Forecasting Revenue Using Supervised Learning on Quarterly Fundamental Variables

STAT 406 Project Report
December 19, 2022

Thiago da Cunha Vasco, Qichen Huang, Victor Zamora

Introduction

One of the many duties of a stock analyst is to forecast revenue growth to project what future earnings will look like, as future earnings are indicative of how a stock should be priced. Stock analysts use a variety of quantitative and qualitative information in order to forecast future revenue growth, such as consumer demand, industry and economic trends, market competition and fundamental variables found in quarterly and annually-released financial statements. In this report, we aim to show how supervised learning models trained on quarterly financial statements can be used to forecast revenue. Through our experiments we also attempt to answer questions such as

1. How do linear statistical learning methods compare to non-linear methods when forecasting revenue using past financial statements as predictors?
2. Does the forecasting performance of a model depend on the size of the firm?
3. What variables are most important when attempting to accurately forecast future revenues?

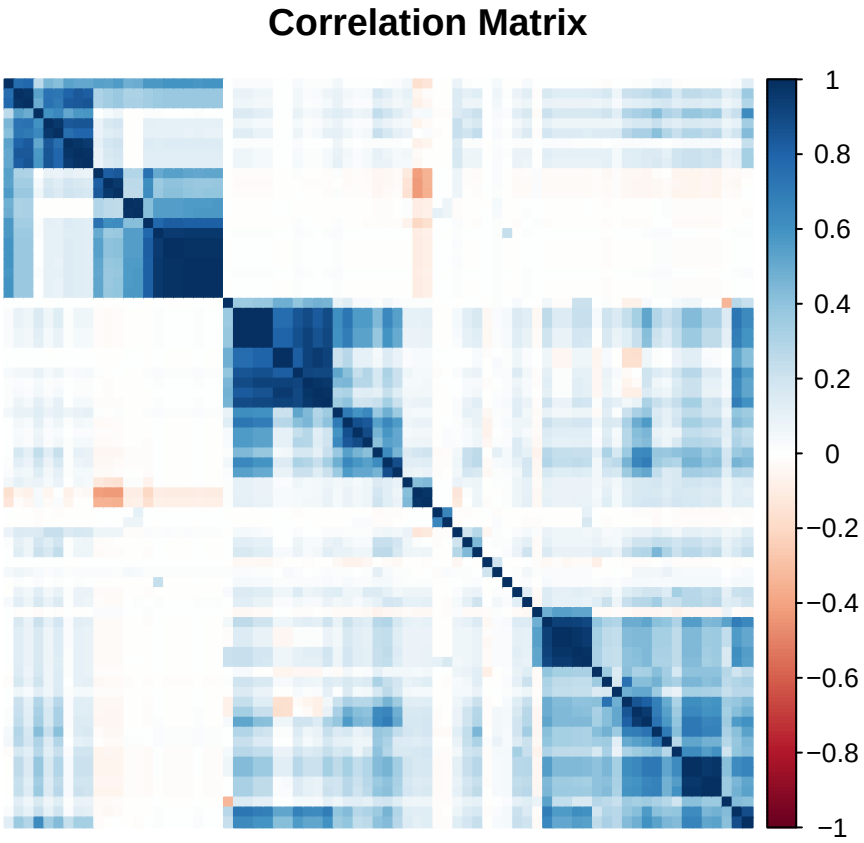
In our experiments, we use data from the North American Fundamental Quarterly (FUNDQ) dataset provided by the Wharton Research Data Service (WRDS). Our dataset contains 157613 observations of 73 variables after preprocessing, and provides information on 2736 North American firms' quarterly financial statements (i.e., balance sheets, income statements, and cash flow statements). The data that we use come from quarters 1990Q1 to 2021Q2 inclusive. Note that a "quarter" in this report is a period of time that is common to all firms, where Q1 is defined as February, March, and April, Q2 is defined as May, June, and July, Q3 is defined as August, September, and October, and Q4 is defined at November, December, and January. This is contrast to a "quarter" of any particular firm (i.e., Apple's Q1 which starts in late September). Our experiment will follow a "kitchen sink" design which means we will use all 73 variables within the dataset as explanatory variables, and let the learning algorithms themselves choose which variables to pay attention to and which to ignore.

This report will begin by first analyzing some characteristics about the dataset, and how to go about preprocessing and restructuring the data so that it is usable within a supervised learning setting. We then go on to discuss the models, metrics, and approaches to evaluating out-of-sample performance that will be used when conducting experiments. Lastly, we discuss our results and analyze the measures of feature importance from each model, and connect our findings to the questions we initially set out answer.

EDA

Within the FUNDQ dataset there are 73 variables which we will use to train our regression model. Many of these variables are likely to be highly colinear with one another (e.g., sales and revenue). Below we plot a correlation matrix to show the extent to which colinearity exists within our data.

```
## corrplot 0.90 loaded
```



Note how there are dozens of moderate-to-highly correlated variables as shown along the diagonal. The reason this is the case is because many variables found within financial statements are an exact linear combination of other variables. For example, think of how total assets is equal to the sum of all assets, or how working capital is the difference between current assets and current liabilities. These relationships lead to almost perfect colinearity, which is why it is a good idea to use a regression method which can be easily regularized. For this reason, we avoid OLS.

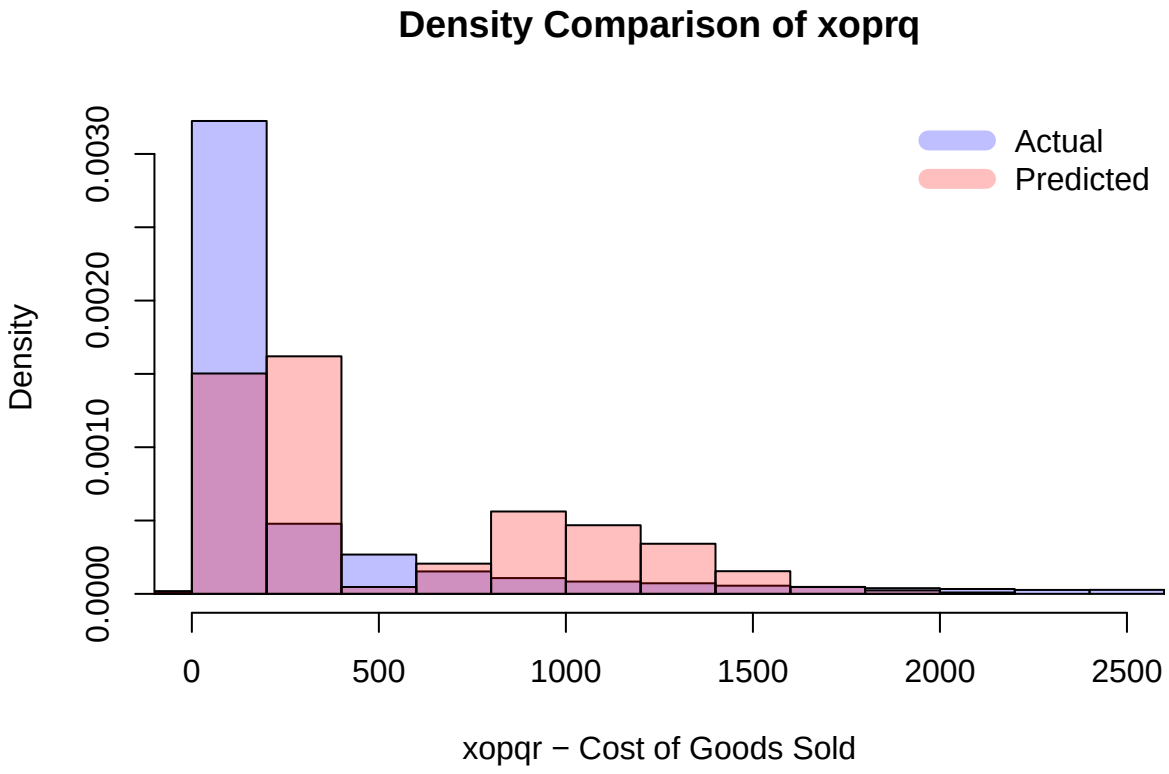
Preprocessing

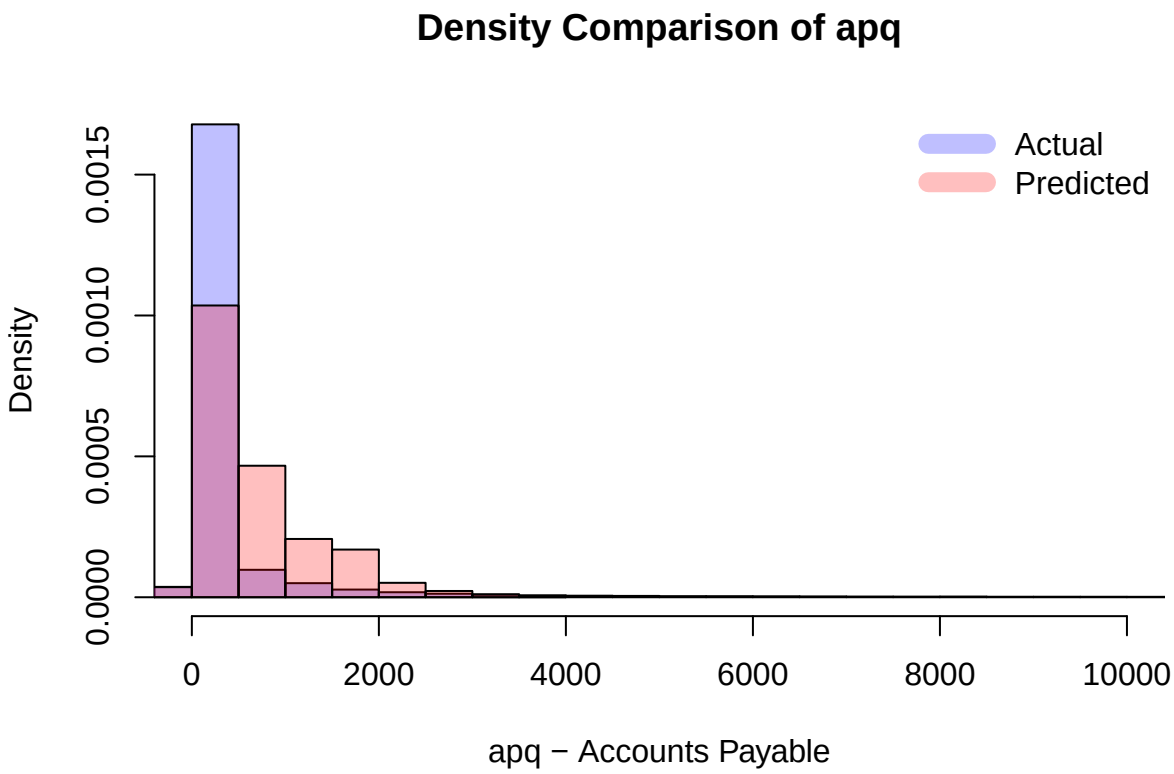
Our data contains 958819 missing values, which comprises just over 4% of the total number of values within our data. In order to deal with missing data, we remove all columns and subsequently all rows which have half or more of its values equal to `NA`. We also remove any observation in which `atq` (total assets), `revtq` (revenue), or `datacqtr` (quarter data is released) are equal to `NA`. However, after having done so we are still left with 738753 missing values. In order to deal with this, we use the following imputation algorithm in order to impute missing values while circumventing the introduction of look-ahead bias into training.

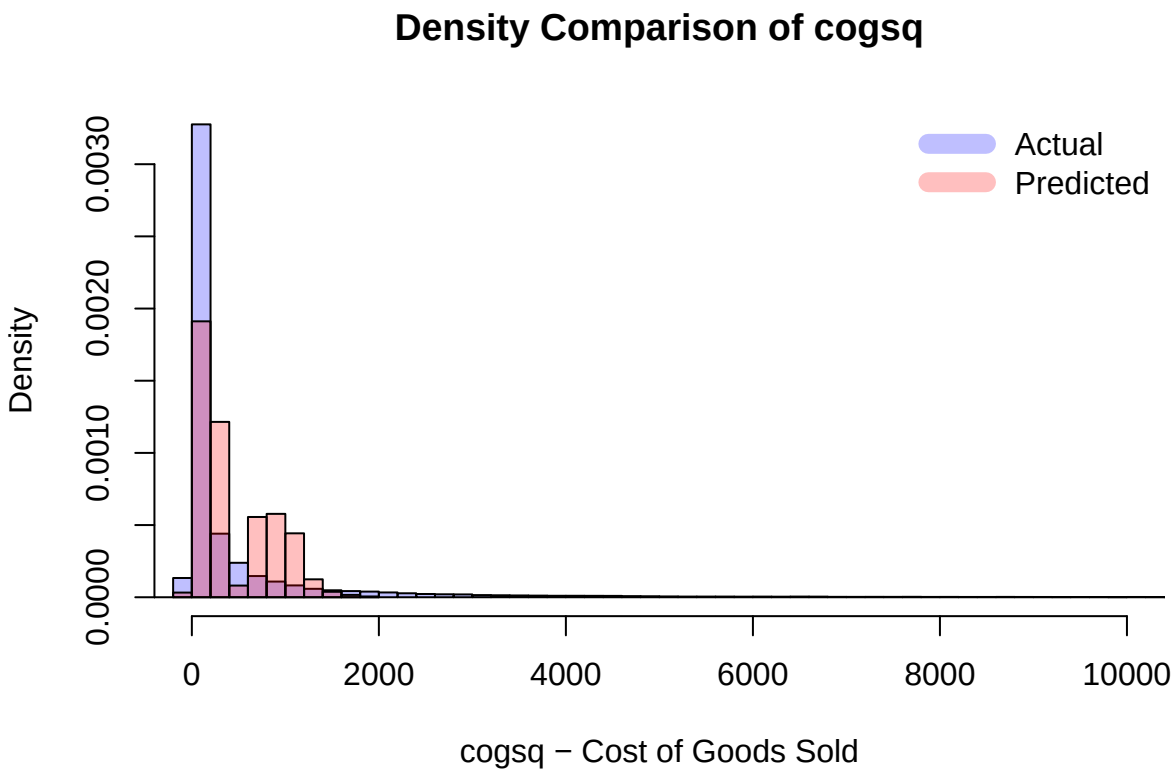
1. Construct a matrix that amalgamates all observations from 1990Q1. Carry out the matrix completion.
2. Construct a new matrix which bind the previous, fully-imputed matrix, with a matrix that amalgamates all observations from the following quarter (e.g., 1990Q2) which still contains missing values. Carry out matrix completion.
3. Repeat step 2 until there are no more quarters left.

This algorithm results in missing values from quarter Q to be predicted using information from quarters $Q-1, Q-2, \cdots$ back until the earliest quarter present within the dataset (1990Q1). We use the `eimpute` (Efficiently IMPUTE Large Scale Incomplete Matrix) package to carry out matrix completion within our

imputation algorithm. We chose this package in particular as its interface is easy to work with, and it is efficient at carrying out large scale matrix imputation. We display the density plots below as a way to qualitatively deduce whether or not the imputation algorithm does a good job of simulating missing values. This is done by comparing the density of the true values with the density of the values predicted by the imputation algorithm for any given variable. Since `revtq`, `saleq`, and `atq` do not have any missing values, we showcase the density comparison plots for the variables `xoprq`, `apq`, and `cogsq` as these are most likely to be the next most important variables. Ideally, we want the two densities within every plot to be similar.




Density Comparison of apq
Actual
Predicted
Density
0.0000
0.0005
0.0010
0.0015
0
2000
4000
6000
8000
10000
apq − Accounts Payable



We can see how on each histogram, there tends to be a high point on the left of the graph with a rightwards skew on both densities. At least from a glance, our plots suggest that our imputation method is effective at imputing missing values in such a way that their distribution resembles that of the values we have present in the dataset.

Results & Analysis

In this report, we attempt to estimate a function $f : (\mathbf{x}_{Q-3}, \mathbf{x}_{Q-2}, \mathbf{x}_{Q-1}) \rightarrow y_Q$, where $Q \in \mathbf{N}$ is any given quarter between 1987Q1 to 2021Q2, $\mathbf{x}_{Q-n}$ is a vector of financial statement variables released n quarters before quarter Q, and y_Q is the revenue of the firm realized at the end of quarter Q. We use two supervised learning approaches to estimate f.

Models

The way we tackle the supervised learning problem of estimating f is by using two models: lasso and random regression forests.

Least absolute shrinkage and selection operator (Lasso)

Lasso solves the minimization problem

$$min_{\beta \in \mathbf{R}^p} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - X\beta)^2 + \lambda ||\beta||_1 \right\}$$

with hyperparameter $\lambda \geq 0$. We choose lasso to be one of our models for a couple of reasons. Firstly, optimizing MSE with L1 regularization acts as a sort of automatic feature selection algorithm which will become useful to us when analyzing feature importance. L1 regularization is also necessary in this scenario as we are working with data that contains high colinearity (see the section on EDA). Lastly, lasso is a linear model which we can use to test if forecasting can be effectively accomplished while disregarding nonlinear relationships. We use the `glmnet` package in order to implement lasso as it provides an easy interface for us to use.

Random Regression Forest

Random Forest is an algorithm which bootstraps low-bias, high variance decision trees and averages them in order to reduce the variance of the ensemble as a whole. The variance is further reduced by decreasing the pairwise correlation between each two trees; this is done by only allowing a subset of all variables to be tried on each split – forcing trees to look significantly different from one another. The `grf` package which we use in our experiments sets `mtry` equal to $\sqrt{p} + 20$ as default. We use this package as it provide an implementation of random regression forest that is significantly faster than the one found in the `randomForest` package. We choose random forest as it can fit nonlinear relationships within a reasonable amount of time, and it is also robust against overfitting. This is in contrast to neural networks, where training can be quite tedious and often unstable.

Evalutation Metrics

We use several metrics to evaluate how our two models perform on out-of-sample data. We first use $MAE = mean(|y_j - \hat{y}_j|)$ as it is commonly used and easy to interpret. In unison with MAE, we also use $RMSE = \sqrt{mean((y_j - \hat{y}_j)^2)}$ as a metric as it allows us to observe which models are more likely to make large errors.

We also utilize mean absolute scaled error, or MASE. We have that $MASE = mean(|q_j|)$, where

$$q_j = \frac{y_j - \hat{y}_j}{\frac{1}{T-1}\sum_{t=2}^{T} |y_t - y_{t-1}|}.$$

This metric shows to what extent our "learned" model outperforms a naive forecasting strategy, in which $\hat{y_Q} = y_{Q-1}$. If $MASE < 1$, our learned model is more effective than the naive strategy, or else not. Aside from being intuitive, another property of this metric is that it is unit free, which allows us to evaluate how our models perform on firms of different sizes. In contrast, if we were to use MSE or MAE for this task, we would find that the errors will almost certainly be lower for forecasts made on smaller firms as they tend to have significantly lower revenues.

Lastly, we employ a metric called PC (Percentage Correct). This metric simply tells us what proportion of the time our model makes a forecast "in the correct direction". We have that $PC = mean(I\{sign(y_t - y_{t-1}) = sign(\hat{y}_t - y_{t-1})\})$, where I is the indicator function. We use this metric as we consider a model which often predicts the direction of revenue growth correctly to be a good model. Note that this metric suffers the same shortcoming as classification accuracy does; if firms are significantly more likely to experience revenue growth than not, then naively predicting positive revenue growth for all observations will grant us a desirable PC score. Thus, this metric should not be used as an absolute measure of a model's ability to detect shifts in direction, but it can be used to compare different models' ability to do so.

Time Series Cross Validation

In order to test our model on historical out-of-sample data, we utilize the time series cross-validation algorithm suggested by Hyndman and Athanasopoulos. The algorithm consists of an expanding window on which our model is trained, to then be tested on a future out-of-sample datapoints. For example, we first train our models on all observations corresponding to all quarters in between 1990Q1 and 2010Q4 inclusive. Then,

we test our model on all observations corresponding to quarter 2011Q1, save these out-of-sample errors, and then include these observations as part of our training data for when evaluating our model on observations corresponding to quarter 2011Q2. This process is repeated until we retrieve the forecast errors from 2021Q2 (the last quarter). The errors collected across all quarters are then used to compute the metrics discussed earlier.

Results

Below we see the out-of-sample performance of our two models. By running our cross validation algorithm on different values of lambda in `seq(1, 150, 0.5)`, we found that setting $\lambda = 57$ resulted in the lowest cross validation score. The lasso scores presented below come from this particular choice of λ. For the random forest model we let `mtry` be equal to its default value, and set the number of trees to be equal to 200.

Model	*MAE*	*RMSE*	*MASE*	*PC*
Random Forest	272.5	1490.7	0.959	0.60
Lasso	286.5	1487.6	1.008	0.55

On all metrics except RMSE, we see that random forest outperforms lasso. In fact, lasso on average had larger absolute errors than the naive forecasting method of always predicting the last value of the time series (MASE > 1.008). This suggests that the relationship between financial statement variables from past quarters and future revenues may be fairly nonlinear. Another thing to note is that random forest regression was capable of correctly guessing the direction of revenue growth 60% of the time, whereas Lasso was only capable of doing so 55% of the time. Why LASSO scores slightly lower on RMSE than random forest despite its underperformance on all of the other metrics may appear to be a mystery at first. However, a clue is given by analyzing how our models perform on different firm sizes. To do this, we partition all out-of-sample observations in our cross validation algorithm into three groups: small, medium and large-sized firms. All observations where revenue is lower than the 33% quantile is deemed to be small. Likewise, observations with revenue in between the 33% quantile and the 66% quantile are deemed to be medium-sized, and observations with revenues larger than the 66% quantile are deemed to be large. We then calculate the metrics for each of these three groups separately. Below, we display MASE and PC by quantile group for random forest and lasso respectively. MSE and MAE are excluded as they are not invariant to firm size.

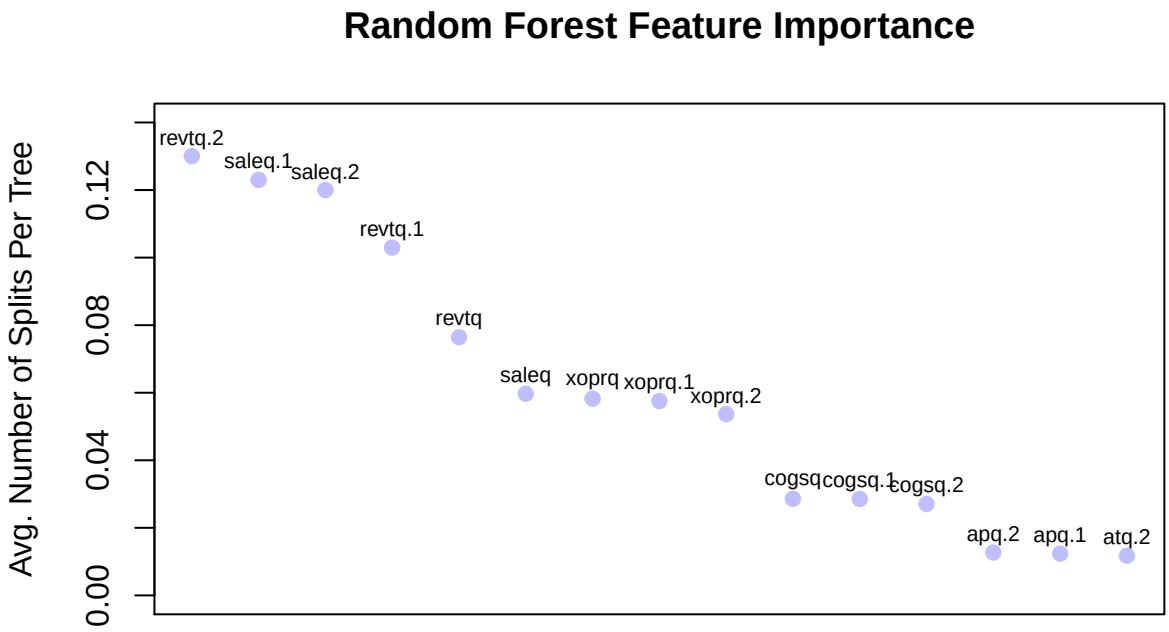
	Quantile	≤ 0.33	$(0.33, 0.66)$	≥ 0.66
Random Forest	*MASE*	0.879	0.920	0.961
	PC	0.554	0.620	0.626
Lasso	*MASE*	2.303	1.068	0.974
	PC	0.485	0.592	0.574

We see that on data deemed to be from "small" firms, random forest has a significantly lower MASE score of 0.879 in comparison with that of lasso, which scores 2.303. As we go up in the quantile groups, we see that the MASE score for both random forest and lasso begin to look similar. We see that on data deemed to be from "large" firms, random forest has a higher MASE score of 0.961, while lasso scores 0.974. These MASE scores suggests that a possible explanation for why lasso has lower out-of-sample MSE than random forest, is because although lasso performs terribly on firms with smaller revenues, it may outperform random forest on firms with the very largest revenues. This outperformance leads to a decrease in MSE that more than makes up for lasso's poor performance on small firms.

It is likely that the relationship between predictors and response are different for firms of different sizes. Random forest, being able to model highly nonlinear relationships, is more capable of taking this into account than lasso which is constrained to being linear. Since lasso is trying to minimize MSE during training while not being flexible enough to model both large and small firms, it simply models larger firms as that provides the greatest reduction in MSE. This would explain the fact that lasso performs terribly on smaller firms, and performs comparably to random forest on firms with larger revenues.

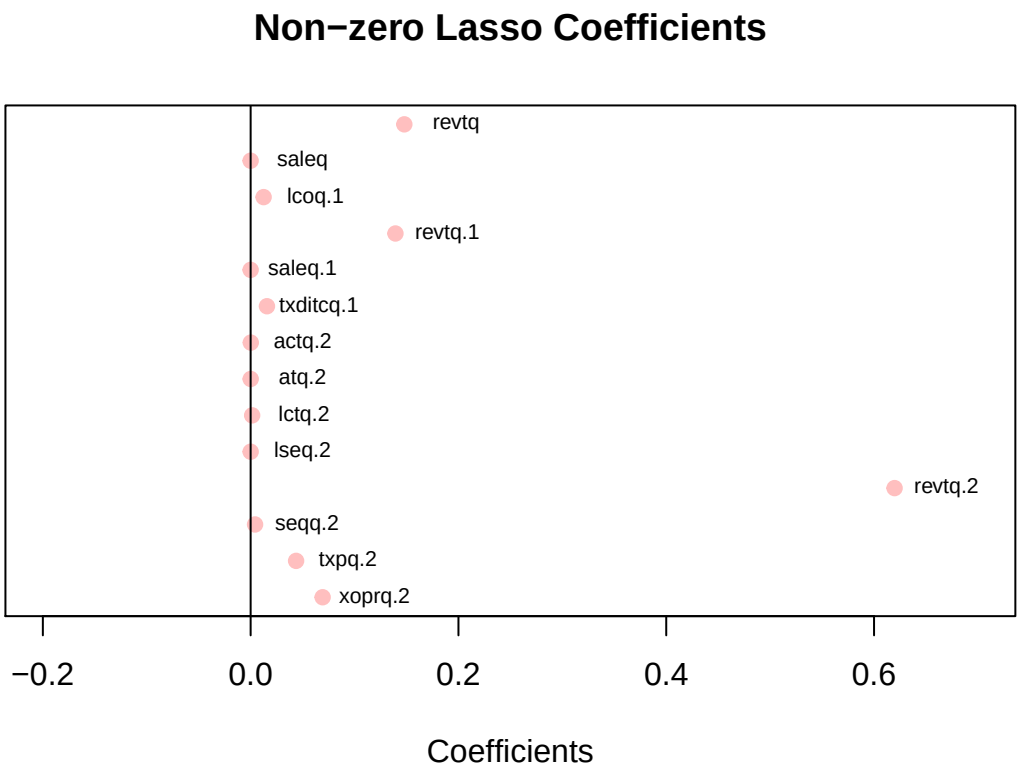
Feature Importance

After obtaining the out-of-sample forecasts from both our models, we now train both models on the entire dataset in order to analyze which features most greatly contributed to each model's performance. Below we show random forest's importance measure for 15 variables, where each variable's score is equal to the average number of times a variables would be split on within each tree. Below are the 15 "most important" features. Note that variables may have ".1" or ".2" appended to their names. This implies that if we are trying to forecast y_Q, those variables were made available during quarters $Q-2$ and $Q-1$ respectively.



As expected, revenue and sales of the most recent periods make the strongest predictor of future revenues. Operating expense (`xoprq`), cost of goods sold (`coqsq`), accounts payable (`apq`) and total assets (`atq`) make up the next strongest predictors.

Below we show the coefficients of the lasso model trained on the entire dataset, while once again setting $\lambda = 57$. This is a proxy for feature importance as lasso sets relatively unimportant features to 0.



Above we see some variables in common with those included in the top 15 most important features in the random forest model: namely, revenue (`revtq`), sales (`saleq`), total assets (`atq`), and operating expenses (`xoprq`). Past revenue is perhaps the most obvious feature to take into account when forecasting future revenue. Sales however, is curiously also included into the calculations. It may be of interest to the model what proportion of revenue derives from selling goods and services to consumers, as opposed to other less stable sources of revenue such as litigation, interest, loyalties and fees. For example, if a company had won a litigation case and was rewarded a significant amount of money, this may artificially bump up the firm's revenue for that quarter. This significant increase in revenue growth however is unlikely to continue onto the next quarter. Total assets is included, perhaps since firms with more assets are more likely to draw in large revenues. We also concluded previously that firm size may significantly change the relationship between predictors and response, so it may be important for the model to keep track of this variable. Lastly, operating expense is a relatively important variable used by both models. Operating expenses include rent, equipment, inventory cost, and funds allocated for R&D. Thus, a firm that has increasing operating expenses is likely to be building up its inventory, renting out locations to sell product, and investing in R&D – all things which are likely to increase future revenues.

Discussion

In this report, we have shown that supervised learning models trained on quarterly financial statements can be used to forecast revenues. This conclusion is supported by the fact that random forest can achieve MASE scores of 0.879 on small firms with minimal tuning and feature preprocessing. Furthermore, we have found that random forest outperforms lasso when forecasting revenues of small-to-medium-sized firms. This can be seen from the second table on page 7, where random forest achieves a MASE score of 0.879 and 0.920 on firms below the 33% quantile and in between the 33% to 66% quantiles respectively, whereas lasso would achieve MASE scores of 2.303 and 1.068 on the same observations. However, we also find that for firms

with revenues above the 66% quantile, lasso is comparable to random forests as both models share similar MASE (0.974 vs 0.961) and RMSE (1487.6 vs 1490.7) scores. We also hypothesize that the reason why lasso performs so poorly on firms with low revenues, is because lasso is optimized using MSE, which punishes large errors more harshly. Thus, because the models are likely to make larger errors when forecasting higher revenues, lasso pays more attention to fitting observations from larger firms rather than smaller ones. This issue may be circumvented by dividing all variables by some constant which represents the relative size of a firm (e.g., dividing by total assets), or by using weighted least squares with L1 regularization, where smaller firms are weighted more heavily to make up for the smaller squared errors. Random forest avoids this issue altogether due to its ability to fit nonlinear relationships. Lastly, we analyze the drivers of forecasting performance, and conclude that aside from revenue itself, sales, total assets, and operating expenses appear to be the most important variables for when forecasting revenue. Sales may be important as it allows models to detect whether or not past increases in revenue is made up of an increase in selling goods and services to consumers, or if revenue growth was fueled by factors unlikely to fuel growth in the future (e.g., revenues from litigation). Total assets may be important as firms that are expanding their assets are likely to experience revenue growth in the nearby future. Finally, increasing operating expenses are an indicator that a firm is spening money on inventory, rent, equipment, and R&D – all of which are likely to result in an increase in revenues in the following quarter. Our results do come with limitations however. We used under 150,000 observations due to lack of computing power, which can be seen as relatively small in comparison to other research that has used close to 10^6 observations from the same FUNDQ dataset. Since increasing the size of the data can lead to great boosts in performance, we acknowledge that the performance of our models observed within our experiments can likely be improved upon by using the entirety of the FUNDQ dataset, although doing so is expensive.

References

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput. Stat. Data Anal. 120, 70–83.

Hyndman, R. J. & Athanasopoulos, G. (2014). Forecasting: Principles and Practice. On Demand Publishing. LLC-Create Space

Amel-Zadeh, A., Calliess, J.-P., Kaiser, D., & Roberts, S. (2020). Machine Learning-Based Financial Statement Analysis. SSRN Working Paper No. 3520684. Social Science Research Network.

Reis, S. R. N., & Reis, A. I. (2013). How to write your first scientific paper. In Proceedings of the 3rd Interdisciplinary Engineering Design Education Conference, IEDEC 2013, 181–186.