This assignment is an extension of the previous assignment CA1 where we generated decision trees for the given dataset. In this assignment (CA2), we need to perform data imputation and find the differences between the outputs/accuracy of the decision trees with and without data imputation.

Approach:

1. My first approach was to try data imputation with mean. I replaced all NA values in all the columns with the mean of the respective columns. I noticed that the performance of most of the decision trees got worse when compared to the performance of decision trees in CA1. Most notably, the decision trees which performed best in CA1 had a huge drop in accuracy (from 78% to 62%).
2. The next approach was to impute the data with median values. I replaced all NA values in all the columns with the median of the respective columns. I noticed that this performed better than the mean imputation, although the accuracy was still relatively low as compared to CA1. The accuracy of the best performing decision tree dropped from 78% to 75%, but the accuracy of some other decision trees increased from 72% to 75%, while the accuracy of some decision trees remained the same.
3. The next approach I used was imputation using kNN method. I applied kNN imputation on all columns and noticed that although the accuracy of the best performing model reduced to 75 percent which is the same as the accuracy for imputation using median, the relative accuracy of the different decision trees was better than what it was for decision trees whose data imputation was done using median values. For some of the trees, the accuracy remained unchanged.

Therefore, I have made a report which highlights only those trees and accuracy values obtained by kNN based imputation.

**Imputation:**

#Inpute data using kNN  
Impute <- kNN(Data, variable= colnames(Data), k=5)  
  
Data <- Impute[1:19]  
  
df <- data.frame(Data[,-1]) #remove ID from dataframe  
Target=ifelse(df$Response==1,'Y','N')   
df <- data.frame(df, Target) #add Target to the dataframe  
df <- df[,-1] #remove Response  
df <- mutate(df,Y1=factor(Y1),Y2=factor(Y2),Y3=factor(Y3),Y4=factor(Y4),Y5=factor(Y5),Y6=factor(Y6),Y7=factor(Y7))  
str(df)

## 'data.frame': 296 obs. of 18 variables:  
## $ Group : num 0 1 1 0 1 0 1 1 0 0 ...  
## $ X1 : num 460 74 58 39 15 47 23 14 56 40 ...  
## $ X2 : num 460 369 0 771 45 141 69 126 0 120 ...  
## $ X3 : num 460 656 0 3960 60 188 92 224 0 160 ...  
## $ X4 : num 50.2 812.5 87.7 92.1 75.2 ...  
## $ X5 : num 9.15 0.88 0.39 26.79 16.6 ...  
## $ X6 : num 2.3 4.1 4.7 3.1 3.6 2.6 7.1 2.4 2.7 2.6 ...  
## $ X7 : num 274 407 946 535 1019 ...  
## $ Y1 : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 1 1 2 2 ...  
## $ Y2 : Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 1 1 1 ...  
## $ Y3 : Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 1 1 1 ...  
## $ Y4 : Factor w/ 2 levels "0","1": 1 2 2 2 2 1 1 1 1 1 ...  
## $ Y5 : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 2 2 ...  
## $ Y6 : Factor w/ 3 levels "0","1","2": 2 3 3 2 3 2 3 2 2 2 ...  
## $ Y7 : Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 1 1 1 ...  
## $ ID\_imp : logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Response\_imp: logi FALSE FALSE FALSE FALSE FALSE FALSE ...  
## $ Target : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...

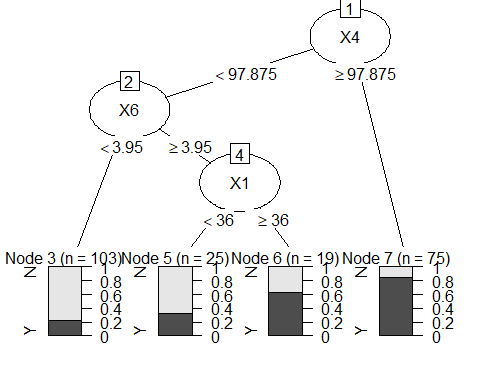
summary(df)

## Group X1 X2 X3   
## Min. :0.0000 Min. : 5.0 Min. : 0 Min. : 0   
## 1st Qu.:0.0000 1st Qu.: 16.0 1st Qu.: 57 1st Qu.: 92   
## Median :1.0000 Median : 39.0 Median : 228 Median : 362   
## Mean :0.6757 Mean : 299.5 Mean : 2336 Mean : 4129   
## 3rd Qu.:1.0000 3rd Qu.: 189.0 3rd Qu.: 771 3rd Qu.: 1572   
## Max. :1.0000 Max. :9743.0 Max. :80919 Max. :143856   
## X4 X5 X6 X7   
## Min. : 21.82 Min. : 0.100 Min. :0.900 Min. : 110.3   
## 1st Qu.: 50.61 1st Qu.: 9.092 1st Qu.:3.200 1st Qu.: 377.5   
## Median : 71.83 Median :20.060 Median :3.700 Median : 684.6   
## Mean : 233.34 Mean :35.439 Mean :3.822 Mean :1418.0   
## 3rd Qu.: 132.38 3rd Qu.:61.970 3rd Qu.:4.125 3rd Qu.:1713.9   
## Max. :6864.00 Max. :99.800 Max. :9.700 Max. :8491.1   
## Y1 Y2 Y3 Y4 Y5 Y6 Y7 ID\_imp   
## 0:144 0:172 0:154 0:144 0:158 0: 4 0:116 Mode :logical   
## 1:152 1:124 1:142 1:152 1:138 1:124 1:180 FALSE:296   
## 2:168   
##   
##   
##   
## Response\_imp Target   
## Mode :logical N:154   
## FALSE:296 Y:142   
##   
##   
##   
##

#split into train and test sets  
set.seed(123)  
sample= sample.split(df$Target,SplitRatio= 0.75)  
train=subset(df, sample==TRUE)  
test= subset(df, sample==FALSE)

1. **All X's, Y's and groups**

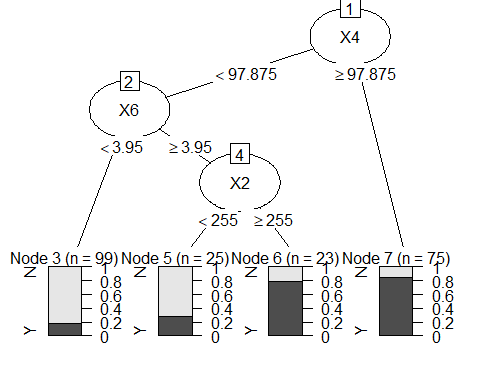
**Before Imputation:**  
df\_with\_all\_X\_and\_Y\_all\_groups <- train  
DT\_Model\_XY\_all\_groups <- rpart(Target~., data=df\_with\_all\_X\_and\_Y\_all\_groups,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))   
plot(as.party(DT\_Model\_XY\_all\_groups))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_XY\_all\_groups,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 32 10  
## Y 6 26  
##   
## Accuracy : 0.7838   
## 95% CI : (0.6728, 0.8711)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.494e-06   
##   
## Kappa : 0.566   
##   
## Mcnemar's Test P-Value : 0.4533   
##   
## Sensitivity : 0.8421   
## Specificity : 0.7222   
## Pos Pred Value : 0.7619   
## Neg Pred Value : 0.8125   
## Prevalence : 0.5135   
## Detection Rate : 0.4324   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7822   
##   
## 'Positive' Class : N   
##

**\*Note: This tree had the best accuracy out of all trees (without imputation). Accuracy = 78.38%**

**After Imputation:**  


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_XY\_all\_groups,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 11  
## Y 7 25  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5117   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6944   
## Pos Pred Value : 0.7381   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7551   
##   
## 'Positive' Class : N   
##

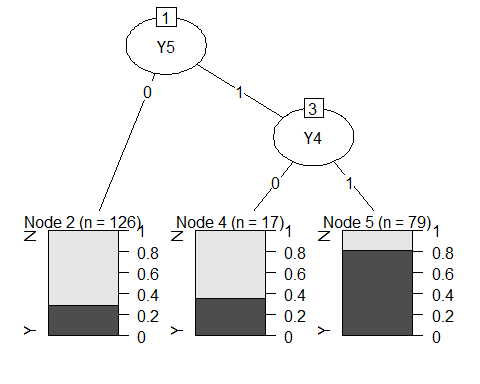
**Accuracy before imputation: 78.38%**

**Accuracy after imputation: 75.68% - reduced accuracy**

1. **all Y's - for all groups**

**Before Imputation:**

df\_excluding\_X <- within(train, rm(X1, X2, X3, X4, X5, X6, X7))  
DT\_Model\_excluding\_X <- rpart(Target~., data=df\_excluding\_X,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_excluding\_X))

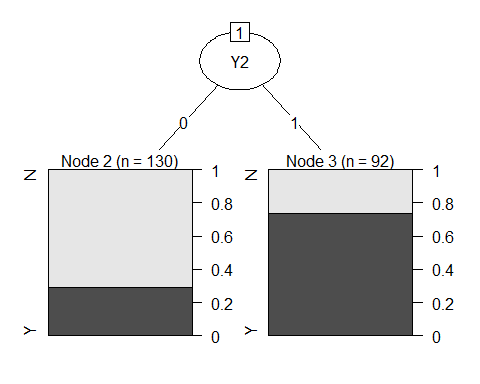


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_X,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 13  
## Y 7 23  
##   
## Accuracy : 0.7297   
## 95% CI : (0.6139, 0.8265)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.0001207   
##   
## Kappa : 0.4567   
##   
## Mcnemar's Test P-Value : 0.2635525   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6389   
## Pos Pred Value : 0.7045   
## Neg Pred Value : 0.7667   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5946   
## Balanced Accuracy : 0.7273   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_excluding\_X))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_X,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 11  
## Y 7 25  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5117   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6944   
## Pos Pred Value : 0.7381   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7551   
##   
## 'Positive' Class : N   
##

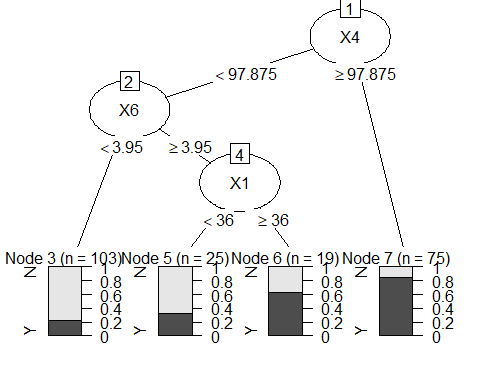
Accuracy **before** Imputation**: 72.97%**

Accuracy **after** Imputation**: 75.68% - improved accuracy**

1. **X's - for all groups**

**Before Imputation:**

df\_excluding\_Y <- within(train, rm(Y1, Y2, Y3, Y4, Y5, Y6, Y7))  
DT\_Model\_excluding\_Y <- rpart(Target~., data=df\_excluding\_Y,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_excluding\_Y))



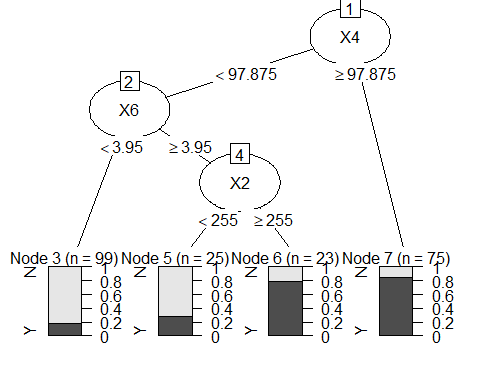
# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_Y,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 32 10  
## Y 6 26  
##   
## Accuracy : 0.7838   
## 95% CI : (0.6728, 0.8711)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.494e-06   
##   
## Kappa : 0.566   
##   
## Mcnemar's Test P-Value : 0.4533   
##   
## Sensitivity : 0.8421   
## Specificity : 0.7222   
## Pos Pred Value : 0.7619   
## Neg Pred Value : 0.8125   
## Prevalence : 0.5135   
## Detection Rate : 0.4324   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7822   
##   
## 'Positive' Class : N   
##

**\*Note: This tree also had the best accuracy out of all trees in CA1. Accuracy = 78.38%. Therefore, we notice that the trees with best accuracy are the ones where we consider all both groups (0,1), with or without Y, as the splits are made on X’s.**

**After Imputation:**

plot(as.party(DT\_Model\_excluding\_Y))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_Y,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 11  
## Y 7 25  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5117   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6944   
## Pos Pred Value : 0.7381   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7551   
##   
## 'Positive' Class : N   
##

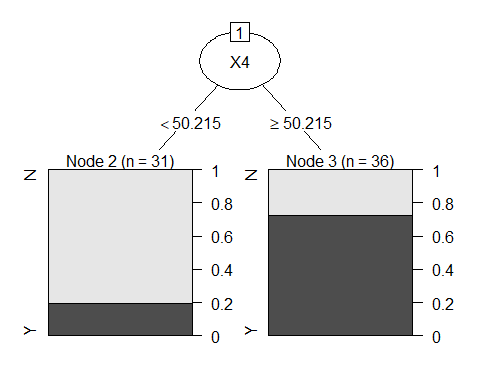
Accuracy **before** Imputation**: 78.38%**

Accuracy **after** Imputation**: 75.68% - reduced accuracy**

1. **All X,Y, group 0**

**Before Imputation:**

df\_all\_XY\_group0 <- train[train$Group == 0,]  
DT\_Model\_all\_XY\_group0 <- rpart(Target~., data=df\_all\_XY\_group0,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_XY\_group0))

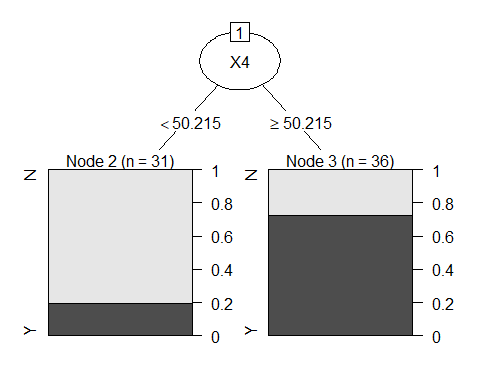


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_XY\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 10 7  
## Y 28 29  
##   
## Accuracy : 0.527   
## 95% CI : (0.4075, 0.6443)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.4541756   
##   
## Kappa : 0.0677   
##   
## Mcnemar's Test P-Value : 0.0007232   
##   
## Sensitivity : 0.2632   
## Specificity : 0.8056   
## Pos Pred Value : 0.5882   
## Neg Pred Value : 0.5088   
## Prevalence : 0.5135   
## Detection Rate : 0.1351   
## Detection Prevalence : 0.2297   
## Balanced Accuracy : 0.5344   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_XY\_group0))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_XY\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 10 7  
## Y 28 29  
##   
## Accuracy : 0.527   
## 95% CI : (0.4075, 0.6443)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.4541756   
##   
## Kappa : 0.0677   
##   
## Mcnemar's Test P-Value : 0.0007232   
##   
## Sensitivity : 0.2632   
## Specificity : 0.8056   
## Pos Pred Value : 0.5882   
## Neg Pred Value : 0.5088   
## Prevalence : 0.5135   
## Detection Rate : 0.1351   
## Detection Prevalence : 0.2297   
## Balanced Accuracy : 0.5344   
##   
## 'Positive' Class : N   
##

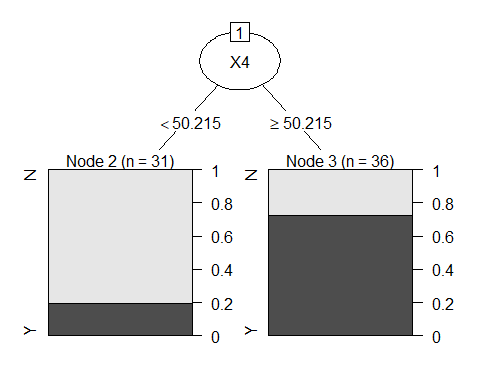
Accuracy **before** Imputation**: 52.7%**

Accuracy **after** Imputation**: 52.7% - No change**

1. **all X, group 0**

**Before Imputation:**

df\_all\_X\_group0 <- train[train$Group == 0,]  
df\_all\_X\_group0 <- within(df\_all\_X\_group0, rm(Y1, Y2, Y3, Y4, Y5, Y6, Y7))  
DT\_Model\_all\_X\_group0 <- rpart(Target~., data=df\_all\_X\_group0,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_X\_group0))

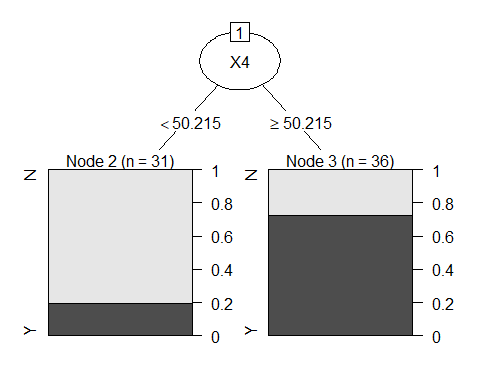


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_X\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 10 7  
## Y 28 29  
##   
## Accuracy : 0.527   
## 95% CI : (0.4075, 0.6443)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.4541756   
##   
## Kappa : 0.0677   
##   
## Mcnemar's Test P-Value : 0.0007232   
##   
## Sensitivity : 0.2632   
## Specificity : 0.8056   
## Pos Pred Value : 0.5882   
## Neg Pred Value : 0.5088   
## Prevalence : 0.5135   
## Detection Rate : 0.1351   
## Detection Prevalence : 0.2297   
## Balanced Accuracy : 0.5344   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_X\_group0))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_X\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 10 7  
## Y 28 29  
##   
## Accuracy : 0.527   
## 95% CI : (0.4075, 0.6443)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.4541756   
##   
## Kappa : 0.0677   
##   
## Mcnemar's Test P-Value : 0.0007232   
##   
## Sensitivity : 0.2632   
## Specificity : 0.8056   
## Pos Pred Value : 0.5882   
## Neg Pred Value : 0.5088   
## Prevalence : 0.5135   
## Detection Rate : 0.1351   
## Detection Prevalence : 0.2297   
## Balanced Accuracy : 0.5344   
##   
## 'Positive' Class : N   
##

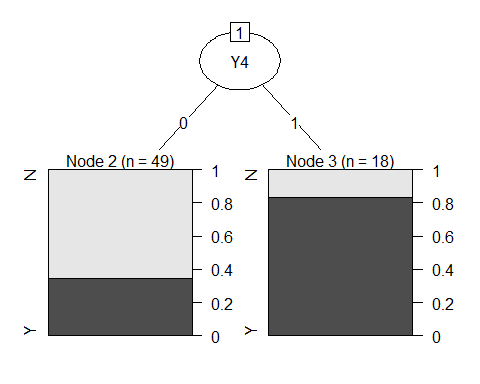
Accuracy **before** Imputation**: 52.7%**

Accuracy **after** Imputation**: 52.7% - No change**

1. **all Y, group 0**

**Before Imputation:**

df\_all\_Y\_group0 <- train[train$Group == 0,]  
df\_all\_Y\_group0 <- within(df\_all\_Y\_group0, rm(X1, X2, X3, X4, X5, X6, X7))  
DT\_Model\_all\_Y\_group0 <- rpart(Target~., data=df\_all\_Y\_group0,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_Y\_group0))

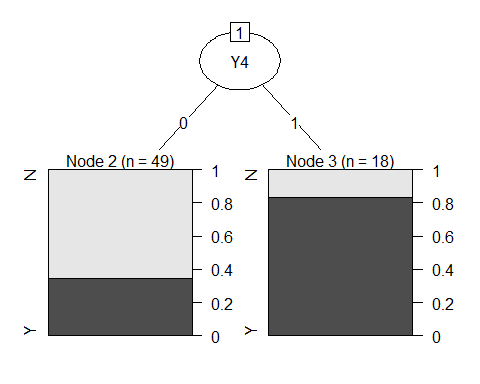


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_Y\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 25 12  
## Y 13 24  
##   
## Accuracy : 0.6622   
## 95% CI : (0.5428, 0.7681)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.006916   
##   
## Kappa : 0.3243   
##   
## Mcnemar's Test P-Value : 1.000000   
##   
## Sensitivity : 0.6579   
## Specificity : 0.6667   
## Pos Pred Value : 0.6757   
## Neg Pred Value : 0.6486   
## Prevalence : 0.5135   
## Detection Rate : 0.3378   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.6623   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_Y\_group0))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_Y\_group0,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 25 12  
## Y 13 24  
##   
## Accuracy : 0.6622   
## 95% CI : (0.5428, 0.7681)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.006916   
##   
## Kappa : 0.3243   
##   
## Mcnemar's Test P-Value : 1.000000   
##   
## Sensitivity : 0.6579   
## Specificity : 0.6667   
## Pos Pred Value : 0.6757   
## Neg Pred Value : 0.6486   
## Prevalence : 0.5135   
## Detection Rate : 0.3378   
## Detection Prevalence : 0.5000   
## Balanced Accuracy : 0.6623   
##   
## 'Positive' Class : N

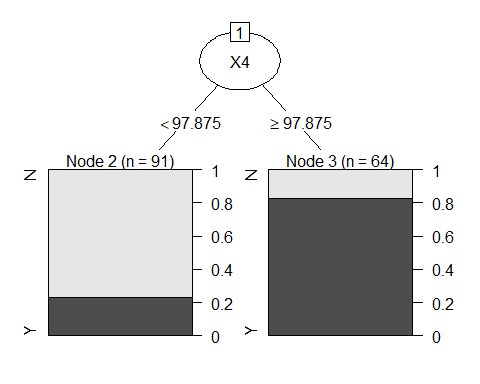
Accuracy **before** Imputation**: 66.22%**

Accuracy **after** Imputation**: 66.22% - No change**

1. **All X,Y, group 1**

**Before Imputation:**

df\_all\_XY\_group1 <- train[train$Group == 1,]  
DT\_Model\_all\_XY\_group1 <- rpart(Target~., data=df\_all\_XY\_group1,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_XY\_group1))

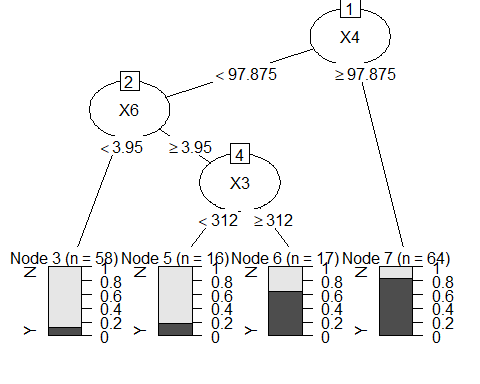


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_X,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 13  
## Y 7 23  
##   
## Accuracy : 0.7297   
## 95% CI : (0.6139, 0.8265)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.0001207   
##   
## Kappa : 0.4567   
##   
## Mcnemar's Test P-Value : 0.2635525   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6389   
## Pos Pred Value : 0.7045   
## Neg Pred Value : 0.7667   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5946   
## Balanced Accuracy : 0.7273   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_XY\_group1))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_excluding\_X,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 11  
## Y 7 25  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5117   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6944   
## Pos Pred Value : 0.7381   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7551   
##   
## 'Positive' Class : N   
##

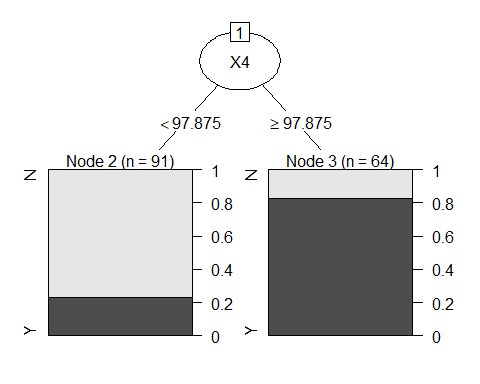
Accuracy **before** Imputation**: 72.97%**

Accuracy **after** Imputation**: 75.68% - improved accuracy**

1. **all X, group 1**

**Before Imputation:**

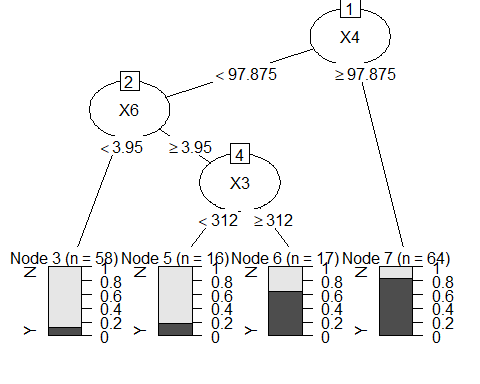
df\_all\_X\_group1 <- train[train$Group == 1,]  
df\_all\_X\_group1 <- within(df\_all\_X\_group1, rm(Y1, Y2, Y3, Y4, Y5, Y6, Y7))  
DT\_Model\_all\_X\_group1 <- rpart(Target~., data=df\_all\_X\_group1,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_X\_group1))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_X\_group1,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 34 14  
## Y 4 22  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5096   
##   
## Mcnemar's Test P-Value : 0.03389   
##   
## Sensitivity : 0.8947   
## Specificity : 0.6111   
## Pos Pred Value : 0.7083   
## Neg Pred Value : 0.8462   
## Prevalence : 0.5135   
## Detection Rate : 0.4595   
## Detection Prevalence : 0.6486   
## Balanced Accuracy : 0.7529   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_X\_group1))

# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_X\_group1,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 31 11  
## Y 7 25  
##   
## Accuracy : 0.7568   
## 95% CI : (0.6431, 0.849)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 1.544e-05   
##   
## Kappa : 0.5117   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.8158   
## Specificity : 0.6944   
## Pos Pred Value : 0.7381   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5135   
## Detection Rate : 0.4189   
## Detection Prevalence : 0.5676   
## Balanced Accuracy : 0.7551   
##   
## 'Positive' Class : N   
##

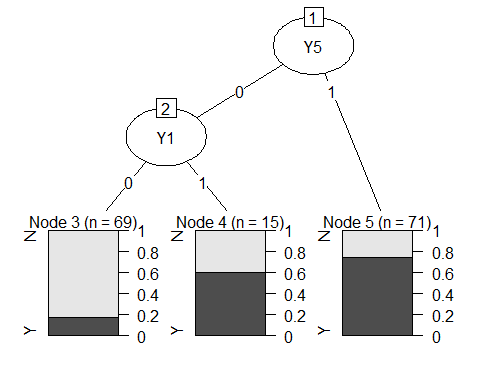
Accuracy **before** Imputation**: 75.68%**

Accuracy **after** Imputation**: 75.68% - No change**

1. **all Y, group 1**

**Before Imputation:**

df\_all\_Y\_group1 <- train[train$Group == 1,]  
df\_all\_Y\_group1 <- within(df\_all\_Y\_group1, rm(X1, X2, X3, X4, X5, X6, X7))  
DT\_Model\_all\_Y\_group1 <- rpart(Target~., data=df\_all\_Y\_group1,   
 control=rpart.control(minsplit=30,   
 minbucket=15,   
 maxdepth=4 ))  
plot(as.party(DT\_Model\_all\_Y\_group1))

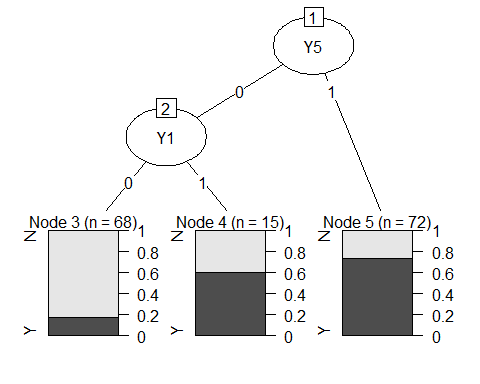


# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_Y\_group1,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 22 6  
## Y 16 30  
##   
## Accuracy : 0.7027   
## 95% CI : (0.5852, 0.8034)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.0007291   
##   
## Kappa : 0.4093   
##   
## Mcnemar's Test P-Value : 0.0550088   
##   
## Sensitivity : 0.5789   
## Specificity : 0.8333   
## Pos Pred Value : 0.7857   
## Neg Pred Value : 0.6522   
## Prevalence : 0.5135   
## Detection Rate : 0.2973   
## Detection Prevalence : 0.3784   
## Balanced Accuracy : 0.7061   
##   
## 'Positive' Class : N   
##

**After Imputation:**

plot(as.party(DT\_Model\_all\_Y\_group1))



# checking accuracy using confusion matrix  
tree.predicted<- predict(DT\_Model\_all\_Y\_group1,test, type='class')  
confusionMatrix(tree.predicted , test$Target)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction N Y  
## N 22 6  
## Y 16 30  
##   
## Accuracy : 0.7027   
## 95% CI : (0.5852, 0.8034)  
## No Information Rate : 0.5135   
## P-Value [Acc > NIR] : 0.0007291   
##   
## Kappa : 0.4093   
##   
## Mcnemar's Test P-Value : 0.0550088   
##   
## Sensitivity : 0.5789   
## Specificity : 0.8333   
## Pos Pred Value : 0.7857   
## Neg Pred Value : 0.6522   
## Prevalence : 0.5135   
## Detection Rate : 0.2973   
## Detection Prevalence : 0.3784   
## Balanced Accuracy : 0.7061   
##   
## 'Positive' Class : N   
##

Accuracy **before** Imputation**: 70.27%**

Accuracy **after** Imputation**: 70.27% - No change**

From the above comparison, we can see that:

1. For **two** decision trees, the accuracy has increased.
2. For **two** decision trees, the accuracy has reduced. These two trees were the best performing when data imputation was not performed. (Accuracy reduced from 78.38% to 75.68%).
3. For **five** decision trees, the accuracy has remained the same after imputation.
4. The highest accuracy after imputation is **75.68%** which was achieved by **five** decision trees.
5. The lowest accuracy is **52.7%** which is the same before imputation as well.