BACHELOR

# Missing binary outcome observations on mixed-type data using a multiply robust imputation method

Farah, Abdullahi A.J.

*Award date:*
2023

*Awarding institution:*
Tilburg University

Link to publication

BACHELOR THESIS

# Missing binary outcome observations on mixed-type data: using a multiply robust imputation method

Abdullahi Farah

June 18, 2023

Thesis advisors: Prof. Dr. Ton de Waal and An-Chiao Liu

## Abstract

This paper introduces a multiply robust imputation framework, `Multiply-Robust-KNN-RF`, for addressing the missing data problem in mixed-type data with a binary outcome variable. $k$-NN and Random Forest algorithms are combined in this framework to support multiple imputations and provide protection against model misspecification. Extensive testing on several mixed-type datasets, followed by an evaluation of the model's performance compared to other single model multiple imputation methods based on a variety of different performance metrics. The proposed method exhibits slightly lower average bias, with a slightly higher average standard error relative to the comparison methods. The `Multiply-Robust-KNN-RF` method, in general, provides a reliable way for handling missing data in mixed-type datasets with a binary outcome variable and demonstrates to be competitive in performance compared to other single model multiple imputation methods.

# Contents

# 1 Introduction

During data collection, researchers inevitably deal with missing observations. This can happen due to various reasons, with item nonresponse being a common reason [1]. Item nonresponse is when a unit respondent opts to provide an incomplete data entry, choosing not to answer one or more survey items [2]. This leads to missing data, which poses a problem for researchers. Furthermore, it is critical to investigate whether these missing observations cause bias because this can significantly affect the inferences that can be made from the data, complicating the analysis of the study [3]. To know this, it is important to understand which type of missing data we are dealing with.

## 1.1 Literature Review

### Missing Data Mechanisms

First of all, we have missing completely at random (MCAR). This is described as the probability that it is not related to any other data characteristic and occurs when the missing data is a random subset of the entire sample. MCAR is the ideal situation as most techniques for handling missing data give unbiased results, however making the assumption that the data is MCAR is unreasonable [4, 5]. Next, we have missing at random (MAR). MAR refers to the likelihood that a data point will be missing is correlated with the observed data. This implies that the values of other observed variables in the data collection can be used to account for the missing data. This is a much more reasonable assumption to make compared to MCAR. If the missing data is neither MCAR nor MAR, then it is missing not at random (MNAR). MNAR refers to the likelihood that a data point will be missing is correlated with the actual missing data. This implies that none of the other observed variables in the data set can account for the missing data. It is important to note the distinction between MAR and MNAR as it can affect the validity of statistical analysis. If MAR, the analysis' validity can still be maintained because it is possible to use statistical methods to account for missing data. If MNAR, this might not be possible, which means that the results might be biased.

## Methods for Missing Data

### Early Developments

To prevent working with missing values, researchers have historically ignored the problem and used complete-case (CC) analysis [6]. Subjects with missing values for any of the model's variables are typically excluded from multivariable modelling in common software programs [7]. This is an approach where listwise deletion is performed on entries that have at least one missing value, leaving only complete cases. But when a sizable part of the data is missing, this approach decreases sample size, making predictions less accurate and statistical tests less powerful [8, 9]. The technique depends on the MCAR assumption because only under MCAR are the parameter estimates consistent. If MCAR conditions are not met, the method can produce biased parameter estimates. The issue is that it does not generate the minimum squared errors for a particular sample size and ignores potentially systematic differences between complete and incomplete cases.

Other missing-data handling techniques were developed to try to preserve the availability of data and maintain statistical power, such as the missingness indicator method (MIM). In this method, a new binary indicator is created to indicate whether the value of an explanatory variable is missing [10, 7]. However, despite the benefit of being able to use all available information in the analysis, the method can be quite flawed. This technique is prone to substantial bias as well as inefficiency when missing values are MAR [11]. Even when missing data is MCAR, it can result in biased associations of the original variables and outcome [4].

**Imputation Techniques**

Besides the aforementioned methods for handling missing values, there exist other strategies that involve imputation. This more sophisticated method of handling missing data can be divided into two categories: statistical and machine learning based methods [12]. Frequently used statistical based techniques are mean/mode imputation, simple regression based imputation and expectation management (EM). To start off with simple mean imputation, the average value of a missing attribute across all observed data (overall sample mean) is used as the imputed value. For the mode imputation, the missing attribute values are filled in by using the attribute value that occurs the most frequently in all of the observed data. Although these simple imputation techniques offer quick and easy ways to fill in missing values, concerns about the distortion of a variable's distribution, underestimation of variability, and compromise of relationships between variables emerge [13]. In simple regression, a variable with missing values acts as the dependent variable in a least squares regression equation, and other reliable factors in the data set are used to predict the missing value. As imputed values may be less accurate if predictor variables are not adequately correlated with variables having missing data, relevant variables must be at least fairly correlated with the variable with missing instances [14]. Just as with mean/mode imputation, underestimation of variation is a limitation to the simple regression imputation method. The EM algorithm handles issues with missing data by taking a maximum probability strategy. The algorithm is a two step iterative approach where the E-step computes the distribution for the missing data using the known values for the observed data and the current parameter estimates; and the M-step updates these parameter estimates using the maximum likelihood approach [12, 15]. The parameter estimates may not be robust however due to some outcome model misspecifications. This is because the information contained in missingness probabilities is ignored by this method, which only depends on the information from a working model predicting the missing values [11].

Frequently used machine learning based techniques are $k$-NN, cluster analysis and random forest. In $k$-NN, the missing value is imputed using a statistical measure (such as the mean) of the k-nearest data points to that data point with the missing value. In cluster analysis, the missing value is imputed by first identifying cluster centroids after grouping a set of related objects into the same cluster, then computing the distance between the missing value and the identified cluster centroids, and finally using the value of the closest centroid as the imputed value. In random forest, multiple decision trees are built using the bootstrapping method and depending on the type of missing value (numerical or categorical), the final predictions are provided by the average or majority votes of each tree's estimate. Further details on $k$-NN and Random Forest in sections 2.1 and 2.2, respectively.

**Multiple Robustness**

These methods are used to create complete data sets so that statistical analyses can be run on these complete data sets. Imputation methods can be implemented in various ways. Specifically, single-value imputation techniques determine what each missing value might have been and impute that value to the data set in place of a single value, outputting a single completed data set [16]. Statistical analyses are then performed on this single complete data set. The outcome of multiple imputation techniques, on the other hand, results in $m \geq 2$ separate complete data sets where the results are then pooled after analysis is performed on each one of the $m$ data sets separately [17]. This approach can be preferred over single imputation in some applications as it better captures the variation in missing data imputation.

The imputation procedure also allows for the simultaneous fitting of multiple models. Multiple robustness refers to this concept, which was developed to counteract model misspecification [18]. Model misspecification refers to all the potential ways that the model could fall short of accurately capturing the current situation. Since any model is only an approximation of reality, it is inevitable to come across misspecified models [19]. This can lead to biased parameter estimates and error terms as a result. When an imputation method is consistent if any one of those models is correctly specified, it is referred to as being multiple robust. By doing this, protection against model misspecification is provided. However, even when inconsistent, these multiply robust procedures still have a tendency to have good numerical performance, even if all models are incorrectly specified. Typically, procedures based on a single model do not work like this.

## 1.2 Scope

The aims of this study are to (1) introduce a multiply robust method combining multiple $k$-NN and Random Forest models for missing binary data imputation on mixed-type data and (2) evaluate the performance of this proposed method compared to single model multiple imputation methods on selected data sets. The motivation to use feature encoded $k$-NN imputation and Random Forest imputation in the proposed method is because these algorithms should, in theory, be able to handle mixed-type data and are fairly accurate and robust. Furthermore, this method supports only supervised classification tasks. This paper then aims to address the following research question:

> *To what extent does the performance of a multiply robust method combining multiple k-NN and Random Forest imputation methods differ in comparison to the performances of other single model multiple imputation methods, for the imputation of missing binary outcome observations on mixed-type data?*

# 2 Methodology

In this section, the missing data problem is addressed using a general multiply robust imputation framework on mixed data with a binary outcome variable based on machine-learning missing data imputation techniques. Specifically, $k$-Nearest Neighbours and Random Forest are used within this framework. For each method, the theoretical implementation is described as well as its extension to produce multiple imputations. Aside from this general framework, Multivariate Imputation by Chained Equations is presented as one of the comparison methods.

In this study, a univariate pattern of categorical data is assumed. This means that given an $n \times p$ data matrix, let $X$ denote the observed variables in the data set, and let $Y$ be the binary target variable. $Y$ can be partitioned even further into $Y^{obs}$, the observed set of values in $Y$, and $Y^{mis}$, the missing set of values in $Y$. Let $R$ be a $n \times p$ response indicator matrix where elements $r_{ij} = 0$ if $Y$ is missing and $r_{ij} = 1$ if $Y$ is observed. We can assume that the data are Missing Completely at Random (MCAR). This is a special case of MAR, where the likelihood of missingness does not depend on either $Y^{obs}$ or $Y^{mis}$ [20, 21]. If MCAR is assumed, then the missing data is viewed as a random sample of the complete data set. This gives that the distribution of $R$ is modelled as follows: $P(R|Y^{obs}, Y^{mis}, \xi) = P(R|\xi)$, with $\xi$ representing some unknown parameters denoting the relationship between $R$ and $Y$ [22].

## 2.1 $k$-Nearest Neighbours

$k$NN algorithms are similarity-based techniques that rely on distance measurements to impute missing data. The Minkowski norm is a popular metric for measuring the distance: $(\Sigma_{i=1}^{n}|x_i - y_i|^p)^{1/p}$. The Euclidean distance is obtained from the Minkowski norm when $p = 2$, which will be used in the remained of this study when discussing $k$-NN methods [23]. For this research, the method used makes a slight modification to the Euclidean distance metric by introducing a weight, denoted as follows:

$$dist(x_i, y_i) = (weight \times ((\Sigma_{i=1}^{n}|x_i - y_i|^2)^{1/2}), \text{ where } weight = \frac{total\ \#\ coordinates\ in\ x_i}{\#\ of\ nonmissing\ coordinates\ in\ y_i}$$

To note, this formula is a general case where multiple values can be missing in multiple columns. As stated previously, a univariate pattern of categorical data is assumed in this study; so, the weight will always be equal to 1. After this calculation, the $k$ smallest distances (which have been specified already) are selected. The values in $Y_j^{mis}$ are then imputed based on the values in $Y_j^{obs}$ that correspond to the rows with the $k$ smallest distances. The method of choice for imputation in this case is a majority vote, essentially the most frequent value among the $k$ nearest neighbours, as we are dealing with categorical variables in this research. For $k$-NN, the ideal value for $k$ is typically chosen as $k = \lfloor\sqrt{n}\rfloor$. This is possible as $k$-NN is conceptually simple and has the advantage of being non-parametric, meaning that it can be used even if the variables are categorical (provided these are properly encoded). $k$-NN methods require less hyper parameter tuning or model specifications compared to the other two methods that will be shortly discussed. Not only that, but the computational performance of this algorithm has been shown to be faster than the other algorithms [24].

## 2.2 Random Forest

The non-parametric Random Forest imputation approach can be used as it is capable of handling any type of input data. By using a random forest that has been trained on observed values to forecast missing values, this method implements an iterative imputation scheme. RF-based imputation techniques do not require the specification of parametric models or the assumption of normality [24]. This proves to be very useful as then the only parameter that needs to be tuned for this algorithm is the number of iterations.

For this research, the method imputes the missing values in an iterative fashion using Random Forests. The algorithm works as follows:

**Step 1** Set a candidate column $Y$, column with the smallest number of missing values

**Step 2** Impute missing values of all non-candidate columns with either its column mean or its column mode, depending on whether the column consists of numerical values or categorical values

**Step 3** Fit Random Forest model, with outcome variable as the candidate column and predictors as the non-candidate columns over all rows where the candidate column values are not missing

**Step 4** Impute missing rows of the candidate column using the predictions from the fitted RF

**Step 5** Repeat steps 1-3 for each column with a missing value

To note, this algorithm is presented for the case when multiple features have missing values, but naturally this also applies in the case where you only have a single feature with missing values.

## 2.3 Multivariate Imputation By Chained Equations (MICE)

Single imputation techniques include $k$-NN and Random Forest imputation, which replace a missing value with a single value. On the other hand, Multivariate Imputation by Chained Equations (MICE) is a popular multiple imputation method [25]. The statistical uncertainty in the imputations is taken into consideration by making multiple imputations as opposed to just one. The missing data are assumed to be MAR or MCAR by MICE. This is crucial since using MICE on data that is not MCAR or MAR (e.g. MNAR) could result in biased estimations, as the missingness is related to the unobserved data. Many multiple imputation techniques that were first created assumed a large joint model for one or more of the variables, for example a joint normal distribution. MICE, on the other hand, performs a sequence of regression models in a different way, modeling each variable with missing data as a function of the other variables in the data. This means that, unlike earlier imputation techniques, MICE does not call for the specification of a multivariate distribution for the missing data.

Additionally, MICE employs a chained equations method, which is particularly flexible and capable of handling complex variables like bounds or survey skip patterns. The imputation model can be specified separately for each data column using the chained equations method. This chained equation method allows for the concatenation of univariate techniques to impute the missing values. This procedure is carried out until convergence: typically the number of iterations needed to reach this is about 10-20.

## 2.4 Multiple Imputation Procedure

In the previous sections, $k$-NN and RF methods were presented for single imputation which produced a single completed data set. In order to extend these methods to produce multiple imputations, the following procedure is executed: Generate $M$ bootstrapped samples of the original data set using random sampling with replacement from the target feature; run $k$-NN or RF at each iteration $m$; and then, output the entire set of $M$ imputed data sets. Each of the $M$ full data matrices with imputed values can now be analysed separately, and the results are pooled.

## 2.5 Multiply Robust KNN-RF

The proposed method combines the multiple imputation $k$-NN and RF methods. This `Multiply Robust KNN-RF` method should then in theory conduce consistent results, assuming that at least one of the models is correctly specified. Even though this cannot be checked, this proposed method fits multiple models; so the more models fitted, the more likely the assumption is met. The described algorithm assumes that the number of models specified is always odd. This requirement allows for a majority vote to be applied that ensures a clear and unambiguous outcome which enhances the robustness of the approach, as it helps balance out any disagreement among the models. If the algorithm also allowed for the specification of an even number of models, then the possibility of having an equal number of votes for different predictions would exist. As a result, there would be uncertainty and it would be challenging to then make a final decision. The algorithm returns $A^{imp}$, containing $M$ full data matrices with imputed values. Each data matrix can then be individually analyzed (or pool and analyze the imputed data sets). The pseudocode for the proposed method can be found in Algorithm 1 of Appendix A.

# 3 Practical Application

## 3.1 Datasets

### Data Description

The following data sets were used in this research: `TitanicSurvival`, detailing the survival status, sex, age and passenger class of passengers in the Titanic disaster of 1912; `SmokeBan`, a study that estimates how smoking prohibitions at work affect the indoor smoking of employees; `PhDPublications`, providing cross-sectional data on the scientific output of biochemistry PhD students; and lastly, `ResumeNames`, containing cross-section data about resume, call-back and employer information for fictitious resumes. The data sets cover a range of sample sizes $n$ and number of dimensions $p$, with a varying degree of numerical variables and categorical variables present in each data set. A target feature $Y$ is selected for each data set, with the condition that is a binary outcome variable. The following variables were chosen as the target features:

1. `TitanicSurvival`: *survived* (Did the passenger survive?)

2. `SmokeBan`: *smoker* (Is the individual a current smoker?)

3. `PhDPublications`: *gender* (Male or Female)

4. `ResumeNames`: *call* (Was the applicant called black?)

### Data Handling & Preprocessing

The first step of preprocessing was to produce complete matrices by eliminating rows that already included missing values. For `TitanicSurvival`, the `age` feature contained missing values; so all passengers with missing age values were dropped. This was the only data set containing already existing missing data. The next step in the preprocessing phase was to prepare the data for modelling by applying feature scaling and feature encoding to each data set. To ensure that all features contribute equally to the model and to avoid features with higher values from dominating the model, feature scaling was used. The data can be changed to a more uniform scale by using feature scaling, which makes it simpler to create accurate and efficient models. Moreover, $k$-NN does not handle features with differing scales very well and this is done as a solution to that problem. The feature scaling technique used was normalization. As for feature encoding, $k$-NN can not handle categorical values. Therefore, it is important to convert the categorical values of these attributes into numerical ones before doing any modelling. As for the Random Forests used in this study, despite no scaling being necessary for this algorithm and it being capable of handling categorical variables, it was opted to perform the same preprocessing steps before applying Random Forest for consistency purposes. The exact values for $n$ and $p$, as well as the $k$-values for `knn-1` ($k_1$) and `knn-2` ($k_2$), after preprocessing are given in Table 1, 2 and 3.

### 3.2 Experimental Setup

A series of computational experiments consisting of seven data imputation models are run on the aforementioned data sets, with the performance of `Multiply-Robust-KNN-RF` compared against the other six data imputation methods. The comparison methods used in this experiment are applications of methods discussed in sections 2.1 - 2.4. Here is an overview of the models used:

1. `Multiply-Robust-KNN-RF`: This proposed method is a multiply robust method that makes use of five different models. Each model imputes the values of the generated missing data and using majority voting, a single value is imputed in each run of the experiment. All of these sub-models are also used as comparison methods.

    (a) $k$-Nearest Neighbors: Multivariate imputation that estimates missing features using nearest samples. Two $k$-NN models were used with different $k$-values:

    i. `knn-1` with $k = \lfloor \sqrt{n} \rfloor$

    ii. `knn-2` with $k = \lfloor n^{1/p} \rfloor$, where $p$ is the number of dimensions

    (b) Random Forest: Non-parametric imputation that estimates missing features using Random Forests in an iterative fashion. Three Random Forest models were used with different $t_{max}$ values (maximum iterations):

    i. `rf-1` with $t_{max} = 1$

    ii. `rf-2` with $t_{max} = 5$

    iii. `rf-3` with $t_{max} = 10$

2. `MICE`: Multivariate imputation that estimates missing values by modeling each feature with missing values as a function of other features in a round-robin fashion, with the number of maximum imputation rounds set to 10. With 50 runs of the experiment, this would total a maximum of 500 imputations. For categorical variables, the strategy used to initialize the missing values is set to 'most frequent'.

To test the performance of these imputation methods, the complete data set is taken as the ground truth and prior to each run, MCAR patterns of missing data in the binary target variable are generated (30%). Totalling 50 runs for each method, after each run the imputed values in $Y^{mis}$ are tested against the actual values in the ground truth. The assessment of the quality of the imputation is based on the following performance metrics: accuracy, bias, SE and confusion matrix. The results are then interpreted in relation to the comparison methods. As the sample size (ranging from $n$ = 915 to 10,000) and number of dimensions (ranging from $p$ = 4 to 46) in the various data sets examined vary substantially, it is expected that this makes them appropriate for performance testing.

All experiments were conducted using Python 3.9.16 and all missing value imputation methods used rely on existing Python implementations. The $k$-NN method was implemented using KNNImputer from `scikit-learn` [26], the Random Forest method was implemented using MissForest from the `missingpy` library [27] and MICE was implemented using IterativeImputer from `scikit-learn` [28]. The proposed method combines multiple KNNImputer/MissForest functions.

## 3.3 Results

### Accuracy

In Figure 1 of Appendix B, the accuracy results for each imputation method per data set across all imputation rounds is presented. Accuracy is a performance metric that captures the amount of correct predictions made by the model in relation to the total number of predictions made. To be specific, the mean accuracy across all imputation rounds is calculated for each data imputation method:

$$Mean\ Accuracy = \frac{1}{M}\Sigma_{m=1}^{M}\frac{\#correct\ predictions}{\#total\ predictions}$$

The benchmark methods and Multiply-Robust-KNN-RF method are compared; the method with the highest average accuracy for each data set is marked in bold (see Table 1). This is done as we are mostly concerned with comparisons between imputation methods within data sets, and not across data sets. Interesting enough, the Multiply-Robust-KNN-RF method obtains the highest mean accuracy score for only the first data set; but only cracks the top 3 for 2 out of the 4 data sets. Comparatively, knn-1 achieves the highest mean accuracy for 3 out of the 4 data sets; while knn-2 and rf-1 have the weakest performances (both having the lowest mean accuracy in 50% of the data sets). In any case, it does not seem that the performance of Multiply-Robust-KNN-RF relative to the comparison methods does not appear to significantly vary in each data set. The accuracy scores for all methods run on the PhDPublications data set are, on average, lower than those for the other data sets. As this data set also has the smallest sample size ($n = 915$), it could be reasoned that small sample sizes are often correlated with more variance, leading to a more likely chance of having insignificant results. However, the sample size of this data set is larger than the general rule of thumb for data collection ($n \geq 30$), so it woud be more likely that this decreased performance is due to poor-quality data and/or the nature of the target feature $Y$ (as it is a sensitive feature). In contrast, the imputation methods seem to overfit on the ResumeNames data set; with most of the methods achieving a mean accuracy score $> 90\%$. Given that this data set comprises 46 characteristics, it can be suggested that this is the result of the dimensionality curse, which results in a decrease in accuracy score quality as dimensionality grows due to an increase in variance. A more plausible explanation is that this is an imbalanced dataset that is highly skewed towards the majority class (92% of the target feature $Y$ is labeled as ”no”). As there are so few examples in the minority class, it is possible to classify every single instance in the test set as the negative class and still have a very high accuracy.

| Name | $n$ | $p$ | $k_1$ | $k_2$ | knn-1 | knn-2 | rf-1 | rf-2 | rf-3 | mice | mr-knn-rf |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TitanicSurvival | 1046 | 4 | 32 | 5 | 0.7719 | 0.7861 | 0.5873 | 0.7765 | 0.7761 | 0.7780 | **0.7885** |
| SmokeBan | 10000 | 7 | 100 | 3 | **0.7584** | 0.7106 | 0.7573 | 0.7235 | 0.7234 | 0.7573 | 0.7409 |
| PhDPublications | 915 | 6 | 30 | 3 | **0.6410** | 0.6185 | 0.5359 | 0.6221 | 0.6223 | 0.6343 | 0.6299 |
| ResumeNames | 4870 | 46 | 69 | 1 | **0.9188** | 0.872 | **0.9188** | 0.9088 | 0.9089 | **0.9188** | 0.9122 |

Table 1: Mean accuracy score for each imputation method on all 4 data sets. The highest mean accuracy score for each data set is in **bold**.

### Bias

The bias of the estimated category proportions of the target variable is also used to assess each imputation method's performance. Calculating the estimator bias, which represents how different the data is in its value distribution compared to before imputation, involves comparing the category proportions before and after imputation. The proportion calculated from the original data set before resampling is the true proportion. After imputation, the proportion of the original data set becomes the estimated proportion. In order to determine the category proportions, the frequency of each element in the data set (original data set for true proportion and imputed data set for estimated proportion) is divided by the sample size (equal across both original and imputed data sets). The true proportion is subtracted from the estimated proportion in order to calculate the bias, which is expressed as $\hat{\theta} - \theta$. The average bias will be the average of the predicted proportion minus the true proportion across all imputations, or equivalently $E[\hat{\theta} - \theta]$.

In Table 2, it is shown that `Multiply-Robust-KNN-RF` has a low positive average bias for `TitanicSurvival` and `ResumeNames`, and a moderate positive average bias for `SmokeBan`; while having a low negative average bias for `PhDPublications`. For the single model multiple imputation methods, they follow a similar structure as that of the proposed method; except for `rf-1`. The average bias for this method is relatively high on `SmokeBan` and `PhDPublications`, making `rf-1` the weakest performing method in terms of bias assessment. In contrast, `rf-2` and `rf-3` (the two other RF models) achieve the lowest average bias on 1 out of the 4 data sets each. This could be explained by the fact that `rf-1` can only have a maximum number of iterations per bootstrapped sample equal to 1. It should be noted that increasing $t_{max}$ may not always result in a reduction in bias. Increasing $t_{max}$, however, may indirectly affect bias by enabling the model to learn more complex patterns and reduce underfitting; giving the model more opportunities to update its parameters and adjust its predictions, which can lead to a lower bias (up to a certain point before overfitting); and providing the model with more opportunities to update its parameters and adjust its predictions. The highest performing method was `knn-2`, achieving the lowest average bias on 50% of the data sets. According to the results for `knn-1` and `knn-2`, it appears that decreasing the value of $k$ lowers the bias. When comparing `Multiply-Robust-KNN-RF` to the other single model multiple imputation methods, the proposed method is found to have a slightly lower average bias relative to the mean of the average bias across the other imputation methods on all data sets. An advantage to the multiply robust method that can be observed in the results is that the high bias of `rf-1` is offset by the other sub-models, further showcasing the robustness of the method.

| Name | $n$ | $p$ | $k_1$ | $k_2$ | `knn-1` | `knn-2` | `rf-1` | `rf-2` | `rf-3` | `mice` | `mr-knn-rf` |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `TitanicSurvival` | 1046 | 4 | 32 | 5 | 0.1290 | 0.0472 | 0.4127 | 0.0426 | **0.0425** | **0.0425** | 0.0894 |
| `SmokeBan` | 10000 | 7 | 100 | 3 | 0.2280 | **0.0914** | 0.2427 | 0.1313 | 0.1313 | 0.2427 | 0.1711 |
| `PhDPublications` | 915 | 6 | 30 | 3 | -0.0228 | -0.0259 | -0.4641 | **-0.0114** | -0.0118 | -0.0195 | -0.0702 |
| `ResumeNames` | 4870 | 46 | 69 | 1 | 0.0812 | **0.0092** | 0.0812 | 0.0606 | 0.0607 | 0.0812 | 0.0680 |

Table 2: Average bias of category 0 for each imputation method on all 4 data sets. The lowest bias for each data set is in **bold**.

### Standard Error

The imputed data sets were also analyzed to obtain the standard errors of the outcome categories using the estimated proportions. The standard error (SE) calculates how accurately any given sample's projected category proportions match the true category proportions. Before the SE could be computed, the MSE was calculated after each imputation. This would be equal to taking the squared difference between the estimated proportion and the true proportion. The average MSE is then equal to the average of the squared differences between the estimated proportion and the true proportion. Now, having calculated the average MSE and the previously calculated average bias, the average variance can be obtained using $Variance = MSE - bias^2$. Finally, taking the square root of the average variance gives the average SE.

The results in Table 3 suggest that `rf-1` has the lowest average SE out of all the imputation methods on 100% of the data sets. Besides this method, `mice` also achieves the lowest average SE recorded alongside `rf-1` on 2 out of the 4 data sets; while `knn-1` is the highest performing model exactly 1 time, together with `rf-1` and `mice`, on `ResumeNames`. Throughout all of the imputation methods on all data sets, it can be seen that the average SEs do not differ greatly and are quite low. It should be noted, however, that the two data sets with the highest sample size also exhibit the lowest average SEs across all data imputation methods. This is to be expected, as increasing $n$ lowers the standard error. What can be observed is that within each $k$-NN sub-model and RF sub-model, decreasing the $k$-value or increasing the number of maximum iterations also slightly increases the average SE in most cases. However, there will come a point where changing these hyper-parameters will not yield much improvement (e.g. increase in SE from 0.0257 to 0.0471 to 0.0472 for `rf-1` - `rf-3`). Again, the proposed method `Multiply-Robust-KNN-RF` does not appear to perform better than some of its individual components, but the results show that it is very close in terms of average SEs.

| Name | $n$ | $p$ | $k_1$ | $k_2$ | `knn-1` | `knn-2` | `rf-1` | `rf-2` | `rf-3` | `mice` | `mr-knn-rf` |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `TitanicSurvival` | 1046 | 4 | 32 | 5 | 0.0354 | 0.0366 | **0.0257** | 0.0335 | 0.0336 | 0.0269 | 0.0324 |
| `SmokeBan` | 10000 | 7 | 100 | 3 | 0.0098 | 0.0157 | **0.0084** | 0.0131 | 0.0129 | **0.0084** | 0.0118 |
| `PhDPublications` | 915 | 6 | 30 | 3 | 0.0573 | 0.0447 | **0.0257** | 0.0471 | 0.0472 | 0.0765 | 0.0504 |
| `ResumeNames` | 4870 | 46 | 69 | 1 | **0.0070** | 0.0112 | **0.0070** | 0.0087 | 0.0087 | **0.0070** | 0.0080 |

Table 3: Average SE of category 0 for each imputation method on all 4 data sets. The lowest standard error for each data set is in **bold**.

### Confusion Matrix

The final evaluation metric used in this study was visualizing a performance of each imputation algorithm in a confusion matrix. Each row of the confusion matrix represents the true instances of a class and each column of the confusion matrix represents the predicted instances of a class. After each imputation, a confusion matrix is determined by computing the amount of $TP$, $TN$, $FP$, $FN$ counts per sample and the average of these values across all imputations is represented in a final confusion matrix. The final confusion matrix is represented with normalization by class support size; which could be beneficial in the event of a class imbalance in order to have a more visual understanding of which class is being incorrectly labeled.

In Appendix C.1 - C.4, confusion matrices for every method on all data sets are presented. `Multiply-Robust-KNN-RF` is highlighted in each subsection with its own figure and the other single model multiple imputation methods are grouped together. By adding the diagonals of each confusion matrix containing the true positive rate ($TPR$) and true negative rate ($TNR$), it can be determined which imputation model correctly predicts each class the best. For the `Titanic-Survival` data set, the proposed method has the highest combination of $TPR$ and $TNR$ (as does `knn-2`). Most of the other imputation methods have similar values in each cell, except for `rf-1`. This method has a $TNR$ of 1, meaning that it never falsely labels the class ”no” as the class ”yes”. However, it always falsely labels the class ”yes” as the class ”no” ($FNR = 1$). Matter of fact, this happens on every data set for `rf-1`. On the `SmokeBan` data set, and more so on the `ResumeNames` data set; the confusion matrices present a very high $TNR$ and $FNR$ across all imputation methods. This makes sense, as these two data sets are the most imbalanced out of the 4 data sets. For the `PhDPublications`, the results suggest that most of the imputation methods have very similar $TPR$, $TNR$, $FPR$, $FNR$ and are quite balanced. When comparing `Multiply-Robust-KNN-RF` to the benchmark methods, the results show that it is outperformed slightly by only `knn-1` and `mice`.

# 4 Discussion

One of the main aims of this paper was to address the missing data problem by introducing a general multiply robust imputation framework on mixed-type data with a binary outcome variable based on $k$-NN and Random Forest. This method is extended to create multiple imputations and it is specifiable to allow for multiple $k$-NN and Random Forest models; increasing the likelihood of any one of these models being correctly specified, and providing protection against model misspecification.

The performance of the multiply robust method is evaluated against six single model multiple imputation methods through extensive experimentation on 4 mixed-type data sets with at least one binary feature present; with all data sets varying in sample size and dimensionality. The results show that `Multiply-Robust-KNN-RF` achieves moderate to high accuracy scores across all data sets and performs well relative to the comparison methods. It performed the best compared to the other benchmark methods on a single occassion, while being at most 2% lower in accuracy compared to the best performing model.

The proposed method is shown to be slightly biased and have low average SEs across most of the data sets, suggesting that this method is quite reliable. `Multiply-Robust-KNN-RF` had a slightly lower average bias relative to the average bias of the other single model methods in all scenarios. The results also show that this method had a slightly higher average standard error relative to the other single model methods in half of the scenarios. For the single model multiple imputation methods, the worst performing model in terms of the highest average bias was `rf-1` for half of the data sets; while it had the lowest average SE across all imputation methods. This bias-variance tradeoff can be found in the results when looking at the various $k$-NN and Random Forest sub models. According to the results, decreasing the value of $k$ or increasing the maximum number of iterations consistently show to lower the bias, but also increase the standard error (thus, an increase in variance).

The confusion matrix provided some insights into the performance of each imputation method for each class. The results suggest that the performance depends on the specific data set, as the imbalanced nature of some of the data sets resulted in very high $TNR$ and $FNR$ values for all imputation methods.. `Multiply-Robust-KNN-RF` outperforms all of the benchmark methods on a single occassion, alongside `knn-2`, as they had the highest combination of $TPR$ and $TNR$. On all other occassions, the proposed method showcases similar results to most of the other comparison models. Except for `rf-1`, which consistently exhibited very high $TNR$ and always falsely labeled the category 1 as the category 0 ($FNR = 1$) across all data sets; this was quite surprising. Due to the performance of the `Multiply-Robust-KNN-RF` method varying across these different evaluation metrics, and no single method consistently outperforming others in all aspects; it would be better practice to provide a list of recommendations for various demands. If one is interested in accuracy, on either a balanced or imbalanced data set, then `knn-1` should be recommended based on the results. If the bias-variance tradeoff is to be considered, then the method that balances these the best is `MICE` as it is the only method to have achieved the lowest average bias and the lowest average SE for at least one data set each. For the method that best evaluates the performance as a representation of actual vs. predicted values, then `knn-2` is recommended.

### 4.1 Limitations and Future Work

A limitation of this study was the lack of model selection and parameter tuning. As this was not the focus of the paper, it was opted to exclude this and implement a specific variation of the proposed method with sub-optimal hyperparameter values and a specific number of models that were somewhat backed up the theory. This was also to reduce time spent on procedures that could have otherwise been used on setting up the experiment and evaluating the methods. However, during the evaluation of the methods, it was apparent that the performance of some of the imputation methods would have benefited from some hyperparameter tuning and it would have been interesting to implement the experimental setup with optimized methods and model selection.

Another limitation of the study is the design of the experiments. The experiment is set up to compare the multiply robust method against other single model methods on the basis of a few performance metrics. There is no study done into the effects of varying the missingness mechanisms, the category proportions of the imbalanced data sets or the missing rate. In future work, this could be implemented. The multiply robust method could have also been implemented differently; instead of using a majority vote, weights could have been established. The multiply robust method could have also been compared to other doubly or multiply robust methods, instead of just single model multiple imputation methods.

## 5 Conclusion

In conclusion, through a series of experiments on 4 mixed-type data sets; the proposed multiply robust imputation framework, `Multiply-Robust-KNN-RF`, demonstrates that it is competitive in performance compared to other single model multiple imputation methods irrespective of the sample size and dimensionality of the data sets used. The method has shown to produce moderate to high accuracy scores. Compared to the other imputation methods, it exhibits reliability with somewhat lower average bias and slightly greater average standard error. Increasing the maximum number of iterations or decreasing the value of $k$ both reduce bias but increase SE, as is the case with the bias-variance tradeoff. Performance of the method differs across evaluation parameters, pointing to the necessity for recommendations based on particular requirements. It is advised to use `knn-1` for accuracy, `MICE` to balance bias and variance, and `knn-2` to assess performance in terms of how well actual values compare to the predictions.

## Acknowledgements

The author thanks his thesis advisors, Prof. Dr. Ton de Waal and An-Chiao Liu, for their help and support during the period of this study and their much-needed feedback.

# A Pseudocode `Multiply-Robust-KNN-RF`

**Algorithm 1** `Multiply-Robust-KNN-RF`

```
Input: Complete data set ($A$), target feature ($Y$), number of bootstrapped samples ($M$), list of numerical columns in data set ($X^{num}$), missing rate ($q$), list of k-values ($k$), list of max iteration values ($t_{max}$)
Output: $A^{imp} \leftarrow M$ complete data sets with $A^{imp,m} = (\mathbf{X}, Y^m)$ at iteration $m$
Initialize $A^{imp}$ as an empty array
$total = k.length + t_{max}.length$
if $total$ is even then
    Raise Error ("Number of total models should be odd")
end if
for $m = 1$ to $M$ do
    Initialize random seed to a new seed value of $m$
    Normalize $A^m[X^{num}]$
    Draw bootstrap samples with replacement from the original data set to create $Y^m = (Y^{obs,m}, Y^{mis,m})$ using a missing rate of $q$
    Initialize $P$ as an empty array
    for each $k\_val$ in $k$ do
        Train KNN model (see section 2.1); replace $Y^{mis,m}$ with predictions from model
        Inverse scale $A^m[X^{num}]$; round imputations to get binary outcomes
        Append $A^m[Y]$ to $P$
    end for
    for each $t$ in $t_{max}$ do
        Train RF model (see section 2.2); replace $Y^{mis,m}$ with predictions from model
        Inverse scale $A^m[X^{num}]$; round imputations to get binary outcomes
        Append $A^m[Y]$ to $P$
    end for
    Transpose the array $P$
    Initialize $MV$ as an empty array
    for each $pred$ in $P$ do
        Count the occurrences of each unique element in $pred$
        $majority \leftarrow$ element with the highest count
        Append $majority$ to $MV$
    end for
    Replace $A^m[Y]$ with values from $MV$
    Append $A^m$ to $A^{imp}$
end for
return $A^{imp}$
```

## B Accuracy plots

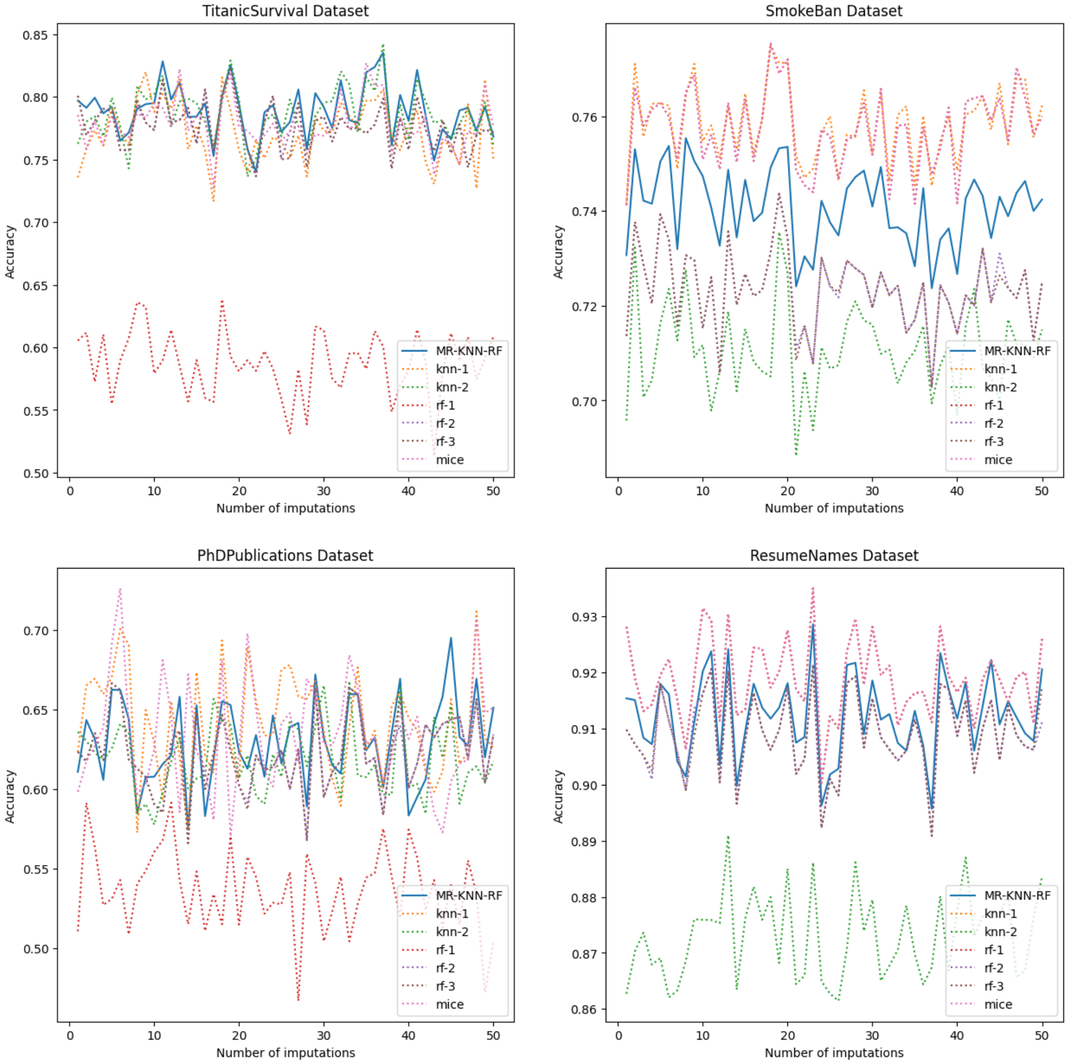

Figure 1: Accuracy plots for each dataset, comparing the proposed method `Multiply-Robust-KNN-RF` to the comparison methods. The x-axis corresponds to the accuracy score at that particular imputation round. The multiply robust method is solid and the single model multiple imputations are dotted.

# C Confusion Matrices

## C.1 TitanicSurvival Dataset

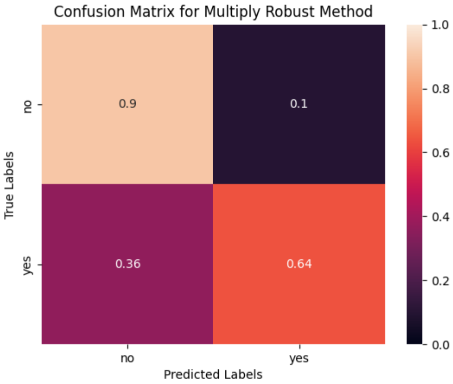

Figure 2: Confusion matrix for `Multiply-Robust-KNN-RF` on the `TitanicSurvival` data set

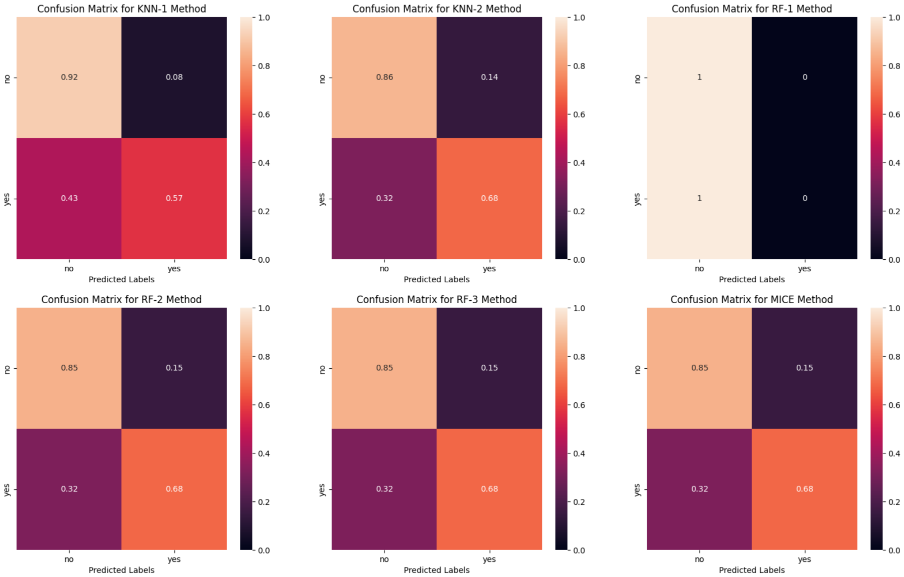

Figure 3: Confusion matrix for all the other single model multiple imputation methods on the `TitanicSurvival` data set

## C.2 SmokeBan Dataset

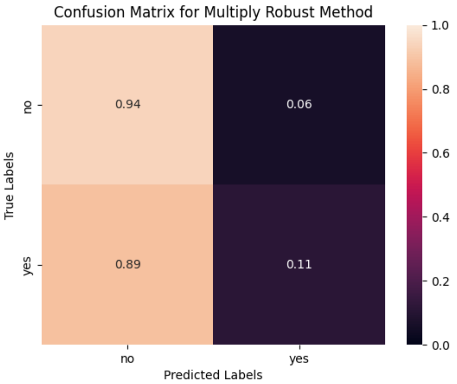

Figure 4: Confusion matrix for `Multiply-Robust-KNN-RF` on the `SmokeBan` data set

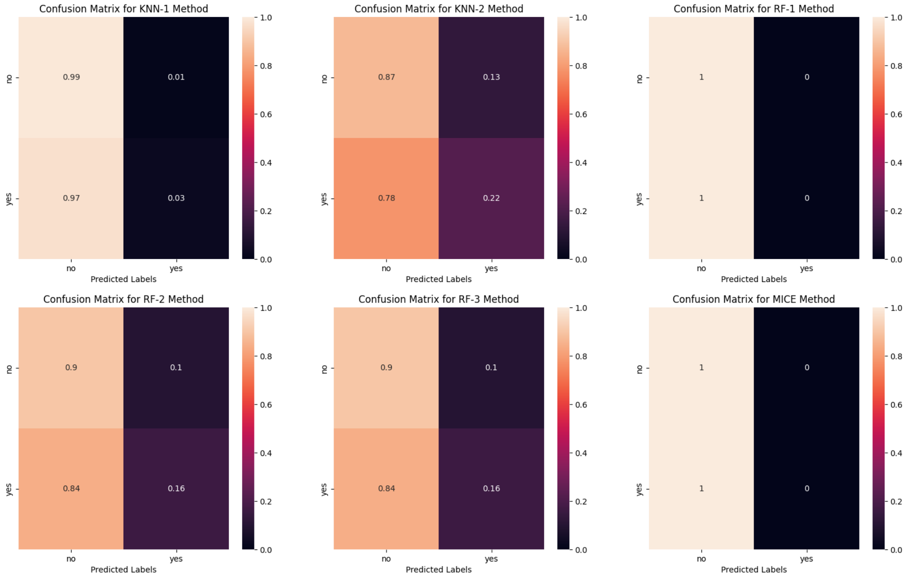

Figure 5: Confusion matrix for all the other single model multiple imputation methods on the `SmokeBan` data set

## C.3 PhDPublications Dataset

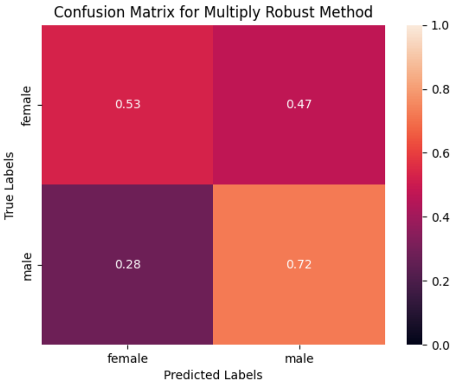

Figure 6: Confusion matrix for `Multiply-Robust-KNN-RF` on the `PhDPublications` data set

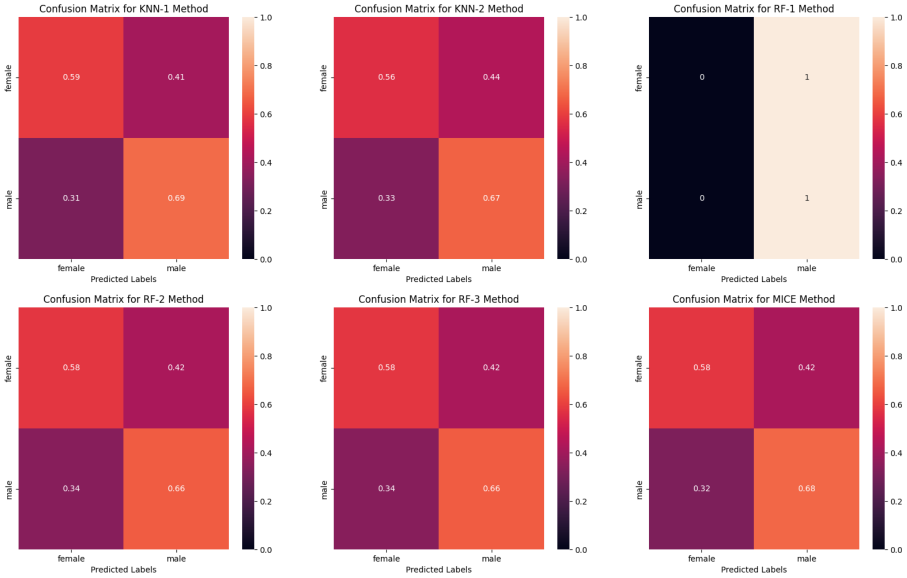

Figure 7: Confusion matrix for all the other single model multiple imputation methods on the `PhDPublications` data set

## C.4 ResumeNames Dataset

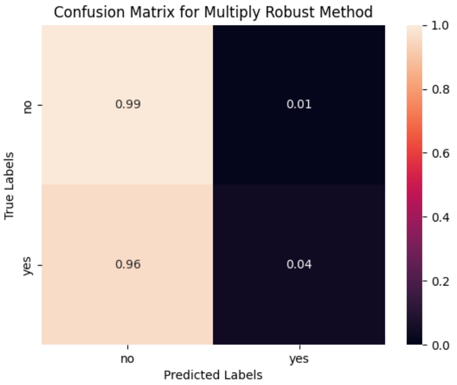

Figure 8: Confusion matrix for `Multiply-Robust-KNN-RF` on the `ResumeNames` data set

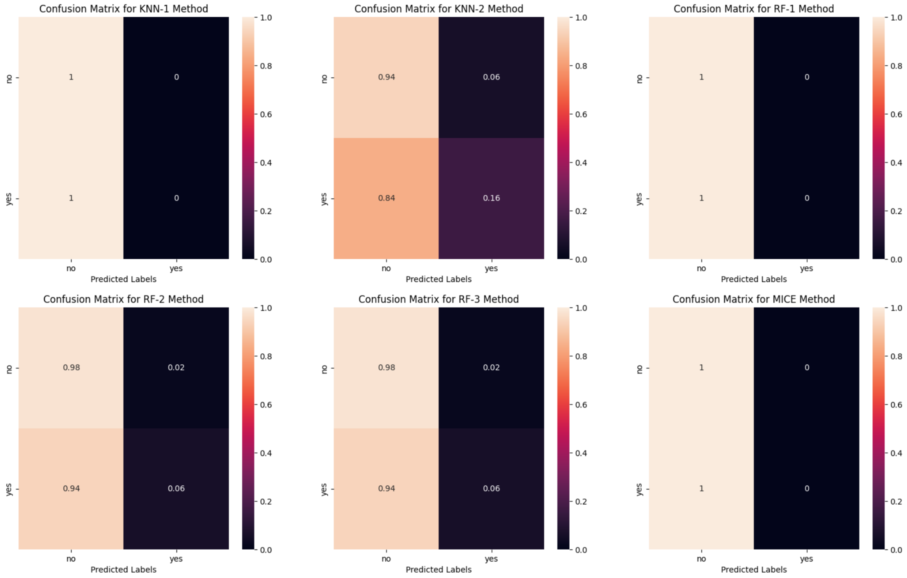

Figure 9: Confusion matrix for all the other single model multiple imputation methods on the `ResumeNames` data set

# References

[1] Reasons for missing data - managing missing data in patient registries - ncbi bookshelf. `https://www.ncbi.nlm.nih.gov/books/NBK493613/`. (Accessed on 04/05/2023).

[2] Patterns of unit and item nonresponse in the cahps® hospital survey - pmc. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1361246/#:~:text=There%20are%20two%20basic%20types,respondent%20is%20eligible%20to%20answer`. (Accessed on 04/05/2023).

[3] Douglas G Altman and J Martin Bland. Missing data. *Bmj*, 334(7590):424–424, 2007.

[4] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[5] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402–406, 2013.

[6] Therese D Pigott. A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383, 2001.

[7] Mirjam J Knol, Kristel JM Janssen, A Rogier T Donders, Antoine CG Egberts, E Rob Heerdink, Diederick E Grobbee, Karel GM Moons, and Mirjam I Geerlings. Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *Journal of clinical epidemiology*, 63(7):728–736, 2010.

[8] Edith D De Leeuw, Joop J Hox, Mark Huisman, et al. Prevention and treatment of item nonresponse. *Journal of Official Statistics*, 19:153–176, 2003.

[9] Mavuto Mukaka, Sarah A White, Dianne J Terlouw, Victor Mwapasa, Linda Kalilani-Phiri, and E Brian Faragher. Is using multiple imputation better than complete case analysis for estimating a prevalence (risk) difference in randomized controlled trials when binary outcome observations are missing? *Trials*, 17(1):1–12, 2016.

[10] Michael P Jones. Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American statistical association*, 91(433):222–230, 1996.

[11] Muhan Zhou, Yulei He, Mandi Yu, and Chiu-Hsieh Hsu. A nonparametric multiple imputation approach for missing categorical data. *BMC medical research methodology*, 17(1):1–12, 2017.

[12] Wei-Chao Lin and Chih-Fong Tsai. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53:1487–1509, 2020.

[13] Zhongheng Zhang. Missing data imputation: focusing on single imputation. *Annals of translational medicine*, 4(1), 2016.

[14] Carol M Musil, Camille B Warner, Piyanee Klainin Yobas, and Susan L Jones. A comparison of imputation techniques for handling missing data. *Western journal of nursing research*, 24(7):815–829, 2002.

[15] Ting Hsiang Lin. A comparison of multiple imputation with em algorithm and mcmc method for quality of life missing data. *Quality & quantity*, 44:277–287, 2010.

[16] Multiple imputation: A flexible tool for handling missing data - pmc. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4638176/#:~:text=Single%2Dvalue%20imputation%20methods%20include,replacing%20missing%20values%20many%20times. (Accessed on 05/23/2023).

[17] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: an optimization approach. *J. Mach. Learn. Res.*, 18(1):7133–7171, 2017.

[18] Sixia Chen and David Haziza. Multiply robust imputation procedures for the treatment of item nonresponse in surveys. *Biometrika*, 104(2):439–453, 2017.

[19] Model misspecification - an overview | sciencedirect topics. https://www.sciencedirect.com/topics/mathematics/model-misspecification. (Accessed on 05/23/2023).

[20] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[21] Yiran Dong and Chao-Ying Joanne Peng. Principled missing data methods for researchers. *SpringerPlus*, 2:1–17, 2013.

[22] Andrew W Lo, Kien Wei Siah, and Chi Heem Wong. *Machine learning with statistical imputation for predicting drug approvals*, volume 60. SSRN, 2019.

[23] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):197–208, 2016.

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

[25] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

[26] sklearn.impute.knnimputer — scikit-learn 1.2.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html#sklearn.impute.KNNImputer. (Accessed on 06/15/2023).

[27] missingpy · pypi. https://pypi.org/project/missingpy/. (Accessed on 05/23/2023).

[28] sklearn.impute.iterativeimputer — scikit-learn 1.2.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html. (Accessed on 06/15/2023).