'A study on Missing Value Imputation (MVI) techniques on multi-variate time series data by adjusting the missing rates using a suitable missingness mechanism over a range of datasets and compare the results of each imputation technique using suitable metrics.'

Module Code:	**BUSN9860**
Module Title:	**Dissertation**
Advisor:	**Professor Shaomin Wu**
Programme of Study:	**MSc Business Analytics**
Surname:	**Baskaran**
First Name:	**Aravindkumar**
Login:	**ab2491**
Required Word Count:	**8,000 – 10,000**
Actual Word Count:	**11,117**

I agree that copies of my report may be used as reference material by the University (please tick)

Yes [√] No [ ]

Ethics (please tick):

[ ] **I HAVE** received ethical approval from the REAG Chair before conducting my research

[ ] **I HAVE NOT** received ethical approval from the REAG Chair before conducting my research

[√] My research did not involve human participants, so I was not requiring to receive ethical approval

Table of Contents

Abstract

Missing values are a constant bane to Data analysts in the pre processing stages of a ML, DL pipeline. Even though ad hoc Imputation methods pre process the data but these ad hoc measures tend to cause bias and affect the reliability of the ML or DL model to be deployed careful considerations must be taken while imputing using Ad hoc measures such as Mean, Mode imputations. List wise deletion could be the most convenient method to treat missing values but would lead to loss of essential information. This analysis uses the concept of Multiple Imputations using different missingness proportion simulated in the range of 15% to 30%, the proportion most encountered by analysts globally. Different datasets from myriad range applications are taken up, simulated to the required missingness proportions. Then state of the art and most frequented MVI (Missing Value Imputation) techniques are used such as kNN imputer, Ensemble Trees – Using ExtraTrees as base estimator, Pattern Based Interpolation and Multiple Iterative Chained Equations (MICE) alongside forward and backward fill and ad hoc measures such as Mean, Mode Imputations. The imputed values are compared using metrics such as MAPE, MAE and MSE. Pattern Based Interpolation seems to the best model amongst the lot with error percentage ranging from 2 – 19%. Some special cases were as well observed where the Tree based, and clustering imputers performed better than Patten Based Interpolation.

1.INTRODUCTION

1.1 BACKGROUND

The possibilities that machine learning has opened in industries like insurance, banking, finance, and sports never cease to astound us. Technology advancement and the evolution of GPUs combined with fast processing rates have given rise to even more reliable algorithms, like LSTM, CNN, GRU, and ANN, which are all part of the larger network of Neural Networks (*Steinkraus, Buck, and Simard, 2005*). The use of neural networks and machine learning algorithms has altered predictive analytics, and this is due to improvements made in Big Data tools and the amount of data generated from sources such as mobile apps, sensors attached to automotive components, and other electronic devices (*Mohammed, Khan, and Bashier, 2016*).

The world of algorithms is catching up with the growth of processors at the rate that Big Data is developing, but as the volume and veracity of the data expanded, some limitations

including cleanness of the data, missing data, data trustworthiness, and other associated issues emerged (*tdwi.org, 2011*). As a result, Data Pre-processing became more important for the algorithms to function in the actual world. Data pre-processing offers a wide range of functions, including scaling down data, cleaning text-based data, converting categorical data to ordinal data, treating missing values by imputing the values that are missing, and more, depending on the application.

This study will concentrate on Missing Value Imputation, the most popular and ambiguous treatment strategy (MVI). Many datasets contain missing values, and different approaches to handling them result in varying performances of the algorithms (*Lin and Tsai ,2019*). Depending on the machine learning task under consideration—such as regression, classification, or time series—the approach would change. The subject of this study is Time Series data, and to assess the optimum imputation strategy for these datasets, we selected a wide range of criteria. The study becomes even more difficult as the number of variables in the dataset rises, and an appropriate MVI technique must be taken into consideration. Imputing missing values in the case of Time Series data would be the most difficult because the values are auto correlated with the preceding values. Data analysts frequently experience difficulties with their analyses because of missing values in the multivariate data. Ad hoc methods like listwise deletion are sometimes used to fill in the gaps left by missing data. When it comes to survey-based data collection, missingness could be due to the subjects leaving early, or they might not want to answer specific questionnaire questions. In addition to reviewing additional methods like machine learning (ML), the authors have successfully dealt with missing values using statistical methods like EM. Another ad-hoc method - Mean-median imputation can result in the loss of critical information in the modelling process, which would be exceedingly inefficient. Regression imputation for each feature, on the other hand, can result in bias if the features that are full are not representative of the entire dataset. Due to unaccounted variability, mean imputation typically makes the modelling process less effective.

This analysis would limit itself to multi-variate time series datasets obtained from well-known dataset repositories like UCI repository, and we would consider various missingness mechanisms and run the datasets through various missing rates. The effectiveness of the imputation would then be assessed using well-known metrics like MAPE (Mean Absolute Percentage Error), and the results would then be compared and analysed.

This paper consists of 5 chapters, the first chapter draws the introduction for the analysis, where the background to the analysis is laid and subsequently the problem statement is defined and then the justification for the research is enlisted. Later, in the section -1 the question the research is penned down to resolve is enlisted. Subsequently, the chapter -2 is the literature survey, where the latest research in the domain is explained in detail and the key learnings is taken from the top-notch academic journals from renounced publishing houses. Chapter – 3 lays the corner stone for the research methodology to be discussed along with emphasis on exploratory data analysis and the assumption encapsulated in this analysis. Chapter -4 discusses the analysis supported by tables and figures and list out the analysis in the order of the research questions. The final chapter, Chapter -5 concludes the analysis by summing up the key findings and stating the limitations if any and then put forth the recommendations for future study in this domain.

Intended audience – This paper would be apt for people from analytics domain, Data Scientists and people from academia who have sound knowledge over statistics, computing, and ML.

1.2 PROBLEM STATEMENT

'*A study on Missing Value Imputation (MVI) techniques on multi-variate time series data by adjusting the missing rates using a suitable missingness mechanism over a range of datasets and compare the results of each imputation technique using suitable metric*.'

Multi variate time series datasets are sourced from multiple online platforms and different MVI techniques right from conventional methods such as Mean, Most Frequent values imputation to Machine learning methods such as K-Nearest Neighbours to Ensemble trees to Pattern Sequencing methods such Interpolation techniques to Iterative Imputer using Bayesian Ridge as the learner method are tested out. The datasets are treated through Multiple Imputation using Missing Complete At Random method, setting missing rates from 15% to 30% with 5% step up in the consequent simulations. The effectiveness of each imputation is screened using MAPE, MAE and MSE as the metrics as these metrics give us the overall idea of the effectiveness of the Imputation across myriad scales of variables in each dataset.

1.3 RESEARCH MOTIVATION / JUSTIFICATION

This section discusses enlists the key motivation behind the research and its relevance to the data analysts and the industry overall.

1.3.1 Academia to Industry knowledge transfer

Time series data frequently have missing values because of inconsistent data collection, which can occur for a variety of reasons, including sensor failure to record data (*Yi, Peter et al., 2019*), server unavailability, and human mistake. Most of these contradictions have been noted for a variety of reasons, as previously said, and each of these situations tends to fit into one of the three mechanisms that we will be addressing in the following sections.

1.3.2 Overcome loss of essential data

According to prior research, if a variable's missing values fraction is between 10% and 15%, it is regarded acceptable; if it is above 15%, however, treatment must be carefully addressed. Eliminating the variable is the simplest and fastest method, but the data may become less detailed and redundant as a result. Therefore, to offset the loss in data granularity and account for higher ML model performance, an effective MVI technique is needed. This study will help businesses that use long-term forecasting and predictive time series analysis to select the best MVI technique to handle missing data.

1.3.3 MVI on Multi variate time series dataset

Time series data is more likely to contain errors for both known and unknown reasons. Since multi-variate time series data represents the actual real-world data, we can infer from a series of the relevant literature that the MVI approaches used in this research have not been tried on multi-variate time series data.

1.3.4 Effectiveness of MVI techniques on different rates

In this analysis datasets would be subjected to different missing rates using a user defined function in python to create missing values. The dataset with different missing rates would subject to the MVI techniques mentioned above and then performance would be studied such that in future it would benefit any academic or Data Analyst on to use which method to cling on to treat dataset which fall under the missing rates analysed upon in this analysis.

1.4 Research Question

This study attempts to answer certain impending questions in the field of Missing Value Imputation (MVI), following are the questions we attempt to answer through this work:

1. Which MVI technique yields better result with regards to multi-variate time series data?
2. What metric is to be gold standardized among different performance evaluation metric?
3. Among the data missing mechanisms such as MCAR, the analysis would try to understand amongst statistical and Machine Learning techniques which method handles which data missing mechanism effectively and efficiently.
4. We try to understand the limitations of these MVI techniques on multi-variate time series dataset?

1.5 Research Aims

The main aim of this study is to analyse different MVI (Missing Value Imputation) techniques on **multiple multi – variate** time series datasets obtained from **UCI data repositories**. Then we compare the results from different MVI techniques using performance evaluation metrics such as **MAPE**.

1. Understanding among the plethora of MVI technique yields better result with regards to multi-variate time series data.
2. Metrics to be gold standardized among different performance evaluation metrics.
3. Simulate MCAR data missing mechanism under different missing rates and try to understand among statistical and Machine Learning techniques which method handles which data missing mechanism effectively and efficiently.
4. We try to understand the performance of these MVI techniques each of the dataset individually and limitations of the techniques on multi-variate.

1.6 Overview: Research Methodology

This study's approach is completely quantitative research. Secondary data sources, including UCI was used to gather the data for this study. Python (Version-3) would also be heavily utilised for pre-processing and MVI method application. The platform on which the codes would be executed and on which the outputs would be graphically plotted would be Google Colab. Matplotlib, Sci-kit Learn, Seaborn, NumPy, DataFrame, and Timeseries generator are a few examples of such packages. A total of 6 datasets are drawn from a variety of applications, including factory productivity, energy production, and environmental emission. Imputation methods differ between statistical and machine learning. In the case of statistical approaches such as EM algorithms (MICE – Multiple Iterative Chained Equations), forward

fill, backward fill, mode, median, and mean are used and K-nearest Neighbour (KNN), Ensemble Trees using ExtraTrees as base estimator (ET) and other MVI techniques such as Pattern Based Interpolation are examples of Machine Learning approaches used in this analysis. If the variable exhibits periodicity and trend smoothness, then look at possibilities like Pattern Sequence Forecasting algorithms. The imputation is then evaluated using direct methods like MAPE, MAE, and MSE.

2. LITERATURE SURVEY

In late part of 1970's Rubin et al., established the concepts of missing data and its relevance to analysis and established the mechanism to missing data. Missing data mechanisms are classified into 3 – namely - Missing Completely At Random (MCAR), Missing At Random (MAR) and Missing Not At Random (MNAR) (*Rubin, 1988)* The missing mechanisms have relevance in statistical terms - MCAR implies that probability of missing values in one of the variables in the dataset is not related to missing values at any other variable in the dataset (MAR is the opposite spectrum of MCAR, the probability of missingness of a variable is related with the other this might lead to ambiguity, but it works based on conditionally missing after controlling all other variables (*Graham, 2009*). MNAR is all about missingness of data in a variable is in association with itself. In a real case where the consequence of missing data is further magnified if it's multi-variate analysis and the values are missing in more than one variable, it is impossible to unambiguously classify any missing data case into the 3 broad spectrum. it is hard to imagine missing data that are entirely unrelated to other variables in the dataset, i.e., purely MCAR. Missing data in real datasets are somewhere on a continuum from MCAR through MAR and to MNAR. In a sense, it may be easiest to think of all missing data as belonging to MAR to some degree because MAR resides in the middle of this continuum. Further details can be found in papers published by Nakagawa and Freckleton – (*Nakagawa and Freckleton, 2008).* Rogiers and Moons et al. explain the differences between single and multiple imputation before going into some of the disadvantages of single imputation, like higher estimate standard errors. The indicator approach, which involves utilising dummy variables encoded as 0 and 1 to indicate that data is missing, has certain downsides, including the possibility that it could cause estimations for the most popular epidemiological study missing mechanisms, the MCAR and MAR, to be incorrect (*Rogier, Geert, Theo, and Karel, 2006*). *Wei-Chao et al*. examines many technical challenges related to dataset selection, missing rates, missing mechanisms, and assessment metrics (*Lin and Tsai ,2019*).

Jerome et al. test out several modifications for multiple imputation in both large and small datasets (*Jerome and Trivellore, 2014*).

James and Gary et al., discuss about developing a new algorithm for treating cross – sectional time series data in the domain of political science and were astounded by the results the software was able to produce *(Honaker and King, 2010).*

Missing values are now regularly encountered throughout the data recording process, and complete data is necessary for further analysis. A thorough analysis of the MVI techniques that are now available, including traditional and algorithm-based techniques that can reduce the bias effects of conventional methods (*Irfan, Adhistya, Igi and Rini, 2016*). Three univariate time series datasets were used in *Neeraj et al's* discussions on the Pattern Sequence Forecasting technique to impute time series datasets. The Pattern Sequence Forecasting algorithm was used to impute features with repetitive and periodic patterns (*Neeraj, Francisco, and Marcus et al., 2018*). Peter and Ian et al. discuss this concept about Multiple datasets were constructed utilising the Multiple Imputation (MI) technique for patients admitted with cardiac problems, and statistical analysis was carried out. (*Peter, Ian, Douglas et al., 2021*).

In relation to politics, James et al. discuss the multi-imputation (MI) methodologies; the paper refers to three shifts. Building MI that considers time trends and correlates across geography and time is the focus of the first shift. Building a novel algorithm that offers superior multiple imputations than the traditional MI approaches is the focus of the remaining two shifts (*Honaker and King, 2010*). For predicting the missing data, *Jangho et al.* applied deep learning, namely MLP (Multi-Layered Perceptron). If the experiment's variable was discovered to be non-linear, the outcomes were better (*Jangho, Juliane, Bhavna., et al, 2022*). *James, Gary,* and others have experimented with the Amelia II software package in R, which uses Bootstrapping to accomplish expectation minimization (*James, Gary, and Matthew, 2011*). The gene expression micro array experiment described by Olga and Michael et al. generated datasets with many missing values. To fill in these datasets, they used machine learning imputation techniques like KNNimpute, SVDimpute (Singular Value Decomposition), and average value imputation by varying the missing value percentage between 1 and 20% under various circumstances (*Olga, Michael, Gavin et al., 2001*).

The flip side imputing with regression techniques usually inflates the correlation amongst the variables (*Joseph, Schafera and Maren, 1998*). Any imputation is just an estimation to the

true values and can tend to deviate a lot from the expected value, any method that ignores uncertainty in the imputation method can lead to type – I error. The authors set the foundation for the concept of Multiple Imputation (MI) using EM algorithm and the concept of Maximum Likelihood in the estimation of missing values. *Tero et al*. conducted a study that is comparable to this one and is based on gene expression. They evaluated the literature on missing values in gene expression datasets and provided us with a comparison of all the MVI approaches applied in this application (*Tero, 2009*). In a totally different application, transfer learning in the LSTM (tLSTM) algorithm was utilised by *Jun, Jack,* and colleagues to impute successive values with extremely high missing rate. Compared to the currently used approaches, the proposed method was more accurate when imputed values were between 25 and 50 percent higher (*Jun, Jack, Yuexiong et at., 2020*). GRUs are employed in deep architecture to fill in the missing data, and it has been discovered that they perform more accurately and with higher imputation quality (*Shi et al., 2021*). In their discussion of missing data in relation to clinical datasets, Peter et al. point out that the absence of significant univariate associations does not prove that the data is MCAR or MAR. The authors have used various methods of multiple imputation (MI) on numerous datasets available for clinical studies. The use of linear mixed models with missing data in the result tends to estimate with less standard errors than the MI when dealing with multiple variate outcomes *(Peter, Ian and Buuren et al., 2021). Parikshit Bansal et al.* describe how missing values are common in analytical systems since the data is combined from many sources throughout a range of time intervals. They may also be vulnerable to missingness due to system failure at different points along the data analytics pipeline. Additionally, the authors' work on time series analysis on dimension n=1 involved the widespread usage of DeepMVI and DynaMMO for Uni variate analysis, with notable results *(Parikshit, Prathamesh and Sunita, 2020).* AICC (Adaptive Imputation Class Centre) is a technique developed by Kritbodin et al. that effectively provides imputations. It is based on determining the weighted distances between the centre and other observed data. They collected about 27 datasets from the UCI repository, and by simulating missingness rates ranging from 10% to 50%, they were able to achieve an average accuracy of 81.48%, which is significantly higher than that of other methods, which was only about 9–14% (*Kritbodin, Charnnarong, Carson et al., 2022*).

From the above it is observed that there is lack of research in multi-variate time series datasets, thereby this study would apply MVI techniques on series of datasets from varied applications, compare the techniques. Thereby conclude the best technique amongst them by

varying the missing rate for a dataset for a range between 15 – 30% over a step of 5%. The best MVI method such as Ensemble trees, Pattern based methods, Nearest Neighbour, MICE and Conventional Imputation **(Mean, bfill, ffill and most frequent**) is assessed for each missingness rate and evaluated with MAPE, MAE and MSE as a metrics.

3. Research Methodology

This section discusses about the data sources description about the data sources, then leads to conceptual map of the methods adopted, then a detailed mathematical description of the MVI methods adopted. Then paves way to exploratory and descriptive analysis carried out before the main analysis. The research methods to be deployed here is the **Quantitative Method**, since this study would encompass aggregating data for statistical analysis specific to the strategy rooted from the research questions framed. Quantitative research method is quite broad, the method can be further classified into Descriptive, Experimental, and relationship-based design approach (*Isadore and Carolyn*, *1998*).

The reason Quantitative research methods are chosen are the **data for this experiment was chosen from structured research instruments** such as data collected from global repository such as UC, Irvine data repository. The **research objectives and questions are clearly addressed** in prior sections even before data collection. The datasets are representative of most of the datasets that analysts would come across in their day-to-day application. Another reason the Quantitative method is chosen is due the cons such as ability to **replicate the study** on any set up due to standardised data collection stages and proper theories set up.

Amongst the Quantitative Design, this analysis would fall under the umbrella of **Experimental Research**, tries to establish the cause and effect of the phenomenon on the datasets and compare the different techniques. Moreover, a control is set on the all the dependent variables on the datasets such as controlling the missingness of the data on all the independent in the dataset. Accordance with this design the subsequent section in this chapter would talk about datasets, key concepts, and other integral concepts about the concepts.

3.1 About the Data

The datasets sourced from UCI repository, the datasets vary in categories right from Air quality datasets, which consists about the concentration of pollutants such as Nitrous oxide,

Carbon monoxide etcetera taken over a range over a year from March 2004 to April 2004 observed at a time interval of one hour. The second dataset is about the energy consumption observed at 10 mins for about 5months. A sensor network was used to keep an eye on the temperature and humidity levels within the home. Around 3.3 minutes later, each wireless node sent the temperature and humidity readings. The wireless data was then averaged across intervals of 10 minutes. Every 10 minutes, m-bus energy metres recorded the energy data. The 3rd datasets talk about the productivity of the labour in the garment sector, which is a very intensive sector, the dataset consists of variables such as idle time, incentives, work in progress, over time etcetera the dataset enlists productivity in the range of 0 to 1. The fourth dataset is pertaining to occupancy estimation of room, the experiment setup consisted numerous sensors and at an interval of 10mins the sensors transmitted data to the receivers. Like this the datasets were collected from myriad range of applications to capture the robustness of real-world data and come up with a diagnostic to mitigate the analysis in case of sensor failures due to unexpected circumstances. The data is publicly available and the link to datasets is presented in the Appendix – 1 section of this analysis.

3.2 Causes for Missing data

The causes for missingness are classified into **Intentional** and **Unintentional. Intentional missingness,** the practise of using various iterations of the same instrument for various subgroups, or matrix sampling, is another instance of purposefully leaving out data. Additionally, missing data that arise via questionnaire routing are on purpose, as are the data (like survival times) that are suppressed at a certain point since the event (like death) hasn't yet happened. Systematically lacking data is a phrase used in a multilevel setting that is similar. This phrase describes variables that are absent for every person in a cluster due to the variable not being measured in that cluster. **Unintentional missingness**; Unintended missing data are unintentional and out of the data collector's control. Examples include the following: the respondent skipped a question, a transmission fault resulted in missing data, certain objects dropped out of the study before it could be completed, resulting in incomplete data, and the respondent was sampled but refused to comply. Sporadically absent data is a related concept in a multilevel environment. This phrase is used when some members of a cluster of individuals have missing values for a variable.

3.3 About the mechanisms and Causes

Since the analysis is about the MVI technique, the missingness mechanism with which the missing data is associated must be investigated, By laying down better emphasis on missingness mechanism would give readers a better clarity.

3.3.1 Missing Completely at Random (MCAR)

This mechanism can be best explained by assuming 2 variables Y and X, the MCAR mechanism can be attributed if the probability of the variable Y cannot be linked to itself or the other variable X. But it doesn't eliminate the probability that the possibility that "missingness" on Y is related to the "missingness" on some other variable X (*Briggs et al., 2003*).

3.3.2 Missing At Random (MAR)

The probability of missing data on Y is unrelated to the value of Y after controlling for other variables in the analysis (say X)

$$P(Y \text{ missing}/Y, X) = P(Y \text{ missing}/X) \quad \textit{(Missing data, 2001)} \qquad (1)$$

This works on the conditional probability, that probability of Y missing on condition that data in Y and X is missing is same as that of Probability of data in Y missing on condition that data in X is missing.

3.3.3 Not Missing At Random (NMAR/MNAR)

MNAR denotes that the likelihood of going missing can change for unknown reasons. For instance, the weighing scale mechanism may eventually wear down, resulting in more missing data over time, but might go unnoticed this could lead to skewed distribution of the data if the heavier objects are measured later in time. According to MNAR, it is possible that the scale will produce more missing values for the heavier objects (as described above), which could be challenging to identify and manage. Similarly, if respondents with weaker opinions participate less frequently, this is an example of MNAR in public opinion research.

3.4 Missing Data Pattern

The missing data in a dataset is said to follow certain patterns theoretically and practically; They are classified into the following: Univariate, Multivariate, Monotone & Non monotone and connected – unconnected. Univariate and multivariate is called respectively if one variable in the dataset is missing is said to be **Univariate and Multivariate** if more than one variable in the dataset is missing. When missing data only affect one variable, this is known

as a univariate missing pattern. The multivariate missing pattern is an extension of the univariate instance and refers to missing data in a group of variables, either for the whole unit or for specific items in a questionnaire. If a variable is absent for a certain subject at every future time occasion as well as at a particular time point, a monotone missing pattern is described as the missing data pattern. The missing data pattern for this subject is known as the non-monotone missing data, in contrast, if a case is missing at one time point and then resurfaces at a later follow-up study. The non-monotone missing pattern merits particular attention since it might lead to greater issues in longitudinal data analysis than the monotone pattern. A **file-matching pattern** is used to describe missing data when variables aren't always observed simultaneously. There are further patterns associated with missing data, including latent-factor patterns that contain variables that are never observed. There are related statistical methods to tackle the effects of each missing data pattern on the quality of longitudinal analysis.

3.5 Assumptions

This section lists out the set of assumptions assimilated in the experimentation, like most of the statistical methods Multi Variate Imputation also works on certain pre assumed notions:

The data missing pattern to be simulated in our analysis is strictly pertaining to **Multi variate** missing data.

Since it is more practical to simulate **Missing Completely At Random** (MCAR) mechanism on the datasets, The observed values are indicative of the complete sample in the absence of missing values if missing data are unrelated to both the missing responses and the set of observed responses then it corresponds to MCAR mechanism.

Imputations methods must be compatible with the further analysis to be performed using imputed datasets, the associations and weights the Imputation models compute must be preserved (*Schafer,1997*). Since the study only focuses on the MVI techniques and evaluates each of the MVI technique. It is assumed that MVI and future analysis are compatible.

Prior Distributions, since the missingness mechanism work on the conditional probability which is the precursor of the Bayes Theorem. Most of the class of MVI follows the Bayes paradigm, according to the paradigm prior distribution quantifies knowledge of the MVI model before any data is seen by the model but this assumes **non-informative prior**

distribution using suitable software packages. Non-informative models **ignore the MVI model parameters**.

3.6 Working of Multiple Imputation

Since, the assumption is Multi Variate data missing, it would be essential to explain the working of Multiple Imputation (MI).

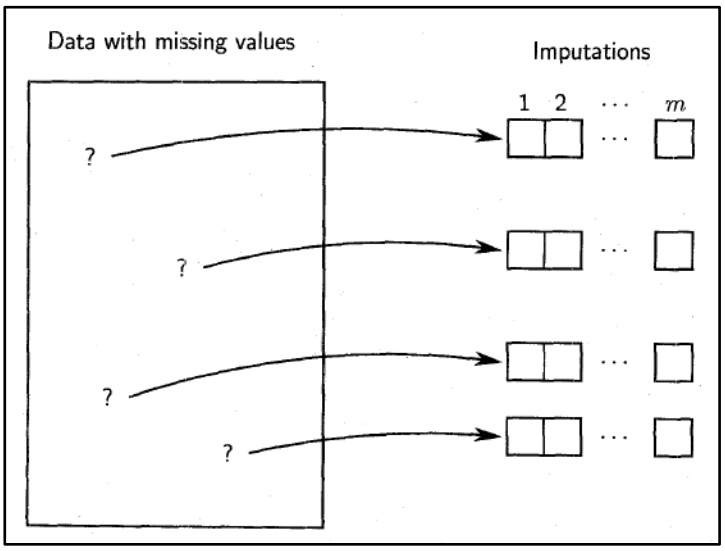

Figure 1 Representation of Multiple Imputation

In the above Figure-1, a dataset with missing values is replaced by set of values m > 1 computed from their predictive distribution. The variance amongst these imputations represents the Uncertainty with which the missing values can be predicted from the observed data points. The efficiency of the Multiple Imputation depends on the number of times the imputation is tried upon as shown in the equation 2.

$$\eta = (1+\gamma/m)^{-1} \qquad (2)$$

where, 'γ' - proportion of the data missing from the dataset and 'm' is the imputation performed.

The relationship could be seen that, the efficiency gets better as the imputations performed increases and has an inverse relationship with missing data proportion. *Schafer et al.*, have done extensive analysis on the efficiency of the MI (Multiple Imputation) across a range of missing proportion and MIs. It was concluded that imputations in the range of 3 -5 ideally proved to be better with an efficiency of 94% *(Schafer and Olsen, 1998*). **With this accordance the MI for each dataset in this study is set to 4.**

With regards to Single Imputation, Multiple Imputation tends to handle uncertainty brought on by the estimated distribution of the variables with missing values well, this should be considered to produce accurate estimations of the standard errors and P-values. This can be accomplished by producing multiple or multiple imputed data sets, each of which contains various imputations based on a random selection from a variety of estimated underlying distributions. These multiple imputed data sets can be created using a variety of techniques. However, since this is a primer, we just cite a few easily accessible publications. Again, each set of imputed data can be examined using conventional analytical methods. Each analysis will result in a multiple regression coefficient (or odds ratio) and associated standard error. The estimates can simply be averaged to provide a pooled estimate of the connection because each estimated association is impartial (presuming that the data are MCAR). The variance of the combined estimate will typically decrease because of the averaging (*Rogier, Geert, Theo, and Karel, 2006)*

3.7 Conceptual Map of Experiment

In the previous section, the assumption undertaken for the experiment is laid out. The conceptual map would give an overview of the things running inside the code and makes it easier to understand how the series of experiments have been done.

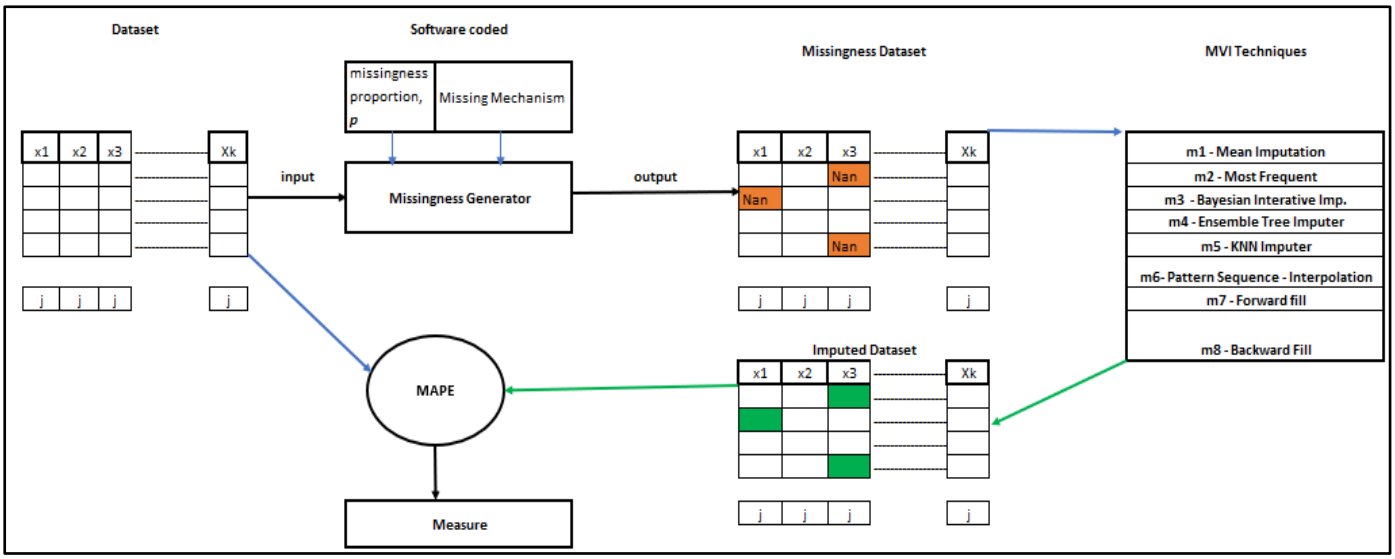

Figure 2 - Process flow of the experiment

The Figure -2 clearly explains about the process flow undertaken in this series of experimentation. The dataset is fed into the missingness generator code along with the parameters for the generator such as Proportion of missingness, ***p*** and mechanism of missingness, '***mecha***'. Once these parameters are fed in the missingness generator, a new

dataset is generated with missing values to the proportion required. Then the new dataset with a certain missing proportion is passed through series of MVI techniques and after passing through MVI techniques at each technique a new dataset created with the missing data imputed by the technique. For each dataset passing through the process, there would be subsequent **4 missing dataset variants** for each missingness rate prefixed in the study (**15%, 20%, 25% and 30%**) and each of these datasets get imputed. Around **4 variants (15%, 20%, 25% and 30%)** are generated. Such that each dataset from UCI would lead to **24 sub datasets**. Then all these 24 datasets are imputed and evaluated with the original dataset using a metrics namely Mean Absolute Percentage Error (MAPE), MAE and MSE undertaken in this study and then these steps are looped through for all the datasets taken up in the study. Then the effects of the different MVI techniques are compared and summarised.

3.8 MVI techniques

This section details about the MVI techniques used in this experiment along with its background and mathematical detailing. Also, the reason why the MVI techniques were chosen amongst the myriad of algorithms that are available in the academia.

3.8.1 Mean Imputation

The first MVI method is the Single Imputation with Mean, Mean Imputation handles with all types of numerical data such as Quantitative data such as scale data which are continuous. The mean imputation takes the mean for each variable in the dataset.

$$X_{i,\ Imputed} = \Sigma X_i / n \qquad \text{- (3)}$$

X_i, the i^{th} variable in the dataset, $X_{i,\ Imputed}$ is the missing value in the i^{th} variable and n is the number of rows in the dataset. Since the mechanism in this study in MCAR the estimate of mean remains unbiased. The main problem associated with Mean Imputation is that imputed data would have very low standard error, i.e., the mean for that variable in the dataset would be same as that of the original dataset thereby imputations themselves are estimates, so standard errors are less. As the standard errors are low, in terms of statistical significance the p-values get lower and more the possibility of occurrence of Type I error.

3.8.2 Mode Imputation/Most Frequent

In this MVI, which is more suitable for categorical data scales, works on the most frequently occurring values in that variable in the dataset and replaces/imputes these values in the missing grids. Just like Mean, even Mode is a measure of central tendency and there by mode

values of the variables in the original dataset is like Imputed dataset which would reduce the standard error which would impact the p – values and paving way to Type I error.

3.8.3 Iterative Imputer

Iterative Imputer is a class multi-Variate imputation technique, also known as MICE (Multi Variate Imputation by Chained Equations). Multi Variate Chained Equations create multiple Imputation there by cover the statistical uncertainty in the imputations. This mechanism is very flexible and can handle different data types. The usage of multiple imputation techniques, especially MICE, is quite versatile and applicable in a variety of contexts. The analyses of multiply imputed data consider the uncertainty in the imputations and produce precise standard errors since multiple imputation entails making numerous predictions for each missing value. Using a sequence of regression models, the MICE approach models each variable with missing data as a function of the other variables in the data. This means that each variable can be modelled according to how it is distributed, with logistic regression being used to represent binary variables and linear regression being used to model continuous variables (*Melissa, Elizabeth, Constantin et al., 2011*).

3.8.3.2 Advantages and Disadvantages of MICE

Keeps the relative distribution similar before and after imputation and it's good to use for Ordinal – Categorical data. In case of nominal categorical data will require conversion to dummy variable, the categorical variable has encoded accordingly. For use of Ordinal categorical data will require. round **()** method, as the outcome will be a floating point.

3.8.4 k Nearest Neighbour – Imputer

k Nearest Neighbour, is a non-parametric imputer works very much like algorithm which are used for classification, which uses proximity to make classifications or predictions about the grouping of an individual data point. The decision of how many neighbours (n-neighbours) to use will involve balancing computing cost, generalizability, and noise. Small K Means more noise/faster, large k = robustness of our results under noise/complexity of calculation. It is often advised to choose an ODD value of K to serve as a tiebreaker in the case of a binary [0,1] imputation.

3.8.4.1 Computing distancing

The distance between the query point and the other data points must be determined to determine which data points are closest to a specific query point. These distance measurements aid in the creation of decision borders, which divide query points into several zones. Decision boundaries are frequently represented using Voronoi diagrams. There are numerous distance measurements available but Euclidean distance is used more often and with accordance to *Surya, Haneen, Ahmad and Omar et al.,* Euclidean distance performs better with existence and without existence of noise in the dataset *(Abu Alfeilat et al., 2019*). The Euclidean distance is calculated based on the Equation – 4, where x and y are the cartesian points in a 2 dimensional where points are in the order of **1 to 'n'**. The distance between the 2 points is represented ***d.***

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \quad (4)$$

3.8.4.2 Selecting k for k-NN

The k value in the k-NN algorithm specifies how many neighbours will be examined to determine a particular query point's classification. The instance will be placed in the same class as its lone nearest neighbour, for instance, if k=1. To avoid either overfitting or underfitting, several values of k must be considered when defining it. Larger values of k may result in strong bias and low variance, while smaller values of k may have high variance but low bias. The selection of k will be heavily influenced by the input data, as data with more outliers or noise will probably perform better with higher values of k. In general, it is advised to use an odd integer for k to prevent classification ties (*Gongde, Hui and David et al., 2004*). With regards to previous statement all odds values for K were run in loops and then it was decided that the analysis run with 'k = 5'.

3.8.5 Interpolation Techniques

Numerical analysis relies heavily on the concept of interpolation. Very frequently, just the values of a function at a collection of points known as nodes, tabular points, or pivotal points are available directly. Interpolation is the process of determining the function's value at any

non-tabular position. If the curve is represented by the g(x), at (N+1) points in a cartesian co-ordinate in a points X_o, X_1----------X_N, spread out in the interval [a, b]. Finding the value of the function new points say for instance X_s, is called Interpolation. Interpolation is categorized into **Polynomial** and **Linear** Interpolation. The function g(x) produces Nth degree polynomial passing through the points it is a called Polynomial Interpolation. This analysis would strictly pertain to Linear Interpolation.

3.8.5.1 Benefits and Limitation – Interpolation

Interpolation can calculate drastic changes in the plots, such as cliffs and fault lines. Well-interpolated dense dots with uniform spacing (flat areas with cliffs). Can change the number of sample points to affect the values of the cells. Estimates cannot be made above or below extreme levels. Not particularly effective in crests.

3.8.6 Ensemble Methods – Base estimator - ExtraTree

Extra Trees also known as Extremely Randomized Tree i.e., providing more variation in building the trees are like RandomForest algorithm the major difference being RandomForest uses **bootstrapping**, that is input data is subsampled with replacement whereas ExtraTree uses the data without any subsampling techniques and uses data as a whole. ExtraTree splits the nodes based on optimization and randomization. The major boost these ExtraTree algorithm is that they take care of Bias – Variance trade off in a better manner by taking the original data instead of the Bootstrap replica and random splitting of the nodes reduces the variance. The randomly split nodes also enable the algorithm to work faster. ExtraTrees algorithm works on the following steps: The entire sample split of data in the training set is used to build each tree in the set of trees, The node split is defined by doing a search in a subset of randomly chosen sqrt-sized characteristics (number of features). Each feature's split is determined at random, and the maximum depth of each tree is set to 1 (*Pierre, Damien, and Louis, 2006*)

Not all ensemble methods use weak learners as their basis estimator. For instance, this analysis uses Scikit-learn to create an ensemble containing the top two classifiers for a certain job. The ensemble's weights can be optimised, and predictions can be made using them. The Voting Classifier would have an advantage over any of them, assuming they have carefully adjusted the classifiers and they all perform similarly.

3.8.6.2 Advantages of ExtraTrees

ExtraTrees has certain edge over the other tree-based algorithm, namely the ExtraTrees are computationally much easy on the system requirement as it is quicker. The node split criterion is easy to interpret as it depends directly on the square root of number of 'n' – number of features in the dataset. The algorithm tides over Bias – Variance trade off using its sampling technique and randomization.

3.8 Metrics – Evaluating the MVI techniques

3.8.1 Mean Absolute Percentage Error (MAPE)

As mentioned in the previous section, there must be a metrics to evaluate each MVI technique on Multiple Imputations (MI) on each dataset in the analysis. MAPE is one among the metrics chosen the reason being MAPE provides the actual percentage between actual ($Xi_{,actual}$) and the Imputations performed by the MVI technique ($Xi_{,Imputed}$) in terms of absolute value in percentage. Since the data in each variable would in different scale – comparing the error would be best suited using MAPE. Another advantage MAPE has over other metrics is its interpretability, analysts and even non-technicians can easily to understand.

$$\text{MAPE} = \frac{1}{N}\Sigma\left|\frac{(Xi,actual - Xi,Impued)}{Xi,actual}\right| \qquad (5)$$

One of the major disadvantages, MAPE yields extremely large percentage errors (outliers), while zero actual values result in infinite MAPEs (*Sungil and Heeyoung, 2016*).

3.8.2 Mean Absolute Error (MAE)

Another metrics chosen for the analysis is MAE, which is average computed over error i.e., difference between Observed and Predicted over an absolute function Because the error value units match the anticipated target value units, mean absolute error (MAE) is a well-liked statistic. For more information, read the next subsection. Unlike RMSE, MAE changes are logical and linear. Due to MSE and RMSE, which penalise greater errors more harshly, the mean error value is inflated or raised because of the square of the error value. In MAE, the scores increase linearly as the number of errors increases rather than being given a different weight for each type of error. The MAE score is determined by averaging the absolute error statistics. A negative integer becomes positive through a mathematical process known as the Absolute. As a result, there may be a positive or negative difference between an expected

value and a forecast value. *(Anomaly Detection and Complex Event Processing over IoT Data Streams, 2022)*. Equation – 6 is the representation of the MAE.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |Xobserved - Xpredicted| \quad (6)$$

3.8.3 Mean Squared Error (MSE)

The last metrics for evaluating the imputations is the MSE, the degree of inaccuracy in statistical models is gauged by the mean squared error, or MSE. Between the observed and projected values, it evaluates the average squared difference. The MSE is equal to 0 when a model is error-free. Its value increases when model error does as well. The mean squared deviation is another name for the mean squared error (MSD). The interpretation is less logical since squared units are used instead of the native data units. There are various benefits to squaring the discrepancies. By squaring the differences, negative differences values are eliminated, and the mean squared error is always greater than or equal to one. It nearly always has a positive value. An MSE of 0 is only produced by a flawless model that is error-free. And that doesn't happen. The equation for MSE is shown in the Equation – 7.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (Xobserved - Xpredicted)^2 \quad (7)$$

4. Findings and Analysis

This section outlines the key analysis performed along with an apt figures and graphs to support the same. Then key findings which would support the research aims and research questions. This section outlines key aspects of the analysis and to be considered as the soul of the analysis.

4.1. Simulating – Multi Imputation

As mentioned earlier multiple Imputation simulated datasets provide better results and in depth understanding about the Imputations performed using plethora of MVI techniques.

```
def simulate_NA(X, prop_miss, mechanism="MCAR", opt=None, p_obs=None, q=None):
    to_torch = torch.is_tensor(X)
    if not to_torch:
        X = X.astype(np.float32)
        X = torch.from_numpy(X)

    else:
        mask = (torch.rand(X.shape) < p_miss).double()

    X_nas = X.clone()
    X_nas[mask.bool()] = np.nan

    return {'X_init': X.double(), 'X_incomp': X_nas.double(), 'mask': mask}
```

Figure 3 - Code Snippet NaN generator

Since the most prevalent missing mechanism mentioned earlier sections is MCAR (Missing Completely At Random). The Figure-3 is the code used in simulating the missingness proportion for each data using the mechanism considered. The user defined function '***simulate_NA***' has the following argument '**X**' – The dataset which must be converted to an array, '**prop_miss' –** The proportion of missingness that is needed for the experiment, '**mechanism**' – missingness mechanism. Returns the dataset in a **NumPy** array and needs to be read into Pandas **DataFrame** after the processing. **Torch** is used to convert the array into a tensor – an algebraic object that is used to map between vectors, scalers etcetera.

The codes used for the analysis are presented in Appendix – 2 of this analysis.

4.2. Exploratory Data Analysis – All Datasets

The datasets are collected from range of application ranging from Air quality dataset to occupancy dataset to garment productivity dataset to Air pollution datasets in Chinese cities and weather stations in the likes of Aotizhongxina and Tetuan city power consumption.

Datasets	No. of variables	No. of variables used	No. of categorical dtyes	No. of continous dtyes	No. of missing data before simulating	No. of Rows
Dataset-1	15	13	0	13	0	9357
Dataset-2	29	28	0	28	0	19735
Dataset-3	15	11	0	0	506	1197
Dataset-4	19	17	0	17	0	10129
Dataset-5	18	10	8(string)	10	10826	35064
Dataset-6	9	8	0	8	0	52416

Table – 1 – Summary of the datasets

Out of the 6 datasets, 2 datasets have in built null values. In dataset-3 there are around 506 values missing in a single variable since in a single variable accounts 50% of the whole dataset. The variable is treated by **dropping the variable**. Whereas in Dataset – 5 around 12 variables account for it, thereby **listwise deletion** is done on these variables since the proportion of missing is less than 3% for each variable. This is shown in the Figure 4 and Figure 5. On the data types grounds as mentioned in Table-1, variables in continuous datatypes fall under 2 categories namely integer and float. Most of the variables in the datasets used in this analysis had float has the datatypes and certain variables fell under the category of **integer datatypes** and **while imputing using MVI techniques care to be adhered** to round these values to the **closest integer** or **typecast** the values to integer. Certain variables in the analysis were to be dropped the reason being these variables in the datasets constituted the timestamps which were of DateTime datatypes. **Special case was observed in Dataset -5** as mentioned in Table -1, **2 of the 8 dropped variables were Timestamps** and the **rest were 'string' datatypes** containing information about the station name from where the atmosphere pollutants concentration was recorded and the zone under which the station falls under. The relationships amongst the independent were tried to establish even though not much of relevance to the analysis but would churn the interest in readers in understanding the relationship amongst these independent variables. For instance, in the Dataset -6, which tries to capture the power consumption in Tentuan city, the **Correlation study** gave certain insights about the known factors in power consumption in the 3 different zones in the dataset, Temperature and power consumption in the zones gave way to increase in power consumption in all zones thereby establishing a Positive relation. Whereas the Relative Humidity (RH) had negative relation to the power consumed in all the zones, even though not a very strong relationship but it could be supported by the fact that on rainy days consumers tend to use less power refer Figure -6. On the similar lines, in the Dataset -1, which captures the factor relating to Air quality over a particular region it can be observed that concentration of certain pollutants can be in positive strong relationship with certain pollutants even though it would be very premature to ascertain the effects there is common saying that causation isn't correlation and further subject matter expertise might be needed to give a proper closure. But from the outset by looking at the correlation matrix as shown in Figure -7, the ambient temperature and concentration of Benzene is found to be related in a positive trend.

date	0
quarter	0
department	0
day	0
team	0
targeted_productivity	0
smv	0
wip	506
over_time	0
incentive	0
idle_time	0
idle_men	0
no_of_style_change	0
no_of_workers	0
actual_productivity	0

No	0
year	0
month	0
day	0
hour	0
PM2.5	925
PM10	718
SO2	935
NO2	1023
CO	1776
O3	1719
TEMP	20
PRES	20
DEWP	20
RAIN	20
wd	81
WSPM	14
station	0

Figure-4 – NaN values in Dataset3 and Dataset5

#	Column	Non-Null Count	Dtype
0	CO(GT)	9357 non-null	float64
1	PT08.S1(CO)	9357 non-null	float64
2	NMHC(GT)	9357 non-null	int64
3	C6H6(GT)	9357 non-null	float64
4	PT08.S2(NMHC)	9357 non-null	float64
5	NOx(GT)	9357 non-null	float64
6	PT08.S3(NOx)	9357 non-null	float64
7	NO2(GT)	9357 non-null	float64
8	PT08.S4(NO2)	9357 non-null	float64
9	PT08.S5(O3)	9357 non-null	float64
10	T	9357 non-null	float64
11	RH	9357 non-null	float64
12	AH	9357 non-null	float64

Figure-5 – DType encountered

	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
Temperature	1.000000	-0.460243	0.477109	0.460294	0.196522	0.440221	0.382428	0.489527
Humidity	-0.460243	1.000000	-0.135853	-0.468138	-0.256886	-0.287421	-0.294961	-0.233022
Wind Speed	0.477109	-0.135853	1.000000	0.133733	-0.000972	0.167444	0.146413	0.278641
general diffuse flows	0.460294	-0.468138	0.133733	1.000000	0.564718	0.187965	0.157223	0.063376
diffuse flows	0.196522	-0.256886	-0.000972	0.564718	1.000000	0.080274	0.044667	-0.038506
Zone 1 Power Consumption	0.440221	-0.287421	0.167444	0.187965	0.080274	1.000000	0.834519	0.750733
Zone 2 Power Consumption	0.382428	-0.294961	0.146413	0.157223	0.044667	0.834519	1.000000	0.570932
Zone 3 Power Consumption	0.489527	-0.233022	0.278641	0.063376	-0.038506	0.750733	0.570932	1.000000

Figure 6 - Correlation matrix - dataset-6

	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)
CO(GT)	1.000000	0.041415	0.128351	-0.031377	0.029939	0.526450	-0.089981
PT08.S1(CO)	0.041415	1.000000	0.170009	0.852659	0.933101	0.278029	0.086931
NMHC(GT)	0.128351	0.170009	1.000000	0.037329	0.110097	-0.004413	0.048832
C6H6(GT)	-0.031377	0.852659	0.037329	1.000000	0.767401	-0.001163	0.512154
PT08.S2(NMHC)	0.029939	0.933101	0.110097	0.767401	1.000000	0.331331	-0.073748
NOx(GT)	0.526450	0.278029	-0.004413	-0.001163	0.331331	1.000000	-0.436083
PT08.S3(NOx)	-0.089981	0.086931	0.048832	0.512154	-0.073748	-0.436083	1.000000
NO2(GT)	0.671140	0.154058	0.103345	-0.010971	0.176569	0.817138	-0.256217
PT08.S4(NO2)	-0.073721	0.845133	0.162689	0.774649	0.874761	0.035580	0.122672
PT08.S5(O3)	0.080316	0.892436	0.101189	0.641306	0.909909	0.461916	-0.208935
T	-0.068952	0.754806	-0.000008	0.971370	0.668984	-0.138457	0.588061
RH	-0.048231	0.745344	0.008288	0.925068	0.585775	-0.053008	0.573513

Figure 7 - Correlation matrix - Dataset-1

4.2 Pre-Processing

Since this analysis talks about the pre-treatment of data often faced by analysts there by not much of emphasis would be given to the pre-treatment the primary reason being treating missing data is by itself an important pre-treatment step. Minimal steps have been taken to handle the same. **To reduce the bias in the estimates**, the **pre-existing null values (NaNs)** in 2 of the datasets have been taken care by **listwise elimination** and **dropping the variables** based on the observed pre-existing missingness percentage in the original datasets.

Another such pre-processing step was required while performing MI using K-nearest neighbour imputer, since this MVI technique works on by **capturing the proximity of data points based on the distances** thereby by it is pertinent that the data be transformed to a common scale otherwise would lead the variable in higher scale being dominant (*Md, Parvez, Pritom et al., 2021*). In this analysis **MinMaxScaler** was put into the task to scale the data. Entire of set of data are transformed to a scale of values ranging 0 to 1 as shown in Figure -8, after which the K-NN imputer is applied upon and then the datasets are **rescaled** to their original scales.

```
scaler = MinMaxScaler(feature_range=(0, 1))
df_knn_15_2 = pd.DataFrame(scaler.fit_transform(df_knn_15_2))
df_knn_20_2 = pd.DataFrame(scaler.fit_transform(df_knn_20_2))
df_knn_25_2 = pd.DataFrame(scaler.fit_transform(df_knn_25_2))
df_knn_30_2 = pd.DataFrame(scaler.fit_transform(df_knn_30_2))
```

Figure 8 - Code snippet - Transforming data.

4.3 Observations from Multiple Imputations

As seen in the section – 4.1 the code snippet for simulating Multiple Imputations of the dataset is shown and explained. This section covers the observations seen during the process and presented in this section.

In the Figure – 9 the original dataset is shown and, in the Figure, – 10 the dataset Imputed using the 15% missing rate using the NaN as shown in the section-4.1. The orange-coloured highlighted grids tell us about the presence of NaN values in the new dataset. The Figure – 11, shows the number values in each removed in the 15% missingness category and confirms the missingness percentage generated by the user defined function which approximates to 14.96% percentage.

	DateTime	Temperature	Humidity	Wind Speed	general diffuse flows	diffuse flows	Zone 1 Power Consumption	Zone 2 Power Consumption	Zone 3 Power Consumption
0	1/1/2017 0:00	6.559	73.8	0.083	0.051	0.119	34055.69620	16128.87538	20240.96386
1	1/1/2017 0:10	6.414	74.5	0.083	0.070	0.085	29814.68354	19375.07599	20131.08434
2	1/1/2017 0:20	6.313	74.5	0.080	0.062	0.100	29128.10127	19006.68693	19668.43373
3	1/1/2017 0:30	6.121	75.0	0.083	0.091	0.096	28228.86076	18361.09422	18899.27711
4	1/1/2017 0:40	5.921	75.7	0.081	0.048	0.085	27335.69620	17872.34043	18442.40964

Figure 9 - First 5 rows of dataset -6 before Imputing

	0	1	2	3	4	5	6	7
0	6.559000	73.800003	0.083000	0.051000	0.119000	34055.695312	16128.875000	20240.962891
1	6.414000	74.500000	0.083000	0.070000	nan	nan	19375.076172	20131.083984
2	6.313000	74.500000	nan	0.062000	0.100000	29128.101562	19006.687500	19668.433594
3	6.121000	75.000000	0.083000	0.091000	0.096000	28228.861328	18361.093750	nan
4	5.921000	75.699997	0.081000	0.048000	0.085000	27335.695312	17872.339844	18442.410156

Figure 10 - 15% missing induced to dataset-6 and generated NaN values highlighted in orange

```
dataset6_mcar_15.isnull().sum()
```

```
0    7780
1    7737
2    7912
3    7966
4    7950
5    7872
6    7973
7    7906
Percentage of generated missing values:  14.96895032051282  %
```

Figure 11 - No. of NaN entries generated for each variable in dataset-6

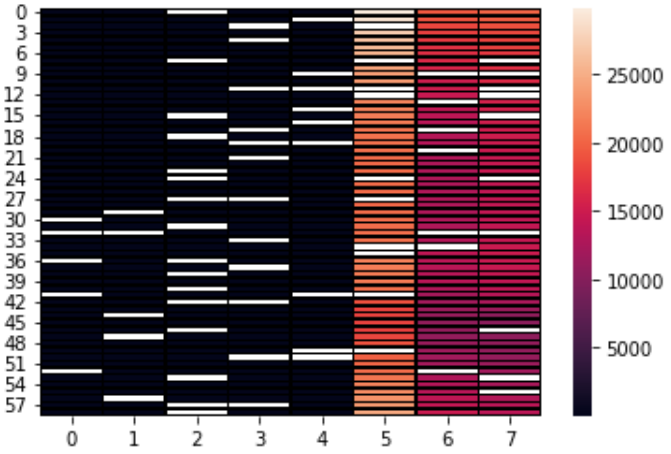

Figure 12 - HeatMap of the NaN vales distribution in Dataset -6-15%missingness

Figure – 12, the random distribution of the NaN values in the 15% missingness in the Dataset-6, the white spaces clearly show the NaN and gradient colour are provided to show the scale of the values in each variable. The variables are represented in the X-axis ranging from 0 -7 i.e., 8 chosen variables and the Y-axis represents the first 59rows taken to plot the heatmap.

4.4 Observations – MVI techniques

4.4.1 Analysing MVI techniques across one Imputation of a Dataset

This section would talk about the MVI techniques across the datasets and try to visualize the imputations performed in the best way possible and try to understand these imputations across datasets and across Multiple Imputations of the datasets. For this instance, **Dataset-1** is chosen at random and the ***variable 'PT08.S1'*** measure of Carbon Monoxide concentration in the air is taken for a series for 100 continuous days. The MI chosen is the **maximum missingness which is 30% variant** of the dataset 1.

The **Figures 13 to 20**, illustrates the Imputations performed by each MVI techniques on a variable taken under study as mentioned above from one variant of the Multiple Imputations performed. In the **below series of Figures**, the **blue line** represents the data points across the original dataset and the **red dotted lines** represent the Imputed values courtesy of MVI techniques considered. From the below figures and Corroborated by the Table-2, **it is seen**

that the least desirable method would be Mode also known as Most Frequent Imputation the reason can be supported by the Figure -14 as well the deviation is way to pronounced and supported by the Table – 2 where it can be clearly seen that all 3 metrics taken into this study have overshot. Whereas **on the flipside Pattern Based Interpolation**, seem to Impute better and corroborated by the Figure – 17 and Table -2.

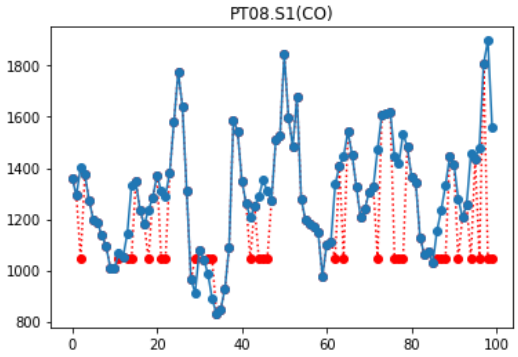

Figure-13 – Mean Imputation

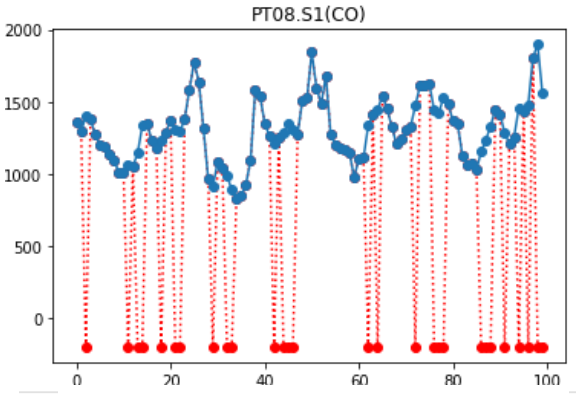

Figure – 14 – Mode Imputation

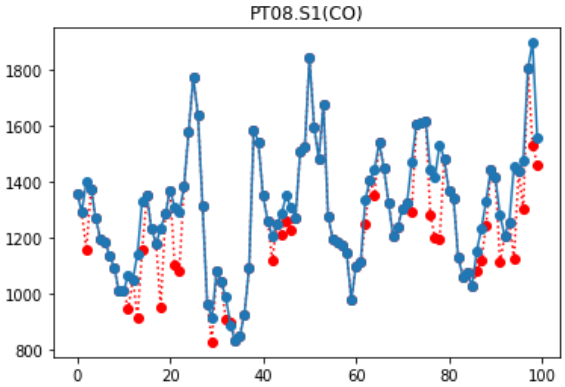

Figure-15 – MICE Imputation

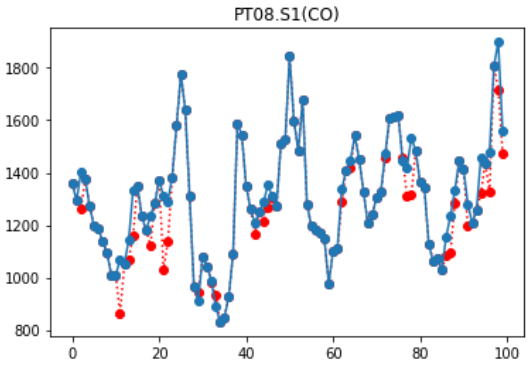

Figure-16 – ExtraTree Imputation

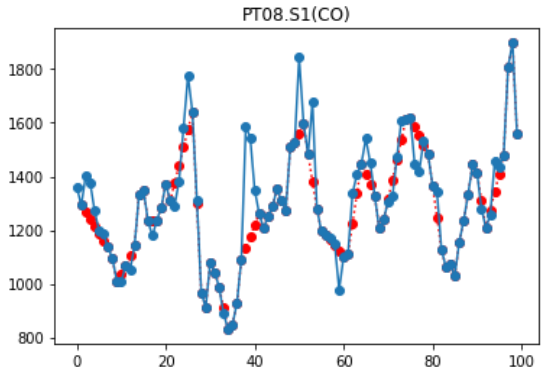

Figure-17 – Pattern Based Interpolation

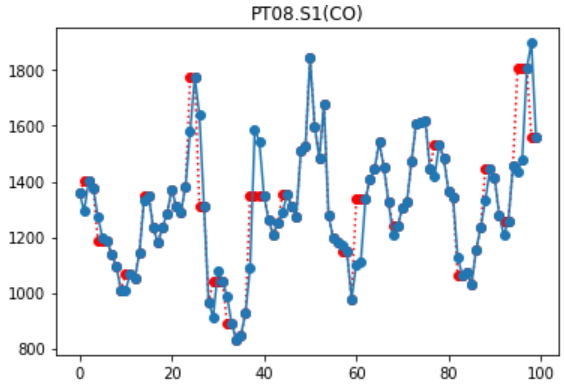

Figure-18 – ffill imputation

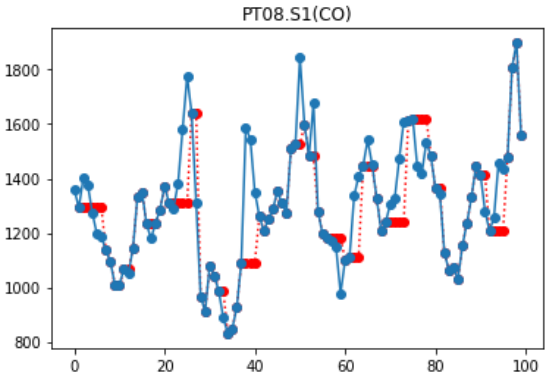

Figure-19 – bfill Imputation

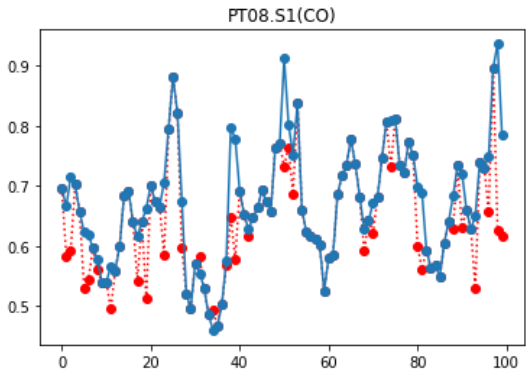

Figure-20 – KNN Imputation

	Dataset -1 - 30%missingness							
	Mean	Most Frequent	MICE	ExtraTree	KNN	Pattern Based Interpolation	Forward Fill	Backward Fill
MAPE (%)	85.6	100	63.7	36.8	63.6	22.3	26.3	21.6
MSE	20733.4	207212.1	3478.3	1480.1	7769.8	2406.9	5644.7	5307.5
MAE	43.1	180.3	18.2	9.2	28.4	11.6	17.8	17.4

Table – 2 – Performance testing using different metrics

4.5 Addressing Research Questions

This section focuses on addressing the Research questions set in the previous with proper tabulations and insights from analysis. This chapter is further divided into 4 sub sections each allocated to answer the readers with questions that had set tone for this analysis.

4.5.1. Best MVI technique

The *Table – 3* illustrates the MVI technique across each Multiple Imputations performed on dataset – 1. For the sake of illustration, MAPE is displayed in these series of tables as MAPE turns up the Imputation error in terms of percentage. The better amongst the series of MVI techniques used on dataset – 1, **Pattern Based Interpolation, Backward fill and Forward fill** techniques seem to work just fine. Lower the error better the quality of the Imputations. Techniques such as KNN, MICE which seemed promising couldn't quite stand up to the expectation set in dataset – 1. Most Frequent value imputation seemed futile in this dataset due to dominance of continuous features. By looking at **Variability of the MVI technique** across multiple Imputations – Pattern Based Interpolation seemed to have lesser range - a measure of variability standing at 12%. Another observation worth mentioning is that in

Backward fill MVI technique at the missingness around 30% the MAPE saw a decline which was indeed an anomaly in the trend.

	Dataset -1 - MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	13%	20%	13%	12%	27%	26%	45%	512%
20% MISSING	17%	28%	22%	18%	41%	52%	76%	701%
25% MISSING	30%	37%	20%	24%	58%	50%	76%	850%
30% MISSING	22%	41%	24%	20%	69%	61%	89%	1001%

Table – 3 – Evaluating Metrics – MAPE – Dataset – 1

	Dataset -2 - MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	2%	2%	3%	2%	4%	3%	8%	4%
20% MISSING	4%	4%	4%	4%	6%	4%	14%	8%
25% MISSING	5%	6%	6%	5%	9%	6%	14%	9%
30% MISSING	6%	6%	6%	6%	9%	7%	15%	9%

Table – 4 – Evaluating Metrics – MAPE – Dataset – 2

	Dataset -3- MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	12%	5%	13%	12%	6%	8%	13%	8%
20% MISSING	16%	8%	16%	15%	10%	11%	22%	9%
25% MISSING	22%	13%	22%	20%	14%	16%	22%	11%
30% MISSING	25%	12%	25%	23%	16%	18%	25%	13%

Table – 5 – Evaluating Metrics – MAPE – Dataset – 3

	Dataset -4- MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	2%	2%	2%	2%	4%	4%	8%	2%
20% MISSING	3%	3%	3%	3%	6%	4%	14%	2%
25% MISSING	4%	4%	4%	4%	8%	5%	14%	3%
30% MISSING	4%	5%	4%	4%	11%	5%	17%	3%

Table – 6 – Evaluating Metrics – MAPE – Dataset - 4

	Dataset -5- MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	4%	12%	4%	3%	17%	21%	35%	15%
20% MISSING	6%	18%	6%	4%	24%	32%	58%	21%
25% MISSING	8%	22%	8%	6%	31%	53%	58%	25%
30% MISSING	9%	30%	10%	7%	40%	55%	70%	31%

Table – 7 – Evaluating Metrics – MAPE – Dataset – 5

	Dataset -6- MAPE							
	Backward Fill	ExtraTree	Forward Fill	Pattern Based Interpolation	MICE	KNN	Mean	Most Frequent
15% MISSING	2%	360%	2%	1%	1609%	734%	3603%	6%
20% MISSING	2%	703%	2%	2%	2224%	1805%	5983%	8%
25% MISSING	3%	1061%	3%	2%	2943%	2180%	5983%	10%
30% MISSING	4%	1620%	5%	3%	4005%	3319%	7140%	12%

Table – 8 – Evaluating Metrics – MAPE – Dataset - 6

The MVI techniques performed in Dataset – 2, had an overall lower MAPE compared to Dataset-1. Even the **ad-hoc imputations methods such as Mode and Mean imputations techniques accounted for lower MAPE, this could be accounted for more integer – 64 data types in the variables and the value could have better frequency.** Another boost that could be observed due to this reason is that clustering and tree-based ensemble performing better. Even all the MVI techniques seems good for dataset-2, the one that seem to be reliable is - **Pattern Based Interpolation techniques**, which can be observed in table – 4. All the percentage were in single digit which is indeed a big boost.

By observing the proceedings in Dataset -3, MICE seemed to outperform all other models along with KNN worked out better tabulated in table – 5. Whereas in Dataset – 4 - Backward fill, Forward fill, Pattern Based – Interpolation and ExtraTree MVI techniques churned out similar MAPE estimates seen in Table – 6. In Dataset – 5 and Dataset – 6 again similarity observed in most of the datasets in this analysis – **Pattern Based Interpolation** had an upper hand over other MVI techniques. One notably observation with respect error percentage in Dataset – 6, ExtraTree, KNN and Mean MVI techniques provided error percentage in order of 1000's.

	Best MVI technique	Average MAPE (%)	Variation Observed over different missingness
Dataset-1	Pattern Based - Interpolation	19%	12%
Dataset-2	Pattern Based - Interpolation, Backward Fill	4%	4%
Dataset-3	ExtraTree	10%	8%
Dataset-4	Pattern Based - Interpolation, Backward Fill, Forward Fill	3%	2%
Dataset-5	Pattern Based - Interpolation	5%	4%
Dataset-6	Pattern Based - Interpolation	2%	2%

Table – 9 – Best models dataset wise w.r.t Average MAPE and Variation.

Table – 9 clearly shows that Pattern Based Interpolation being the best amongst the datasets. Out of the 6 datasets, Pattern Based Imputation method bested out in 5 datasets. This clearly shows that Pattern Based Interpolation could be possible avenue for Data Analysts working in Time Series analysis to explore to mitigate from Missing values. **Another observation which is worth mentioning** and had been addressed in the prior sections is that – In presence of **larger proportion Integer datatype** in the variables causes the **Pattern Based Interpolation to deviate bit more than the usual**. In presence **integer datatypes dominance in the dataset**, it would imperative or suggestive to **use Clustering or Ensemble Tree based algorithm using ExtraTree as the base estimator** supported by the performance of the same MVIs in dataset 3.

4.5.2 Metrics – Proposing new approach

As mentioned earlier this study has focused on 3 metrics – MAPE, MSE and MAE to evaluate Multiple Imputations. But all the metrics corroborated the same. The magnitude varied but all the 3 metrics conveyed the same. MAPE had the best in advantage as even a lay man with some basic math can understand the deviation but the ease in understanding does come with a catch. One major drawback observed during this analysis is that if the actual value or the observed values approach Zero the MAPE yields higher values or tends towards Infinity as shown in Figure 21. In the Figure 21, all the MVIs have overshot a lot and upon deep diving it is found that the problem is ascertained to the way by which MAPE is computed, in case there happens to be zeroes as entries in the original dataset, the true values take up the denominator in the MAPE calculation thereby leading to a proportional much larger MAPE. This drawback wasn't observed in MAE and MSE but there is a trade-off to be noticed in terms of Interpretability MAPE is much easier to interpret whereas MSE would be least desired for interpretability. In terms of treading through different of scales of data observed in the variable MAPE does an impeccable job. By looking at the shortcomings and advantages seen in both MAPE and MAE a new approach could be derived to alleviate these shortcomings.

There by scaling the data to a scale between 0 to 1 and transforming using an Exponential transformer as shown Figure – 22 and then calculating the error percentage can lead to a standardized error metric across different datasets. Computationally this could be demanding but they would provide with reliable error

metrics. As could be observed from the Figure – 22, these transformations provided nonzero estimates thereby can troubleshoot the issues faced in MAPE.

```
MAPE FOR DATASET2 30% MISSING - Mean 142059094253459.2
MAPE FOR DATASET2 30% MISSING - Mode 1301820521188.3828
MAPE FOR DATASET2 30% MISSING - Iterative Imp 160204971865570.84
MAPE FOR DATASET2 30% MISSING - Ensemble Trees 63411865049449.414
MAPE FOR DATASET2 30% MISSING - KNN 60844252005299.1
MAPE FOR DATASET2 30% MISSING - Interpolation 35502058229455.164
MAPE FOR DATASET2 30% MISSING - ForwardFill 37887108488498.51
MAPE FOR DATASET2 30% MISSING - BackwardFill 35448996670079.76
MAPE FOR DATASET2 20% MISSING - Mean 123015843131936.05
MAPE FOR DATASET2 20% MISSING - Mode 966035099921.4728
MAPE FOR DATASET2 20% MISSING - Iterative Imp 110470890217758.8
```

Figure 21 – Higher MAPE values due to zeroes observed in original data.

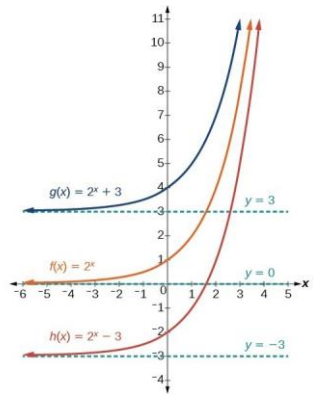

Figure 22 – Exponential Transforms

In Figure 22, the transformation functions g(x), f(x) and h(x) don't have a zero has a unit root thereby by setting the range between 0 to 1. These transformers churn out data void of zero.

5. Conclusion and Recommenda

This section summarizes the conclusi recommendations for future works and further enhance the contribution to the domain of Missing Value treatment.

5.1 Conclusion

In this analysis, an experimental framework for simulating Multiple missing rates in the range of 15% to 30% at 5% step up for each dataset taken into the analysis. A total of 6 datasets were taken up and missing rates were simulated. Taking into the missingness variants into account a total of 24 variants were put thorough MVI techniques to impute the missing data and then after these imputations the Imputations are compared to the original datasets. The

uncertainty or the errors of imputation were evaluated using 3 prominent metrics – MAPE, MAE and MSE. The missingness were simulated successfully within this stipulated range due to realistic conditions.

First, the datasets were successfully simulated and the expected missingness were achieved successfully, upon succeeding in this phase then the variants of the datasets are passed through MVI techniques.

Second, from the plethora of MVI techniques – Pattern Based Interpolation bested in 83% of the datasets with an average MAPE varying from 2% to 19%. Out of the 6 datasets Pattern Based Interpolation bested in 5. Similarly, Pattern Based Interpolation deviated from its performance in an Integer datatype dominant dataset.

Third, in an Integer datatype dominant dataset – Tree based, and Clustering based MVI techniques performed better besting out Pattern Based Interpolation. The MAPE achieved in this dataset was around 10%.

Fourth, upon performing Multiple Imputations – the imputations were evaluated using the 3 metrics and the advantages pertaining to the 3 metrics were discussed in detail. MAPE and MAE bested out due to its interpretability. **MAPE is better due to its independency to the scale of the variables**.

Fifth, an approach is proposed to evaluate the Imputations – such as Scaling the data in the range 0 to 1 and then transforming data using Exponential Transformers as discussed in section – 4.5.2.

5.2 Understanding limitations of MVI

As mentioned earlier, Imputations are nothing but filling in the missing values with best possible values thereby all the filling values come with some uncertainty and bias. Multiple models were tried and best amongst from pool of MVI techniques is chosen based on the metrics set up in the initial phase of the study. This section lists out the limitations faced while performing imputations.

Inability to generalise - the performance of MVIs varied based on datasets even though the best MVIs is judged by the one performing the best in most of the datasets chosen for the study but still there isn't a clear-cut winner. Pattern Based Interpolation failed to best in datasets with strong presence of Integer datatype.

Another major drawback that high end Imputations methods have is that they consume a lot of time and require high end computation prowess, this was observed even during this analysis even with datasets of 10K entries the MVI methods such as kNN and Ensemble Tree consumed much of RAM requirements and caused cells to crash.

The major focus was on point estimates of the Multiple Imputations rather than point estimates on single imputations these leave with certain drawbacks like inability to run in Data Pipelines while productionizing and scaling the process to large-scale real-world datasets could be bit of a challenge.

5.3 Recommendations

The new approach to evaluate the imputations could be tested out in the future works alongside some state-of-the-art MVI techniques using Deep Learning algorithms such as LSTM and other ensemble approach such as Bagging, boosting etcetera for which state of the art advanced computers would also be required.

Since the missingness mechanism used is strictly restricted to MCAR but real time datasets could be combination of different mechanism such as MAR and MNAR there by those possible avenues could be ventured out.

Much larger multi-variate time series datasets with around 50 – 60 independent variables and more than 100K entries could be tested out in the future analysis. Incorporating datasets with Categorical independent variables can bring in more insights which this analysis failed to capture.

The Multiple Imputations impact on downstream ML model implementation could also be a avenue to explore. ML models could be tested out before and after the MVI technique thereby the improvement could be analysed.

i. References

Steinkraus, D., Buck, I. and Simard, P., 2005. Using GPUs for machine learning algorithms. Eighth International Conference on Document Analysis and Recognition (ICDAR'05).

Mohammed, M., Khan, M. and Bashier, E., 2016. Machine Learning.

Transforming Data with Intelligence. 2022. Press Releases | Transforming Data with Intelligence. [online] Available at: <http://www3.tdwi.org/Articles/List/Press-Releases.aspx?m=1&Page=2> [Accessed 20 April 2022]

Lin, W. and Tsai, C., 2019. Missing value imputation: a review and analysis of the literature (2006–2017). Artificial Intelligence Review, 53(2), pp.1487-1509.

Schafer, J. and Olsen, M., 1998. Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective. Multivariate Behavioural Research, 33(4), pp.545-571.

Roger, M., Li, Z., Clarke, P., Song, C., Hu, J., Feng, W. and Yi, L., 2020. Joint Inversion of Geodetic Observations and Relative Weighting—The 1999 Mw 7.6 Chi-Chi Earthquake Revisited. Remote Sensing, 12(19), p.3125.

Rubin, D., 1988. An Overview of Multiple Imputation. Statistics in Medicine, 10(4), pp.585-598.

Graham, J., 2009. Missing Data Analysis: Making It Work in the Real World. Annual Review of Psychology, 60(1), pp.549-576.

Nakagawa, S. and Freckleton, R., 2008. Missing inaction: the dangers of ignoring missing data. Trends in Ecology & Evolution, 23(11), pp.592-596.

Honaker, J. and King, G., 2010. What to Do about Missing Values in Time-Series Cross-Section Data. American Journal of Political Science, 54(2), pp.561-581.

Pratama, I., Permanasari, A., Ardiyanto, I. and Indrayani, R., 2016. A Review of Missing Values Handling Methods on Time-Series Data. IEEE.

Bokde, N., Beck, M., Martínez Álvarez, F. and Kulat, K., 2018. A novel imputation methodology for time series based on pattern sequence forecasting.

Austin, P., White, I., Lee, D. and van Buuren, S., 2021. Missing Data in Clinical Research: A Tutorial on Multiple Imputation. Canadian Journal of Cardiology, 37(9), pp.1322-1331.

Donders, A., van der Heijden, G., Stijnen, T. and Moons, K., 2006. Review: A gentle introduction to imputation of missing values. Journal of Clinical Epidemiology, 59(10), pp.1087-1091.

Reiter, J. and Raghunathan, T., 2007. The Multiple Adaptations of Multiple Imputation. Journal of the American Statistical Association, 102(480), pp.1462-1471.

Park, J., Muller, J., Arora, B., Faybishenko, B., Pastorello, G., Varadharajan, C., Sahu, R. and Agarwal, D., 2022. Long-Term Missing Value Imputation for TimeSeries Data Using Deep Neural Networks.

Honaker, J., King, G. and Blackwell, M., 2011. AmeliaII: A Program for Missing Data. Journal of Statistical Software, 45(7).

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R., 2001. Missing value estimation methods for DNA microarrays.

Aittokallio, T., 2009. Dealing with missing values in large-scale studies: microarray data imputation and beyond. Briefings in Bioinformatics, 11(2), pp.253-264.

Ma, J., Cheng, J., Ding, Y., Lin, C., Jiang, F., Wang, M. and Zhai, C., 2022. Transfer learning for long-interval consecutive missing values imputation without external features in air pollution time series.

Shi, Z., Wang, S., Yue, L., Pang, L., Zuo, X., Zuo, W. and Li, X., 2021. Deep dynamic imputation of clinical time series for mortality prediction. Information Sciences, 579, pp.607-622.

Bansal, P., Deshpande, P. and Sarawagi, S., 2021. Missing value imputation on multidimensional time series. Proceedings of the VLDB Endowment, 14(11), pp.2533-2545.

Phiwhorm, K., Saikaew, C., Leung, C., Polpinit, P. and Saikaew, K., 2022. Adaptive multiple imputations of missing values using the class center. Journal of Big Data, 9(1).

Brewer, J., Newman, I. and Benz, C., 1998. Qualitative-Quantitative Research Methodology: Exploring the Interactive Continuum. Contemporary Sociology, 28(2), p.245.

Briggs, A., Clark, T., Wolstenholme, J. and Clarke, P., 2003. Missing.... presumed at random: cost-analysis of incomplete data. Health Economics, 12(5), pp.377-392.

Schafer, J.L., 1997. Analysis of incomplete multivariate data. CRC press.

Alison, P., 2001. Missing Data. SAGE Publications, pp.100-130.

Azur, M., Stuart, E., Frangakis, C. and Leaf, P., 2011. Multiple imputation by chained equations: what is it and how does it work? International Journal of Methods in Psychiatric Research, 20(1), pp.40-49.

Abu Alfeilat, H., Hassanat, A., Lasassmeh, O., Tarawneh, A., Alhasanat, M., Eyal Salman, H. and Prasath, V., 2019. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. Big Data, 7(4), pp.221-248.

Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K., 2004. Using kNN model for automatic text categorization. Soft Computing, 10(5), pp.423-430.

Geurts, P., Ernst, D. and Wehenkel, L., 2006. Extremely randomized trees.

Kim, S. and Kim, H., 2016. A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting, 32(3), pp.669-679.

2022. Anomaly Detection and Complex Event Processing over IoT Data Streams.

Ahsan, M.M.; Mahmud,M.A.P.; Saha, P.K.; Gupta, K.D.;Siddique, Z. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance.Technologies 2021, 9, 52.https://doi.org/10.3390/technologies9030052

j. Appendix

Appendix – 1 – Link to data source

Dataset – 1 - https://archive.ics.uci.edu/ml/datasets/Air+Quality

Dataset -2 - https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction

Dataset – 3 – https://archive.ics.uci.edu/ml/datasets/Productivity+Prediction+of+Garment+Employees

Dataset – 4 - https://archive.ics.uci.edu/ml/datasets/Room+Occupancy+Estimation

Dataset – 5 - https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data

Dataset – 6 - https://archive.ics.uci.edu/ml/datasets/Power+consumption+of+Tetouan+city

Appendix – 2 – Codes for MVI methods

```
[ ] dataset_1.isnull().sum()

[ ] dataset_1.shape

[ ] dataset_1.columns

[ ] dataset_1_continous = dataset_1[['CO(GT)', 'PT08.S1(CO)', 'NMHC(GT)', 'C6H6(GT)',
        'PT08.S2(NMHC)', 'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)', 'PT08.S4(NO2)',
        'PT08.S5(O3)', 'T', 'RH', 'AH']]
```

```
dataset_1.isnull().sum()
```

```
[ ] dataset_1_continous =  np.array(dataset_1_continous)
```

MCAR mechanism with different missing rates

15% missing rate with MCAR mechanism

+ Code + Text

```
[ ] dataset1_miss_mcar = produce_NA(dataset_1_continous, p_miss=0.15, mecha="MCAR")

[ ] dataset1_mcar = dataset1_miss_mcar['X_incomp']
    Miss_15_1 = pd.DataFrame(dataset1_miss_mcar['mask'])

[ ] dataset1_mcar_15 = pd.DataFrame(dataset1_mcar)

[ ] dataset1_mcar_15.isnull().sum()
```

```
dataset1_mcar_15.head().style.highlight_null(null_color='orange')
```

	0	1	2	3	4	5	6	7	8	9
0	2.600000	nan	150.000000	11.881723	1045.500000	166.000000	nan	113.000000	nan	nan
1	2.000000	1292.250000	112.000000	nan	954.750000	103.000000	1173.750000	92.000000	1558.750000	nan
2	2.200000	1402.000000	88.000000	8.997817	939.250000	131.000000	1140.000000	114.000000	1554.500000	1074.000000
3	2.200000	1375.500000	80.000000	9.228796	948.250000	172.000000	1092.000000	122.000000	1583.750000	1203.250000
4	1.600000	1272.250000	51.000000	6.518224	835.500000	131.000000	1205.000000	116.000000	nan	1110.000000

20% missing rate with MCAR mechanism

```
[ ]
```

```
[ ] dataset1_miss_mcar_20 = produce_NA(dataset_1_continous, p_miss=0.20, mecha="MCAR")
    dataset1_mcar_15 = pd.DataFrame(dataset1_mcar).head().style.highlight_null(null_color='orange')
```

```
[ ] dataset1_mcar_20 = dataset1_miss_mcar_20['X_incomp']
```

```
[ ] dataset1_mcar_20_np = dataset1_mcar_20.numpy()
```

```
[ ] dataset1_mcar_20 = pd.DataFrame(dataset1_mcar_20_np)
```

```
[ ] dataset1_mcar_20.head().style.highlight_null(null_color='orange')
```

0s completed at 4:45 AM

25% missing rate using MCAR mechanism

```
[ ] dataset1_miss_mcar_25 = produce_NA(dataset_1_continous, p_miss=0.25, mecha="MCAR")
```

```
[ ] dataset1_mcar_25 = dataset1_miss_mcar_25['X_incomp']
```

```
[ ] dataset1_mcar_25_np = dataset1_mcar_25.numpy()
```

```
[ ] dataset1_mcar_25 = pd.DataFrame(dataset1_mcar_25_np)
```

```
[ ] dataset1_mcar_25.head().style.highlight_null(null_color='orange')
```

30% Missingness

```
[ ]  dataset1_miss_mcar_30 = produce_NA(dataset_1_continous, p_miss=0.30, mecha="MCAR")
```

```
[ ]  dataset1_mcar_30 = dataset1_miss_mcar_30['X_incomp']
```

```
[ ]  dataset1_mcar_30_np = dataset1_mcar_30.numpy()
```

```
[ ]  dataset1_mcar_30 = pd.DataFrame(dataset1_mcar_30_np)
```

```
[ ]  dataset1_mcar_30.head().style.highlight_null(null_color='orange')
```

MCAR mechanism with different missing rates-

```
[ ]  dataset1_mean_mcar_15 = SimpleImputer().fit_transform(dataset1_mcar)
     dataset1_mean_mcar_20 = SimpleImputer().fit_transform(dataset1_mcar_20_np)
     dataset1_mean_mcar_25 = SimpleImputer().fit_transform(dataset1_mcar_25_np)
     dataset1_mean_mcar_30 = SimpleImputer().fit_transform(dataset1_mcar_30)
     dataset1_mean_mcar_15 = pd.DataFrame(dataset1_mean_mcar_15)
     dataset1_mean_mcar_20 = pd.DataFrame(dataset1_mean_mcar_20)
     dataset1_mean_mcar_20 = pd.DataFrame(dataset1_mean_mcar_25)
     dataset1_mean_mcar_30 = pd.DataFrame(dataset1_mean_mcar_30)
```

Mode Imputation

```
[ ]  dataset1_mode_mcar_15 = SimpleImputer(strategy='most_frequent').fit_transform(dataset1_mcar)
     dataset1_mode_mcar_20 = SimpleImputer(strategy='most_frequent').fit_transform(dataset1_mcar_20_np)
     dataset1_mode_mcar_25 = SimpleImputer(strategy='most_frequent').fit_transform(dataset1_mcar_25_np)
     dataset1_mode_mcar_30 = SimpleImputer(strategy='most_frequent').fit_transform(dataset1_mcar_30_np)
     dataset1_mode_mcar_15 = pd.DataFrame(dataset1_mode_mcar_15)
     dataset1_mode_mcar_20 = pd.DataFrame(dataset1_mode_mcar_20)
     dataset1_mode_mcar_25 = pd.DataFrame(dataset1_mode_mcar_25)
     dataset1_mode_mcar_30 = pd.DataFrame(dataset1_mode_mcar_30)
```

Bayesian

```
dataset1_itr_mcar_15 = IterativeImputer().fit_transform(dataset1_mcar)
dataset1_itr_mcar_20 = IterativeImputer().fit_transform(dataset1_mcar_20_np)
dataset1_itr_mcar_25 = IterativeImputer().fit_transform(dataset1_mcar_25_np)
dataset1_itr_mcar_30 = IterativeImputer().fit_transform(dataset1_mcar_30_np)
dataset1_itr_mcar_15 = pd.DataFrame(dataset1_itr_mcar_15)
dataset1_itr_mcar_20 = pd.DataFrame(dataset1_itr_mcar_20)
dataset1_itr_mcar_25 = pd.DataFrame(dataset1_itr_mcar_25)
dataset1_itr_mcar_30 = pd.DataFrame(dataset1_itr_mcar_30)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/impute/_iterative.py:701: Convergence
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/impute/_iterative.py:701: Convergence
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/impute/_iterative.py:701: Convergence
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/impute/_iterative.py:701: Convergence
  ConvergenceWarning,
```

Ensemble Tree - RandomForest

```
[ ] from sklearn.ensemble import ExtraTreesRegressor
```

```
[ ] rf = ExtraTreesRegressor(n_estimators=10, random_state=0)
```

```
[ ] dataset1_rand_mcar_15 = IterativeImputer(estimator = rf,random_state=0, max_iter=50).fit_transform(dataset1_mcar)
    dataset1_rand_mcar_20 = IterativeImputer(estimator = rf,random_state=0, max_iter=50).fit_transform(dataset1_mcar_20_np)
    dataset1_rand_mcar_25 = IterativeImputer(estimator = rf,random_state=0, max_iter=50).fit_transform(dataset1_mcar_25_np)
    dataset1_rand_mcar_30 = IterativeImputer(estimator = rf,random_state=0, max_iter=50).fit_transform(dataset1_mcar_30_np)
    dataset1_rand_mcar_15 = pd.DataFrame(dataset1_rand_mcar_15)
    dataset1_rand_mcar_20 = pd.DataFrame(dataset1_rand_mcar_20)
    dataset1_rand_mcar_25 = pd.DataFrame(dataset1_rand_mcar_25)
    dataset1_rand_mcar_30 = pd.DataFrame(dataset1_rand_mcar_30)
```

KNN Imputer for MCAR mechanism - Different missing rates

```
[ ] from sklearn.impute import KNNImputer
```

```
[ ] from sklearn.impute import KNNImputer
    from sklearn.preprocessing import MinMaxScaler

    #Define a subset of the dataset
    df_knn_15 = dataset1_mcar.copy()
    df_knn_20 = dataset1_mcar_20_np.copy()
    df_knn_25 = dataset1_mcar_25_np.copy()
    df_knn_30 = dataset1_mcar_30_np.copy()

    # Define scaler to set values between 0 and 1

    scaler = MinMaxScaler(feature_range=(0, 1))
    df_knn_15 = pd.DataFrame(scaler.fit_transform(df_knn_15))
    df_knn_20 = pd.DataFrame(scaler.fit_transform(df_knn_20))
    df_knn_25 = pd.DataFrame(scaler.fit_transform(df_knn_25))
```

```
# Define KNN imputer and fill missing values
knn_imputer = KNNImputer(n_neighbors=5, weights='uniform', metric='nan_euclidean')
dataset_1_knn_imputed_15 = pd.DataFrame(knn_imputer.fit_transform(df_knn_15))
dataset_1_knn_imputed_20 = pd.DataFrame(knn_imputer.fit_transform(df_knn_20))
dataset_1_knn_imputed_25 = pd.DataFrame(knn_imputer.fit_transform(df_knn_25))
dataset_1_knn_imputed_30 = pd.DataFrame(knn_imputer.fit_transform(df_knn_30))
```

Rescaling the dataset

```
[ ] dataset_1_knn_imputed_15 = pd.DataFrame(scaler.inverse_transform(knn_imputer.fit_transform(df_knn_15)))
    dataset_1_knn_imputed_20 = pd.DataFrame(scaler.inverse_transform(knn_imputer.fit_transform(df_knn_20)))
    dataset_1_knn_imputed_25 = pd.DataFrame(scaler.inverse_transform(knn_imputer.fit_transform(df_knn_25)))
    dataset_1_knn_imputed_30 = pd.DataFrame(scaler.inverse_transform(knn_imputer.fit_transform(df_knn_30)))
```

```
] dataset1_mcar_15 = pd.DataFrame(dataset1_mcar)
  dataset1_mcar_20 = pd.DataFrame(dataset1_mcar_20_np)
  dataset1_mcar_25 = pd.DataFrame(dataset1_mcar_25_np)
  dataset1_mcar_30 = pd.DataFrame(dataset1_mcar_30_np)
```

```
] linear_interpolation_15 = dataset1_mcar_15.interpolate(method='linear')

  # Plot imputed data
  linear_interpolation_15[1][:100].plot(color='red', marker='o', linestyle='dotted')
  dataset_1['PT08.S1(CO)'][:100].plot(title='PT08.S1(CO)', marker='o')
```

Forward fill and Backward fill

```
[ ] # Ffill imputation
    ffill_imputation_15 = dataset1_mcar_15.fillna(method='ffill')

    # Plot imputed data
    ffill_imputation_15[1][:100].plot(color='red', marker='o', linestyle='dotted')
    dataset_1['PT08.S1(CO)'][:100].plot(title='PT08.S1(CO)', marker='o')
```

```
[ ] from sklearn.metrics import mean_absolute_percentage_error,mean_squared_error,mean_absolute_error
```

```
[ ] MAPE = mean_absolute_percentage_error
    MAE  = mean_absolute_error
    MSE  = mean_squared_error
```

```
[ ] Mean_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:],dataset1_mean_mcar_30)  #dataset1_mean_mcar_15
    Mode_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:],dataset1_mode_mcar_30)  #df_knn_scaled
    Itr_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:],dataset1_itr_mcar_30)
    Rand_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:],dataset1_rand_mcar_30)
    knn_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_30)
    Intp_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_30.fillna(0))
    ffill_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_30.fillna(0))
    bfill_abs_percentage_30 = MAPE(dataset_1.iloc[:,2:].dropna(),bfill_imputation_30.fillna(0))
```

```
[ ]  Mean_abs_percentage_30 = MAE(dataset_1.iloc[:,2:],dataset1_mean_mcar_30)  #dataset1_mean_mcar_15
     Mode_abs_percentage_30 = MAE(dataset_1.iloc[:,2:],dataset1_mode_mcar_30)  #df_knn_scaled
     Itr_abs_percentage_30 = MAE(dataset_1.iloc[:,2:],dataset1_itr_mcar_30)
     Rand_abs_percentage_30 = MAE(dataset_1.iloc[:,2:],dataset1_rand_mcar_30)
     knn_abs_percentage_30 = MAE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_30)
     Intp_abs_percentage_30 = MAE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_30.fillna(0))
     ffill_abs_percentage_30 = MAE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_30.fillna(0))
     bfill_abs_percentage_30 = MAE(dataset_1.iloc[:,2:].dropna(),bfill_imputation_30.fillna(0))


     Mean_abs_percentage_15 = MAE(dataset_1.iloc[:,2:],dataset1_mean_mcar_15)  #dataset1_mean_mcar_15
     Mode_abs_percentage_15 = MAE(dataset_1.iloc[:,2:],dataset1_mode_mcar_15)  #df_knn_scaled
     Itr_abs_percentage_15 = MAE(dataset_1.iloc[:,2:],dataset1_itr_mcar_15)
     Rand_abs_percentage_15 = MAE(dataset_1.iloc[:,2:],dataset1_rand_mcar_15)
     knn_abs_percentage_15 = MAE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_15)
     Intp_abs_percentage_15 = MAE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_15.fillna(0))
     ffill_abs_percentage_15 = MAE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_15.fillna(0))
     bfill_abs_percentage_15 = MAE(dataset_1.iloc[:,2:].dropna(),bfill_imputation_15.fillna(0))

     Mean_abs_percentage_20 = MAE(dataset_1.iloc[:,2:],dataset1_mean_mcar_20)  #dataset1_mean_mcar_15
     Mode_abs_percentage_20 = MAE(dataset_1.iloc[:,2:],dataset1_mode_mcar_20)  #df_knn_scaled
     Itr_abs_percentage_20 = MAE(dataset_1.iloc[:,2:],dataset1_itr_mcar_20)
```

```
Rand_abs_percentage_15 = MSE(dataset_1.iloc[:,2:],dataset1_rand_mcar_15)
knn_abs_percentage_15 = MSE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_15)
Intp_abs_percentage_15 = MSE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_15.fillna(0))
ffill_abs_percentage_15 = MSE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_15.fillna(0))
bfill_abs_percentage_15 = MSE(dataset_1.iloc[:,2:].dropna(),bfill_imputation_15.fillna(0))

Mean_abs_percentage_20 = MSE(dataset_1.iloc[:,2:],dataset1_mean_mcar_20)  #dataset1_mean_mcar_15
Mode_abs_percentage_20 = MSE(dataset_1.iloc[:,2:],dataset1_mode_mcar_20)  #df_knn_scaled
Itr_abs_percentage_20 = MSE(dataset_1.iloc[:,2:],dataset1_itr_mcar_20)
Rand_abs_percentage_20 = MSE(dataset_1.iloc[:,2:],dataset1_rand_mcar_20)
knn_abs_percentage_20 = MSE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_20)
Intp_abs_percentage_20 = MSE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_20.fillna(0))
ffill_abs_percentage_20 = MSE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_20.fillna(0))
bfill_abs_percentage_20 = MSE(dataset_1.iloc[:,2:].dropna(),bfill_imputation_20.fillna(0))

Mean_abs_percentage_25 = MSE(dataset_1.iloc[:,2:],dataset1_mean_mcar_25)  #dataset1_mean_mcar_15
Mode_abs_percentage_25 = MSE(dataset_1.iloc[:,2:],dataset1_mode_mcar_25)  #df_knn_scaled
Itr_abs_percentage_25 = MSE(dataset_1.iloc[:,2:],dataset1_itr_mcar_25)
Rand_abs_percentage_25 = MSE(dataset_1.iloc[:,2:],dataset1_rand_mcar_25)
knn_abs_percentage_25 = MSE(dataset_1.iloc[:,2:],dataset_1_knn_imputed_25)
Intp_abs_percentage_25 = MSE(dataset_1.iloc[:,2:].dropna(),linear_interpolation_25.fillna(0))
ffill_abs_percentage_25 = MSE(dataset_1.iloc[:,2:].dropna(),ffill_imputation_25.fillna(0))
```