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 X1, X2.Xk variables. If X1 has missing values, then it will be regressed on other variables X2 to Xk. The missing values in X1 will be then replaced by predictive values obtained. Similarly, if X2 has missing values, then X1, X3 to Xk 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

```
## load data
data <- iris
# remove the last categorical column
dataWolastCol <- data[,-5]

#introduce 2% missing values using prodNa
iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)


## mean data impuation
# takes dataframe as input and replace NA values with mean of the
    column
meanDataImpuation <- function(df){
  i <- 1
  for(col in names(df)){
   localVector <-  df[,i]
   localVector[is.na(localVector)]<- mean(localVector[!is.na(
    localVector)])
   df[,i] <- localVector
    i <- i + 1;
  }
  return(df)
}
## call meanDataImuation with iris.mis
iris.meanImpuated_2 <- meanDataImpuation(iris.mis_2)
```

```r

## calculate RMSE
rmseMean_2 <- rmse(iris.meanImpuated_2, iris[,-5], na.rm = TRUE)

# print RMSE
print(rmseMean_2)

#function used for data nomalization
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

# normalized imputed data
iris.meanImpuated_2_n <- as.data.frame(lapply(iris.meanImpuated_2, normalize))
# extract the categorical data
iris_train_target <- iris[, 5]

# perform KNN on imputed data (k = 13 as total record
#are 150 and sqaure root of 150 #is almost 13
iris.pred_2 <- knn(train = iris_train_test, test=iris_train_test,
cl = iris_train_target, k =13)

# check the accuracy of knn
table(iris_train_target, iris.pred_2)

# Repeat the above process by introducing 5%, 10%, 15%, 20% and 25% missing values
```

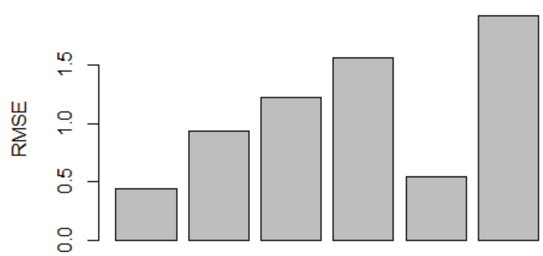

Figure 1: RMSE error distribution from 2% to 25% missing data for mean

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

```
## load data
data <- iris
# remove the last categorical column
dataWolastCol <- data[,-5]

#introduce 2% missing values using prodNa
iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)


## call knn with iris.mis
iris.knnImpuated_2 <- kNN(iris.mis_2)

## take first four columns
iris.knnImpuated_2 <- iris.knnImpuated_2[,1:4]

## calculate RMSE for imputate values with original values
rmseKnn_2 <- rmse(iris.meanknnImpuated_2, dataWolastCol, na.rm =
    TRUE)

# print RMSE
print(rmseKnn_2 )
#function used for data nomalization
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

# normalized imputed data
iris.knnImpuated_2_n <- as.data.frame(lapply(iris.knnImpuated_2,
    normalize))
# extract the categorical data
iris_train_target <- iris[, 5]

# perform KNN on imputed data (k = 13 as total record
#are 150 and square root of 150 #is almost 13
iris.pred_2 <- knn(train = iris_train_test,
test=iris_train_test, cl = iris_train_target, k =13)

# check the accuracy of knn
table(iris_train_target, iris.pred_2)

# Repeat the above process by introducing 5%, 10%, 15%, 20% and 25%
    missing values
```

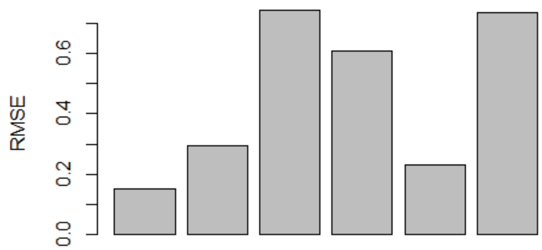

Figure 2: RMSE error distribution from 2% to 25% missing data for KNN

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

```
## load data
data <- iris
# remove the last categorical column
dataWolastCol <- data[,-5]

#introduce 2% missing values using prodNa
iris.mis_2 <- prodNA(dataWolastCol, noNA = 0.02)

## call pmm imputation with iris.mis
iris.pmmImpuated_2 <- mice(iris.mis_2, m=5, maxit = 50, method = '
    pmm', seed = 500)

iris.pmmImpuated_2 <- complete(iris.pmmImpuated_2,1) # iris.
    pmmImpuated_2$data[,1:4]

## calculate RMSE
rmsepmm_2 <- rmse(iris.pmmImpuated_2, dataWolastCol, na.rm = TRUE)

#print RMSE

print(rmsepmm_2)


#function used for data nomalization
```

```r
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

# normalized imputed data
iris.knnImpuated_2_n <- as.data.frame(lapply(iris.knnImpuated_2,
    normalize))
# extract the categorical data
iris_train_target <- iris[, 5]

# perform KNN on imputed data (k = 13 as total record
#are 150 and square root of 150 #is almost 13
iris.pred_2 <- knn(train = iris_train_test,
test=iris_train_test, cl = iris_train_target, k =13)

# check the accuracy of knn
table(iris_train_target, iris.pred_2)

# Repeat the above process by introducing 5%, 10%, 15%, 20% and 25%
     missing values
```

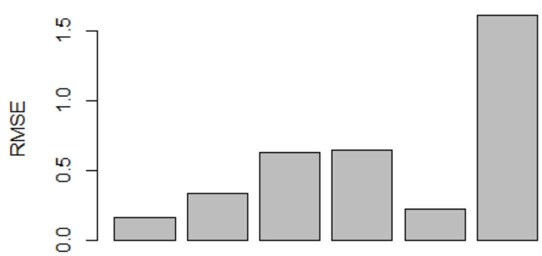

Figure 3: RMSE error distribution from 2% to 25% missing data for PMM

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

```
## install packages
install.packages("VIM")
install.packages("mice")
install.packages("missForest")
install.packages("imputeR")
install.packages("hydroGOF")
install.packages("gmodels")
install.packages("class")

## import libraries
library(mice)
library(missForest)
library(VIM)
library(imputeR)
library(hydroGOF)
library(gmodels)
library(class)

path <- ".."
show(path)

```

```r
#load data
data <- iris

# get summary
summary(data)

dataWolastCol_1 <- as.data.frame(data[,1])
dataWolastCol_3 <- as.data.frame(data[,3])
##create missing data 2,5,10, 15,20,25
iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)

iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)

iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)

iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)

iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)

iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)

iris.mis_2 <- cbind(iris.mis_2_0,data[,2],iris.mis_2_1, data[,4])
iris.mis_5 <- cbind(iris.mis_5_0,data[,2],iris.mis_5_1, data[,4])
iris.mis_10 <- cbind(iris.mis_10_0,data[,2],iris.mis_10_1, data[,4])
iris.mis_15 <- cbind(iris.mis_15_0,data[,2],iris.mis_15_1, data[,4])
iris.mis_20 <- cbind(iris.mis_20_0,data[,2],iris.mis_20_1, data[,4])
iris.mis_25 <- cbind(iris.mis_25_0,data[,2],iris.mis_25_1, data[,4])


## mean data impuation
# takes dataframe as input and replace NA values with mean of the column
meanDataImpuation <- function(df){
  i <- 1
  for(col in names(df)){
    # meanVal <- mean(df[,0], na.rm = TRUE)
    print(i)
    print(df[, i])
    localVector <-  df[,i]
    localVector[is.na(localVector)]<- mean(localVector[!is.na(localVector)])
    df[,i] <- localVector
    i <- i + 1;
  }

  print(df)

```

```r
}
## call meanDataImuation with iris.mis
iris.meanImpuated_2 <- meanDataImpuation(iris.mis_2)
iris.meanImpuated_5 <- meanDataImpuation(iris.mis_5)
iris.meanImpuated_10 <- meanDataImpuation(iris.mis_10)
iris.meanImpuated_15<- meanDataImpuation(iris.mis_15)
iris.meanImpuated_20<- meanDataImpuation(iris.mis_20)
iris.meanImpuated_25 <- meanDataImpuation(iris.mis_25)

## calculate RMSE
rmseMean_2 <- rmse(iris.meanImpuated_2, iris[,-5], na.rm = TRUE)
rmseMean_5 <- rmse(iris.meanImpuated_5, iris[,-5], na.rm = TRUE)
rmseMean_10 <- rmse(iris.meanImpuated_10, iris[,-5], na.rm = TRUE)
rmseMean_15<- rmse(iris.meanImpuated_15,iris[,-5], na.rm = TRUE)
rmseMean_20 <- rmse(iris.meanImpuated_20, iris[,-5], na.rm = TRUE)
rmseMean_25 <- rmse(iris.meanImpuated_25, iris[,-5], na.rm = TRUE)

print(rmseMean_2)
print(rmseMean_5)
print(rmseMean_10)
print(rmseMean_15)
print(rmseMean_20)
print(rmseMean_25)
rmseVecpt <- c(rmseMean_2, rmseMean_5, rmseMean_10,
rmseMean_15, rmseMean_20, rmseMean_25)

##plot RMSE
barplot(rmseVecpt)


##plot RMSE
barplot(rmseMean_2, ylab = "RMSE", main="RMSE error distribution for 2%  missing data")
barplot(rmseMean_5 , ylab = "RMSE", main="RMSE error distribution for 5%  missing data")
barplot(rmseMean_10, ylab = "RMSE", main="RMSE error distribution for 10%  missing data")
barplot(rmseMean_15, ylab = "RMSE", main="RMSE error distribution for 15%  missing data")
barplot(rmseMean_20, ylab = "RMSE", main="RMSE error distribution for 20%  missing data")
barplot(rmseMean_25, ylab = "RMSE", main="RMSE error distribution for 25%  missing data")



total_RMSE_2 <-  rmseMean_2[1] + rmseMean_2[2]
+ rmseMean_2[3] + rmseMean_2[4]
total_RMSE_5 <- rmseMean_5[1] + rmseMean_5[2]
+ rmseMean_5[3] + rmseMean_5[4]
total_RMSE_10 <- rmseMean_10[1] + rmseMean_10[2]
+ rmseMean_10[3] + rmseMean_10[4]
total_RMSE_15 <- rmseMean_15[1] + rmseMean_15[2]
+ rmseMean_15[3] + rmseMean_15[4]
total_RMSE_20 <- rmseMean_20[1] + rmseMean_20[2]
+ rmseMean_20[3] + rmseMean_20[4]
total_RMSE_25 <- rmseMean_25[1] + rmseMean_25[2]
```

```
+ rmseMean_25[3] + rmseMean_25[4]

total_rmse <- c(total_RMSE_2,total_RMSE_5, total_RMSE_10,
total_RMSE_15, total_RMSE_20,total_RMSE_25)

per_col <- c(2,5,10,15,20,25)

rmse_df <- data.frame(percentage = per_col,error = total_rmse);
barplot(rmse_df$error,  ylab = "RMSE",
main="RMSE error distribution from 2% to 25%  missing data")


## calculate classification error
lebal <- iris[,5]


#set.seed(9850)

#gp<- runif(nrow(iris))

#iris_r <- iris[order(gp)]
iris_r <- iris_r[,c(1,2,3,4)]
iris_c_2 <-rbind(iris_r, iris.meanImpuated_2)
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

iris_n <- as.data.frame(lapply(iris_c_2, normalize))

iris_train <- iris_n[1:150,]
iris_train_test <- iris_n[151:300,]
iris_train_target <- iris[, 5]

iris.meanImpuated_5_n <- as.data.frame(lapply(iris.meanImpuated_5,
    normalize))
iris.meanImpuated_10_n <- as.data.frame(lapply(iris.meanImpuated_
    10, normalize))
iris.meanImpuated_15_n <- as.data.frame(lapply(iris.meanImpuated_
    15, normalize))
iris.meanImpuated_20_n <- as.data.frame(lapply(iris.meanImpuated_
    20, normalize))
iris.meanImpuated_25_n <- as.data.frame(lapply(iris.meanImpuated_
    25, normalize))
iris.meanImpuated_2_n <- as.data.frame(lapply(iris.meanImpuated_2,
    normalize))


iris.pred_2 <- knn(train = iris_train_test,
test=iris_train_test, cl = iris_train_target, k =20)
iris.pred_5 <- knn(train = iris.meanImpuated_5_n,
test=iris.meanImpuated_5_n, cl = iris_train_target, k =20)
iris.pred_10 <- knn(train = iris.meanImpuated_10_n,
test=iris.meanImpuated_10_n, cl = iris_train_target, k =20)
iris.pred_15 <- knn(train = iris.meanImpuated_15_n,
test=iris.meanImpuated_15_n, cl = iris_train_target, k =20)
iris.pred_20 <- knn(train = iris.meanImpuated_20_n,
test=iris.meanImpuated_20_n, cl = iris_train_target, k =20)
```

```
iris.pred_25 <- knn(train = iris.meanImpuated_25_n,
test=iris.meanImpuated_25_n, cl = iris_train_target, k =20)
iris.pred <- knn(train = iris_train, test=iris_train,
cl = iris_train_target, k =20)

table(iris_train_target, iris.pred_2)
table(iris_train_target, iris.pred_5)
table(iris_train_target, iris.pred_10)
table(iris_train_target, iris.pred_15)
table(iris_train_target, iris.pred_20)
table(iris_train_target, iris.pred_25)
```

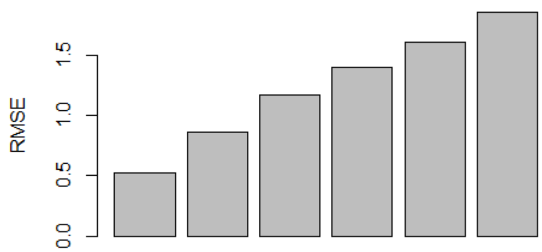

Figure 4: RMSE error distribution from 2% to 25% missing data for mean MNAR

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

```
## install packages
install.packages("VIM")
install.packages("mice")
install.packages("missForest")
install.packages("imputeR")
install.packages("hydroGOF")

## import libraries
library(mice)
library(missForest)
library(VIM)
```

```
library(imputeR)
library(hydroGOF)
library(class)
path <- ".."
show(path)

#load data
data <- iris

# get summary
summary(data)

dataWolastCol <- data[,-5]
##create missing data 2,5,10, 15,20,25


dataWolastCol_1 <- as.data.frame(data[,1])
dataWolastCol_3 <- as.data.frame(data[,3])
##create missing data 2,5,10, 15,20,25
iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)

iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)

iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)

iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)

iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)

iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)

iris.mis_2 <- cbind(iris.mis_2_0,data[,2],iris.mis_2_1, data[,4])
iris.mis_5 <- cbind(iris.mis_5_0,data[,2],iris.mis_5_1, data[,4])
iris.mis_10 <- cbind(iris.mis_10_0,data[,2],iris.mis_10_1, data[,4])
iris.mis_15 <- cbind(iris.mis_15_0,data[,2],iris.mis_15_1, data[,4])
iris.mis_20 <- cbind(iris.mis_20_0,data[,2],iris.mis_20_1, data[,4])
iris.mis_25 <- cbind(iris.mis_25_0,data[,2],iris.mis_25_1, data[,4])


## call meanDataImuation with iris.mis
iris.meanImpuated_2 <- kNN(iris.mis_2)
iris.meanImpuated_5 <- kNN(iris.mis_5)
iris.meanImpuated_10 <- kNN(iris.mis_10)
iris.meanImpuated_15<- kNN(iris.mis_15)
iris.meanImpuated_20<- kNN(iris.mis_20)
iris.meanImpuated_25 <- kNN(iris.mis_25)

```

```
iris.meanImpuated_2 <- iris.meanImpuated_2[,1:4]
iris.meanImpuated_5 <- iris.meanImpuated_5[,1:4]
iris.meanImpuated_10 <- iris.meanImpuated_10[,1:4]
iris.meanImpuated_15<- iris.meanImpuated_15[,1:4]
iris.meanImpuated_20<- iris.meanImpuated_20[,1:4]
iris.meanImpuated_25 <- iris.meanImpuated_25[,1:4]

rmseMean_2 <- rmse(iris.meanImpuated_2,
dataWolastCol, na.rm = TRUE)
rmseMean_5 <- rmse(iris.meanImpuated_5,
dataWolastCol, na.rm = TRUE)
rmseMean_10 <- rmse(iris.meanImpuated_10,
dataWolastCol, na.rm = TRUE)
rmseMean_15<- rmse(iris.meanImpuated_15,
dataWolastCol ,na.rm = TRUE)
rmseMean_20 <- rmse(iris.meanImpuated_20,
dataWolastCol, na.rm = TRUE)
rmseMean_25 <- rmse(iris.meanImpuated_25,
dataWolastCol, na.rm = TRUE)

print(rmseMean_2)
print(rmseMean_5)
print(rmseMean_10)
print(rmseMean_15)
print(rmseMean_20)
print(rmseMean_25)
rmseVecpt <- c(rmseMean_2, rmseMean_5, rmseMean_10,
rmseMean_15, rmseMean_20, rmseMean_25)

barplot(rmseVecpt)


##plot RMSE
barplot(rmseMean_2, ylab = "RMSE", main="RMSE error distribution for 2%  missing data")
barplot(rmseMean_5 , ylab = "RMSE", main="RMSE error distribution for 5%  missing data")
barplot(rmseMean_10, ylab = "RMSE", main="RMSE error distribution for 10%  missing data")
barplot(rmseMean_15, ylab = "RMSE", main="RMSE error distribution for 15%  missing data")
barplot(rmseMean_20, ylab = "RMSE", main="RMSE error distribution for 20%  missing data")
barplot(rmseMean_25, ylab = "RMSE", main="RMSE error distribution for 25%  missing data")



total_RMSE_2 <-  rmseMean_2[1] + rmseMean_2[2] + rmseMean_2[3] + rmseMean_2[4]
total_RMSE_5 <- rmseMean_5[1] + rmseMean_5[2] + rmseMean_5[3] + rmseMean_5[4]
total_RMSE_10 <- rmseMean_10[1] + rmseMean_10[2] + rmseMean_10[3] + rmseMean_10[4]
total_RMSE_15 <- rmseMean_15[1] + rmseMean_15[2] + rmseMean_15[3] + rmseMean_15[4]
total_RMSE_20 <- rmseMean_20[1] + rmseMean_20[2] + rmseMean_20[3] +
```

```
      rmseMean_20[4]
total_RMSE_25 <- rmseMean_25[1] + rmseMean_25[2] + rmseMean_25[3] +
      rmseMean_25[4]

total_rmse <- c(total_RMSE_2,total_RMSE_5, total_RMSE_10,
total_RMSE_15, total_RMSE_20,total_RMSE_25)

per_col <- c(2,5,10,15,20,25)

rmse_df <- data.frame(percentage = per_col,error = total_rmse);
barplot(rmse_df$error,  ylab = "RMSE",
main="RMSE error distribution from 2% to 25%  missing data")



## calculate classification error
lebal <- iris[,5]


#set.seed(9850)

#gp<- runif(nrow(iris))

#iris_r <- iris[order(gp)]
iris_r <- iris_r[,c(1,2,3,4)]
iris_c_2 <-rbind(iris_r, iris.meanImpuated_2)
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

iris_n <- as.data.frame(lapply(iris_c_2, normalize))

iris_train <- iris_n[1:150,]
iris_train_test <- iris_n[151:300,]
iris_train_target <- iris[, 5]

iris.meanImpuated_5_n <- as.data.frame(lapply(iris.meanImpuated_5,
     normalize))
iris.meanImpuated_10_n <- as.data.frame(lapply(iris.meanImpuated_
     10, normalize))
iris.meanImpuated_15_n <- as.data.frame(lapply(iris.meanImpuated_
     15, normalize))
iris.meanImpuated_20_n <- as.data.frame(lapply(iris.meanImpuated_
     20, normalize))
iris.meanImpuated_25_n <- as.data.frame(lapply(iris.meanImpuated_
     25, normalize))
iris.meanImpuated_2_n <- as.data.frame(lapply(iris.meanImpuated_2,
     normalize))


iris.pred_2 <- knn(train = iris_train_test,
test=iris_train_test, cl = iris_train_target, k =20)
iris.pred_5 <- knn(train = iris.meanImpuated_5_n,
test=iris.meanImpuated_5_n, cl = iris_train_target, k =20)
iris.pred_10 <- knn(train = iris.meanImpuated_10_n,
test=iris.meanImpuated_10_n, cl = iris_train_target, k =20)
iris.pred_15 <- knn(train = iris.meanImpuated_15_n,
```

```r
test=iris.meanImpuated_15_n, cl = iris_train_target, k =20)
iris.pred_20 <- knn(train = iris.meanImpuated_20_n,
test=iris.meanImpuated_20_n, cl = iris_train_target, k =20)
iris.pred_25 <- knn(train = iris.meanImpuated_25_n,
test=iris.meanImpuated_25_n, cl = iris_train_target, k =20)
iris.pred <- knn(train = iris_train, test=iris_train,
cl = iris_train_target, k =20)

table(iris_train_target, iris.pred_2)
table(iris_train_target, iris.pred_5)
table(iris_train_target, iris.pred_10)
table(iris_train_target, iris.pred_15)
table(iris_train_target, iris.pred_20)
table(iris_train_target, iris.pred_25)
```

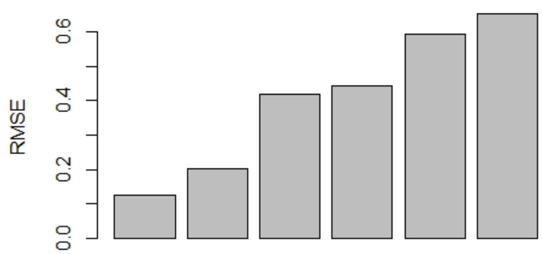

Figure 5: RMSE error distribution from 2% to 25% missing data for KNN MNAR

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

```r
## install packages
install.packages("VIM")
install.packages("mice")
install.packages("missForest")
install.packages("imputeR")
install.packages("hydroGOF")

## import libraries
```

```
library(mice)
library(missForest)
library(VIM)
library(imputeR)
library(hydroGOF)
library(class)
path <- ".."
show(path)

#load data
data <- iris

# get summary
summary(data)

dataWolastCol <- data[,-5]
##create missing data 2%,5%,10%, 15%,20%,25%
dataWolastCol_1 <- as.data.frame(data[,1])
dataWolastCol_3 <- as.data.frame(data[,3])
##create missing data 2,5,10, 15,20,25
iris.mis_2_0 <- prodNA(dataWolastCol_1, noNA = 0.04)
iris.mis_2_1 <- prodNA(dataWolastCol_3, noNA = 0.04)

iris.mis_5_0 <- prodNA(dataWolastCol_1, noNA = 0.1)
iris.mis_5_1 <- prodNA(dataWolastCol_3, noNA = 0.1)

iris.mis_10_0 <- prodNA(dataWolastCol_1, noNA = 0.2)
iris.mis_10_1 <- prodNA(dataWolastCol_3, noNA = 0.2)

iris.mis_15_0 <- prodNA(dataWolastCol_1, noNA = 0.3)
iris.mis_15_1 <- prodNA(dataWolastCol_3, noNA = 0.3)

iris.mis_20_0 <- prodNA(dataWolastCol_1, noNA = 0.4)
iris.mis_20_1 <- prodNA(dataWolastCol_3, noNA = 0.4)

iris.mis_25_0 <- prodNA(dataWolastCol_1, noNA = 0.5)
iris.mis_25_1 <- prodNA(dataWolastCol_3, noNA = 0.5)


iris.mis_2 <- cbind(iris.mis_2_0,data[,2],iris.mis_2_1, data[,4])
iris.mis_5 <- cbind(iris.mis_5_0,data[,2],iris.mis_5_1, data[,4])
iris.mis_10 <- cbind(iris.mis_10_0,data[,2],iris.mis_10_1, data
    [,4])
iris.mis_15 <- cbind(iris.mis_15_0,data[,2],iris.mis_15_1, data
    [,4])
iris.mis_20 <- cbind(iris.mis_20_0,data[,2],iris.mis_20_1, data
    [,4])
iris.mis_25 <- cbind(iris.mis_25_0,data[,2],iris.mis_25_1, data
    [,4])


## call pmm inputation with iris.mis
iris.pmmImpuated_2 <- mice(iris.mis_2, m=5, maxit = 50, method = '
    pmm', seed = 500)
iris.pmmImpuated_5 <- mice(iris.mis_5, m=5, maxit = 50, method = '
    pmm', seed = 500)
iris.pmmImpuated_10 <- mice(iris.mis_10, m=5, maxit = 50, method =
```

```
    'pmm', seed = 500)
iris.pmmImpuated_15<- mice(iris.mis_15, m=5, maxit = 50, method = 'pmm', seed = 500)
iris.pmmImpuated_20<- mice(iris.mis_20, m=5, maxit = 50, method = 'pmm', seed = 500)
iris.pmmImpuated_25 <- mice(iris.mis_25, m=5, maxit = 50, method = 'pmm', seed = 500)

iris.pmmImpuated_2 <- complete(iris.pmmImpuated_2,1) # iris.pmmImpuated_2$data[,1:4]
iris.pmmImpuated_5 <- complete(iris.pmmImpuated_5,1)#iris.pmmImpuated_5$data[,1:4]
iris.pmmImpuated_10 <- complete(iris.pmmImpuated_10,1)#iris.pmmImpuated_10$data[,1:4]
iris.pmmImpuated_15<- complete(iris.pmmImpuated_15,1)#iris.pmmImpuated_15$data[,1:4]
iris.pmmImpuated_20<- complete(iris.pmmImpuated_20,1)#iris.pmmImpuated_20$data[,1:4]
iris.pmmImpuated_25 <- complete(iris.pmmImpuated_25,1)#iris.pmmImpuated_25$data[,1:4]

rmsepmm_2 <- rmse(iris.pmmImpuated_2, dataWolastCol, na.rm = TRUE)
rmsepmm_5 <- rmse(iris.pmmImpuated_5, dataWolastCol, na.rm = TRUE)
rmsepmm_10 <- rmse(iris.pmmImpuated_10, dataWolastCol, na.rm = TRUE)
rmsepmm_15<- rmse(iris.pmmImpuated_15,dataWolastCol ,na.rm = TRUE)
rmsepmm_20 <- rmse(iris.pmmImpuated_20, dataWolastCol, na.rm = TRUE)
rmsepmm_25 <- rmse(iris.pmmImpuated_25, dataWolastCol, na.rm = TRUE)

print(rmsepmm_2)
print(rmsepmm_5)
print(rmsepmm_10)
print(rmsepmm_15)
print(rmsepmm_20)
print(rmsepmm_25)
rmseVecpt <- c(rmsepmm_2, rmsepmm_5, rmsepmm_10,
rmsepmm_15, rmsepmm_20, rmsepmm_25)

barplot(rmseVecpt)



##plot RMSE
barplot(rmsepmm_2, ylab = "RMSE", main="RMSE error distribution for 2% missing data")
barplot(rmsepmm_5 , ylab = "RMSE", main="RMSE error distribution for 5% missing data")
barplot(rmsepmm_10, ylab = "RMSE", main="RMSE error distribution for 10% missing data")
barplot(rmsepmm_15, ylab = "RMSE", main="RMSE error distribution for 15% missing data")
barplot(rmsepmm_20, ylab = "RMSE", main="RMSE error distribution for 20% missing data")
barplot(rmsepmm_25, ylab = "RMSE", main="RMSE error distribution for 25% missing data")
```

```
total_RMSE_2 <-  rmsepmm_2[1] + rmsepmm_2[2] + rmsepmm_2[3] +
    rmsepmm_2[4]
total_RMSE_5 <- rmsepmm_5[1] + rmsepmm_5[2] + rmsepmm_5[3] +
    rmsepmm_5[4]
total_RMSE_10 <- rmsepmm_10[1] + rmsepmm_10[2] + rmsepmm_10[3] +
    rmsepmm_10[4]
total_RMSE_15 <- rmsepmm_15[1] + rmsepmm_15[2] + rmsepmm_15[3] +
    rmsepmm_15[4]
total_RMSE_20 <- rmsepmm_20[1] + rmsepmm_20[2] + rmsepmm_20[3] +
    rmsepmm_20[4]
total_RMSE_25 <- rmsepmm_25[1] + rmsepmm_25[2] + rmsepmm_25[3] +
    rmsepmm_25[4]

total_rmse <- c(total_RMSE_2,total_RMSE_5, total_RMSE_10,
total_RMSE_15, total_RMSE_20,total_RMSE_25)

per_col <- c(2,5,10,15,20,25)

rmse_df <- data.frame(percentage = per_col,error = total_rmse);
barplot(rmse_df$error ,  ylab = "RMSE",
main="RMSE error distribution from 2% to 25%  missing data")



## calculate classification error
lebal <- iris[,5]


#set.seed(9850)

#gp<- runif(nrow(iris))

#iris_r <- iris[order(gp)]
iris_r <- iris_r[,c(1,2,3,4)]
iris_c_2 <-rbind(iris_r, iris.pmmImpuated_2)
normalize <- function(x){
  return( (x -min(x))/ (max(x) - min(x)))
}

iris_n <- as.data.frame(lapply(iris_c_2, normalize))

iris_train <- iris_n[1:150,]
iris_train_test <- iris_n[151:300,]
iris_train_target <- iris[, 5]

iris.pmmImpuated_5_n <- as.data.frame(lapply(iris.pmmImpuated_5,
    normalize))
iris.pmmImpuated_10_n <- as.data.frame(lapply(iris.pmmImpuated_10,
    normalize))
iris.pmmImpuated_15_n <- as.data.frame(lapply(iris.pmmImpuated_15,
    normalize))
iris.pmmImpuated_20_n <- as.data.frame(lapply(iris.pmmImpuated_20,
    normalize))
iris.pmmImpuated_25_n <- as.data.frame(lapply(iris.pmmImpuated_25,
```

```
    normalize))
iris.pmmImpuated_2_n <- as.data.frame(lapply(iris.pmmImpuated_2,
    normalize))


iris.pred_2 <- knn(train = iris_train_test,
test=iris_train_test, cl = iris_train_target, k =20)
iris.pred_5 <- knn(train = iris.pmmImpuated_5_n,
test=iris.pmmImpuated_5_n, cl = iris_train_target, k =20)
iris.pred_10 <- knn(train = iris.pmmImpuated_10_n,
test=iris.pmmImpuated_10_n, cl = iris_train_target, k =20)
iris.pred_15 <- knn(train = iris.pmmImpuated_15_n,
test=iris.pmmImpuated_15_n, cl = iris_train_target, k =20)
iris.pred_20 <- knn(train = iris.pmmImpuated_20_n,
test=iris.pmmImpuated_20_n, cl = iris_train_target, k =20)
iris.pred_25 <- knn(train = iris.pmmImpuated_25_n,
test=iris.pmmImpuated_25_n, cl = iris_train_target, k =20)
iris.pred <- knn(train = iris_train,
test=iris_train, cl = iris_train_target, k =20)

table(iris_train_target, iris.pred_2)
table(iris_train_target, iris.pred_5)
table(iris_train_target, iris.pred_10)
table(iris_train_target, iris.pred_15)
table(iris_train_target, iris.pred_20)
table(iris_train_target, iris.pred_25)
```

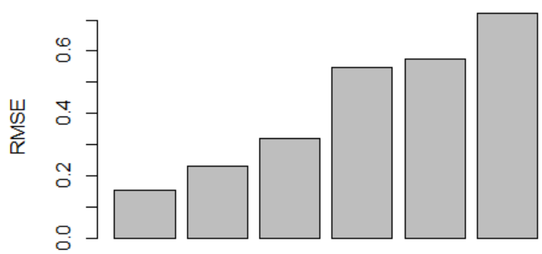

Figure 6: RMSE error distribution from 2% to 25% missing data for PMM MNAR

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