

# Data Imputation methods

Amitsingh Pardeshi

November 16, 2017

## 1 Repository

Please find the link to the git repository- [Project Link](#)

## 2 Data Imputation

Missing data can be a not so trivial problem when analysing a dataset and accounting for it is usually not so straightforward either.

If the amount of missing data is relatively small compared to the size of the dataset then leaving few sample features could be the best strategy. However, if the amount of missing data increases then we can not afford to ignore the record showing missing values.

## 3 Few Strategies/methods to impute missing values

- Mean data Imputation: This is one of the simple method to impute missing values where the NA(missing values ) will be replaced by the mean value of the given feature.
- KNN Imputation: KNN defines for each sample or individual a set of K-nearest neighbors and then replaces the missing data for a given variable by averaging (non-missing) values of its neighbors.
- Predictive mean matching: Predictive mean matching (PMM) is an attractive way to do multiple imputation for missing data, especially for imputing quantitative variables that are not normally distributed. Suppose we have  $X_1, X_2, \dots, X_k$  variables. If  $X_1$  has missing values, then it will be regressed on other variables  $X_2$  to  $X_k$ . The missing values in  $X_1$  will be then replaced by predictive values obtained. Similarly, if  $X_2$  has missing values, then  $X_1, X_3$  to  $X_k$  variables will be used in prediction model as independent variables. Later, missing values will be replaced with predicted values.

- Regression Imputation: A regression model is estimated to predict observed values of a variable based on other variables, and that model is then used to impute values in cases where that variable is missing. In other words, available information for complete and incomplete cases is used to predict whether a value on a specific variable is missing or not. Fitted values from the regression model are then used to impute the missing values.

## 4 Experiment

1. Use Iris dataset.
2. Introduce 2%, 5%, 10%, 15%, 20%, 25% missing values into the dataset.
3. Use three imputation methods (mean, knn and pmm ) to impute the missing values.
4. Determine the performance of each method using RMSE and supervised classification error using knn.
5. Compare the results.

## 5 Code Snippet and Generated Output

### 5.1 Mean Data imputation method

```

1 ## load data
2 data <- iris
3 # remove the last categorical column
4 dataWolastCol <- data[, -5]
5
6 #introduce 2% missing values using prodNa
7 iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)
8
9
10 ## mean data imputation
11 # takes dataframe as input and replace NA values with mean of the
    column
12 meanDataImputation <- function(df){
13   i <- 1
14   for(col in names(df)){
15     localVector <- df[, i]
16     localVector[is.na(localVector)] <- mean(localVector[!is.na(
        localVector)])
17     df[, i] <- localVector
18     i <- i + 1;
19   }
20   return(df)
21 }
22 ## call meanDataImputation with iris.mis
23 iris.meanImputed_2 <- meanDataImputation(iris.mis_2)

```

```

24 ## calculate RMSE
25 rmseMean_2 <- rmse(iris.meanImputed_2, iris[, -5], na.rm = TRUE)
26
27 # print RMSE
28 print(rmseMean_2)
29
30 #function used for data nomalization
31 normalize <- function(x){
32   return( (x -min(x))/ (max(x) - min(x)))
33 }
34
35 # normalized imputed data
36 iris.meanImputed_2_n <- as.data.frame(lapply(iris.meanImputed_2,
37   normalize))
38 # extract the categorical data
39 iris_train_target <- iris[, 5]
40
41 # perform KNN on imputed data (k = 13 as total record
42 #are 150 and sqaure root of 150 #is almost 13
43 iris.pred_2 <- knn(train = iris_train_test, test=iris_train_test,
44   cl = iris_train_target, k =13)
45
46 # check the accuracy of knn
47 table(iris_train_target, iris.pred_2)
48
49 # Repeat the above process by introducing 5%, 10%, 15%, 20% and 25%
   missing values

```

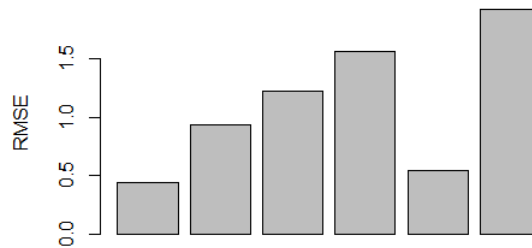


Figure 1: RMSE error distribution from 2% to 25% missing data for mean

Percentage missing data	RSME
2	0.4624256
5	0.8210702
10	1.2937844
15	1.4622056
20	0.4351662
25	1.9010289

## 5.2 Knn imputation method

```

1 ## load data
2 data <- iris
3 # remove the last categorical column
4 dataWolastCol <- data[, -5]
5
6 #introduce 2% missing values using prodNa
7 iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)
8
9
10 ## call knn with iris.mis
11 iris.knnImputed_2 <- knn(iris.mis_2)
12
13 ## take first four columns
14 iris.knnImputed_2 <- iris.knnImputed_2[, 1:4]
15
16 ## calculate RMSE for imputate values with original values
17 rmseKnn_2 <- rmse(iris.meanknnImputed_2, dataWolastCol, na.rm =
    TRUE)
18
19 # print RMSE
20 print(rmseKnn_2)
21 #function used for data nomalization
22 normalize <- function(x){
23   return( (x - min(x)) / (max(x) - min(x)))
24 }
25
26 # normalized imputed data
27 iris.knnImputed_2.n <- as.data.frame(lapply(iris.knnImputed_2,
    normalize))
28 # extract the categorical data
29 iris_train_target <- iris[, 5]
30
31 # perform KNN on imputed data (k = 13 as total record
32 #are 150 and square root of 150 #is almost 13
33 iris.pred_2 <- knn(train = iris_train_test,
34 test=iris_train_test, cl = iris_train_target, k =13)
35
36 # check the accuracy of knn
37 table(iris_train_target, iris.pred_2)
38
39 # Repeat the above process by introducing 5%, 10%, 15%, 20% and 25%
    missing values

```

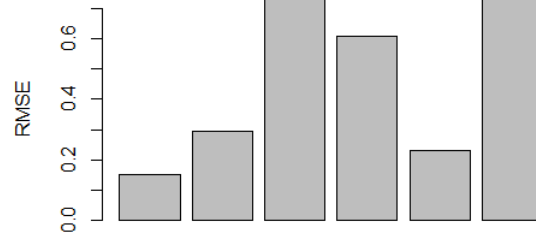


Figure 2: RMSE error distribution from 2% to 25% missing data for KNN

Percentage missing data	RSME
2	0.09576408
5	0.31250259
10	0.41557885
15	0.86278124
20	0.08008516
25	1.21798363

### 5.3 PMM imputation method

```

1 ## load data
2 data <- iris
3 # remove the last categorical column
4 dataWolastCol <- data[, -5]
5
6 #introduce 2% missing values using prodNa
7 iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)
8
9 ## call pmm imputation with iris.mis
10 iris.pmmImpuated_2 <- mice(iris.mis_2, m=5, maxit = 50, method = '
    pmm', seed = 500)
11
12 iris.pmmImpuated_2 <- complete(iris.pmmImpuated_2, 1) # iris .
    pmmImpuated_2$data[, 1:4]
13
14 ## calculate RMSE
15 rmsepmm_2 <- rmse(iris.pmmImpuated_2, dataWolastCol, na.rm = TRUE)
16
17 #print RMSE
18
19 print(rmsepmm_2)
20
21
22 #function used for data nomalization

```

```

23 normalize <- function(x){
24   return( (x -min(x))/ (max(x) - min(x)))
25 }
26
27 # normalized imputed data
28 iris.knnImputed_2.n <- as.data.frame(lapply(iris.knnImputed_2,
29   normalize))
29 # extract the categorical data
30 iris_train_target <- iris[, 5]
31
32 # perform KNN on imputed data (k = 13 as total record
33 #are 150 and square root of 150 #is almost 13
34 iris.pred_2 <- knn(train = iris_train_test,
35 test=iris_train_test, cl = iris_train_target, k =13)
36
37 # check the accuracy of knn
38 table(iris_train_target, iris.pred_2)
39
40 # Repeat the above process by introducing 5%, 10%, 15%, 20% and 25%
   missing values

```

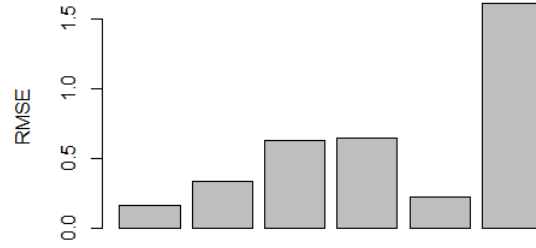


Figure 3: RMSE error distribution from 2% to 25% missing data for PMM

Percentage missing data	RSME
2	0.1545490
5	0.3702967
10	0.4734299
15	0.7294551
20	0.2470372
25	1.4299442

## 6 Few Observations

- Across all the above methods, as the percentage of missing values increases in the given dataset, the RMSE error increases which indicates that the

level of accuracy decreases.

- Knn data imputation performs better than predictive mean matching imputation method.
- Mean imputation is on the bottom side as far as accuracy of imputed values are concerned.

## 7 Scheme for not random missing values

- Often the missing data is classified into two major categories, MAR(Missing at Random) and second is MNAR (Missing not at random).
- MNAR case arises when there is a specific pattern for random data which is nothing but probability of missing values in specific features are greater than others.
- For example, if you are collecting personal information for a group of population. Then some people might not be comfortable disclosing their salary and age.
- In the below section, I am also considering the same where I am considering Sepal.Length equivalent to salary in peoples dataset and Petal.Length is equivalent to age in peoples dataset.
- Because of the above assumptions I introduced missing values in only Sepal.Length and Petal.Length columns and re ran the above experiment.

### 7.1 Mean dataimputation MNAR method

```
1 ## install packages
2 install.packages("VIM")
3 install.packages("mice")
4 install.packages("missForest")
5 install.packages("imputeR")
6 install.packages("hydroGOF")
7 install.packages("gmodels")
8 install.packages("class")
9
10 ## import libraries
11 library(mice)
12 library(missForest)
13 library(VIM)
14 library(imputeR)
15 library(hydroGOF)
16 library(gmodels)
17 library(class)
18
19 path <- ".."
20 show(path)
21
```

```

22 #load data
23 data <- iris
24
25 # get summary
26 summary(data)
27
28 dataWolastCol_1 <- as.data.frame(data[,1])
29 dataWolastCol_3 <- as.data.frame(data[,3])
30 ##create missing data 2,5,10, 15,20,25
31 iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
32 iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)
33
34 iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
35 iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)
36
37 iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
38 iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)
39
40 iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
41 iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)
42
43 iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
44 iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)
45
46 iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
47 iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)
48
49 iris.mis_2 <- cbind(iris.mis_2_0,data[,2],iris.mis_2_1, data[,4])
50 iris.mis_5 <- cbind(iris.mis_5_0,data[,2],iris.mis_5_1, data[,4])
51 iris.mis_10 <- cbind(iris.mis_10_0,data[,2],iris.mis_10_1, data
52   [,4])
53 iris.mis_15 <- cbind(iris.mis_15_0,data[,2],iris.mis_15_1, data
54   [,4])
55 iris.mis_20 <- cbind(iris.mis_20_0,data[,2],iris.mis_20_1, data
56   [,4])
57 iris.mis_25 <- cbind(iris.mis_25_0,data[,2],iris.mis_25_1, data
58   [,4])
59
60 ## mean data impuation
61 # takes dataframe as input and replace NA values with mean of the
62   column
63 meanDataImpuation <- function(df){
64   i <- 1
65   for(col in names(df)){
66     # meanVal <- mean(df[,0], na.rm = TRUE)
67     print(i)
68     print(df[, i])
69     localVector <- df[,i]
70     localVector[is.na(localVector)]<- mean(localVector[!is.na(
71       localVector)])
72     df[,i] <- localVector
73     i <- i + 1;
74   }
75   print(df)

```



```

73 }
74 ## call meanDataImputation with iris.mis
75 iris.meanImputed_2 <- meanDataImputation(iris.mis_2)
76 iris.meanImputed_5 <- meanDataImputation(iris.mis_5)
77 iris.meanImputed_10 <- meanDataImputation(iris.mis_10)
78 iris.meanImputed_15 <- meanDataImputation(iris.mis_15)
79 iris.meanImputed_20 <- meanDataImputation(iris.mis_20)
80 iris.meanImputed_25 <- meanDataImputation(iris.mis_25)
81
82 ## calculate RMSE
83 rmseMean_2 <- rmse(iris.meanImputed_2, iris[, -5], na.rm = TRUE)
84 rmseMean_5 <- rmse(iris.meanImputed_5, iris[, -5], na.rm = TRUE)
85 rmseMean_10 <- rmse(iris.meanImputed_10, iris[, -5], na.rm = TRUE)
86 rmseMean_15 <- rmse(iris.meanImputed_15, iris[, -5], na.rm = TRUE)
87 rmseMean_20 <- rmse(iris.meanImputed_20, iris[, -5], na.rm = TRUE)
88 rmseMean_25 <- rmse(iris.meanImputed_25, iris[, -5], na.rm = TRUE)
89
90 print(rmseMean_2)
91 print(rmseMean_5)
92 print(rmseMean_10)
93 print(rmseMean_15)
94 print(rmseMean_20)
95 print(rmseMean_25)
96 rmseVecpt <- c(rmseMean_2, rmseMean_5, rmseMean_10,
97 rmseMean_15, rmseMean_20, rmseMean_25)
98
99 ##plot RMSE
100 barplot(rmseVecpt)
101
102
103 ##plot RMSE
104 barplot(rmseMean_2, ylab = "RMSE", main="RMSE error distribution
    for 2% missing data")
105 barplot(rmseMean_5, ylab = "RMSE", main="RMSE error distribution
    for 5% missing data")
106 barplot(rmseMean_10, ylab = "RMSE", main="RMSE error distribution
    for 10% missing data")
107 barplot(rmseMean_15, ylab = "RMSE", main="RMSE error distribution
    for 15% missing data")
108 barplot(rmseMean_20, ylab = "RMSE", main="RMSE error distribution
    for 20% missing data")
109 barplot(rmseMean_25, ylab = "RMSE", main="RMSE error distribution
    for 25% missing data")
110
111
112
113 total_RMSE_2 <- rmseMean_2[1] + rmseMean_2[2]
114 + rmseMean_2[3] + rmseMean_2[4]
115 total_RMSE_5 <- rmseMean_5[1] + rmseMean_5[2]
116 + rmseMean_5[3] + rmseMean_5[4]
117 total_RMSE_10 <- rmseMean_10[1] + rmseMean_10[2]
118 + rmseMean_10[3] + rmseMean_10[4]
119 total_RMSE_15 <- rmseMean_15[1] + rmseMean_15[2]
120 + rmseMean_15[3] + rmseMean_15[4]
121 total_RMSE_20 <- rmseMean_20[1] + rmseMean_20[2]
122 + rmseMean_20[3] + rmseMean_20[4]
123 total_RMSE_25 <- rmseMean_25[1] + rmseMean_25[2]

```

```

124 + rmseMean_25[3] + rmseMean_25[4]
125
126 total_rmse <- c(total_RMSE_2, total_RMSE_5, total_RMSE_10,
127 total_RMSE_15, total_RMSE_20, total_RMSE_25)
128
129 per_col <- c(2, 5, 10, 15, 20, 25)
130
131 rmse_df <- data.frame(percentage = per_col, error = total_rmse);
132 barplot(rmse_df$error, ylab = "RMSE",
133 main="RMSE error distribution from 2% to 25% missing data")
134
135
136 ## calculate classification error
137 lebal <- iris[, 5]
138
139
140 #set.seed(9850)
141
142 #gp <- runif(nrow(iris))
143
144 #iris_r <- iris[order(gp)]
145 iris_r <- iris_r[, c(1, 2, 3, 4)]
146 iris_c_2 <- rbind(iris_r, iris.meanImputed_2)
147 normalize <- function(x){
148   return( (x - min(x)) / (max(x) - min(x)))
149 }
150
151 iris_n <- as.data.frame(lapply(iris_c_2, normalize))
152
153 iris_train <- iris_n[1:150,]
154 iris_train_test <- iris_n[151:300,]
155 iris_train_target <- iris[, 5]
156
157 iris.meanImputed_5_n <- as.data.frame(lapply(iris.meanImputed_5,
158   normalize))
159 iris.meanImputed_10_n <- as.data.frame(lapply(iris.meanImputed_
160   10, normalize))
161 iris.meanImputed_15_n <- as.data.frame(lapply(iris.meanImputed_
162   15, normalize))
163 iris.meanImputed_20_n <- as.data.frame(lapply(iris.meanImputed_
164   20, normalize))
165 iris.meanImputed_25_n <- as.data.frame(lapply(iris.meanImputed_
166   25, normalize))
167 iris.meanImputed_2_n <- as.data.frame(lapply(iris.meanImputed_2,
168   normalize))
169
170
171 iris.pred_2 <- knn(train = iris_train_test,
172 test=iris_train_test, cl = iris_train_target, k = 20)
173 iris.pred_5 <- knn(train = iris.meanImputed_5_n,
174 test=iris.meanImputed_5_n, cl = iris_train_target, k = 20)
175 iris.pred_10 <- knn(train = iris.meanImputed_10_n,
176 test=iris.meanImputed_10_n, cl = iris_train_target, k = 20)
177 iris.pred_15 <- knn(train = iris.meanImputed_15_n,
178 test=iris.meanImputed_15_n, cl = iris_train_target, k = 20)
179 iris.pred_20 <- knn(train = iris.meanImputed_20_n,
180 test=iris.meanImputed_20_n, cl = iris_train_target, k = 20)

```

```

175 iris.pred_25 <- knn(train = iris.meanImputed_25_n,
176 test=iris.meanImputed_25_n, cl = iris_train_target, k =20)
177 iris.pred <- knn(train = iris_train, test=iris_train,
178 cl = iris_train_target, k =20)
179
180 table(iris_train_target, iris.pred_2)
181 table(iris_train_target, iris.pred_5)
182 table(iris_train_target, iris.pred_10)
183 table(iris_train_target, iris.pred_15)
184 table(iris_train_target, iris.pred_20)
185 table(iris_train_target, iris.pred_25)

```

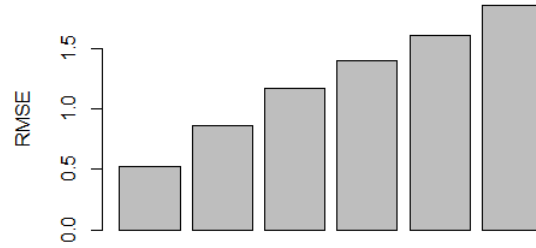


Figure 4: RMSE error distribution from 2% to 25% missing data for mean MNAR

Percentage missing data	RSME
2	0.6125603
5	0.6932605
10	1.2239339
15	1.3436273
20	1.5709511
25	1.8317603

## 7.2 KNN dataimputation MNAR method

```

1 ## install packages
2 install.packages("VIM")
3 install.packages("mice")
4 install.packages("missForest")
5 install.packages("imputeR")
6 install.packages("hydroGOF")
7
8 ## import libraries
9 library(mice)
10 library(missForest)
11 library(VIM)

```

```

12 library(imputeR)
13 library(hydroGOF)
14 library(class)
15 path <- ".."
16 show(path)
17
18 #load data
19 data <- iris
20
21 # get summary
22 summary(data)
23
24 dataWolastCol <- data[, -5]
25 ##create missing data 2,5,10, 15,20,25
26
27
28 dataWolastCol_1 <- as.data.frame(data[, 1])
29 dataWolastCol_3 <- as.data.frame(data[, 3])
30 ##create missing data 2,5,10, 15,20,25
31 iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
32 iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)
33
34 iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
35 iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)
36
37 iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
38 iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)
39
40 iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
41 iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)
42
43 iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
44 iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)
45
46 iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
47 iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)
48
49 iris.mis_2 <- cbind(iris.mis_2_0, data[, 2], iris.mis_2_1, data[, 4])
50 iris.mis_5 <- cbind(iris.mis_5_0, data[, 2], iris.mis_5_1, data[, 4])
51 iris.mis_10 <- cbind(iris.mis_10_0, data[, 2], iris.mis_10_1, data
52   [, 4])
53 iris.mis_15 <- cbind(iris.mis_15_0, data[, 2], iris.mis_15_1, data
54   [, 4])
55 iris.mis_20 <- cbind(iris.mis_20_0, data[, 2], iris.mis_20_1, data
56   [, 4])
57 iris.mis_25 <- cbind(iris.mis_25_0, data[, 2], iris.mis_25_1, data
58   [, 4])
59
60 ## call meanDataImputation with iris.mis
61 iris.meanImputed_2 <- kNN(iris.mis_2)
62 iris.meanImputed_5 <- kNN(iris.mis_5)
63 iris.meanImputed_10 <- kNN(iris.mis_10)
64 iris.meanImputed_15 <- kNN(iris.mis_15)
65 iris.meanImputed_20 <- kNN(iris.mis_20)
66 iris.meanImputed_25 <- kNN(iris.mis_25)

```

```

65 iris.meanImputed_2 <- iris.meanImputed_2[,1:4]
66 iris.meanImputed_5 <- iris.meanImputed_5[,1:4]
67 iris.meanImputed_10 <- iris.meanImputed_10[,1:4]
68 iris.meanImputed_15 <- iris.meanImputed_15[,1:4]
69 iris.meanImputed_20 <- iris.meanImputed_20[,1:4]
70 iris.meanImputed_25 <- iris.meanImputed_25[,1:4]
71
72 rmseMean_2 <- rmse(iris.meanImputed_2,
73 dataWolastCol, na.rm = TRUE)
74 rmseMean_5 <- rmse(iris.meanImputed_5,
75 dataWolastCol, na.rm = TRUE)
76 rmseMean_10 <- rmse(iris.meanImputed_10,
77 dataWolastCol, na.rm = TRUE)
78 rmseMean_15 <- rmse(iris.meanImputed_15,
79 dataWolastCol, na.rm = TRUE)
80 rmseMean_20 <- rmse(iris.meanImputed_20,
81 dataWolastCol, na.rm = TRUE)
82 rmseMean_25 <- rmse(iris.meanImputed_25,
83 dataWolastCol, na.rm = TRUE)
84
85 print(rmseMean_2)
86 print(rmseMean_5)
87 print(rmseMean_10)
88 print(rmseMean_15)
89 print(rmseMean_20)
90 print(rmseMean_25)
91 rmseVecpt <- c(rmseMean_2, rmseMean_5, rmseMean_10,
92 rmseMean_15, rmseMean_20, rmseMean_25)
93
94 barplot(rmseVecpt)
95
96
97 ##plot RMSE
98 barplot(rmseMean_2, ylab = "RMSE", main="RMSE error distribution
   for 2% missing data")
99 barplot(rmseMean_5, ylab = "RMSE", main="RMSE error distribution
   for 5% missing data")
100 barplot(rmseMean_10, ylab = "RMSE", main="RMSE error distribution
   for 10% missing data")
101 barplot(rmseMean_15, ylab = "RMSE", main="RMSE error distribution
   for 15% missing data")
102 barplot(rmseMean_20, ylab = "RMSE", main="RMSE error distribution
   for 20% missing data")
103 barplot(rmseMean_25, ylab = "RMSE", main="RMSE error distribution
   for 25% missing data")
104
105
106
107 total_RMSE_2 <- rmseMean_2[1] + rmseMean_2[2] + rmseMean_2[3] +
   rmseMean_2[4]
108 total_RMSE_5 <- rmseMean_5[1] + rmseMean_5[2] + rmseMean_5[3] +
   rmseMean_5[4]
109 total_RMSE_10 <- rmseMean_10[1] + rmseMean_10[2] + rmseMean_10[3] +
   rmseMean_10[4]
110 total_RMSE_15 <- rmseMean_15[1] + rmseMean_15[2] + rmseMean_15[3] +
   rmseMean_15[4]
111 total_RMSE_20 <- rmseMean_20[1] + rmseMean_20[2] + rmseMean_20[3] +

```

```

rmseMean_20[4]
112 total_RMSE_25 <- rmseMean_25[1] + rmseMean_25[2] + rmseMean_25[3] +
rmseMean_25[4]
113
114 total_rmse <- c(total_RMSE_2, total_RMSE_5, total_RMSE_10,
115 total_RMSE_15, total_RMSE_20, total_RMSE_25)
116
117 per_col <- c(2, 5, 10, 15, 20, 25)
118
119 rmse_df <- data.frame(percentage = per_col, error = total_rmse);
120 barplot(rmse_df$error, ylab = "RMSE",
121 main="RMSE error distribution from 2% to 25% missing data")
122
123
124
125 ## calculate classification error
126 lebal <- iris[, 5]
127
128
129 #set.seed(9850)
130
131 #gp <- runif(nrow(iris))
132
133 #iris_r <- iris[order(gp)]
134 iris_r <- iris_r[, c(1, 2, 3, 4)]
135 iris_c_2 <- rbind(iris_r, iris.meanImputed_2)
136 normalize <- function(x){
137   return( (x - min(x)) / (max(x) - min(x)))
138 }
139
140 iris_n <- as.data.frame(lapply(iris_c_2, normalize))
141
142 iris_train <- iris_n[1:150,]
143 iris_train_test <- iris_n[151:300,]
144 iris_train_target <- iris[, 5]
145
146 iris.meanImputed_5_n <- as.data.frame(lapply(iris.meanImputed_5,
normalize))
147 iris.meanImputed_10_n <- as.data.frame(lapply(iris.meanImputed_
10, normalize))
148 iris.meanImputed_15_n <- as.data.frame(lapply(iris.meanImputed_
15, normalize))
149 iris.meanImputed_20_n <- as.data.frame(lapply(iris.meanImputed_
20, normalize))
150 iris.meanImputed_25_n <- as.data.frame(lapply(iris.meanImputed_
25, normalize))
151 iris.meanImputed_2_n <- as.data.frame(lapply(iris.meanImputed_2,
normalize))
152
153
154 iris.pred_2 <- knn(train = iris_train_test,
155 test=iris_train_test, cl = iris_train_target, k = 20)
156 iris.pred_5 <- knn(train = iris.meanImputed_5_n,
157 test=iris.meanImputed_5_n, cl = iris_train_target, k = 20)
158 iris.pred_10 <- knn(train = iris.meanImputed_10_n,
159 test=iris.meanImputed_10_n, cl = iris_train_target, k = 20)
160 iris.pred_15 <- knn(train = iris.meanImputed_15_n,

```

```

161 test=iris.meanImputed_15_n, cl = iris_train_target, k =20)
162 iris.pred_20 <- knn(train = iris.meanImputed_20_n,
163 test=iris.meanImputed_20_n, cl = iris_train_target, k =20)
164 iris.pred_25 <- knn(train = iris.meanImputed_25_n,
165 test=iris.meanImputed_25_n, cl = iris_train_target, k =20)
166 iris.pred <- knn(train = iris_train, test=iris_train,
167 cl = iris_train_target, k =20)
168
169 table(iris_train_target, iris.pred_2)
170 table(iris_train_target, iris.pred_5)
171 table(iris_train_target, iris.pred_10)
172 table(iris_train_target, iris.pred_15)
173 table(iris_train_target, iris.pred_20)
174 table(iris_train_target, iris.pred_25)

```

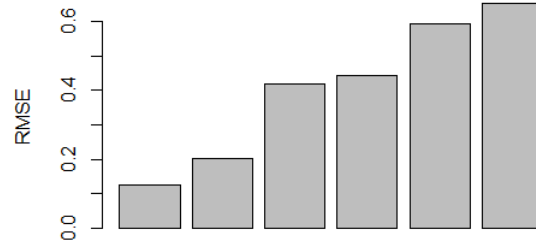


Figure 5: RMSE error distribution from 2% to 25% missing data for KNN MNAR

Percentage missing data	RSME
2	0.1124197
5	0.2694590
10	0.2868216
15	0.5059823
20	0.4941634
25	0.6307102

### 7.3 PMM dataimputation MNAR method

```

1 ## install packages
2 install.packages("VIM")
3 install.packages("mice")
4 install.packages("missForest")
5 install.packages("imputeR")
6 install.packages("hydroGOF")
7
8 ## import libraries

```

```

9 library(mice)
10 library(missForest)
11 library(VIM)
12 library(imputeR)
13 library(hydroGOF)
14 library(class)
15 path <- ".."
16 show(path)
17
18 #load data
19 data <- iris
20
21 # get summary
22 summary(data)
23
24 dataWolastCol <- data[, -5]
25 ##create missing data 2%,5%,10%, 15%,20%,25%
26 dataWolastCol_1 <- as.data.frame(data[, 1])
27 dataWolastCol_3 <- as.data.frame(data[, 3])
28 ##create missing data 2,5,10, 15,20,25
29 iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
30 iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)
31
32 iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
33 iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)
34
35 iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
36 iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)
37
38 iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
39 iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)
40
41 iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
42 iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)
43
44 iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
45 iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)
46
47
48 iris.mis_2 <- cbind(iris.mis_2_0, data[, 2], iris.mis_2_1, data[, 4])
49 iris.mis_5 <- cbind(iris.mis_5_0, data[, 2], iris.mis_5_1, data[, 4])
50 iris.mis_10 <- cbind(iris.mis_10_0, data[, 2], iris.mis_10_1, data
51   [, 4])
52 iris.mis_15 <- cbind(iris.mis_15_0, data[, 2], iris.mis_15_1, data
53   [, 4])
54 iris.mis_20 <- cbind(iris.mis_20_0, data[, 2], iris.mis_20_1, data
55   [, 4])
56 iris.mis_25 <- cbind(iris.mis_25_0, data[, 2], iris.mis_25_1, data
57   [, 4])
58
59 ## call pmm imputation with iris.mis
60 iris.pmmImputed_2 <- mice(iris.mis_2, m=5, maxit = 50, method = '
61   pmm', seed = 500)
62 iris.pmmImputed_5 <- mice(iris.mis_5, m=5, maxit = 50, method = '
63   pmm', seed = 500)
64 iris.pmmImputed_10 <- mice(iris.mis_10, m=5, maxit = 50, method =

```



```

      'pmm', seed = 500)
60 iris.pmmImputed_15<- mice(iris.mis_15, m=5, maxit = 50, method = '
      pmm', seed = 500)
61 iris.pmmImputed_20<- mice(iris.mis_20, m=5, maxit = 50, method = '
      pmm', seed = 500)
62 iris.pmmImputed_25 <- mice(iris.mis_25, m=5, maxit = 50, method =
      'pmm', seed = 500)
63
64 iris.pmmImputed_2 <- complete(iris.pmmImputed_2,1) # iris .
      pmmImputed_2$data[,1:4]
65 iris.pmmImputed_5 <- complete(iris.pmmImputed_5,1)#iris .
      pmmImputed_5$data[,1:4]
66 iris.pmmImputed_10 <- complete(iris.pmmImputed_10,1)#iris .
      pmmImputed_10$data[,1:4]
67 iris.pmmImputed_15<- complete(iris.pmmImputed_15,1)#iris .
      pmmImputed_15$data[,1:4]
68 iris.pmmImputed_20<- complete(iris.pmmImputed_20,1)#iris .
      pmmImputed_20$data[,1:4]
69 iris.pmmImputed_25 <- complete(iris.pmmImputed_25,1)#iris .
      pmmImputed_25$data[,1:4]
70
71 rmsepmm_2 <- rmse(iris.pmmImputed_2, dataWolastCol, na.rm = TRUE)
72 rmsepmm_5 <- rmse(iris.pmmImputed_5, dataWolastCol, na.rm = TRUE)
73 rmsepmm_10 <- rmse(iris.pmmImputed_10, dataWolastCol, na.rm = TRUE
      )
74 rmsepmm_15<- rmse(iris.pmmImputed_15,dataWolastCol ,na.rm = TRUE)
75 rmsepmm_20 <- rmse(iris.pmmImputed_20, dataWolastCol, na.rm = TRUE
      )
76 rmsepmm_25 <- rmse(iris.pmmImputed_25, dataWolastCol, na.rm = TRUE
      )
77
78 print(rmsepmm_2)
79 print(rmsepmm_5)
80 print(rmsepmm_10)
81 print(rmsepmm_15)
82 print(rmsepmm_20)
83 print(rmsepmm_25)
84 rmseVecpt <- c(rmsepmm_2, rmsepmm_5, rmsepmm_10,
85 rmsepmm_15, rmsepmm_20, rmsepmm_25)
86
87 barplot(rmseVecpt)
88
89
90
91 ##plot RMSE
92 barplot(rmsepmm_2, ylab = "RMSE", main="RMSE error distribution for
      2% missing data")
93 barplot(rmsepmm_5, ylab = "RMSE", main="RMSE error distribution
      for 5% missing data")
94 barplot(rmsepmm_10, ylab = "RMSE", main="RMSE error distribution
      for 10% missing data")
95 barplot(rmsepmm_15, ylab = "RMSE", main="RMSE error distribution
      for 15% missing data")
96 barplot(rmsepmm_20, ylab = "RMSE", main="RMSE error distribution
      for 20% missing data")
97 barplot(rmsepmm_25, ylab = "RMSE", main="RMSE error distribution
      for 25% missing data")

```

```

98
99
100
101 total_RMSE_2 <- rmsepmm_2[1] + rmsepmm_2[2] + rmsepmm_2[3] +
    rmsepmm_2[4]
102 total_RMSE_5 <- rmsepmm_5[1] + rmsepmm_5[2] + rmsepmm_5[3] +
    rmsepmm_5[4]
103 total_RMSE_10 <- rmsepmm_10[1] + rmsepmm_10[2] + rmsepmm_10[3] +
    rmsepmm_10[4]
104 total_RMSE_15 <- rmsepmm_15[1] + rmsepmm_15[2] + rmsepmm_15[3] +
    rmsepmm_15[4]
105 total_RMSE_20 <- rmsepmm_20[1] + rmsepmm_20[2] + rmsepmm_20[3] +
    rmsepmm_20[4]
106 total_RMSE_25 <- rmsepmm_25[1] + rmsepmm_25[2] + rmsepmm_25[3] +
    rmsepmm_25[4]
107
108 total_rmse <- c(total_RMSE_2, total_RMSE_5, total_RMSE_10,
109 total_RMSE_15, total_RMSE_20, total_RMSE_25)
110
111 per_col <- c(2, 5, 10, 15, 20, 25)
112
113 rmse_df <- data.frame(percentage = per_col, error = total_rmse);
114 barplot(rmse_df$error, ylab = "RMSE",
115 main = "RMSE error distribution from 2% to 25% missing data")
116
117
118
119 ## calculate classification error
120 lebal <- iris[, 5]
121
122
123 #set.seed(9850)
124
125 #gp <- runif(nrow(iris))
126
127 #iris_r <- iris[order(gp)]
128 iris_r <- iris_r[, c(1, 2, 3, 4)]
129 iris_c_2 <- rbind(iris_r, iris.pmmImputed_2)
130 normalize <- function(x){
131   return( (x - min(x)) / (max(x) - min(x)))
132 }
133
134 iris_n <- as.data.frame(lapply(iris_c_2, normalize))
135
136 iris_train <- iris_n[1:150,]
137 iris_train_test <- iris_n[151:300,]
138 iris_train_target <- iris[, 5]
139
140 iris.pmmImputed_5_n <- as.data.frame(lapply(iris.pmmImputed_5,
    normalize))
141 iris.pmmImputed_10_n <- as.data.frame(lapply(iris.pmmImputed_10,
    normalize))
142 iris.pmmImputed_15_n <- as.data.frame(lapply(iris.pmmImputed_15,
    normalize))
143 iris.pmmImputed_20_n <- as.data.frame(lapply(iris.pmmImputed_20,
    normalize))
144 iris.pmmImputed_25_n <- as.data.frame(lapply(iris.pmmImputed_25,

```

```

    normalize))
145 iris.pmmImputed_2_n <- as.data.frame(lapply(iris.pmmImputed_2,
    normalize))
146
147
148 iris.pred_2 <- knn(train = iris.train.test,
149 test=iris.train.test, cl = iris.train.target, k =20)
150 iris.pred_5 <- knn(train = iris.pmmImputed_5_n,
151 test=iris.pmmImputed_5_n, cl = iris.train.target, k =20)
152 iris.pred_10 <- knn(train = iris.pmmImputed_10_n,
153 test=iris.pmmImputed_10_n, cl = iris.train.target, k =20)
154 iris.pred_15 <- knn(train = iris.pmmImputed_15_n,
155 test=iris.pmmImputed_15_n, cl = iris.train.target, k =20)
156 iris.pred_20 <- knn(train = iris.pmmImputed_20_n,
157 test=iris.pmmImputed_20_n, cl = iris.train.target, k =20)
158 iris.pred_25 <- knn(train = iris.pmmImputed_25_n,
159 test=iris.pmmImputed_25_n, cl = iris.train.target, k =20)
160 iris.pred <- knn(train = iris.train,
161 test=iris.train, cl = iris.train.target, k =20)
162
163 table(iris.train.target, iris.pred_2)
164 table(iris.train.target, iris.pred_5)
165 table(iris.train.target, iris.pred_10)
166 table(iris.train.target, iris.pred_15)
167 table(iris.train.target, iris.pred_20)
168 table(iris.train.target, iris.pred_25)

```

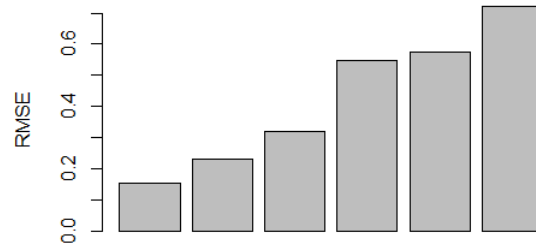


Figure 6: RMSE error distribution from 2% to 25% missing data for PMM MNAR

Percentage missing data	RSME
2	0.1573870
5	0.3257673
10	0.3561408
15	0.5447880
20	0.8192819
25	0.8099624