Jeannie Thai (11292475) Blosher Brar (11258419)

Comparison of methods for handling missing data on the performance of classification algorithms

Context

Over the last few years, the availability of data has increased tremendously, leading to opportunities for building algorithms that rely on data to perform analysis. However, the problem of missing data has remained a challenge in data mining. There are multiple techniques used to deal with this problem, the easiest of which is removing the data points with missing values. However, it is important to select the most appropriate method based on the dataset. Training a model with missing data can drastically impact statistical inferences and performance measures. There are many reasons why data can be missing. An investigation on the causes of missing values is a good starting point. For example, missing values can arise because the age of a client is unknown or no phone transactions by a client are recorded. It is important to develop a deep understanding of the data. This will be beneficial to determine the best methods of handling missing values.

Rubin (1976) classified missing data into three types: MCAR (Missing Completely at Random) and MAR (Missing at Random) and MNAR (Missing Not at Random). Although they may appear to be similar, they have very different consequences when it comes to correcting missing values and performing data analysis. MCAR is when data is randomly distributed across all observations, with no underlying causes. In other words, the probability of being missing is the same for all data points and some of the data is missing due to a random stochastic process. The second type is MAR, which is when the probability of being missing differs between subgroups in a dataset. This can arise in a variety of situations but can be due to a poorly designed study or biased samples chosen from a population. For example, in a given questionnaire without the option for Not-Applicable answers, males would be systematically less likely to answer questions related to menstrual cycles and hence the overall questionnaire would have missing data for males and questions related to menstrual cycles. According to Burren (2018), “MAR is more general and more realistic than MCAR.” The third type of missing values are MNAR, which means that the probability of missing varies for reasons that are unknown. An example of this would be in public opinion research if those with weaker opinions respond less often than those with stronger opinions and hence lead to a biased dataset whose inherent bias may be overlooked. Coony (2018) advises that MNAR is the most complex missing value situation, however, an effective strategy may be to find more data about the cause for the missingness, or perform a what-if analysis to measure sensitivity of the results under various scenarios.

Overall, missing data can significantly reduce the information contained in a dataset and therefore distort the predictive power of a model. However, datasets without any missing data is

often an impossibility, and hence the prevailing scientific practice is to downplay the missing data. By using appropriate techniques to deal with this missing data, one hopes to be more transparent and mitigate against the problem of missing data. This paper will investigate a few of these techniques and their impact with various machine learning models.

Description of methods

There are multiple approaches one can take when it comes to handling missing values, each of which affects the end result in a different way. This paper will focus on three main methods.

1) The simplest method is to discard the data altogether, known as *listwise deletion*. However, Buuren (2018) indicates that this may cause additional complexities in interpretation, and the "occurrence of missing data has long been considered a sign of sloppy research." Listwise deletion is often used without explicitly mentioning its use or arguing its pros and cons (Burren, 2018). This method suffers from loss of information and can lead to biased estimates if the data is not MCAR. Furthermore, it reduces the sample size, which might not be representative of the population. Especially when a large proportion of data is missing, statistical results show that large standard errors are obtained (Karangwa, 2013). Therefore, the listwise deletion should be avoided if possible.
2) Another method of handling missing values is *single imputation* using mean, median or mode. Each missing value in the variable is imputed based on the non-missing values of that variable. This means every missing value in the variable will have a common value (mean, median, mode). It is an easy and simple strategy in which the sample size is preserved. However, it reduces the variance of the feature and the correlation between features. For a categorical variable, imputation using the mode can be used to observe the highest frequency. We will be using mean imputation on continuous feature variables.
3) Another missing value method is *Classification and Regression Trees (CART) Imputation*. It seeks to approximate the distribution of a univariate conditional outcome using multiple predictors and a tree structure to represent the partitions of binary predictor splits. Regression trees have a built-in methodology to handle missing values using surrogate decisions, in which the values of other features are used to perform a split between observations with missing values. A feature with missing values will be used as the dependent variable and the remaining of the features as potential splitting variables. The variable with split point yielding the best surrogate split is used to make predictions for the feature with missing values. The benefits of using CART imputation are that it can handle interactions, non-linear relations, and complex distributions. Overall, CART imputation is an efficient choice that preserves the correlation between features and requires little

effort of treatment on missing values when training a CART model that handles this imputation automatically.

Classification is a common supervised learning problem used to make predictions for categorical response variables. Standard data pre-processing steps like checking for correlations among explanatory variables, and searching for explanatory variables which were the most important, were conducted to ensure model accuracy. We used three machine learning models to tackle binary classification.

1) Logistic Regression is a simple model for modelling binary data. The model returns a predicted probability of the response variable of interest between 0 and 1. It assumes linearity of the independent variables and log odds, as well as little or no multicollinearity among independent variables.
2) Classification and Regression Trees (CART) is another classification method that can be used to predict a binary response variable. It uses a tree structure to represent data, with each leaf node containing an output variable used to make a prediction. Default settings from RPART function were used when training the model. Note that the CART imputation was automatically handled in the CART model.
3) Random Forest is an ensemble learning method that uses the concept of *trees* to classify new predictions using input vectors. It can handle high dimensional problems and oftens leads to high performance measures without the need to tune hyperparameters. To reduce the runtime, the number of trees used is 20 instead of the default 500 trees, in the RandomForest package.

Research Question

The purpose of this paper is to evaluate the sensitivity/robustness of different classification models to different missing value methods. The classification models used for the purpose of this study are logistic regression, CART and random forest. Different scenarios were simulated based on the percentage of missing values and the respective imputation method will be applied to each replication. The performance of each scenario was compared based on the AUC (Area Under the Curve), where the model with high AUC is preferred.

Monte Carlo Simulation

In order to investigate the various methods, we used Monte Carlo simulations to generate data following a logistic distribution to model classification with balanced classes (1 for occurrence and 0 otherwise). The dataset has a sample size of 1000 and contains 10 features; 3 binary variables following a binomial distribution and 7 continuous variables following a normal distribution. The coefficients were randomly chosen and a random error term with normal distribution was added to create noise to the dataset and to further distinguish the performance

between each classification model. The simulations were repeated 1000 times. A random seed was used to ensure reproducibility and consistent comparisons between methods. The simulations were done to know the true model with complete data and to control different confounding factors, including pattern of missingness, and the proportion and relevance of missing values. The logistic regression equation is defined as below:

$$Y_i = B_0 + B_1 X_{1,i} + \ldots + B_{10} X_{10,i} + \varepsilon_i$$

Missing values were randomly generated under the assumption of MCAR. This form may not hold in real datasets, nonetheless it is a general and commonly assumed scenario for comparison between different missing value methods (Saar-Tsechansky, Provost, 2007). We generated two levels of missing data, 15% and 30% on each of the four continuous variables (x_3, x_5, x_7, x_8) that are statistically significant from the regression model, with a p-value less than 0.05 on average over all simulations. For clarity, when we mention 15% or 30% missing data, we refer to *each* of the four columns, and not the whole dataset. In this context, we assumed that missing values are *only* present in the training set. In principle, "missing data can occur in the training data only" (Valdiviezo, Aelst, 2015). For example, in transportation, the variable "seat belt used" contains missing values for bicycles, but in unseen data, there are no longer records of bicycles possibly due to the winter season. Though missing data most often appear in both training and test sets, listwise deletion is not applicable when missing values are present in test cases (Valdiviezo, Aelst, 2015). Furthermore, a complete test set will "avoid an extra source of variability in the performance measures" (Valdiviezo, Aelst, 2015). Therefore, the relative performance of different missing value treatments was evaluated for each model using a complete validation set. The dataset was randomly partitioned into training and validation sets. We used 70% of the data for training and 30% for validation. Missing values were treated before applying classification models. The average ROC (Receiver Operating Characteristic) curve AUC scores across 1000 iterations were used as a measure of performance. It specifies the probability that a randomly chosen positive instance is higher than a random negative instance. A model with predictions that are 100% correct has an AUC of 1.0. Accuracy scores were not used as it produced equivalent ranking predictions.

Results

The mean predictive performance of 21 models with 1000 iterations each were evaluated as summarized in Table 1. In general, the models with no missing data perform better than with missing data. This is expected, since Machine Learning models tend to perform better when they have more information to *learn* from. The models are better at making predictions and capable of distinguishing between the binary classes. When dealing with missing values, both mean and CART imputation have higher average AUC scores for each model than the listwise deletion method. Evidently, listwise deletion performs poorly due to the amount of information lost in the training set. We observed that the lowest performances of each model is when we used listwise

deletion with 30% missing data. Other studies have also demonstrated the poor performance of listwise deletion on real datasets (Karangwa, Kotze, 2013). As the proportion of missing data increases from 15% to 30%, the average AUC decreases across each model. In other words, predictions made were less accurate and some information is lost on the relationship between the binary response and the covariates. When the missing rate is low, there is a smaller difference in performance for each missing value method. However, when the missing rate is at 30%, this difference is higher.

Table 1: Comparison of Average AUC on classification algorithms

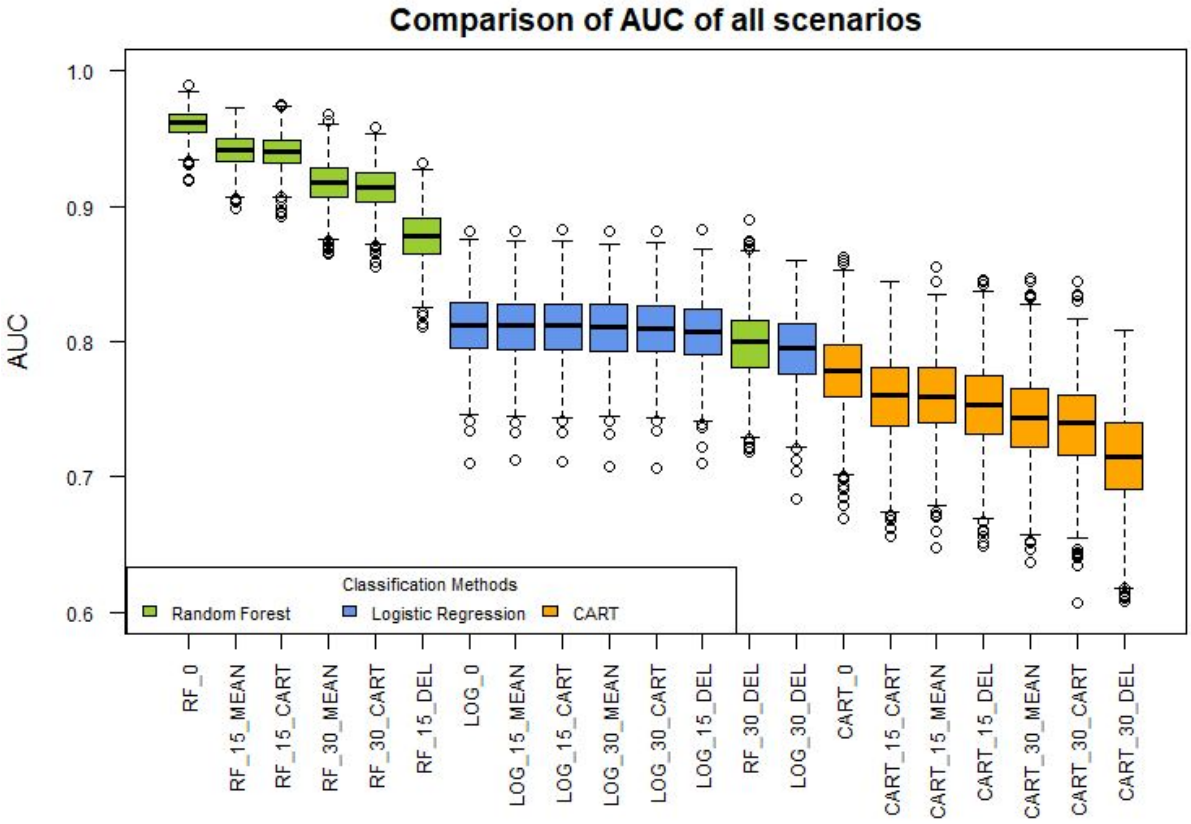
Proportion of Missing	Missing Value Method	Logistic Regression	CART	Random Forest
0%	N/A	0.8114	0.7769	0.9614
15%	Listwise Deletion	0.8063	0.7528	0.8768
15%	Mean Imputation	0.8108	0.7594	0.9410
15%	CART Imputation	0.8107	0.7587	0.9398
30%	Listwise Deletion	0.7939	0.7123	0.7972
30%	Mean Imputation	0.8101	0.7439	0.9169
30%	CART Imputation	0.8093	0.7388	0.9132

Both mean and CART imputation on each model display similar performances, with mean imputation performing slightly better. Therefore, both methods are reasonable choices for our simulated datasets. Using the mean has its advantages, since it is a simple method that requires little coding and has a lower computation time. In addition, since the data is MCAR and that there is little to no correlation between each feature, the estimation of the mean of each feature column remains unbiased. However, mean imputation should be used with caution. Although it may appear to be the best method in our simulated datasets, this method reduces variability in the data. Other studies have shown that CART imputation led to high performance measures due to its attractive properties such as its robustness to outliers and ability to deal with multicollinearity and skewed distributions (Saar-Tsechansky, Provost, 2007).

The AUC scores of 1000 simulations for each method/model are summarized in Figure 1. The ranking from highest to lowest AUC based on median is equivalent to the ranking based on the mean. There are some outliers/datasets on each model outside the whiskers. This is very

reasonable to expect in our simulations due to noise, the different missing values generated, and the stochastic process of running simulations.

Figure 1: Box Plot of all scenarios from highest to lowest AUC based on median



Interestingly, CART models have the lowest overall AUC scores. While CART is an intuitive technique to fit trees, it has some drawbacks: "it is highly unstable due to its hierarchical nature and tends to produce selection bias towards continuous and categorical features with many possible splits and missing values" (Valdiviezo, Aelst, 2015). Moreover, we used the default parameters (min split, max depth, max surrogate from RPART function), which possibly led to overfitting.

As we see from the box plot, Random Forest models perform significantly better than logistic and CART models. Random Forest was developed to decrease the correlation among trees by adjusting the splitting process (Valdiviezo, Aelst, 2015). It averages out sampling variability, making it an ideal model for predictions. It is robust and can handle large amounts of missing data. One drawback of tree models is that they can be difficult to interpret unless the trees are relatively small. Surprisingly, the Random Forest model with 30% missing data and listwise deletion (RF_30_DEL) has a lower overall AUC than the logistic models. This suggests that in our simulations, the model's performance quickly deteriorates when a large proportion of

data is missing and a relatively poor method was used to handle missing data, reinforcing the idea that listwise deletion should be avoided at all cost.

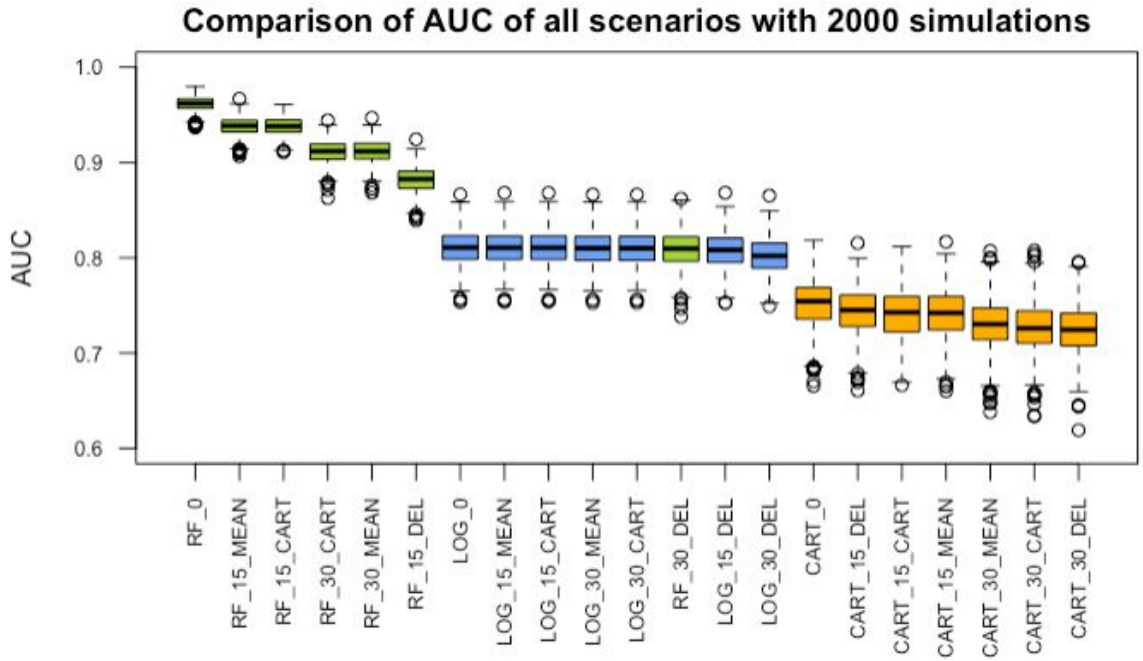
The logistic regression displays similar results across the model 0%, 15% and 30% missing values. Since our data generated is based on logistic regression with a binary response variable, on average, the estimated parameters from each model are consistent with the true model with no missing data. Furthermore, the absence of high correlation between features made it simpler for the model to make consistent predictions even when there are missing values. This might not be the case with real datasets, where it does not necessarily follow a logistic distribution and there is considerably more noise. The method used was the logistic regression model:

$$Prob\{Y = 1|X\} = \frac{1}{1 + \exp(-X\beta)}$$

As such, logistic regression is able to adequately capture the true underlying data structure even with listwise deletion. Although the mean imputation and CART imputation improve the AUC on average for logistic regression, it is a significantly smaller improvement when compared with CART and Random Forest.

To further investigate the role of simulations on the AUC, we re-ran the code with 2000 simulations instead of the 1000 simulations run initially. As can be seen in Figure 2, the resulting AUC scores for 2000 simulations were similar to the ones with 1000 simulations except that the variability of AUC scores was reduced by a significant amount. This was an expected result, due to the Law of Large Numbers, where we expect the stochasticity to be reduced as the sample size increases. The average AUC scores of 1000 simulations versus 2000 simulation was essentially the same with no drastic differences between imputation methods or modelling methods.

Figure 2: Box Plot of all scenarios from highest to lowest AUC based on median



To understand the effect of simulations on AUC scores, we decided to compare the AUC score of multiple models across 1000 simulations. This way we can get a better understanding of the stochastic process in which simulations are conducted. Figure 3 illustrates the proportion of times each logistic model outperforms the other logistic models. We counted the number of times the AUC score is higher for the reference model on the x-axis to the models displayed in the bar columns. It is clear that not one method consistently exceeds another method based on AUC, since it strongly depends on the proportion and location of missing values. For example, the logistic model with 30% missing data/listwise deletion (LOG_30_DEL) outperforms the model with 15% missing data/listwise deletion almost 20% of the time (200/1000 simulations) as indicated by the green column. It is important to note that the model compared to itself is always 100%. The LOG_30_DEL always outperforms the model itself as indicated by the yellow bar column.

Figure 3: Logistic Models: Proportion of times each model outperforms other models

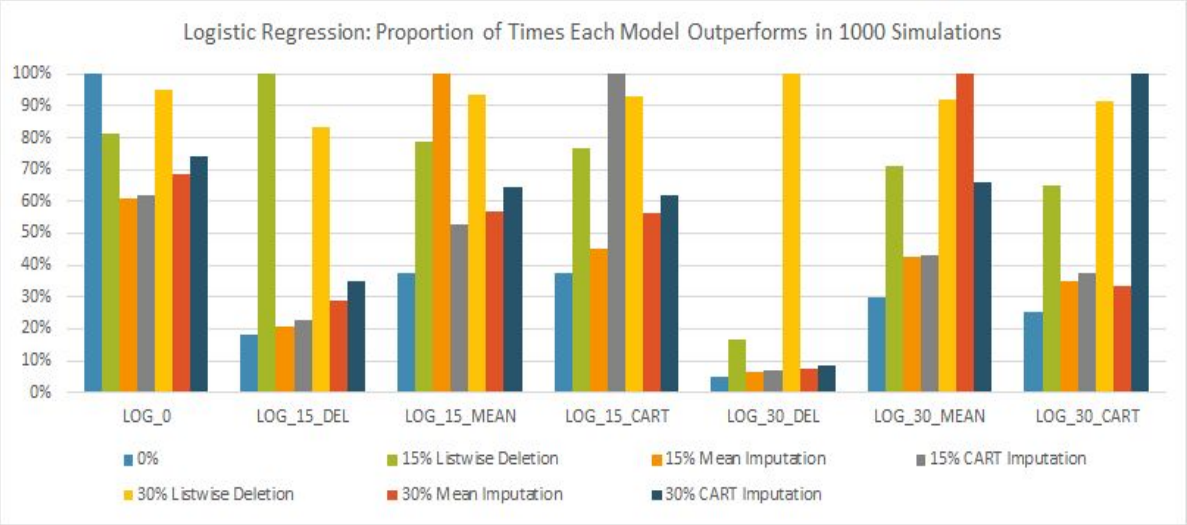


Figure 4: CART models: Proportion of times each model outperforms other models

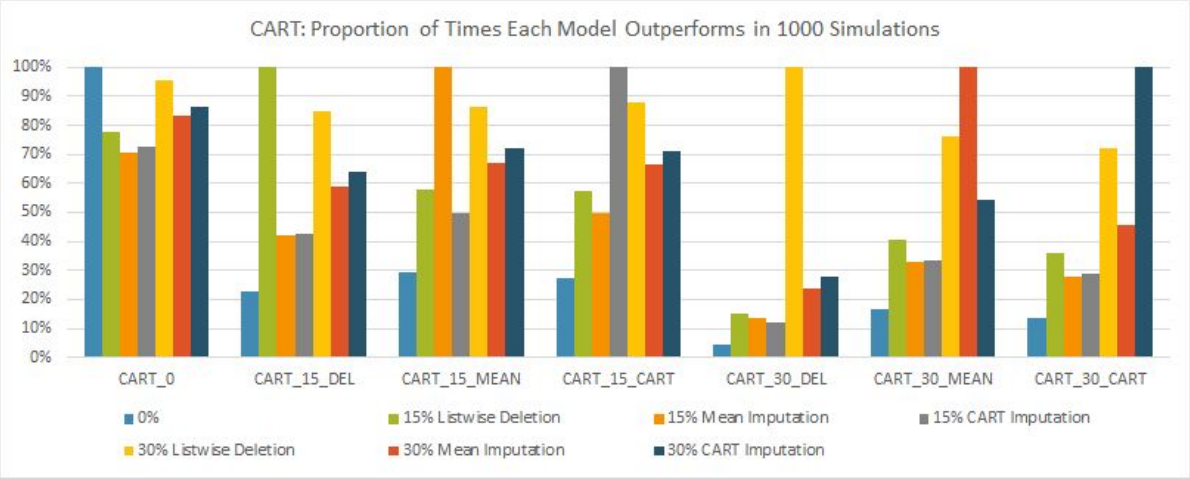
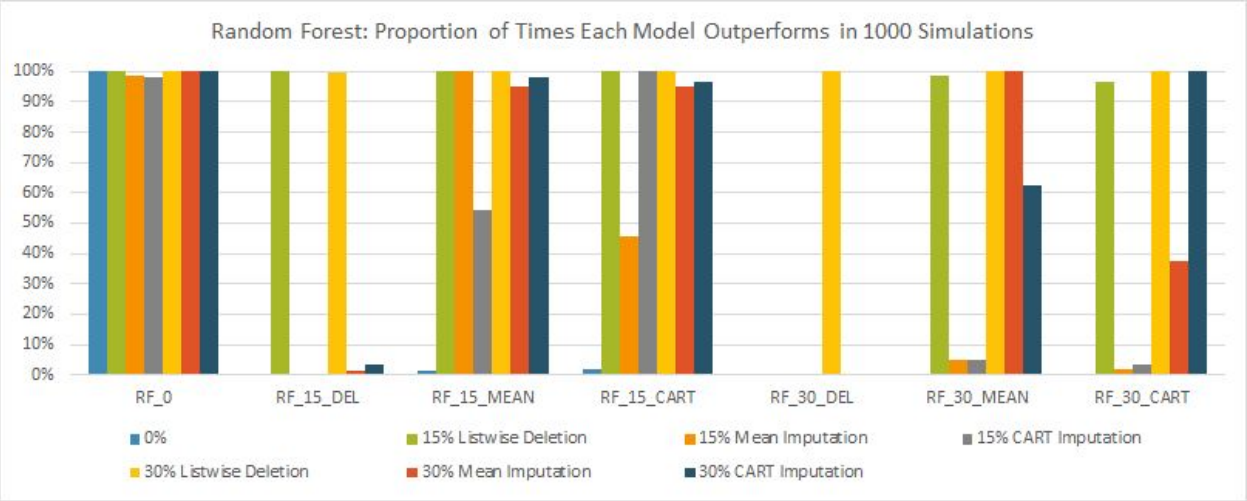


Figure 5: Random Forest models: Proportion of times each model outperforms other models



Note that in Figure 5, the Random Forest model with 30% missing data and listwise deletion (RF_30_DEL) never outperforms other Random Forest models (except itself as indicated by the yellow bar). The AUC score for this method is always worse in all 1000 iterations. Therefore, we can be confident that this is an extremely poor method for our data.

As can be seen in Figures 3,4 and 5, the lowest AUC is not always observed for the best fitting model due to the stochasticity in Monte Carlo simulations. The training set and its patterns of missingness in every iteration are unique, hence the performance can vary widely between iterations. In every iteration, all models consist of a random partition of training and validation set and missing values under MCAR. The correlation between features and feature importance is also affected. Each iteration has different relationships between the target and covariates. We initially believed that missing data leads to inaccurate predictions. We observed that for each true model (LOG_0, CART_0, RF_0), the AUC scores are not always higher compared to models with missing data. The noise in some iterations gives a sense of false performance measure. As a result, looking at the mean AUC, gives an overall direction for each missing value method.

It is important to realize that our results focus on classification trees and do not generalize to other models. Furthermore, the results are based on the limitations of the simulations performed. Results can vary under different settings in terms of sample size, missingness of values, relationship between variables and the choice of learning algorithm.

Conclusion

Missing values in datasets are a reality that will likely be encountered when conducting research in the data science field. Although the method of simply deleting rows with missing values is the easiest, it is hardly an appropriate method considering the consequences of a biased or incomplete dataset. Missing values make it difficult to extract useful information and can lead

to misleading predictions and incorrect statistical inferences. We emphasize the importance of understanding the behaviour of missing data and the choice of a method for handling missing data should complement the type of missing data being considered. Since our study is limited to MCAR data and classification models, researchers should choose a missing value method based on careful consideration of the advantages and disadvantages of different methods for their specific dataset.

This paper presents three different methods to handling missing data and the resulting performance under three different machine learning methods. 1000 and 2000 Monte Carlo simulations were used to create two datasets that tested the various methods. The results we received were a mixture of what we expected and also what we did not expect. The conclusion we can draw from this paper is that the problems associated with missing values follow a stochastic process. Although the average AUC was always higher for a model in which missing values were accounted for and imputed, this was not always the case for individual AUC scores for each simulation. In fact, sometimes the models with missing values were able to capture the true data structure better than the one whose missing values had been estimated. This is purely due to the stochasticity associated with a dataset. In other words, the impact of missing values varies across datasets due to randomness.

Looking beyond the results of this paper, we acknowledge that our simulated results lack generalizability due to the limited artificial simulation patterns and three missing value methods (Valdiviezo, Aelst, 2015). A future improvement should be to extend our analysis to mimic real datasets. Perhaps, one could also conduct simulations using a continuous response variable and see how missing data impacts machine learning models that deal with regression problems or even clustering. Nevertheless, the impact of missing values in the context of classification should not be overlooked, and one should understand the availability of tools and methods to deal with missing values.

References

Burren, S.V. (2018). *Flexible Imputation of Missing Data*. Chapman & Hall

H. Cevallos Valdiviezo, S. Van Aelst. (2015). Tree-based prediction on incomplete data using imputation or surrogate decisions, Information Sciences, Volume 311, Pages 163-181, ISSN 0020-0255

I. Karangwa, D. Kotze. (2013). Using the Markov Chain Monte Carlo Method to Make Inferences on Items of Data Contaminated by MissingValues, American Journal of Theoretical and Applied Statistics. Vol. 2, No. 3, 2013, pp.48-53. doi: 10.11648/j.ajtas.20130203.12

Karangwa, D. Kotze. (2013). Using the Markov Chain Monte Carlo Method to Make Inferences on Items of Data Contaminated by Missing Values, American Journal of Theoretical and Applied Statistics. Vol. 2, No. 3, 2013, pp.48-53. doi: 10.11648/j.ajtas.20130203.12

Lane F. Burgette, Jerome P. Reiter. (2010). Multiple Imputation for Missing Data via Sequential Regression Trees, *American Journal of Epidemiology*, Volume 172, Issue 9, 1 November 2010, Pages 1070–1076, https://doi.org/10.1093/aje/kwq260

Maytal Saar-Tsechansky and Foster Provost. (2007). Handling Missing Values when Applying Classification Models. J. Mach. Learn. Res. 8 (12/1/2007), 1623–1657, https://dl.acm.org/doi/pdf/10.5555/1314498.134553

Rubin, D.B. (1976) Inference and Missing Data. *Biometrika*, 63(3), 581-590