# Missing Value Imputation for Multivariate Data Using Data Depth

Dr. Mahesh Barale
Department of Statistics
Central University of Rajasthan
Ajmer, India
Email: baralemahesh12@gmail.com

Aman Bashir Sheikh
Department of Statistics
Central University of Rajasthan
Ajmer, India
Email: aman.b.sheikh119@gmail.com

***Abstract*—The problem of missing data occurs in various practical situations and imputation of missing data becomes important task. This study addresses the prevalent issue of missing values in multivariate data. We are surrounded by missing data. Problems created by missing data in statistical analysis have long been swept under the carpet. These times are now slowly coming to an end. Despite the existence of numerous data imputation methods, this research introduces the innovative utilization of data depth to handle missing data in multivariate datasets. Data depth is gaining recognition in the field of multivariate analysis. By integrating data depth with established techniques, the aim is to increase the accuracy and robustness of imputation method, thereby enhancing overall data analysis outcomes.**

***Index Terms*—Class Based Imputation,Data Depth,k-max, Missing Values, Multivariate Data, Normalized Root Mean Square Error,**

## I. Introduction

When data for one or more variables in a dataset are missing, it's referred to as missing data. A number of factors, including incorrect data entry, broken equipment, and survey non-response, can result in these missing values. A respondent may decline to comment on a question due to privacy concerns, or they may misinterpret the question. In statistical analysis, missing values present challenges as they affect the accuracy and reliability of results. Missing data are questions without answers or variables without observations.

Multivariate data consist of measurements or observations on more than one characteristic or variable for each unit or individual in the dataset. Missing values in datasets significantly impact analysis accuracy and interpretability. Consider the available datasets in R, such as banknoten and liver disorder data (BUPA). Suppose there is some level of missingness in banknote dimension data; its presence distorts classification accuracy and hinders feature importance assessment, affecting algorithm performance. Similarly, in liver disorder diagnosis data, missing values challenge diagnostic accuracy, feature importance identification, patient risk stratification, and treatment planning. Absence of critical variables compromises diagnostic precision and predictor identification, hindering risk stratification and treatment planning. Prudent handling of missing values, through imputation or model adjustments, is essential for reliable insights and informed decisions in both classification tasks and clinical practice. Effective addressing of missing values enhances analysis reliability and utility, ultimately benefiting patient care and outcomes.

In the literature, numerous methods exist for the imputation of missing values, such as mean imputation( [Little and Rubin, 2019])( [Donders et al., 2006]), K-nearest neighbor (KNN) imputation( [Troyanskaya et al., 2001]), and random forest imputation. In mean imputation, the missing values in a column are replaced with the mean of the non-missing values. While straightforward, this method has several limitations. By imputing the same value for all missing data points, mean imputation ignores the relationships between variables, potentially leading to biased or unrealistic results. Additionally, since the mean is highly sensitive to extreme values, the imputed values may be influenced by outliers, resulting in misleading imputations.

To address these limitations, more sophisticated methods like KNN and random forest imputation( [Beretta and Santaniello, 2016]) have been used. The KNN imputation technique uses Euclidean distance to find the nearest neighbors and imputes missing values based on their values. However, it does not account for the variance-covariance of the data, which could improve precision. Incorporating the variance-covariance matrix, such as through the Mahalanobis depth function( [Neto, 2008]), enhances accuracy in imputation tasks. On the other hand, random forest imputation( [Pujianto et al., 2019])( [Stekhoven and Bühlmann, 2012]) is computationally intensive for large datasets. As the dataset size increases, the imputation process can become excessively time-consuming.(see III-D)

A non-parametric imputation method using data depth( [Mozharovskyi et al., 2019]) has shown greater precision than traditional techniques. This method identifies the data point with the maximum depth to impute missing values. However, when three or more distinct points on the convex hull share the same depth, selecting the replacement value becomes challenging. To address this, our method averages the $k$-points with the maximum depth to replace the missing value, thereby achieving more precise results (see III-C1).

*Class centre based missing value imputation* In CCMVI,

the class center is defined as the mean of data samples within a specific class, similar to the centroid concept in the K-means algorithm. This center represents the class content, and Euclidean distances between each data sample and the class center are measured to establish a threshold.Operates in two modes. The first mode identifies the imputation threshold based on the distances between class centers and their corresponding data samples. The second mode applies this threshold for imputation( [Tsai et al., 2018]).
In our class based imputation approach , we first identify and separate the columns of qualitative variables, then perform imputation for each qualitative variable individually, ensuring that their unique characteristics are preserved during the process. (see II-A3)

## II. Methodology

This study uses the Bupa and banknoten datasets to evaluate imputation methods. The proposed method leverages data depth to identify central tendencies and impute missing values. Comparative methods include K-Max, Class-based imputation, KNN, and Random Forest. We evaluate these methods using Normalized Root Mean Squared Error (NRMSE) and computational time.

### A. Proposed Method

*1) K-max method:* Let $X$ be a random vector in $\mathbb{R}^d$, represented as $X = (x_1, x_2, \ldots, x_n)^T$, where each $x_i$ is a sample point in $\mathbb{X}$. Consider $X$ as a data matrix containing missing values. We distinguish between "incomplete rows" (incomplete-row$(i)$), which have missing entries, and "complete rows" (complete-row$(i)$), which contain all values.

The simplest method for imputing missing values involves:

- Initially, we classify the data into complete and incomplete rows based on the presence of missing entries in incomplete-row$_{(i)}$.
- Next, we employ Mean Imputation to address missing values in the dataset $X$. This involves replacing missing entries with the column mean, creating a modified dataset denoted as $X_{\text{Mid}}$.
- We then construct a dummy matrix, $X_{\text{Dummy}}$, of the same dimension as $X_{\text{Mid}}$, where incomplete rows are replicated to maintain consistency.
- By assessing the depth of both $X_{\text{Dummy}}$ and $X_{\text{Mid}}$, we identify the $k$ points with the maximum depth.
- Finally, we replace missing values in the original dataset with corresponding values from $X_{\text{Mid}}$ based on the maximum depth, effectively completing the imputation process.

*2) Iterative Imputation:* The iterative imputation method is crafted to adeptly manage missing values within a dataset, denoted as matrix $X$. Through a series of iterations, it endeavors to steadily approximate missing entries until either reaching *convergence* or exhausting a predetermined *iteration limit*. Here's a concise breakdown of its core procedures

**Algorithm 1** missing_info_kmax

```
$x \leftarrow$ Data with Missing value
$mid \leftarrow$ Mean Imputed Matrix
$depth\_fun \leftarrow$ Depth Function
$k \leftarrow$ k point which having maximum depth
function MISSING_INFO_KMAX($x, mid, depth\_fun, k$)
    $x \leftarrow$ MATRIX($x$)
    $mid \leftarrow$ MATRIX($mid$)
    $row\_miss \leftarrow$ ROWS_WITH_MISSING_VALUES($x$)
    $location\_miss \leftarrow$ LOCATION_OF_MISSING_VALUES($x$)
    $location\_non\_miss \leftarrow$ LOCATION_OF_NON_MISSING_VALUES($x$)
    $incomplete\_rows \leftarrow$ ROWS_WITH_ANY_MISSING_VALUES($x$)
    $complete\_data \leftarrow$ COMPLETE_ROWS($x$)
    $incomplete\_data \leftarrow$ INCOMPLETE_ROWS($x$)
    $miss\_ind \leftarrow$ LOCATION_OF_MISSING_VALUES($incomplete\_data$)
    $imp\_data \leftarrow$ CREATE_EMPTY_MATRIX_WITH_SAME_DIMENSIONS_AS($mid$)
    for $i \leftarrow 1$ to NUMBER_OF_ROWS($incomplete\_data$) do
        $loc\_miss\_inc \leftarrow$ LOCATION_OF_MISSING_VALUES_IN_ROW($incomplete\_data[i,]$)
        $dummy\_matrix \leftarrow$ CREATE_MATRIX_WITH_INCOMPLETE_ROW_REPLACED_BY_MID($incomplete\_data[i,], mid$)
        $Est\_Meth \leftarrow$ "$moment$"
        $depth\_function \leftarrow$ CALCULATE_DEPTH_FUNCTION($depth\_fun, dummy\_matrix, mid, Est\_Meth$)
        $max\_depth\_indices \leftarrow$ TOP_K_INDICES($depth\_function, k$)
        $max\_depth\_points \leftarrow$ SELECT_POINTS_BY_INDICES($mid, max\_depth\_indices$)
        if $k == 1$ then
            $imputed\_value \leftarrow$ VALUE_OF_MAX_DEPTH_POINT_FOR_MISSING_LOCATIONS($loc\_miss\_inc, max\_depth\_points$)
        else
            $value \leftarrow$ CALCULATE_COLUMN_MEANS($max\_depth\_points$)
            $imputed\_value \leftarrow$ VALUES_FOR_MISSING_LOCATIONS_IN_COLUMN_MEANS($loc\_miss\_inc, value$)
        end if
        $row\_miss\_i \leftarrow$ ROW_MISS_INDEX_FOR_ROW($i$)
        UPDATE_IMPUTED_DATA_AND_MID($row\_miss\_i, loc\_miss\_inc, imputed\_value, imp\_data, mid$)
    end for
    $imputed\_data \leftarrow imp\_data$
    return ($imputed\_data, err$)
end function
```

The method primarily comprises two steps: Data Pre-processing & Imputation, followed by Checking Convergence Criteria.

- **Data Pre-processing and Imputation:**
    1) Identify incomplete rows containing missing values and classify the data as complete and incomplete.
    2) Impute the mean at the location of missing values in the provided dataset and designate it as $X_{\text{Mid}}$.
    3) For each incomplete row, construct the dummy matrix ($X_{\text{Dummy}}$) and compute the depth $D(X_{\text{Dummy}}, X_{\text{Mid}})$.
    4) Identify $k$ points with the highest depth corresponding to $X_{\text{Mid}}$ and replace the missing value by the mean of those $k$ values.
    5) Update the imputed dataset with the newly estimated values (new imputed data).
- **Convergence Criteria:**
    1) Monitor the maximum absolute difference between consecutive imputed datasets.
    2) Terminate the iteration if the maximum difference falls below a predetermined threshold or if the maximum number of iterations is reached.

**Algorithm 2** Iterative Imputation

```
$x \leftarrow$ Data with Missing value
$mid \leftarrow$ Mean Imputed Matrix
$depth\_fun \leftarrow$ Depth Function
$k \leftarrow$ k point which having maximum depth
$num\_iter \leftarrow$ Number of iteration
$req_err \leftarrow$ Specified Errors
function ITERATIVE_IMPUTATION($x, depth\_fun, k, num\_iter, req\_err$)
    $x \leftarrow$ MATRIX($x$)
    $location\_miss \leftarrow$ LOCATION_OF_MISSING_VALUES($x$)
    $incomplete\_row \leftarrow$ ROWS_WITH_ANY_MISSING_VALUES($x$)
    $incomplete\_col \leftarrow$ COLUMNS_WITH_ANY_MISSING_VALUES($x$)
    $complete\_data \leftarrow$ COMPLETE_ROWS($x$)
    $incomplete\_data \leftarrow$ INCOMPLETE_ROWS($x$)
    $miss\_ind \leftarrow$ LOCATION_OF_MISSING_VALUES($incomplete\_data$)
    $mean\_imp \leftarrow$ IMPUTE_MEAN($x, type = $"$columnwise$")
    $imp\_old \leftarrow mean\_imp$
    $iter \leftarrow 0$
    $dif \leftarrow \infty$
    while ($iter < num\_iter$ AND MAX($dif$) $> req\_err$) do
        $imp\_data \leftarrow$ CREATE_EMPTY_MATRIX_WITH_SAME_DIMENSIONS_AS($mean\_imp$)
        for $i \leftarrow 1$ TO NUMBER_OF_ROWS($incomplete\_data$) do
            $loc\_miss\_inc \leftarrow$ LOCATION_OF_MISSING_VALUES_IN_ROW($incomplete\_data[i,]$)
            $dummy\_matrix \leftarrow$ CREATE_MATRIX_WITH_INCOMPLETE_ROW_REPLACED_BY_MEAN($incomplete\_data[i,], mean\_imp$)
            $Est\_Meth \leftarrow$ "$moment$"
            $depth\_function \leftarrow$ CALCULATE_DEPTH_FUNCTION($depth\_fun, dummy\_matrix, mean\_imp, Est\_Meth$)
            $max\_depth\_indices \leftarrow$ TOP_K_INDICES($depth\_function, k$)
            $max\_depth\_points \leftarrow$ SELECT_POINTS_BY_INDICES($mean\_imp, max\_depth\_indices$)
            if $k == 1$ then
                $imputed\_val \leftarrow max\_depth\_points$
            else
                $imputed\_val \leftarrow$ CALCULATE_COLUMN_MEANS($max\_depth\_points$)
            end if
            $imp\_data[i,] \leftarrow imputed\_val$
        end for
        $missing\_vals \leftarrow$ VALUES_AT_MISSING_INDICES($imp\_data, miss\_ind$)
        $imputed\_data \leftarrow x$
        $imputed\_data[location\_miss] \leftarrow missing\_vals$
        $dif \leftarrow$ ABSOLUTE_DIFFERENCE($imputed\_data, mean\_imp$)
        $mean\_imp \leftarrow imputed\_data$
        $iter \leftarrow iter + 1$
    end while
    return $imputed\_data$
end function
```

*3) Class Based Imputation:* This method is specializes in imputing missing values in dataset which contain categorical variable.

**Input Variables**

- **data:** - Data Contain Missing Value(provided Data)
- **depth fun:** - Depth function guiding the imputation process.
- **k :** - Number of points with maximum depth considered.
- **num-iter :**- Maximum iterations allowed for imputation.
- **col-id:**- Column indices identifying categorical variables.

**The function proceeded following these steps as:**

- Iteratively, the function addresses each unique category within the categorical variable
- splits the dataset accordingly to the active category.
- With these steps, the function continued.removes the category variable from the segmented data and performs *iterative imputation*(Method 2) on it.
- went back into the original dataset with the imputed values added.

**Algorithm 3** CB Imputation

```
function IMPUTE_COLUMNWISE($data, depth\_fun, k, num\_iter, col\_id$)
    $imp\_data \leftarrow$ CREATE_MATRIX_WITH_ALL_ZEROS(rows = NUMBER_OF_ROWS($data$), cols = NUMBER_OF_COLUMNS($data$))
    $categories \leftarrow$ UNIQUE_VALUES_IN_COLUMNS($data, col\_id$)
    $n.cat \leftarrow$ LENGTH($categories$)
    for category in $categories$ do
        $category\_data \leftarrow$ SELECT_ROWS_WHERE_COLUMN_EQUALS_CATEGORY($data, col\_id[1]$, category)
        $mod\_data \leftarrow$ REMOVE_CATEGORY_COLUMNS($category\_data, col\_id$)
        $imp\_class \leftarrow$ iterative_imputation($mod\_data, depth\_fun, k, est\_meth = 1, num\_iter, req\_err = 0.001$)
        $imp\_data$[SELECT_ROWS_WHERE_COLUMN_EQUALS_CATEGORY_INDEX($data, col\_id[1]$, category), imp_class
        $imp\_data[, col\_id] \leftarrow$ SELECT_COLUMN_VALUES($data, col\_id$)
    end for
    return $imp\_data$
end function
```

## III. Simulation Study

### *A. Data Description*

For our performance study, we will be looking at two datasets from the *R package ddalpha*.

| **Name** | **Columns** | **Rows** |
|---|---|---|
| Bupa | 6 | 345 |
| Banknoten | 7 | 200 |

The **bupa** dataset consists of 345 records, each containing 6 attributes. These characteristics include age, gender, and drinking habits, in addition to blood test findings such as bilirubin, alkaline phosphate, and SGOT levels. Based on test findings and patient data, each record represents a patient and is intended to anticipate complications with the liver.On the other hand, the **banknoten** dataset, which is typically used for classification, includes 200 observations and 7 variables. It gives banknote dimensions in detail, including length, breadth, width, and diagonal length. Two extra columns provide more dimensional data. The difference between real and counterfeit banknotes occurs in the seventh column.

- Produced missing value by $(MCAR)$
- Missing percentage 5, 10, and 15
- Depth Function : $Mahalanobis$

### *B. Normalized Root Mean Squared Error*

Normalized Root Mean Squared Error (NRMSE) is a measure of the differences between predicted and observed values in a dataset, normalized by the range, mean, or standard deviation of the observed values. It provides a dimensionless quantity that facilitates comparison across different datasets or models.

The formula for NRMSE is:

$$NRMSE = \sqrt{\frac{mean(X_{original} - X_{imputed})^2}{var(X_{original})}}$$

Where,

- $X_{imputed}$ represents the predicted matrix,
- $X_{original}$ represents the observed Data,
- n is the number of observations. *(without missing value)*

## Why Normalize RMSE?

Normalizing RMSE allows for:

- **Comparison Across Different Scales:** Since NRMS dimensionless, it can be used to compare the performa of models across different datasets with varying scal
- **Interpreting Error Relative to Data Scale:** It helps in understanding how significant the error is relative to the magnitude of the observed data. For instance, an RMSE of 5 might be acceptable for a dataset with values ranging from 0 to 100, but not for one ranging from 0 to 10.

## Interpretation

- **NRMSE = 0**: Perfect model with no error.
- **0 < NRMSE < 1**: Error is less than the range/mean of the observed data.
- **NRMSE > 1**: Error exceeds the range/mean of the observed data, indicating a relatively poor model.

By providing a normalized metric, NRMSE offers a standardized way to assess model performance, facilitating easier comparison and interpretation of prediction errors.

### *C. Performance Study*

*1) **NRMSE At Different K-Values**:* Instead of using a single point, our method takes the average of $k$-points, which have maximum depth, to improve the accuracy of imputation when dealing with missing values. When there are several locations on the convex hull that have the same depth and it is difficult to choose just one, this method works especially well. These points are averaged to improve the quality of imputation and produce estimates which can be considered more reliable.

TABLE I
AVERAGE NRMSE AT DIFFERENT K-VALUES

| **K-Values (Maximum Depth)** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **NRMSE** | **0.47287** | 0.47238 | 0.47252 | **0.47165** | 0.47264 |

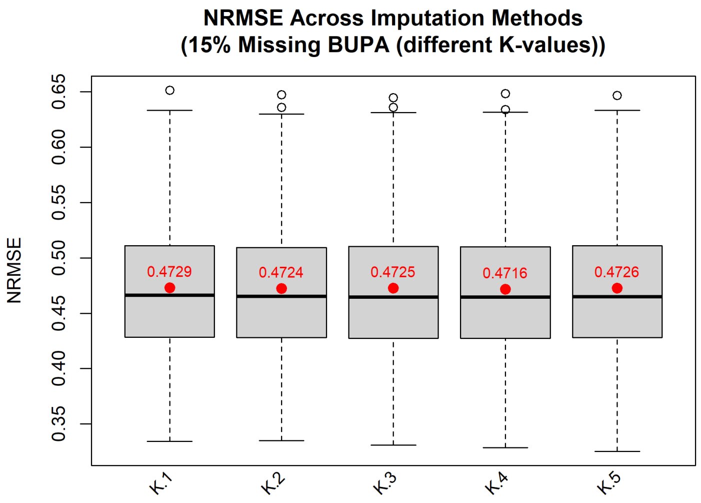

Fig. 1. **Average NRMSE At Different K-Values**

The method maintains low NRMSE scores throughout a range of $k$-values, with only minimal fluctuations, as can be seen in the boxplot for NRMSE values at different $k$-values. This consistency shows how employing many points with maximum depth is more reliable than depending on only one, resulting in more stable and accurate imputation.

*2) **Bupa Dataset at 5% Missing Value**:* First, we analyzed the Bupa dataset at different percentages of missing data under $MCAR$.

TABLE II
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | **0.453497** | 0.455851 | 0.458311 | 0.495776 | 0.493108 | 0.548099 |

At 5% missing values under MCAR, the "K_MAX" approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.453497**. The least accurate approach for this dataset is the "MEAN IMPUTATION" method, which has the highest NRMSE value of **0.548099**.

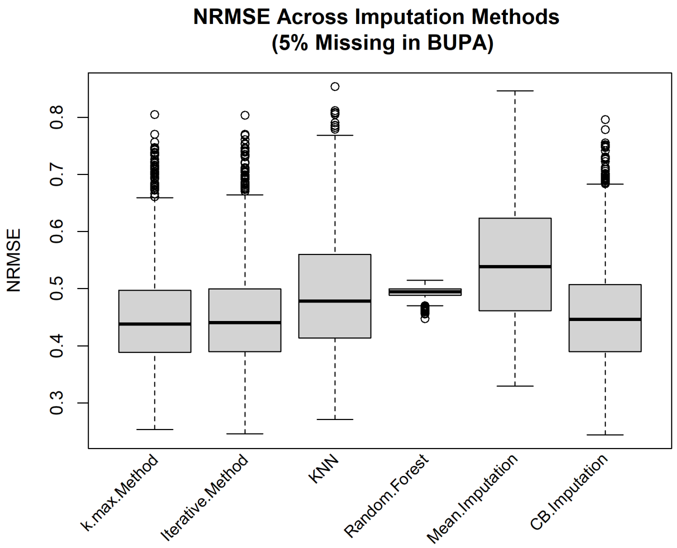

Fig. 2. **Bupa with** 5% **Missing Value**

*3) Average NRMSE for Bupa data with 10% missing values:* At 10% missing values under MCAR, the ”K_max” approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.464624**.

The least accurate approach for this dataset is the ”MEAN IMPUTATION” method, which has the greatest NRMSE value **0.555629**

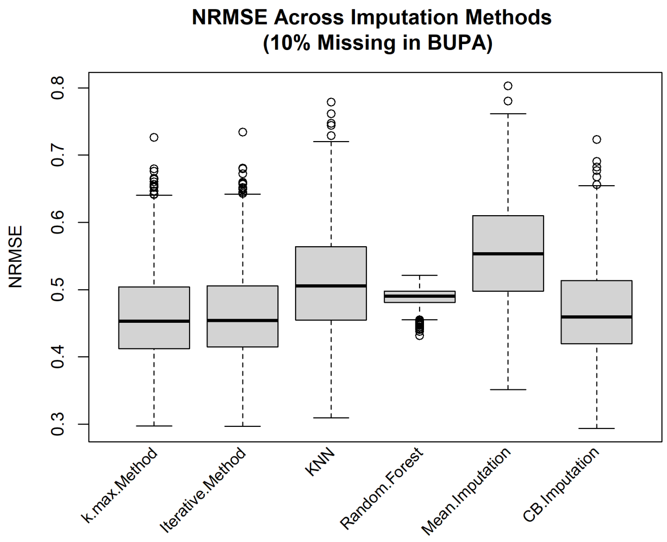

Fig. 3. **bupa with** 10% **missing value**

TABLE III
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | **0.464624** | 0.46668 | 0.47095 | 0.51278 | 0.487748 | **0.55629** |

*4) bupa dataset at 15% missing value:* At 15% missing values under MCAR, the ”K_max” approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.473875**.

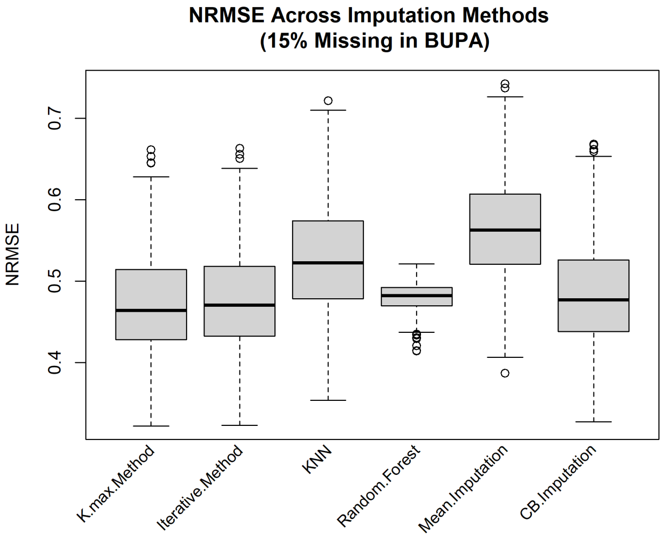

Fig. 4. **bupa with** 15% **missing value**

TABLE IV
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | **0.473875** | 0.477528 | 0.483791 | 0.526566 | 0.480133 | 0.56305 |

The least accurate approach for this dataset is the ”MEAN IMPUTATION” method, which has the greatest NRMSE value **0.56305**

*5) banknoten with* 5% *missing value:* The ”IMPUTE CLASS” approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.005688**.

The least accurate approach for this dataset is the ”MEAN IMPUTATION” method, which has the greatest NRMSE value of **0.011695**.

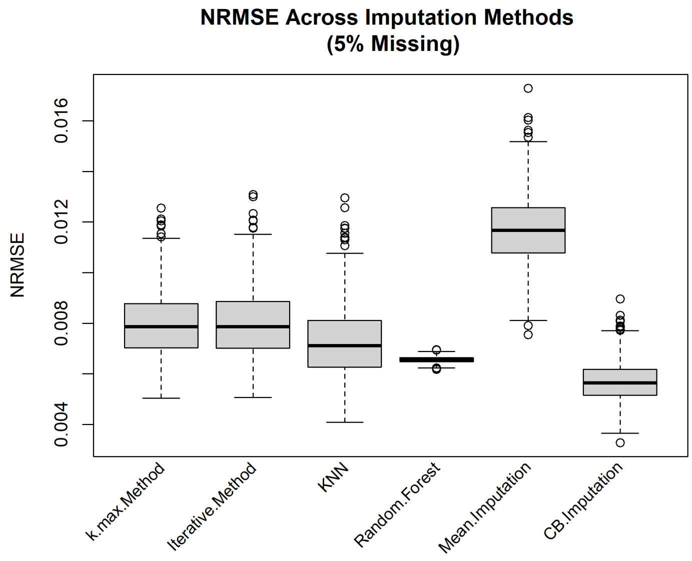

Fig. 5. **banknoten with** 5% **missing value**

TABLE V
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | 0.007964 | 0.007971 | **0.005688** | 0.007226 | 0.006557 | **0.011695** |

*6) **banknoten with 10% missing value**:* The ”IMPUTE CLASS” approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.005953**. The least accurate approach for this dataset is the ”MEAN IMPUTATION” method, which has the greatest NRMSE value **0.01173**.

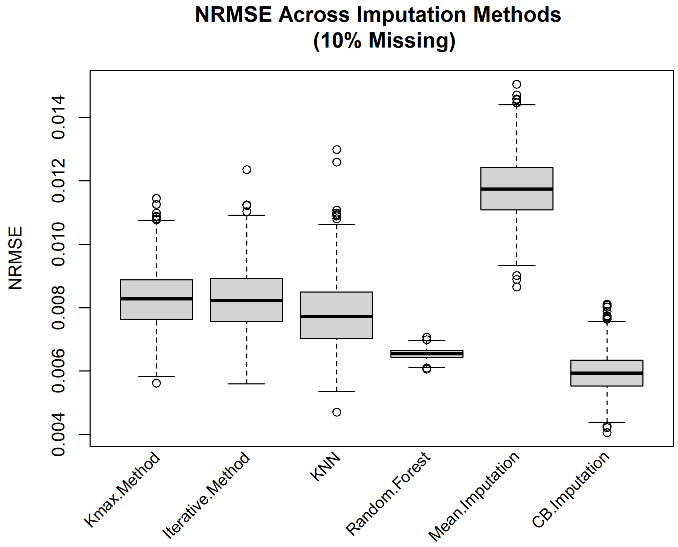

Fig. 6. **banknoten with** 10% **missing value**

TABLE VI
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | 0.008281 | 0.008267 | 0.005953 | 0.007795 | 0.006549 | 0.01173 |

*7) **banknoten with 15% missing value**:* The ”IMPUTE CLASS” approach appears to be the most accurate of the methods mentioned for the given data, as seen by its lowest NRMSE score of **0.005953**.

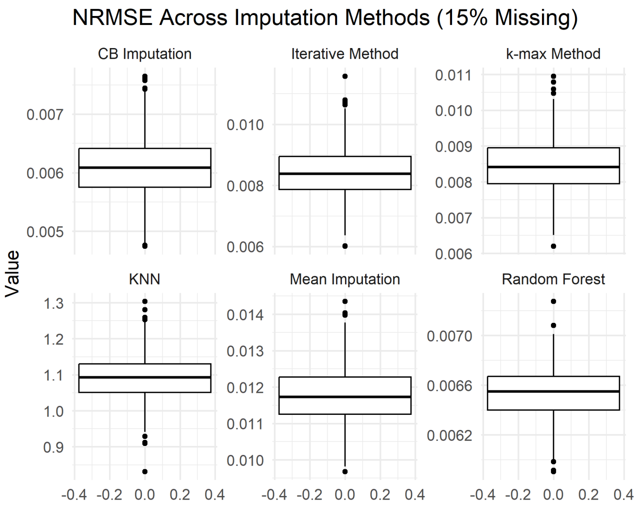

Fig. 7. **banknoten with** 15% **missing value**

TABLE VII
AVERAGE NRMSE

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | 0.008281 | 0.008267 | 0.005953 | 0.007795 | 0.006549 | 0.01173 |

The least accurate approach for this dataset is the ”MEAN IMPUTATION” method, which has the greatest NRMSE value **0.01173**

## *D. Time Comparision*

We saw that there are not many variations in the average NRMSE (Normalized Root Mean Squared Error) of our created approach and Random Forest, among other methods, from Figures 1–7. In light of this, we also examine each method’s efficiency with respect to computational time, which is the second factor in our analysis.

We determine the average time required for imputation by each approach in order to evaluate computational efficiency. We compare the imputation time required by different approaches in an effort to learn more about each method’s usefulness than just its predictive power. This assessment is essential to find the approach that generates correct data and works well under reasonable time limitations.Here we are considering the data generated from multivariate normal distribution with parameter with parameter $\mu = (0, 0, 0, 0, 0)$ & $\Sigma = diag(1, 1, 1, 1, 1)$

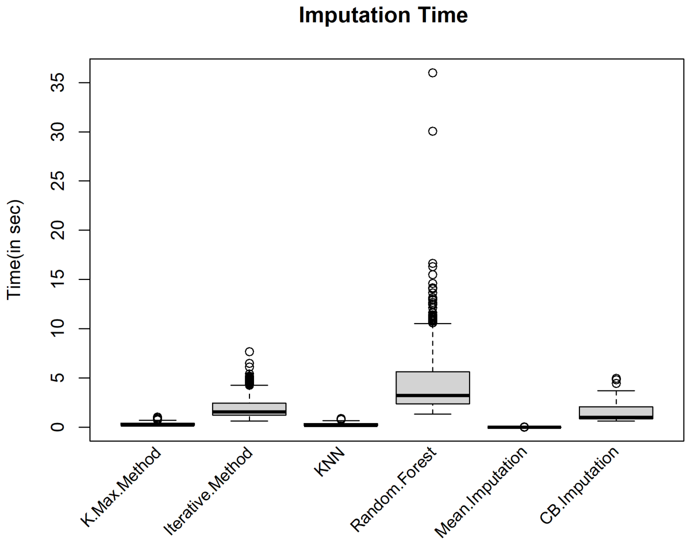

Fig. 8. **Normal distribution with 10% missing value**

TABLE VIII
AVERAGE TIME FOR IMPUTATION

| METHOD | K-MAX | ITERATIVE IMPUTATION | IMPUTE CLASS | KNN | RANDOM FOREST | MEAN IMPUTATION |
|---|---|---|---|---|---|---|
| NRMSE | 0.265664 | 1.97226 | 1.3320110 | 0.22018676 | 4.307334 | 0.0037 |

## IV. DISCUSSION

### A. Key Findings

Our study shows that *K-Max method* is the best alternative method for the BUPA dataset and achieves the lowest Normalized root mean square error (NRMSE). This shows that the model is robust and that using a model with good data can reduce errors. Also for banknote data with class-based Imputation, the best method is *class-based Imputation*. This method is suitable for data with a categorical structure, storing specific information within categories and considering relationships between categories.

## V. CONCLUSION

The results presented in the previous Section provide a comprehensive overview of the imputation techniques and their performances.

Firstly, considering the results on the BUPA dataset, it is evident that *K-Max Method* yields the lowest Normalized Root Mean Square Error (NRMSE). This demonstrates that *K-Max Method* is the most effective method among both our proposed methods and the existing imputation techniques. Its superior performance indicates its robustness and reliability in handling missing data in the BUPA dataset. The effectiveness of *K-Max Method* can be attributed to its ability to leverage the underlying patterns in the data more efficiently, thereby minimizing the error.

Additionally, for the Banknotan dataset, which includes class labels, a different approach is required. Given the structured nature of this dataset, the *Class Based Imputation* method proves to be the most effective for imputing missing values. The *Class Based Imputation* approach is specifically designed to handle datasets with classes, making it particularly well-suited for the Banknotan data. This method ensures that the imputation process respects the underlying class structure, leading to more accurate and meaningful data reconstruction. The effectiveness of the *Class Based Imputation* technique resides in its ability to take into account inter-class relationships and variations, which are essential for preserving the integrity of datasets containing information that is categorical or class-specific.

In summary, the analysis shows that *K-Max Method* is the best overall imputation technique for general datasets like BUPA, while *Class Based Imputation* is optimal for datasets with class structures, such as the Banknotan data. These findings emphasize the importance of selecting imputation methods that align with the specific characteristics of the dataset to achieve the best results.

Our proposed methods have demonstrated superior performance compared to widely used techniques like mean imputation, kNN imputation, and random forest imputation. They not only exhibit higher accuracy, evidenced by lower RMSE values across different levels of missing data, but also perform more efficiently in terms of computation time.

*K-Max Method* significantly outperforms traditional methods on the BUPA dataset, while *Class Based Imputation* effectively handles datasets with class structures, preserving class-specific information. The reduced computational time required for our methods further highlights their practicality for large-scale applications.

In conclusion, our proposed methods offer both better accuracy and significant time savings, making them advantageous over traditional imputation techniques. This dual benefit underscores their potential for adoption in various data-driven fields, enhancing the quality and efficiency of data analysis.

## Acknowledgments

I would like to express my deepest gratitude to my guide, Dr. Mahesh Barale, for her invaluable guidance, constant support, and encouragement throughout this research. Her insights and feedback have been instrumental in shaping this work.

Special thanks to my family for their unwavering support and encouragement. His belief in me and his constant motivation have been a driving force behind the completion of this work.

## References

[Beretta and Santaniello, 2016] Beretta, L. and Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16:197–208.

[Donders et al., 2006] Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091.

[Little and Rubin, 2019] Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

[Mozharovskyi et al., 2019] Mozharovskyi, P., Josse, J., and Husson, F. (2019). Nonparametric imputation by data depth. *Journal of the American Statistical Association*.

[Neto, 2008] Neto, M. R. (2008). The concept of depth in statistics. *Tech. rep.*

[Pujianto et al., 2019] Pujianto, U., Wibawa, A. P., Akbar, M. I., et al. (2019). K-nearest neighbor (k-nn) based missing data imputation. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, pages 83–88. IEEE.

[Stekhoven and Bühlmann, 2012] Stekhoven, D. J. and Bühlmann, P. (2012). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.

[Troyanskaya et al., 2001] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

[Tsai et al., 2018] Tsai, C.-F., Li, M.-L., and Lin, W.-C. (2018). A class center based approach for missing value imputation. *Knowledge-Based Systems*, 151:124–135.