Data Science Post Block Assignment 3: Task A

Sarel Vermaak [190980235], Kebaabetswe Tlhoaele [28816749], Lize Mostert [23537140],
Difedile Rasenyalo [28294882] & Isabel de Waal [20805055]

1. Introduction

Background

Missing data is a common data quality issue encountered during the data exploration and cleaning phases of the CRISP-DM process. Missing values can be introduced during data integration, which can usually be easily fixed to resolve the missing value issues [1]. On the other hand, missing values can also be introduced during the data generation or collection phases – these are more difficult to deal with. There are three typical approaches for dealing with missing values. One is to simply remove instances or features that contain them. This is not the best approach as it could lead to the loss of valuable information and lead to bias during inference [2]. Another method is to convert the missing value into a new feature, but this has shown to lead to serious inference problems [3 according to 2]. Finally, one can impute the missing values. Imputation is generally a good idea if a significant portion of the data contains missing values for a few features. Generally, imputation above 60% of missing values are not recommended [1].

Objectives

Task A: The objective was to evaluate the effectiveness of a baseline imputation versus that of a Naïve Bayes imputation on the performance of classification models. For this study we also decided to look at the influence of the proportion of missing values on the effectiveness of the imputation method. This will be done by comparing the performance of two classification models who have been trained on the control data and the imputed data and then tested with unchanged data.

2. Methodology

In this study, we leveraged R Studio and Python as the primary tools for imputation, modelling, and evaluation.

Data Exploration: The first phase of the analysis involved exploring the Nursery dataset to gain a comprehensive understanding of their structure, variables, and content.

Data Pre-processing: For this experiment, the pre-processing consisted of splitting the data into a 30:70 test: train split. The testing dataset was then reserved for later use. Three copies of the training data-subset were made and were induced with missing values in varying proportions (10%, 40% and 70%). Following this the induced missing values were then imputed using mode imputation and Naïve Bayes imputation. After imputation there were 7 training sets namely: control, mode_10, mode_40, mode_70, nb_10, nb_40, nb_70. These were all tested with the same test set.

Modelling: K-Nearest Neighbours (k-NN) Classifier and a Classification Tree (CT) were used to evaluate the effectiveness of the data imputation by comparing the performance of the models that were trained on the imputed datasets to models trained on the control dataset. Each modelling approach required additional pre-processing steps and hyper parameter selection:

Classification tree: Due to the nature of the model used in python, the features had to be converted into factors. The optimal tree depth was determined by comparing the training and testing F1 scores of the control datasets at various values for` tree_depth`. There was no clear point at which performance started to deteriorate, so the point at which the performance stopped improving was chosen as the optimal tree depth.

k-Nearest Neighbour: Due to the nature of the k-NN model, the categorical features had to be converted to factors prior to modelling. Two methods were utilized for establishing the k value in the training and testing datasets: the Grid Search method and cross-validation of the F1 score. The Grid Search involved evaluating model performance across a defined range of k values, initially spanning from 2 to 95 and later narrowed down to 2 to 16. Despite adjustments, the optimal k value remained consistent at 5. It was decided to keep this hyperparameter constant in all the models, in order to better evaluate the effect of the imputation.

After these pre-processing steps the same modelling and evaluation approach was used: 7 individual models were trained each using one of the 7 different training sets (control, mode_10, mode_40, mode_70, nb_10, nb_40, nb_70) These were all tested using the same testing set (which is 30% of the original dataset). After training the performance of the models were evaluated by calculating the accuracy, precision, recall and F1 scores.

Visualization: Bar plots were constructed in Microsoft Excel 365 to clearly visualise the difference in performance of the two models when trained with data that was imputed in two different ways.

3. Results and Discussion

Although the full 'suite' of classification performance metrics was calculated, the accuracy and F1 scores were deemed to be the most informative and is what will be discussed in this section. Accuracy in machine learning serves as a measure of overall correctness, representing the ratio of correctly classified instances to the total instances. Conversely, the F1 score, which ranges from 0 to 1, represents the harmonic mean of precision and recall, both crucial evaluation metrics.

It is important to note that the prediction target was a heavily skewed multiclass problem. When reporting accuracy and F1 scores, this is a weighted average across the 5 potential prediction classes. This weight is based on the proportion of values in each class, i.e. larger classes will contribute more to the score.

Classification Tree results: Figure 1A&B (right) shows the accuracy (A) and F1(B) scores of the CT. Generally, the results were quite interesting. Figure 1 (right) shows the performance metrics of each of the classification tree for each imputation method per proportion of missing values. As can be seen by the accuracy and F1 Score graphs (Figure 1A &B), both techniques performed very well at 10% missing values, with performance degrading as more missing values were introduced and imputed. The mode imputation and NB imputation performed remarkably similar at all levels of missing values tested (Accuracy and F1 score within 3% difference). Surprisingly, the classifier still predicted with remarkable accuracy at 70% missing values imputed. The reason could possibly be because of the skew prediction class and that the missing values were introduced randomly, which meant that on average the skewed class distributions would be maintained, and mode imputation would still guess the correct prediction class quite often.

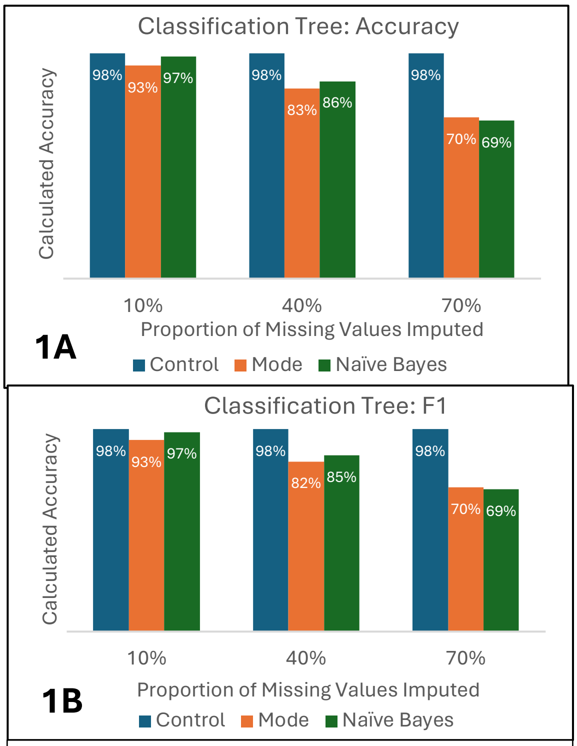

Figure 1 A&B: Bar charts visualising the Accuracy (A) and F1 Scores(B) for the classification tree model.

k-Nearest Neighbour results: During the analysis of accuracy (Figure 2A) levels across various datasets with differing proportions of missing values, noticeable instabilities were observed. Specifically, the accuracy trends varied across different datasets, with K-NN initially exhibiting high accuracy for the mode data at 10%, but significantly underperforming for mode 40% and mode 70%. A similar trend was noted for Naïve Bayes. The model's performance decreased notably for Naïve Bayes at 40% and 70% missing values. The F1 score (Figure 2B) mirrored the accuracy trends, with the K-NN model performing well initially at 10% missing values but experiencing a decline for mode and Naïve Bayes at 40% and 70%. It's important to note that a higher F1 score indicates better model performance, yet in this scenario, there was a significant decrease in K-NN performance. This decline could be attributed to the skewed distribution of data and the sensitivity of K-NN to missing values. Overall, it would seem that the Naïve Bayes imputation performed much better that the mode imputation for the k-NN model.

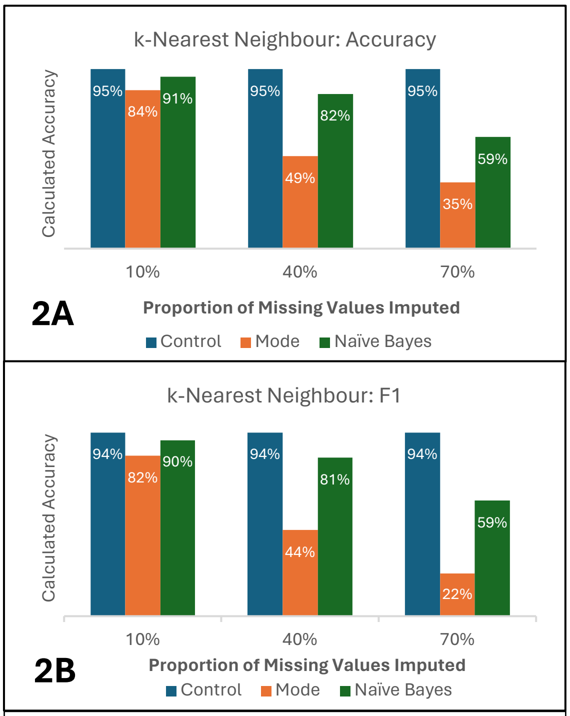

Figure 2 A&B: Bar charts visualising the Accuracy (A) and F1 Scores(B) for the classification tree model.

Farhangfar *et.al.* conducted a similar (albeit infinitely more complex) study [3]. In it they tested 6 different imputation methods with 6 different classification models using the aggregated scores of 15 datasets (of which the 'Nursery" dataset used in this assignment was one). As their study was much more nuanced and they used a unique way to measure model performance, direct comparisons were tricky. What was clear from their study however was that different imputation methods worked better for some models than others, and that there was no 'universally better' imputation method. This was also observed during this study and supporting our findings. Of all the imputation methods in their study, the mean imputation performed the worst. For the k-NN model, we found similar results whereas we did not see this response in the classification tree model. This could simply be explained by the different methodologies used for calculating the performance. Farhangfar *et.al.* measured their performance by calculating the classification error which was based on a zero-one loss [3].

4. **Conclusion**

In conclusion, the effectiveness of an imputation method is highly dependent on the type of model to be used. In this study it was found that the Naïve Bayes imputation suited the k-NN model better whereas the classification tree was more robust and the different imputation techniques performed about the same. In these cases, the simpler mean imputation would be recommended.

5. **References**

[1] J. D. Kelleher, D. Aoife, and M. N. Brian, "Data Exploration," in Fundamentals of Machine Learning for Predictive Data Analytics, 2nd ed, Cambridge, Massachusetts: The MIT Press, 2020, p. 63

[2] A. Farhangfar, L. Kurgan, and J. Dy, "Impact of imputation of missing values on classification error for discrete data," Pattern Recognition, vol. 41, no. 12, pp. 3692–3705, Dec. 2008. doi:10.1016/j.patcog.2008.05.019

[3] J. L. Schafer, Analysis of incomplete multivariate data, Aug. 1997. doi:10.1201/9781439821862