**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** 
   1. **Background**

Missing data is a common data quality issue encountered during the data exploration and cleaning phases. If features have hissing values, it is important to determine why these values are missing. 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 may 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].

There are different imputation methodologies, of which we will be focusing on the following two:

*Baseline imputation (data driven):* Imputing the variable simply based on the mode(categorical) or mean/median(numerical).

*Model Imputation:* a model is used to perform parameter estimation.

Studies have found that the ‘best’ imputation method depends on the type of data in the dataset, the proportion of missing values and even the distribution of the features have an influence on the type of imputation that is considered ‘best’ [2].

* 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 and Python as the primary tools for imputation, modelling, and evaluation. The methodology involved several key steps:

**Data Exploration:** The first phase of the analysis involved exploring the Nursery dataset to gain a comprehensive understanding of their structure, variables, and content. As the main aim of this experiment is to evaluate the effectiveness of two different imputation methods, it was essential that the data be as clean as possible, as not to confuse any possible insights.

**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.

**Modelling:** K-Nearest Neighbours Classifier and a Classification Tree was used to evaluate the effectiveness of the data Imputation.

k-Nearest Neighbour: [insert chosen k= 5 across the data sets and method here]

Determining the optimal k value is pivotal for addressing the objectives outlined in this report, which include comparing the efficacy of baseline imputation against Naïve Bayes imputation on classification model performance. Additionally, the aim is to examine how the proportion of missing values influences the effectiveness of imputation methods, specifically mode value imputation, for K-Nearest Neighbours (K-NN) and Classification Tree models.

In the context of the Nursery dataset, where the target variable pertains to classifying nursery applications into categories like "not recommend," "priority," and so forth, the optimal k value holds significance. This value aids in dividing the data into clusters, reflecting different parental occupations, financial statuses, and social and health aspects. Understanding these clusters can inform school enrolment decisions, especially during periods of high application rates in Ljubljana, Slovenia during the 1980s. The decision-making process for nursery admissions relied on factors such as parental occupation, family structure, financial standing, and social and health circumstances.

In determining the optimal k value for the K-NN classifier, several methods can be employed. One approach involves calculating the square root of the dataset's instance count to derive k. However, selecting a high k value may lead to underfitting, while a low k value can result in overfitting, compromising the model's predictive performance. Therefore, it's crucial to strike a balance between capturing data patterns effectively and avoiding excessive complexity in decision boundaries.

In this research paper, 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.

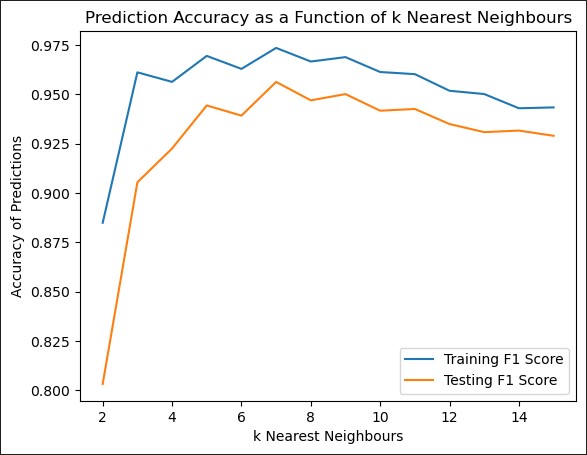


Figure 1:

It was found that the testing F1 score performs similarly to the training F1 score and initially peaking at K= 5. Why this was the preferred method to apply optimal K is because the data set had already been varied and to see how best the model performs as a scientific practice would be to maintain one factor or aspect of the variable constant across. This may not be ideal for other kinds of datasets where a prediction between an accurate diagnosis is life threatening or not.

The investigation revealed that the testing F1 score closely matched the training F1 score, with both reaching a peak at K=5. This choice of selecting K=5 as the optimal value was preferred because the dataset had already undergone sufficient variation, and maintaining consistency in one aspect of the variable aligns with scientific practice. However, it's important to note that this approach may not be suitable for datasets where accurate predictions are critical, such as those involving life-threatening diagnoses.

The K\_ Nearest Neighbours (K-NN) classifier was used as one of the models to train on and validate the performance of the imputation techniques. We used the K-NN classifier in the R studio and Python to determine how K-NN responds to the various data sets namely, original training sets, original testing, mode data at 10, mode at 40, mode at 70, naïve bayes at 10, naïve bayes at 40, naïve bayes at 70 of the public nursery data set.

The implication of the K-NN is that the model is a simple model that works for both categorical and numerical data sets. This implementation, therefore, required that the data sets be handled in a way the k- NN algorithm could provide outputs into whether the algorithm could effectively learn the data set and predict outcomes through the performance metrics of the accuracy and the F1 score. The K-NN was therefore compared to the control and between the different imputed data of the same Nursery data set. The below steps were the coding methodology utilised in R studio to build the K-NN.

* Importing of the data sets,
* Converting the character data into a factor,
* Converting the factor of the variables within the data sets as numeric,
* Handling missing values if any are identified,
* Training the data and testing data sets,
* Determining the optimal value of k,
* Creating predications,
* Developing the K\_NN model from the trained, test and target vectors, and
* Calculating the performance metrices and the confusions matrix.

All the above steps were taken for each of the data sets mentioned above, however noteworthy without ease, particularly for the mode 40 data set, and the naïve bayes data sets at the 40, 70 percentages of imputation.

We then trained the model on the various datasets and tested against the non-imputed base table. For each dataset we recorded the accuracy and F1 score, which was then used to compare model performance across different datasets. We trained on the base dataset (no missing values induced, or missing values imputed) and report the accuracy and F1 score for this model as the baseline values. These baseline values are then compared to the paired imputed datasets (Same % of missing values imputed using Mode value and Naïve Bayes) against the K-NN Accuracy and F1 score values.

1. **LIMITATION**

Using R Studio for this exercise posed several challenges, primarily due to encountering missing values in the target variable (class) within the original training dataset, as well as in other datasets, including other variables. These missing values were unexpected, considering the initial collation of the dataset was done without any such gaps. Addressing these missing values became necessary as K-NN generated error messages due to their presence, preventing the application of the algorithm.

The occurrence of missing values in R Studio could be attributed to various factors. Firstly, there might have been issues with data file encoding, leading to misinterpretation of certain characters or symbols. Secondly, incorrect interpretation of data types by R, such as numeric values being stored as character strings, could have contributed. Additionally, special characters or non-standard symbols might have caused misinterpretation. Lastly, file corruption or data file structure issues could have played a role in R misreading the data.

Classification tree: [insert parameter settings here]

**Visualization:** Line plots were constructed 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**

In the report, it's essential to highlight the significance of accuracy and F1 score evaluation metrics, particularly within the context of classification tasks, as they offer distinct insights into model performance. 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.

During the analysis of accuracy 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.

Similarly, the F1 score 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. Missing values impact distance calculations, thus influencing the classification or regression outcome, aligning with previous research findings. K-NN is expected to decline should there more missing data.

Figure 2: K-NN Accuracy performance metric graph comparing the control, mode and naïve bayes imputation techniques.

Figure 3: K-NN F1 performance metric graph comparing the control, mode and naïve bayes imputation techniques.

1. **Conclusion**

In conclusion, the analysis of the

1. **References**
2. AML textbook, page 64
3. Impact of imputation of missing values on classification error for discrete data - article
4. J.L. Shafer, Analysis of Incomplete Multivariate Data, Chapman and Hall, London, 1997.