**PREDICTIVE MODEL FOR HANDLING MISSING DATA**

**Abstract :**

Missing data presents a problem in many fields ,including machine learning. The data can be missing at random, in recurring patterns, or in large sections. Incomplete datasets can lead to misleading conclusions. It is important to identify, mark and handle missing data when developing machine learning models in order to get the very best performance. Performance of machine learning model is based imputation method. This method will helpful to discuss if there are patterns in the missing data. This is by far one of the best and most efficient method for handling missing data. Depending on the class of data that is missing, one can either use a regression model or classification to predict missing data. This works by turning missing features to labels themselves and now using columns without missing values to predict columns with missing values.

**Introduction :**

In datasets have a large percentage of missing values that directly impacts their usefulness in yielding high accuracy classifiers when used for training in supervised machine learning. While missing value imputation methods have been shown to work well with smaller percentages of missing values.

**There are some method to handling imputation**

1. Deleting Rows
2. Replacing with mean/median/mode
3. Filling Null Values
4. **k-Nearest Neighbour (KNN) Imputation**

**Deleting Rows :**

This method commonly used to handle the null values. Here, we either delete a particular row if it has a null value for a particular feature and a particular column if it has more than 70-75% of missing values. Removing the data will lead to loss of information which will not give the expected results while predicting the output.

### Pros:

* Complete removal of data with missing values results in robust and highly accurate model
* Deleting a particular row or a column with no specific information is better, since it does not have a high weightage.

### Cons:

* Loss of information and data
* Works poorly if the percentage of missing values is high (say 30%), compared to the whole dataset

**Replacing with mean/median/mode**

We can calculate the mean, median or mode of the feature and replace it with the missing values. Replacing with the above three approximations are a statistical approach of handling the missing values. This method is also called as leaking the data while training. Another way is to approximate it with the deviation of neighbouring values. This works better if the data is linear.

### Pros:

* This is a better approach when the data size is small
* It can prevent data loss which results in removal of the rows and columns

### Cons:

* Imputing the approximations add variance and bias
* Works poorly compared to other multiple-imputations method

## Filling Null Values

Sometimes rather than dropping NA values, you’d rather replace them with a valid value. This value might be a single number like zero, or it might be some sort of imputation or interpolation from the good values.

### Pros:

* Less possibilities with one extra category, resulting in low variance after one hot encoding — since it is categorical
* Negates the loss of data by adding an unique category

### Cons:

* Adds less variance
* Adds another feature to the model while encoding, which may result in poor performance

**k-Nearest Neighbour (KNN) Imputation**

For k-Nearest Neighbour imputation, the missing values are based on a KNN algorithm. These values are obtained by using similarity-based methods that rely on distance metrics (Euclidean distance, etc). They can be used to predict both discrete and continuous attributes. The main disadvantage of using KNN imputation is that it becomes time-consuming when analyzing large datasets because it searches for similar instances through all the dataset.

In this particular dataset, taking into account the person’s age, sex, class etc, we will assume that people having same data for the above mentioned features will have the same kind of fare.

Unfortunately, the SciKit Learn library for the K – Nearest Neighbour algorithm in Python does not support the presence of the missing values. Otherwise separate python code for do KNN on dataset but limitation is after imputing with KNN some of missing values remains.

But fancy imputation method supports presence of the missing values.

Another algorithm which can be used here is Random Forest. This model produces a robust result because it works well on non-linear and the categorical data. It adapts to the data structure taking into consideration of the high variance or the bias, producing better results on large datasets.

### Pros:

* Does not require creation of a predictive model for each attribute with missing data in the dataset
* Correlation of the data is neglected

### Cons:

* Is a very time consuming process and it can be critical in data mining where large databases are being extracted
* Choice of distance functions can be Euclidean, Manhattan etc. which is do not yield a robust result

Now predicting missing values using predictive models.

**Predictive Model For Handling Missing Data :**

we can predict the nulls with the help of a machine learning algorithm. This method may result in better accuracy, unless a missing value is expected to have a very high variance. Depending on the class of data that is missing, one can either use a regression model or classification to predict missing data. This works by turning missing features to labels themselves and now using columns without missing values to predict columns with missing values.

**Example :**

If we predicting a disease based on patient (symptoms). It is sensitive issue. For suppose data was missed but filled with mean/median/mode. Model predicts patient has disease. If missing value filled with the help of pattern (Using machine learning models). Machine learning means algorithms learn patterns from data by training and predicts the outcome. Rather than academic projects, coming to real world problems. This is suggestible method to fill missing values.

## Conclusion

Every dataset we come across will almost have some missing values which need to be dealt with. But handling them in an intelligent way and giving rise to robust models is a challenging task. We have gone through a number of ways in which nulls can be replaced. It is not necessary to handle a particular dataset in one single manner. One can use various methods on different features depending on how and what the data is about. Having a small domain knowledge about the data is important, which can give you an insight about how to approach the problem.

**References:**

[1] https://res.mdpi.com/data/data-02-00008/article\_deploy/data-02-00008.pdf?filename=&attachment=1

[2] https://www.oreilly.com/learning/handling-missing-data

[3] https://towardsdatascience.com/the-tale-of-missing-values-in-python-c96beb0e8a9d

[4] https://machinelearningmastery.com/handle-missing-data-python/