Case Study 5, Unit 10 - Working with Missing Data
Daniel Davieau, Lei Jiang, Kristen Rollins
17 March 2020

Introduction

In a data scientist's ideal world, every dataset would be clean and complete, but this is certainly not the case in the real world. Missing values often plague datasets, and it can be difficult to determine why those values came to be missing, as well as the best imputation method to use. In this case we will explore types of missing data that may be present, and experiment with how imputation can affect the results of our models in these different scenarios.

Background

The types of missing data we will encounter in this case study are Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR). In the MCAR scheme, data values are missing for no particular reason, i.e. it is totally random as to why some values are missing. In the MAR scheme, data are missing in only a subset of the data, which is determined by some other value. In the MNAR situation, the missing data actually contain information, i.e. there is definitely a pattern as to why the data are missing. It can be difficult to determine which case we are dealing with, but it is important to understand the differences so we can more easily recognize them and deal with them appropriately.

In this case study we will utilize the Boston Housing dataset, which is commonly used for introductory data science projects. This is because it is not too large (506 rows, 13 columns) and it is a very clean dataset (no missing values). In this case we will simulate missing values occurring in different quantities from each of the three missing data scenarios. We will then impute the missing values, run a linear regression model to predict median home value, and compare to the original regression model where no data values were missing. In this way we can attempt to evaluate the effect of missing data on model performance.

Step/Question 1 - Baseline

We first fit a linear regression model to 70% of the original Boston Housing dataset, and tested this model on the other 30% of the data. In this model we included all 12 available regressors to predict the MEDV variable, which is the median value of homes in a given town, as was done in the provided code. We did notice from a heatmap of the variables (Figure 4 in the Appendix) that the variables RM and LSTAT are somewhat highly correlated with each other (-0.614). This could contribute to multicollinearity which we would want to explore further if we were building a final model for prediction. However, we continued to fit the model on all variables so that we could make valid comparisons between different missing value scenarios.

When we tested this linear model, we measured several metrics to assess the fit. We measured Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) as loss functions, which are indications of how far off the prediction was from the

actual target value for each test point. For these loss functions, lower values are better. These are all relative metrics, so they will have more meaning once we compare them to other models fit on the same test set. We also measured R squared to indicate the goodness of fit of our model, which is on a scale of 0 to 1, where 1 is best and means that 100% of the variance of the target was explained by the regression variables. For this baseline model that used the complete Boston Housing dataset, we achieved an MAE of 3.604, MSE of 24.098, RMSE of 4.909, and R squared of 0.705. These baseline metrics are also displayed in Table 1.

Table 1. Baseline Model - Loss and Goodness of Fit

Data	**Imputation**	**MAE**	**MSE**	**RMSE**	R^2
Original	None	3.604571	24.098505	4.909023	0.704940

Step/Question 2 - MCAR

Next we selected a single column to remove different percentages of values from and then impute. We chose the column DIS somewhat randomly out of variables that had nonzero feature importance in the baseline model, seen in Figure 5 in the Appendix. We also experimented with removing values from the NOX column which has a very large coefficient, but choosing that column ultimately produced similar results. So we removed 1, 5, 10, 20, 33, and 50 percent of values (chosen at random) from the DIS column, and imputed them to the mean of the remaining values in the column. We decided to impute to the mean for simplicity, and because we knew we could continue using this method for all the missing value scenarios. The results of these models on MCAR data are found in Table 2, while their comparison to metrics of the baseline model can be found in Table 3.

Table 2. Raw Metrics for Models on Imputed MCAR Data

Data	**Imputation**	**MAE**	**MSE**	**RMSE**	R^2
Original	None	3.604571	24.098505	4.909023	0.704940
1% MCAR	Mean	3.638791	24.316646	4.931191	0.702269
5% MCAR	Mean	3.553488	23.977540	4.896687	0.706421
10% MCAR	Mean	3.547616	24.027511	4.901786	0.705809
20% MCAR	Mean	3.641010	25.303311	5.030240	0.690188
33% MCAR	Mean	3.707851	26.073462	5.106218	0.680759
50% MCAR	Mean	3.624329	25.662844	5.065851	0.685786

Table 3. MCAR Models Compared to Baseline

Data	Imputation	MAE diff	MSE diff	RMSE diff	R^2 diff
1% MCAR	Mean	0.034219	0.218141	0.022168	-0.002671
5% MCAR	Mean	-0.051083	-0.120964	-0.012336	0.001481
10% MCAR	Mean	-0.056955	-0.070994	-0.007236	0.000869
20% MCAR	Mean	0.036439	1.204806	0.121217	-0.014752
33% MCAR	Mean	0.103280	1.974957	0.197195	-0.024181
50% MCAR	Mean	0.019757	1.564339	0.156828	-0.019154

We found from Table 3 that when 1%, 20%, 33%, and 50% of values were Missing Completely at Random, the loss and R^2 metrics showed worse performance (increase in losses, decrease in R squared). This is what we expected, since information was removed and imputed from those datasets. However, we found that when 5% and 10% of values were missing completely at random, the performance actually improved slightly from the baseline. We did not expect this, and believe the phenomena might be similar to a case of overfitting, or occurred based on data that happened to be in the test set. We could use cross-validation to determine whether this was a fluke or whether imputing these cases actually improves the performance consistently.

Step/Question 3 - MAR

Next we simulated a Missing at Random (MAR) scenario. To do this, we chose the column INDUS as a control, so that for rows where the INDUS was greater than 7 (the median of the INDUS values so that sufficient data would be available), we removed 10, 20, or 30 percent of values in the DIS and NOX variables. We then imputed those missing values to the mean of the respective column. The results of simulating and imputing the MAR scenario are found in Table 4 (raw metrics) and Table 5 (comparison to baseline).

Table 4. Raw Metrics for Models on Imputed MAR Data

Data	Imputation	MAE	MSE	RMSE	R^2
Original	None	3.604571	24.098505	4.909023	0.704940
10% MAR	Mean	3.630117	24.335170	4.933069	0.702042
20% MAR	Mean	3.611678	24.237506	4.923160	0.703238
30% MAR	Mean	3.619581	24.641156	4.963986	0.698296

Table 5. MAR Models Compared to Baseline

Data	Imputation	MAE diff	MSE diff	RMSE diff	R^2 diff
10% MAR	Mean	0.025546	0.236665	0.024046	-0.002898
20% MAR	Mean	0.007107	0.139001	0.014137	-0.001702
30% MAR	Mean	0.015010	0.542652	0.054963	-0.006644

In Table 5 we see that imputing for the Missing at Random case resulted in worse behavior than the baseline (higher loss functions, lower R squared). We would have expected the results to increase as the percent of missing values increased, but this was not strictly the case since the 20% missing had better results than the 10% missing. Again we could investigate whether this was by chance or not by performing cross-validation.

Step/Question 4 - MNAR

Finally we simulated a Missing Not at Random (MNAR) case, which seems to be the most severe situation for missing values. In this case, we removed the whole first quartile of the DIS variable, effectively removing the lowest 25% of the values in that column. This simulates systematic missing values, though in practice we would not know this had happened. Thus we continued imputing to the mean since we probably would not have any extra information to indicate that the missing data points were lower values. The results for this model are found in Table 6 (raw metrics) and Table 7 (comparison to baseline).

Table 6. Raw Metrics for Model on Imputed MNAR Data

Data	Imputation	MAE	MSE	RMSE	R^2
Original	None	3.604571	24.098505	4.909023	0.704940
25% MNAR	Mean	3.667832	25.719009	5.071391	0.685099

Table 7. MNAR Model Compared to Baseline

Data	Imputation	MAE diff	MSE diff	RMSE diff	R^2 diff
25% MNAR	Mean	0.063261	1.620504	0.162368	-0.019841

From Table 7 we see that the performance of a linear regression model fit on the imputed MNAR data was worse than the baseline model. We expected this model on the MNAR data to be the worst model overall, but it was the second worst model after the 33% MCAR case. Thus our prediction was close in this case.

Step/Question 5 - Summary

The imputation approach we used throughout this study was imputation to the mean. That is, after we removed chunks of data from a particular column, we calculated the mean of the remaining values and substituted that in for the missing values. This is a very simple approach, but it is actually a fairly common solution when working with missing data. We chose to keep our approach simple so that we would not bias our results such that we would leak information. For example, in the MNAR scenario, we knew that the first quartile had been removed, so the mean was not very appropriate as it would overestimate all the values. However, in practice we probably would not know which type of missing data had occurred, and would have to do further exploration to determine the best imputation approach. Our primary goal was not to determine the best imputation approach for various missing data schemes, but rather see the effect of different types of missing data when an imputation scheme is kept constant. Thus we imputed to the mean in each case.

Table 8. All Models Compared to Baseline

Data	Imputation	MAE diff	MSE diff	RMSE diff	R^2 diff
1% MCAR	Mean	0.034219	0.218141	0.022168	-0.002671
5% MCAR	Mean	-0.051083	-0.120964	-0.012336	0.001481
10% MCAR	Mean	-0.056955	-0.070994	-0.007236	0.000869
20% MCAR	Mean	0.036439	1.204806	0.121217	-0.014752
33% MCAR	Mean	0.103280	1.974957	0.197195	-0.024181
50% MCAR	Mean	0.019757	1.564339	0.156828	-0.019154
10% MAR	Mean	0.025546	0.236665	0.024046	-0.002898
20% MAR	Mean	0.007107	0.139001	0.014137	-0.001702
30% MAR	Mean	0.015010	0.542652	0.054963	-0.006644
25% MNAR	Mean	0.063261	1.620504	0.162368	-0.019841

The results from all our 10 models are combined into Table 8, which shows the differences between the models with missing data and the baseline linear regression model, for the four metrics of MAE, MSE, RMSE, and R squared. We also visualized some of the metrics in Figures 1, 2, and 3. In these visualizations we chose to focus on the MSE loss function and the R squared goodness of fit measure. We chose MSE from the three available loss functions because it is on a larger scale and thus easier to differentiate between models, and we also found that the MAE and RMSE showed very similar relationships between the models.

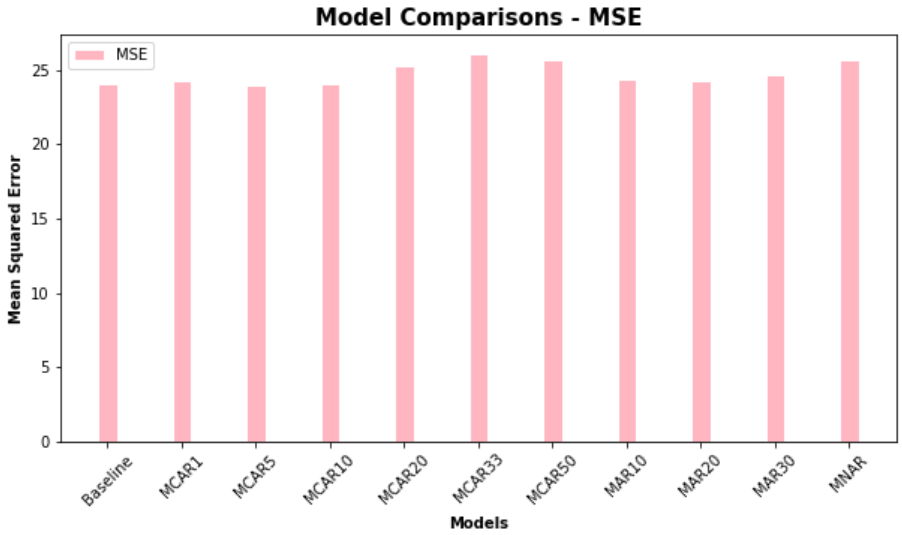

Figure 1. MSE Comparison for All Models

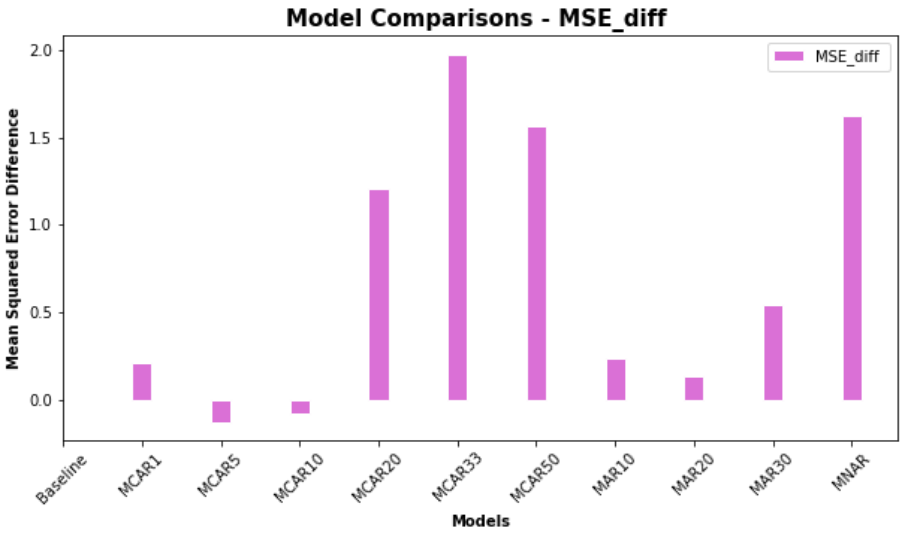

Figure 2. MSE Differences from Baseline Model

In Figure 1 we can see the raw Mean Squared Error (MSE) loss for each of the models. From this we observed some differences, but no drastic changes. The largest difference from the baseline was the 33% MCAR case, which showed about an 8.2% increase in MSE. This performance difference is certainly noticeable, but not so much to deem the MCAR model to be worthless. If that much data was missing from an actual dataset, we might be lucky to only suffer an 8.2% increase in MSE.

Nevertheless, to compare the models more closely, we created Figure 2 which zooms in to show the differences in MSE from the baseline model, and Figure 3 which shows the differences in R squared. We expected the MSE losses to increase steadily from left to right (and the R squared values to decrease) since they seem to be in an order of increased severity

of missingness. We did observe that models in the 20%, 30% and 50% MCAR and the MNAR scenarios performed the worst, and those missing values did have substantial effect on the models. However, an unexpected result was that the 5% and 10% MCAR models showed improvement over the baseline in both MSE and R squared. The improvement seen for these models was very slight, however, so we believe this was likely due to chance since theoretically a model should not perform better when it has less information. We could perform cross-validation to confirm our suspicions. Finally we observed that the Missing at Random (MAR) had better performance than we expected in relation to other models. We believe this happened because we removed 10% of values when another variable hit a threshold, which only occurred in half of the whole dataset (since we chose a median value for the threshold). Thus values were only removed from 5% of the total dataset (and 10% and 15% for the other MAR models), so the performance for these MAR models were better than we had originally anticipated.

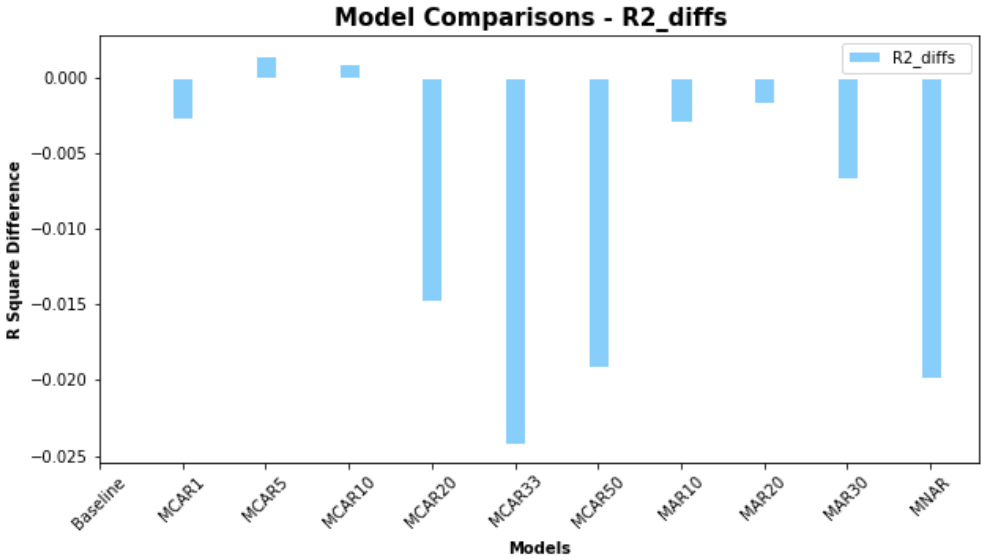

Figure 3. R Squared Differences from Baseline Model

In summary, we found that in all but two cases, removing and imputing values to the mean resulted in decreased performance compared to a baseline linear model using the complete dataset. The worst cases were the 33% MCAR and the MNAR scenarios. Also, within a certain missing data schema (i.e. MCAR or MAR) a greater percentage missing generally (though not strictly) corresponded with results further from the baseline model. We expected more strict sequences of decreasing performance (i.e. the 50% MCAR model would be worse than the 33% MCAR model), but this was not always the case. We also observed that two models (5% and 10% MCAR) actually improved slightly over the baseline model, but we believe this occurred by chance because models theoretically should not perform better with less information. These results apply to using a single imputation method (imputing to the mean) across all missing data scenarios. Other imputation methods could be more effective for various

missing data scenarios, but overall we were able to observe that missing values can have a substantial impact on the performance of models even after they are imputed.

Conclusions

Through our investigation we found that missing data does have an effect on model performance, to varying degrees. Further use of cross-validation could build more confidence as to the exact extent of that effect, but we did observe a decrease in performance for all but two models that had imputed some missing data. We found that models with large quantities of Missing Completely at Random (MCAR) as well as Missing Not at Random (MNAR) data experienced the greatest drop in performance. However, even in the worst of these models, the decline was not drastic. The MSE only increased by around 2 units (when the baseline was fairly high at 24, so it was only an 8.2% increase), and the R squared only decreased by about 0.025. So we concluded that missing values will have an effect, but it seems that even a simple imputation method such as imputing to the mean can help salvage a dataset with many missing values. We believe more advanced imputation methods could improve the models further, to bring them even closer to achieving the performance of the baseline model that had the advantage of no missing data.

References

- Python code provided by professor Blanchard
- Missing data slide deck from professor Blanchard

Appendix

Supplemental Figures/Tables

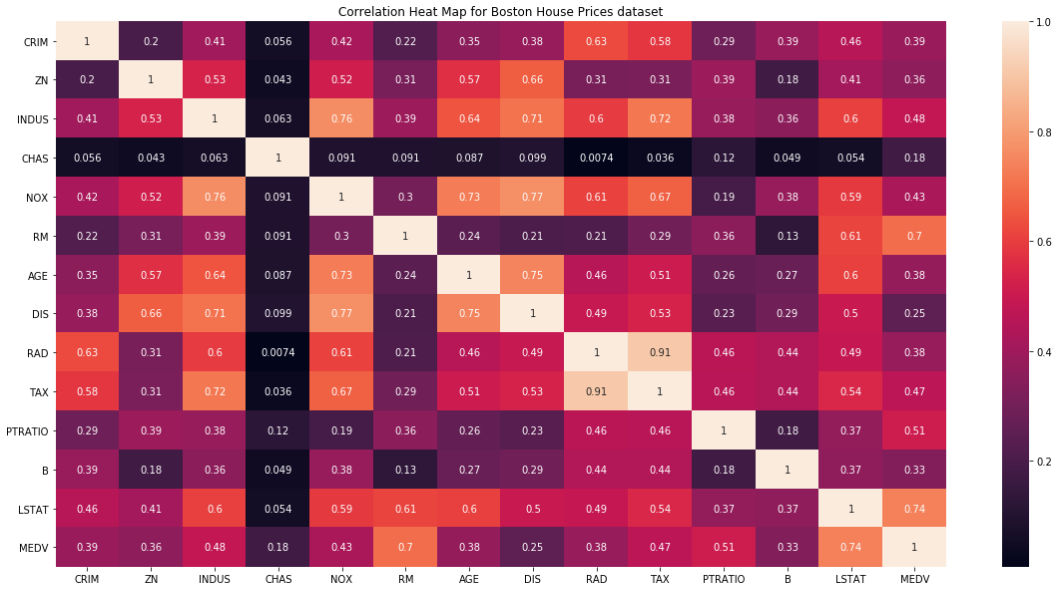

Figure 4. Correlation Heatmap (Absolute Values)

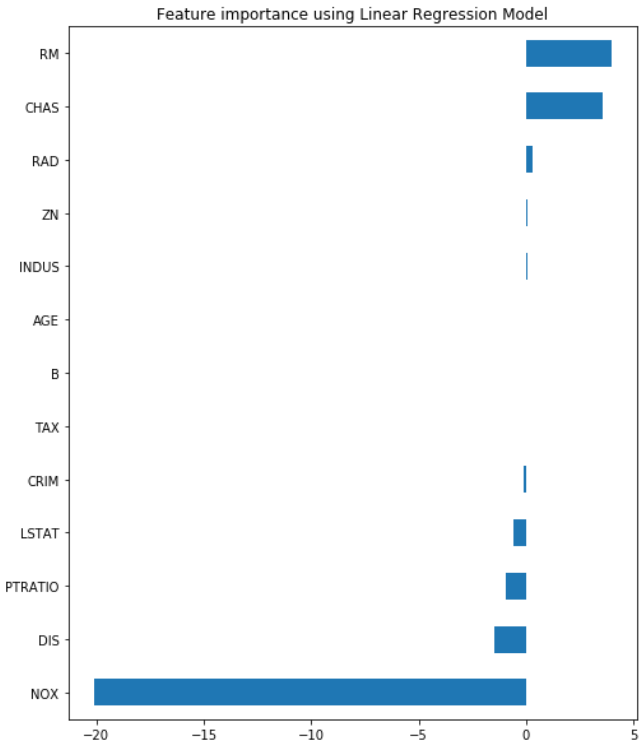

Figure 5. Feature Importance of Linear Regressors

Boston Housing Attribute Information

- 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
- MEDV Median value of owner-occupied homes in $1000's

Codebase

Our full codebase has been included in a file called lJiang_kRollins_dDavieauCaseStudy5 Update.ipynb.

Assignment

- **Step 1:** Use sklearn.datasets to get the Boston Housing dataset. Fit a linear regressor to the data as a baseline. There is no need to do Cross-Validation. We will simply be exploring the change in results.
- **Question 1:** What is the loss and what are the goodness of fit parameters? This will be our baseline for comparison.
- **Step 2:** (repeat for each percentage value below) Select 1%, 5% 10%, 20%, 33%, and 50% of your data in a single column [hold that column selection constant throughout all iterations] (Completely at random), replace the original value with a NaN (i.e., "not a number" – ex., np.nan) and then perform an imputation for the missing values.
- **Question 2:** In each case [1%, 5%, 10%, 20%, 33%, 50%] perform a fit with the imputed data and compare the loss and goodness of fit to your baseline. [Note: you should have (6) models to compare against your baseline at this point.]
- **Step 3:** Take two columns and create data "Missing at Random" when controlled for a third variable (i.e., if Variable Z is > 30, then Variables X, Y are randomly missing). Use your preferred imputation method to fill in 10%, 20% and 30% of your missing data.
- **Question 3:** In each case [10%, 20%, 30%] perform a fit with the imputed data and compare the loss and goodness of fit to your baseline. [Note: you should have (9) models to compare against your baseline at this point.]
- **Step 4:** Create a "Missing Not at Random" pattern in which 25% of the data is missing for a single column.
- **Question 4:** Perform a fit with the imputed data [25%] and compare the loss and goodness of fit to your baseline. [Note: you should have (10) models to compare against your baseline at this point.]
- **Step 5:** Describe your imputation approach and summarize your findings. What impact did the missing data have on your baseline model's performance?