Predictive Imputation in Stock Data: A Comparative Analysis of Random Forest, Linear Regression, Bootstrapping, and Predictive Mean Matching

C. Beattie, Sam Rudenberg, and K. Shah

December 15, 2023

Abstract

We investigated different methods, such as random forest and arithmetic algorithms, for imputing for NA values in S&P 500 quarterly reports. Using a dataframe of 15,882 observations for 338 of the original 500 equities, we use different predictive models within the “mice” package to impute values for common financial ratios, such as book ratio, earnings per share, and market capitalization. Each observation was standardized as a percent change relative to the oldest observation for the respective equity. Accuracies are calculated as the percentage of accurate predictions within a predetermined range above and below the true value. Our results found that the random forest consistently outperforms the others, reaching an accuracy above 35% at the 100-level threshold.

I - Introduction

Data manipulation is almost always a step of the complete data application process. Handling "not available" (NA) data points is typically something we think very little of as they are discarded more often than not. However, NA's may need to be imputed in some applications. Prediction has become an extremely popular topic within the investment world as accurate models can theoretically yield enticing volumes of alpha. However, rather than predicting future price points and returns, we compare different supervised learning models to accurately predict missing values in quarterly reports. This motivation fueled our effort to answer the research question, "How do different predictive models from the *mice* package, such as random forest, differ in predictive accuracy regarding different prediction thresholds?".

II - Methods

Utilizing the *Rblpapi* package in R, we gathered financial data from Bloomberg Terminal for all equities currently in the S&P500. Our choice of indicators was guided by previous literature (Huang et al. 2021, 11). The data was exported to a CSV and manipulated in RStudio. The original CSV had 62758 rows and 23 columns. We then removed all rows containing NAs, reducing the data frame to 15882 observations corresponding to 338 of the original 500 equities. While counterintuitive to our study's goals, removing NAs was critical. We needed a complete dataset to separate into training and test sets. Observations for each equity were standardized as a percent change relative to the first observation for each equity, i.e. the oldest quarterly report. The first observation for each equity was set to zero. Then, we analyzed patterns of NAs in the original dataset and created methods to randomly reproduce these patterns in a copy of the newly standardized data frame.

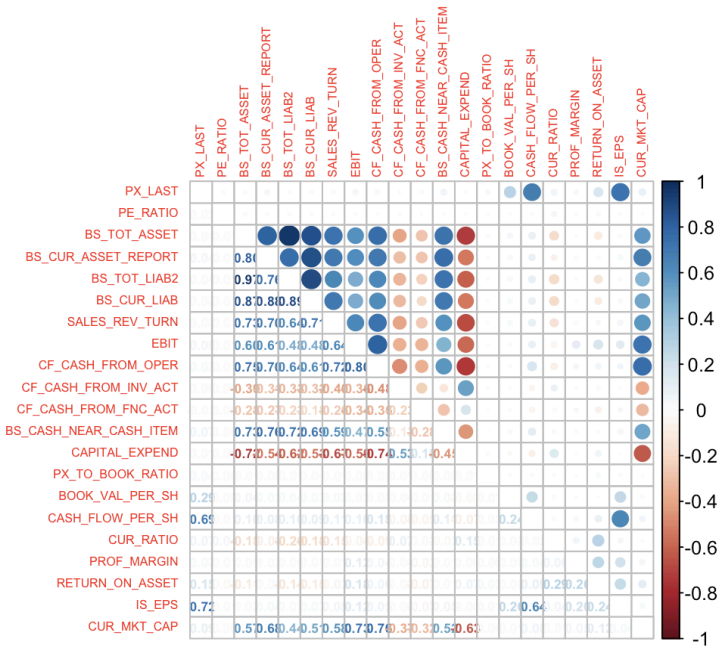
At this point, the data was prepared for analysis. We used the Multivariate Imputation by Chained Equations (mice) package to test the accuracy of built-in predictive imputation models. More generally, mice functions inspect the pre-existing patterns of missing values and then impute m times producing m copies of the original dataset in one dataframe. We used predictive mean matching (pmm), random forest imputations (rf), linear regression ignoring model error (norm.nob), and linear regression using bootstrap (norm.boot) with $m = 4$. Predictive mean matching predicts a given NA based upon matches in the observed data (Buuren 2003). Via random sampling of the observed matches, pmm will impute the prediction (Buuren 2003). As for the random forest method, forests were generated to impute values where NAs pre-existed at random within each numeric column of the original dataframe. As previously specified, imputation was conducted four times ($m = 4$). Specifically, the call to rf implements Breiman's random forest algorithm (Buuren 2003). The norm.nob method imputes data in each cell previously occupied by NA using "linear regression analysis without accounting for the uncertainty of the model parameters" (Buuren 2003, 125). And finally, norm.boot imputes data using a similar linear regression method that incorporates bootstrapping (Buuren 124).

Each of the $m = 4$ imputation cycles were stored in the same data frame. We created subsets and found the mean for each [row, col] pair of these subsets to create a more accurate prediction. Cross-validation techniques inspired this approach.

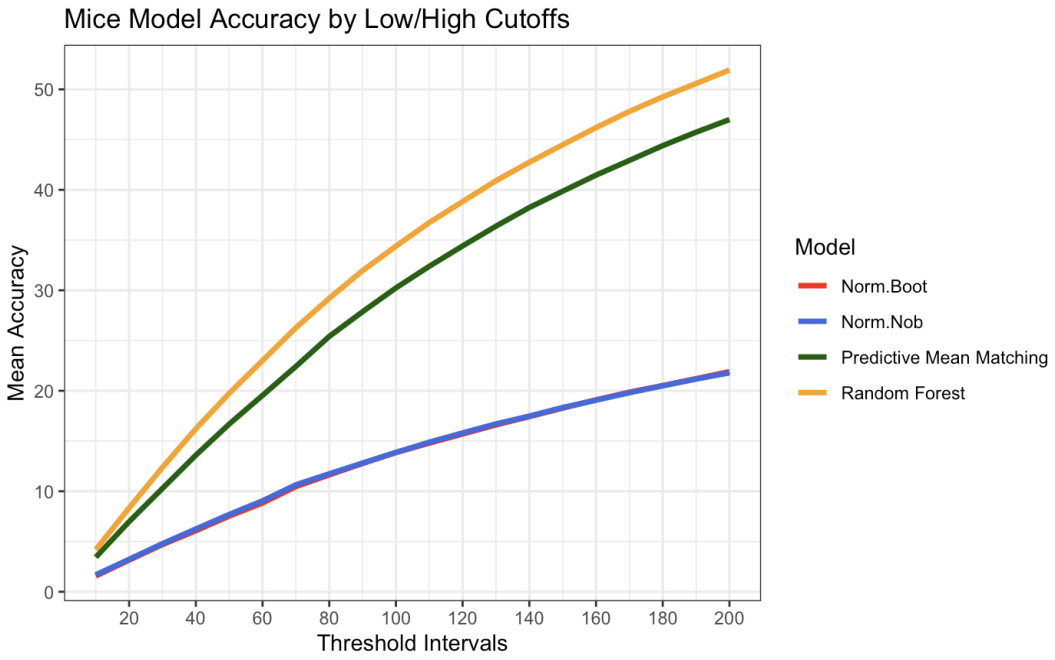
The metric we used to assess the success of each model was percent accuracy. The models are not able to reliably predict exact values, so to account for error, we introduced a threshold range. Each model was evaluated at multiple threshold ranges, and the results were stored in a data frame. This data frame was then used to output graphics and determine the results of the study.

III - Results

With 21 unique column variables, it is difficult to assess the contribution of each predictor toward the imputation of NA values. To find correlations between variables and assess any underlying relationships, we created a corrplot (seen below).



While some variables are strongly correlated or not, we note that most variables have little to no correlation. This may prove to be a challenge for the predictive imputation methods. After running the models, we found that the order from most to least accurate model was random forest, pmm, norm.boot, and norm.nob. When the threshold is small, all models have relatively the same accuracy. However, as the threshold interval is increased, the curves for norm.boot and norm.nob begin to flatten out and become much less accurate than pmm and random forest. The curves for norm.boot and norm.nob are very close together, but looking closely norm.nob begins as a better predictor. Yet, norm.boot is more accurate at a threshold interval of 160 and above.



We ultimately conclude that the rf and pmm methods perform at increasingly higher accuracies respective to norm.nob and norm.boot as the threshold for accurate predictions expands.

IV - Discussion

While this analysis of the 'mice' package is somewhat novel in terms of our threshold of accuracy approach, the findings are extremely limited. First, we assume that our method of standardization was not mathematically biased. Furthermore, our NA patterns were placed within the clean data via an algorithm based on a personal review of the dirty data. This may also bias our methods. And finally, the results must be taken with a grain of salt. While the random forest model does break the 50% accuracy mark, the threshold for this prediction mean was set at plus-minus 200. This is an incredibly large margin for error. We must also consider the real-world implications of these imputations. While all cells represented a percent change, incorrectly predicting a missing value would result in the potential loss of capital in the real world. Depending on volumes, the discussion could involve hundreds, thousands, etc.

This is to say that further research must be conducted before significant conclusions are made. This project's findings might lay the groundwork for a wider spectrum of imputation contexts within other fields. The tools we have developed and tested will never be considered a replacement for true data, but they could prove to be very useful in datasets that may only be missing a small percentage of information.

V - References

Buuren, Stef van. 2003. “Package ‘Mice.’” https://github.com/amices/mice.

Buuren, Stef van, and Karen Groothuis-Oudshoorn. 12/012011. “Mice: Multivariate Imputation of Chained Equations in R.” *Journal of Statistical Software* 45 (3): 1–67.

Eddelbuettel, Dirk. 2022. “Package ‘Rblpapi.’” R Interface to “Bloomberg.” https://dirk.eddelbuettel.com/code/rblpapi.html.

Huang, Yuxuan, Luiz Fernando Capretz, and Danny Ho. 2021. “Machine Learning for Stock Prediction Based on Fundamental Analysis.” *Electrical and Computer Engineering Publications*, no. IEEE Symposium Series on Computational Intelligence (SSCI) (December): 1–11. https://doi.org/10.1109/SSCI50451.2021.9660134.