Model performance on the testing set

Turning to the MSFEs calculated for the 10-year testing set, it is surprising to observe that the Regression Tree (RT) model outperforms the Random Forest (RF) model, achieving the smallest MSFEs for both monthly (with only 0.0377) and quarterly frequencies (with 0.0698) and presenting the overall best MSFE, even better than MSFEs of all models on the validation set. This unexpected result suggests that, in certain scenarios, the simpler structure of the RT model may yield more accurate predictions than the RF model. RTs are capable of capturing interactions between predictors, which can be crucial in modeling real-world phenomena. However, for yearly frequencies with fewer possible data points, the RF model continues to perform better than other models, showcasing its efficacy in handling longer-term trends and dealing with fewer data situations.

Remarkably, on the testing set, the historical autoregressive (AR) model outshines all other historical models, demonstrating superior out-of-sample forecasting. This is particularly evident in the monthly, quarterly, and yearly datasets, where AR models with specific lags exhibit notable performance. For the general model, as mentioned earlier, the RT model excels in monthly and quarterly datasets, while the RF model demonstrates superior performance in annual data. This pattern underscores the importance of model selection based on the specific characteristics and temporal resolutions of the data at

hand. In contrast, the stepwise model has the worst performance in the out-of-sample testing set and is close to the worst in the validation set. This emphasizes the limitations of the stepwise model in real-world scenarios, possibly due to its inability to consider interactions between predictors effectively.

4.2.3 Overal results

In general, the smallest MSFE is recorded as 0.0377 for the regression tree model with the general model that combines all predictors. Conversely, the largest MSFE is recorded as 0.3017 for the forward stepwise model, which selects the best predictors from among all 23 predictors, specifically for the annual dataset. For all the models in all frequencies, the model using the monthly dataset usually achieves the smallest MSFEs in both the validation set and testing set due to the abundance of data available in monthly from January 1927 to December 2012, facilitating more effective model training. This is in contrast to the small dataset with just 95 data points in total for the annual dataset. For only the lag model scenarios, the AR(p) model has the worst performance in the validation set but has the best performance in the out-of-sample testing set. The discrepancy might be attributed to the specific lag structures captured by the AR(p) model, which may align well with the underlying patterns in the testing set. In contrast, with all covariates included, the traditional model is outperformed by two machine learning methods (RT and RF), suggesting that the complexity and adaptability of these two machine learning approaches contribute to their superior performance in capturing non-linear relationships within the data. This flexibility is crucial when dealing with real-world datasets that may exhibit complex, non-linear patterns. The results emphasize the importance of thoughtful model selection, considering complexity and adaptability for optimal predictive performance.

4.2.4 Compare with the previous projects

To assess the effectiveness of out-of-sample forecasting and one step ahead rolling window forecast, we compared the MSFEs across all models and frequencies in the testing dataset with the in-sample testing evaluation. Specifically, we also conducted forecasts for Regression Tree (RT) and Random Forest (RF) models on quarterly and annual data, and the results are detailed in Table 5.

The findings indicate that, in general, rolling forecasts in the testing set yield superior results compared to the train-test split in-sample forecasting, particularly in the monthly

dataset and for all OLS models, for RT and RF for all covariates, the historical models using monthly data for RT and RF exhibit slightly better performance with the train-test split method. The RT model with all covariates stands out with the smallest MSFE, recorded as 0.03771. However, for annual rolling forecasts, only the stepwise model and RF all-model scenario show worse performance compared to the train-test split and in-sample forecasting. Remarkably, the stepwise model in rolling forecasts performs the worst overall, with the highest MSFE reaching 0.301731.

In the quarterly dataset, the train-test split method appears to outperform the rolling forecast in most models, displaying smaller MSFEs. This trend holds for almost all models, except for RT all and Autoregressive (AR) models.

In the train test split and in-sample forecast projects, RF models for lag and RF for all demonstrated the smallest MSFEs compared to other models, except for the annual dataset where the stepwise model slightly outperformed the RF model with a very close MSFE. However, in the rolling forecast analysis, machine learning models outperformed traditional OLS models across all frequencies. Notably, the RT model with all covariates achieved the smallest MSFEs, recording 0.03771 for the monthly dataset and 0.06987 for the quarterly dataset. For the annual dataset, the RF model with all covariates exhibited the best performance, yielding an MSFE of 0.1643.

Table 5: Compare MSFEs results for all models at all frequencies to the previous projects

Model		Rolling forecast out of sample			Train test splits- previous		
		Monthly	Quarterly	Annual	Monthly	Quarterly	Annual
OLS	AR(p)	**0.0441**	0.0836	0.1753	0.0471	0.0939	0.1825
	Stepwise	**0.04749**	0.07455	**0.3017**	0.0537	0.0695	0.1445
RT	Historical	0.04696	0.083646	0.17756	**0.0456**	0.082	0.264
	General	**0.0377**	**0.0698**	0.1775	0.04538	0.081	0.190
RF	Historical	0.0466	0.08513	0.1753	**0.0439**	0.078	0.178
	General	**0.0407**	0.0759	**0.1643**	0.04536	0.060	0.148

4.3 Diebold-Mariano test result

In addition to the MSFE results, we also conducted Diebold-Mariano (DM) tests across various models at different frequencies, aiming to assess the forecast accuracy of models in comparison with others. All models with the same frequency will be compared pairwise

and here we utilized the squared loss function. The outcome is visually depicted through a heat map in Figure 7 for monthly, quarterly and annual datasets. With the null hypothesis (H_0): There is no significant difference in forecasting accuracy between the two models being compared. These maps illustrate the p-values associated with the null hypothesis of the forecasts. The low p-values (green color) show statistical significance ($p-values < 0.05$) that the model along the y-axis is significantly more accurate than the forecast along the x-axis. On the other hand, shades closer to red (p-values approaching 0.1) suggest there is no evidence against the null hypothesis.

In Figure 7 (a), the results of the DM test for all models on the monthly dataset reveal that both the general Regression Tree model (RT) and Random Forest with all covariates exhibit greater accuracy compared to the Regression Tree and Random Forest models using only lagged of excess returns. The traditional OLS models, including AR(3) and the Forward stepwise model, show slightly superior performance to the Random Forest historical model, and also the Forward stepwise model proves the more accurate than the RT historical model. While the MSFE results favor the AR(3) model over the Forward stepwise model, the DM test provides no evidence supporting the superiority of AR(3) over the Stepwise model.

Moving on to the quarterly dataset in Figure 7 (b), the Random Forest general model, incorporating all covariates, outperforms the RF historical model relying solely on the lagged combination of excess returns. Additionally, the Stepwise model demonstrates a slight advantage over both the RT and RT historical models. This underscores the importance of extending models beyond historical data solely based on excess returns. No clear evidence emerges regarding whether RT is superior to the RF model or vice versa in forecasting quarterly excess returns.

For the annual dataset in Figure 7 (c), the RF model with all covariates exhibits slightly better performance than the RF historical model. No conclusive evidence emerges regarding the comparison between machine learning models and OLS models.

In summary, across various cases, models incorporating all covariates consistently outperform historical models relying solely on lagged excess returns as predictors.
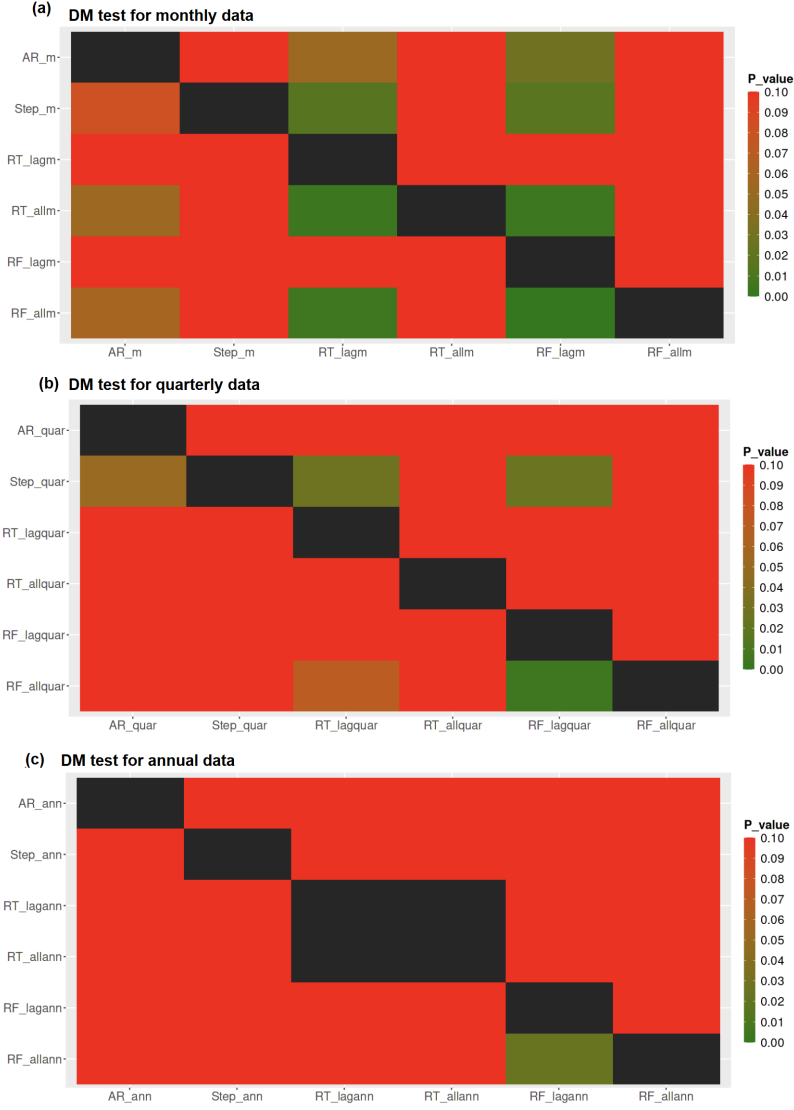
Figure. 7.

Figure 7: Results from DM test

4.4 Rolling forecast visualization

Figure 8 presents the predicted values on the testing dataset for select models based on their Mean Squared Forecast Errors (MSFEs) and Diebold-Mariano (DM) test results. The highlighted models for visualization include the Regression Tree (RT) general models, characterized by the inclusion of all covariates. These models exhibit superior performance in both monthly and quarterly frequencies compared to all other models. Additionally, we explore their performance in the annual dataset for comparative analysis. To provide a contrast, we also examine the Random Forest (RF) general model's forecast in the annual dataset, recognized as the top performer in this context. Finally, we consider the Forward Stepwise model, which, despite its overall weaker performance with the highest MSFE, is included for comprehensive comparison.

In Figure 8, it is evident that the forecast generated by the RT general model for
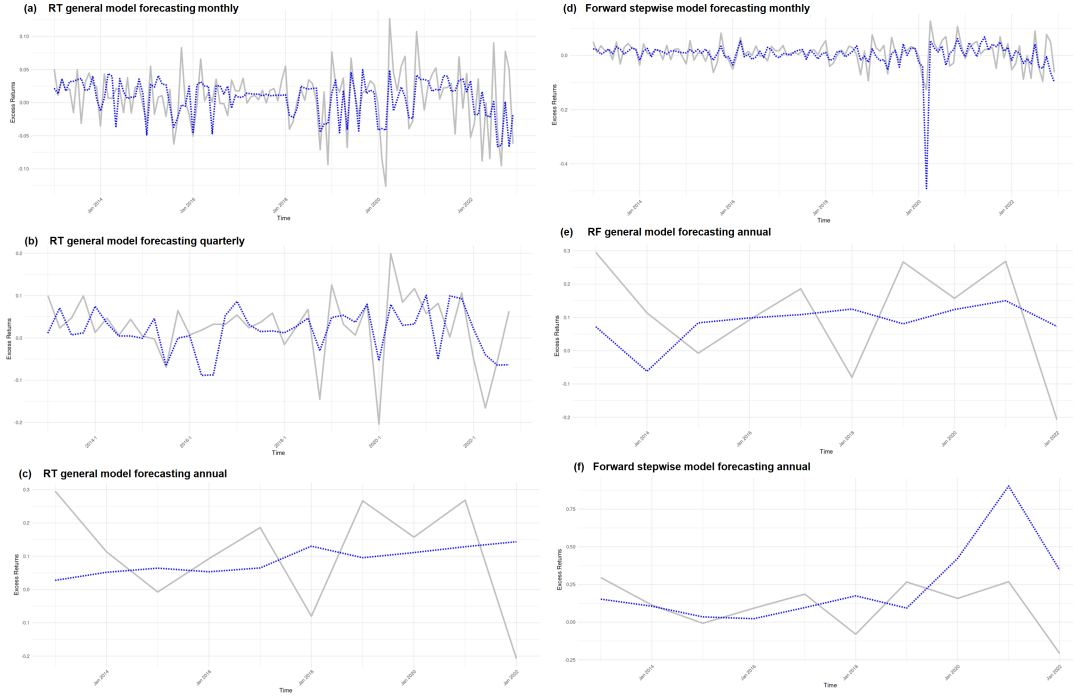
Figure 8: Forecast visualization

monthly and quarterly frequencies (Figure 8 (a), (b)) aligns well with the actual data, effectively capturing the dynamics of the time series excess return. However, in the annual dataset, where only 95 data points are available compared to the 1152 data points in the monthly frequency, the RT general model appears less fitting, seemingly reflecting an average across the entire period (Figure 8 (c)). This trend is also noticeable in the RF general model's forecast (Figure 8 (e)), where the model does not fit as well but tends to average the data. Nevertheless, the RF model demonstrates a better fit for annual data in the initial and final periods, closely resembling the actual data. This improved fit contributes to its smaller MSFE compared to the RT general model.

Conversely, the Forward Stepwise model appears to overfit, performing well in in-sample forecasting, resulting in the smallest MSFEs (Table 5) in the monthly dataset. However, its out-of-sample forecast and visualization (Figure 8 (d) for monthly frequency and Figure 8 (f) for annual frequency) reveal challenges. Although it generally fits well or even outperforms the RT general model, it exhibits extremes during peaks, surpassing or falling below the actual values. This behavior contributes to a higher MSFE for the Forward Stepwise model.

25

5 Conclusion

In this extensive exploration of excess stock return forecasting, our comprehensive comparison between historical models, leveraging lagged excess returns, and general models incorporating all covariates as predictors has provided valuable insights. Through the application of three primary methods—Ordinary Least Squares (OLS) regression, Regression Tree, and Random Forest—we delved into the nuanced dynamics of each approach, engaging in meticulous hyperparameter tuning guided by information criteria to optimize forecasting accuracy. The Mean Squared Forecast Error (MSFE) emerged as a robust metric for evaluating forecast quality, revealing that, on the validation set, the historical Random Forest model consistently outperformed, demonstrating superior performance across frequencies, even surpassing the more complex general model. However, it is crucial to acknowledge the limitations observed in our study. The Forward Stepwise model, while robust in in-sample forecasting, exhibited poor performance on the testing set, suggesting susceptibility to overfitting. This highlights a potential area for improvement, emphasizing the need for enhanced techniques to address overfitting issues and enhance model generalization.

Our MSFE results on the testing set underscored the general Regression Tree model's superiority, particularly in monthly and quarterly frequencies. Notably, machine learning methods consistently outperformed traditional OLS models, showcasing their effectiveness, especially in rolling out-of-sample forecasts. The Diebold-Mariano Test reinforced the overall superiority of the general machine learning models (models with all covariates) over OLS and their historical counterparts.

Looking forward, the study opens avenues for future research. Further investigation into methods to mitigate overfitting, exploring advanced machine learning techniques, and incorporating real-time data could enhance forecasting accuracy. Additionally, assessing the models' performance under varying economic conditions and market dynamics would provide a more comprehensive understanding of their robustness.

In conclusion, our study not only reveals effective strategies for accurately predicting excess stock returns across various frequencies but also recognizes the nuanced strengths and limitations of each modeling approach. As we look towards the future, ongoing refinement and innovation in forecasting methodologies will continue to be essential in navigating the complexities of financial markets.

Bibliography

Ken Aho, DeWayne Derryberry, and Teri Peterson. *Model selection for ecologists: the worldviews of AIC and BIC.* Ecological Society of America, 2014.

Nagesh Singh Chauhan. *Decision Tree Algorithm, Explained*, 2022.

James D Hamilton. *Time Series Analysis.* Princeton University Press, Princeton, New Jersey, 1994.

James D Hamilton. *An introduction to statistical learning with applications in R.* Springer, 2013.

Klaus Ackermann Christoph Bergmeir Hansika Hewamalage. *Forecast evaluation for data scientists: common pitfalls and best practices*, 2023.

Marc Hofmann. *ggplot2: Elegant Graphics for Data Analysis.* CRAN, 2022. URL `https://github.com/marc-hofmann/lmSubsets.R`.

Wei-Yin Loh. *Classification and regression trees*, 2011.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2021. URL `https://www.R-project.org/`.

Ashwin Raj. *A Quick Guide to Random Forest Regression*, 2020.

Ivo Welch and Amit Goyal. *A Comprehensive Look at The Empirical Performance of Equity Premium Prediction*, 2008.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. *dplyr: A Grammar of Data Manipulation*, 2022. URL `https://CRAN.R-project.org/package=dplyr`. R package version 1.0.9.
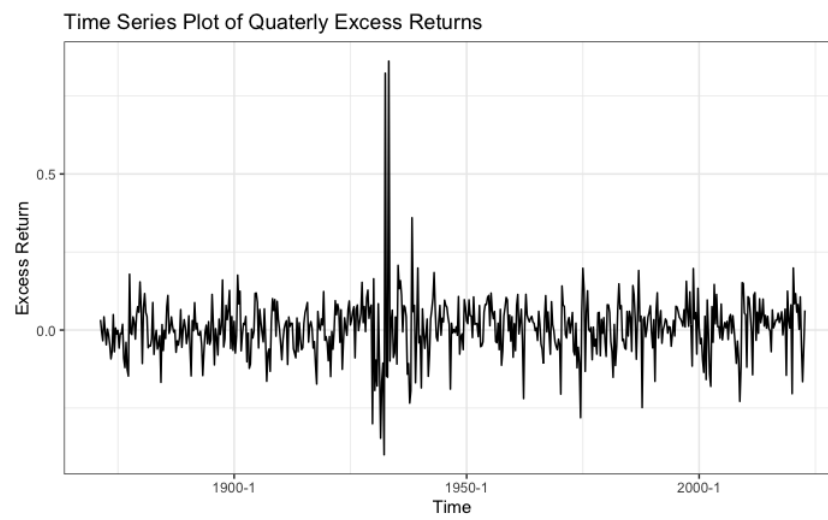
Appendix
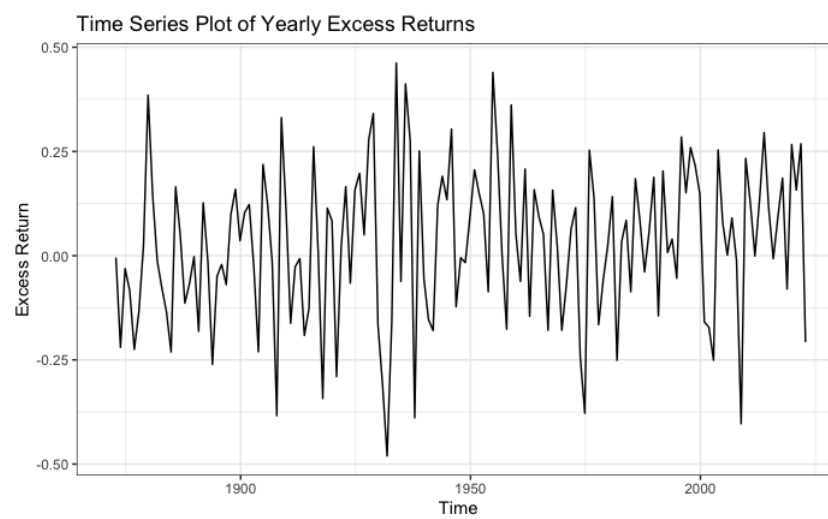

Figure 9: Plot of time series excess returns
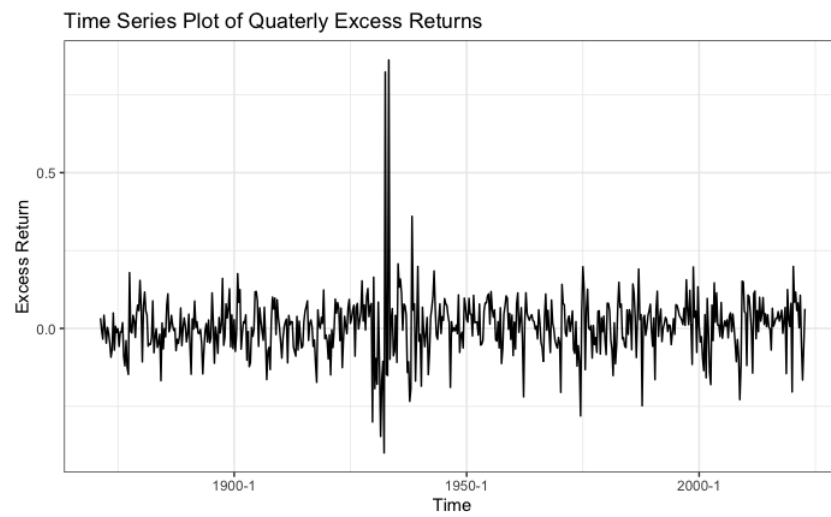

Figure 10: Plot of time series excess returns

Figure 11: Plot of time series excess returns