**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. Often missing values can creep in during data integration, in these cases, the integration errors can be 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 various ways to deal 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.

1. **Methodology**

In this study, we leveraged R Studio, Orange 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%). These together with the training control was then used during the modelling. 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. 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. These were all tested using the same testing set. After training the performance of the models were evaluated by calculating the accuracy, precision, recall and F1 scores.

*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.

*k-Nearest Neighbour:* k-NN works on a distance calculation, the categorical features had to be converted to factors. K was decided to be the square root of the total number of observations within the data set.

For the k-NN model there was a limitation in using R. During pre-processing it was found that some of the data still contained missing values. After meticulous searching and de-bugging attempts, the source of these missing values could not be determined. This issue was not found when using Orange data mining software, so this modelling portion was done using orange.

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.

1. 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. 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.

**1A**

**1B**

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

After pre-processing, the optimal number of clusters were to be selected for the K-means clustering.

In the report, it's crucial to summarize the significance of accuracy and F1 score evaluation metrics, particularly in the context of classification tasks, as they offer distinct insights into model performance. Accuracy in machine learning denotes the ratio of correctly classified instances to the total instances, serving as a measure of overall correctness. On the other hand, the F1 score represents the harmonic mean of precision and recall, both essential evaluation metrics. Ranging from 0 to 1, a higher F1 score indicates better model performance.

**2A**

Upon examining the accuracy levels across different datasets with varying proportions of missing values, noticeable fluctuations are observed. Notably, the accuracy trends vary for different datasets, with K-NN showing high accuracy initially for the naïve Bayes data, followed by a decline and subsequent increase. This indicates a gradual establishment of correctness proportions across the mode datasets at 10, 40, and 70.

**2B**

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

1. Conclusion

In conclusion, the analysis of the

This was showed to be quite sensitive to the skew class distribution, and different results may be obtained from datasets with more even class distributions and non-random missing values.

1. References
2. [1]
3. 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 J.L. Shafer, Analysis of Incomplete Multivariate Data, Chapman and Hall, London, 1997.