## **Introduction**

Data quality remains one of the major issues in almost every data quality project. However, poor quality data is not only concern of the handler, but is also costly to companies that try to make use of it. According to an Experian survey, 77% of respondents believe poor quality data is actively affecting their bottom line. One of the most common issues impacting that bottom line is missing data, which makes handling missing data a requirement for almost every data science project. It also makes handling missing data correctly the difference between an average data wrangler and a practiced data scientist.

Missing data can be a result of a variety of causes, and not all missing data necessarily impact your ability to effectively perform the analysis required to solve your problem. Most missing data is defined into three categories: missing completely at random, missing at random and missing not at random. Missing completely at random (MCAR) is a scenario where there is no relationship between the missing data and any other variable in your data. Missing at random (MAR) means the data has some relationship to other variables in your relationship that can be controlled for. The third group, missing not at random (MNAR), is a group where the missing value is dependent on the actual unobserved value. For example, in a survey that requires income, the level of someone’s income may be the cause of that value being missing, since people making over or less than a certain amount may not be interested in disclosing their income.

In this case study, we will be exploring the impact of different levels and types of missing data on a linear regression problem and demonstrate how various imputation strategies affect the performance of our model. Beginning with MCAR values, we compare performance between a baseline and varying levels of missing data. We see that our performance was affected the most when the largest amount of data was removed (50% of values from CRIM). We observe a similar degradation in model fit for our MAR scheme - as more data is removed, the MSE generally increases while the decreases. Finally, we demonstrate that our MNAR scheme also causes our model fit to decrease when compared to our baseline.

Below we detail those findings, first by providing some background about our data and methodologies used for imputing the missing values. We then discuss the results in detail and our conclusions about this process.

**Background**

For this case study, we will be working with the Boston housing data set made available through Python’s sklearn library. This data set is a collection of 506 homes sampled in 1978 and is maintained by Carnegie Mellon University. This data set is comprised of 14 total variables described in Table 1 below.

**Table 1**: List of Features in our dataset

|  |  |
| --- | --- |
| **Feature Name** | **Feature Description** |
| CRIM | per capita crime rate by town |
| ZN | proportion of residential land zoned for lots over 25,000 sq.ft. |
| INDUS | proportion of non-retail business acres per town |
| CHAS | Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) |
| NOX | nitric oxides concentration (parts per 10 million) |
| RM | average number of rooms per dwelling |
| AGE | proportion of owner-occupied units built prior to 1940 |
| DIS | weighted distances to five Boston employment centres |
| RAD | index of accessibility to radial highways |
| TAX | full-value property-tax rate per $10,000 |
| PTRATIO | pupil-teacher ratio by town |
| B | 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town |
| LSTAT | % lower status of the population |
| PRICE | Median value of owner-occupied homes in $1000's |

For this analysis we will be using this data to predict the median value of the homes using 12 of the variables. For this analysis, we will not be using the feature “B” in our model, as we do not want to introduce race a feature for predicting the median value of our home. For 12 other features and target values there are no missing fields in this data, so we will have to manually manipulate the data for this case study, which we discuss more in the methodology.

For this study we will be exploring various imputation techniques for handling missing data. Without imputation, most regression tools will automatically apply list-wise deletion to all rows with missing data. This means that if only a single value from the 12 features above is missing, the entire observation will not be used determining the model weights. This results in a loss of information that could be helpful when trying to make predictions on new data. In order to save that information, the empty values can be filled using either techniques from two categories, single imputation and multiple imputation.

Single imputation is a strategy of filling the missing values using a value determined by the existing cased from this variable. This includes things like mean imputation or last observation carried forward. This has largely been considered an insufficient method as it often reduces variability and can lead to bias in your data. Most statisticians prefer the use of multiple imputation techniques.

Multiple imputation is a strategy designed to maintain the variability when filling in missing values. It begins be creating multiple imputed data sets using a selected technique for imputing each set differently. We select a means of filling these variables that maintains the variability within the dataset based on the existing relationships in the data. We then perform our regression on each of the imputed data sets. Once we have all the different results from each regression, they are combined using Rubin’s rules, which are designed to pool parameter estimates and regression coefficients.

Below we explore several of these imputation techniques and their impact on the performance of our regression model.

## **Method**

In order to compare the impact of a variety of missing data schemes we first must understand how a simple linear regression fits the data as provided. We already observed that the data is not missing any data, so we remove the attribute B (proportion of blacks by town) and fit a linear regression model using all of the data. We record the baseline mean squared error (MSE) and r-squared ( for future comparison.

For the imputation in all of the exercises we chose to use IterativeImputer from sklearn. IterativeImputer models each feature that is missing values as a function of the other features, and uses that estimate for imputation. IterativeImputer works in a round-robin fashion by identifying a feature with missing values y and other feature columns are treated as inputs x. A regressor is fit on (X, y) for known y and is used to predict missing values of y. Next the algorithm iteratively selects another feature with missing data and uses the imputed data from the last iteration to inform the current step. Our MCAR and MNAR schemes introduce missing data into a single feature, therefore for these imputations the advanced stepwise features of IterativeImputer will not be of any use. Effectively, we are using a simple regression scheme in a single pass to impute our missing data. However, our MAR scheme removes data from two features and thus the stepwise imputation will be effective. [2]

First, we are tasked with introducing a MCAR scheme to the dataset. We write a function that will select a column and set a percentage of the values in the column to null. We iterate through the function to create data sets that have 1%, 5%, 10%, 33%, and 50% of the data missing, respectively. The selection of the column is arbitrary and can be updated in the function for further investigation, however we chose the first feature, which is CRIM (per capita crime rate by town) for our evaluation. In each iteration, we impute the missing values and record the MSE and for comparison against our baseline and each prior iteration.

Secondly, we create a version of the dataset introducing a MAR scheme. This was achieved by introducing randomly selected null values by first controlling for another feature. We decide to partition the data for those rows with a value of NOX (nitric oxide concentration) that is greater than 60% of the maximum value for this feature. We arrived at 60% through trial and error to ensure that we had enough candidate rows left to make an impact by removing data. We found that the 60% threshold gives us 269 candidate rows. For these candidate rows we selected RM (average number of rooms per dwelling) and AGE (proportion of owner-occupied units built prior to 1940) as the features where we would introduce null values. We randomly selected 10%, 20%, and 30% from our candidate rows and removed data in these features. We then imputed the data and fit a linear regression model for each dataset to measure the MSE and for comparison against our baseline and each prior iteration.

Finally, we create a version of the dataset that introduces a MNAR scheme. We selected the CRIM (per capita crime rate by town) as the feature to manipulate. We identify the top 25% quartile and set all values to null. We then impute the data, train a regressor, and record the MSE and against the baseline.

## **Results**

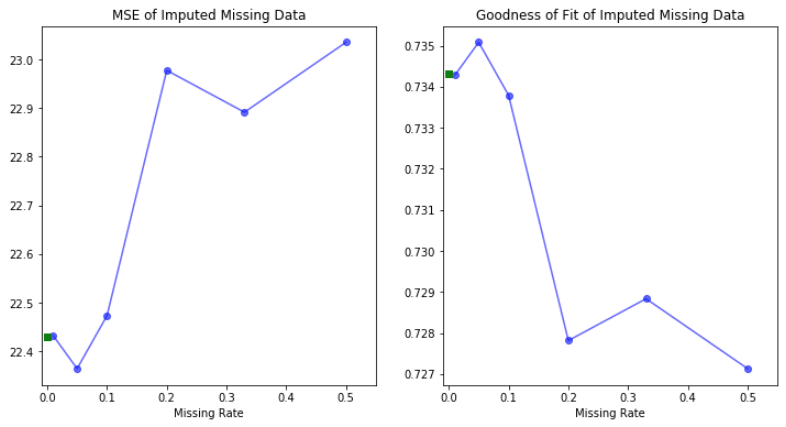
The baseline results from performing a simple linear regression on the raw data with the B (proportion of blacks by town) feature removed are shown in Table 2.

**Table 2**: Baseline MSE and for Full Dataset with B Removed

|  |  |
| --- | --- |
| **Mean Squared Error (MSE)** | **Baseline** |
| 22.4297 | 0.7343 |

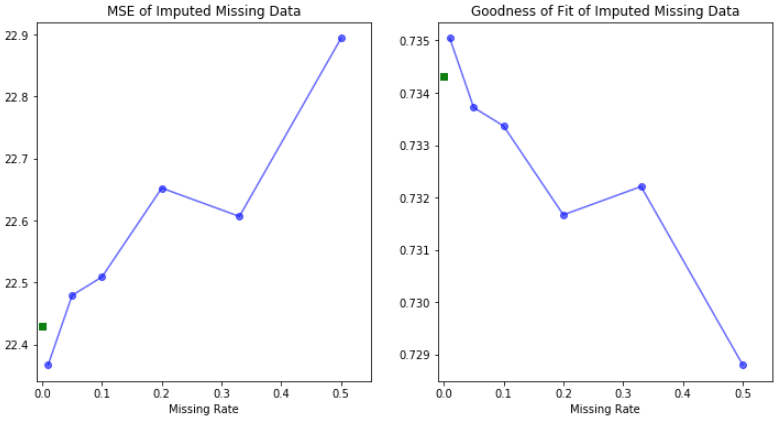
The results from the MCAR scheme are shown in Figure 1 below.

**Figure 1**: Results of Iterations Introducing MCAR



The baseline MSE and are shown by the green block. It’s quite clear in the general trend that MCAR reduces the accuracy of our models as the amount of imputed data in CRIM increases. The evidence is a general increase in MSE and decrease in . We do note that there are interesting improvements in fit as an increase in missing values are introduced, which is counterintuitive at first glance. This is observed via pairwise comparisons between the baseline and 5% missing as well as 20% and 33% missing. We attribute this to randomly removing the values from CRIM. In some cases the values removed might not be representative of the expected results when imputing the missing values via the IterativeImputer regression on the other features. This would have the effect of actually tightening the fit. We choose to perform another pass through our MCAR function using a different random seed to support this argument. The results of the additional pass through are shown in Figure 2 below.

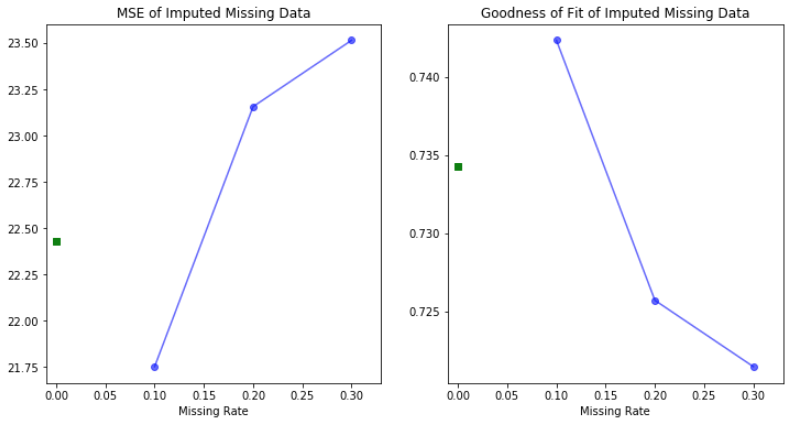
**Figure 2**: Results of Iterations Introducing MCAR (Different Random Seed)



The additional pass exhibits different anomalies as described above, which supports our argument. Furthermore, the general increase in MSE and decrease in follow the expected trends and our initial analysis.

The results of our MAR scheme are shown in Figure 3 below.

**Figure 3**: Results of Iterations Introducing MAR



Our baseline is shown by the green block with no missing data and our iterations proceed to be plotted with 10%, 20%, and 30% missing data respectively. We note that the model fit actually improves with 10% of RM and AGE set to null after accounting for NOX. Similar to our MCAR analysis, we can attribute this to the random selection of the candidate values for removal. As more of the candidate values are removed, we observe that the model fit degrades as we would expect, as evidenced by the increase in MSE and reduction in .

The results of the MNAR scheme further demonstrate the impact of missing values in model fit. Table 3 below shows our baseline against our MNAR result where we replaced the top 25% of values in CRIM (per capita crime rate by town) with nulls.

**Table 3**: Baseline MSE and Compared with MNAR Scheme

|  |  |  |
| --- | --- | --- |
|  | **Mean Squared Error (MSE)** |  |
| **Baseline** | 22.4297 | 0.7343 |
| **MNAR Scheme** | 23.0288 | 0.7272 |

This shows a degradation in fit between the two datasets as evidenced by the increasing MSE and reduction in .

## **Conclusions**

In conclusion, it’s clear that the imputation of missing data will never restore the information that would be provided if accurately sampled from within a population. However, as most statistical learning methods require complete data, it’s very hard to remove entire observations from consideration. Data scientists should be diligent in their approach to reduce bias and document their methods when explaining their research.

We can also see from our results that low percentages of imputed data may be safe to use for general conclusions, since the MSE and metrics are not highly impacted until after the 10% threshold for imputed data. Choosing a 10% or lower maximum threshold for imputing data in future data analyses may prove to be a good rule of thumb to adopt until such a time as an official body, either governmental regulatory body or professional association, provide a generally accepted threshold guideline for data scientists to use.

## **Future Work**

Our future work should include a deeper comparison of how various imputation methods affect missing data for a given circumstance. Our research in this case study included simple regression and a 2 step stepwise regression. There are many more imputation schemes available, including: mean, median, hot deck, cold deck, stochastic regression, tree based, and neural networks (including deep learning). Our research could compare these methods with the missing schemes discussed in this paper: MCAR, MAR, and MNAR, in order to determine which imputation methods are the most effective given the data. Additional research could also explore the implications of imputation from a personal bias perspective. For example - naively using a previously imputed dataset for inference into an application in which it wasn’t intended. Our research could introduce imputations to explore a specific question of interest, and then measure its effectiveness against a different question of interest.

## **References**

1. Experian. (2013). *The State of Data Quality: An Experian Data Quality White Paper*. Retrieved from <https://www.experian.com/assets/decision-analytics/white-papers/the%20state%20of%20data%20quality.pdf>
2. sklearn.impute.IterativeImputer. (n.d.) Retrieved July 14, 2019 from <https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html>

## 

## **Appendix - Code**