WESTERN MICHIGAN UNIVERSITY

STAT 6850

Applied Data Mining

Fall 2014

Instructor: Dr. J.C. Wang

Case Study #3

Using tree models to classify

Diabetes in Pima Indians Data

Antonio Giraldi

Saurabh Rajratn Kulkarni

Shoruk Mansour

Joan Martinez

Milton Soto Ferrari

Mustafa Yildiz

Description Summary

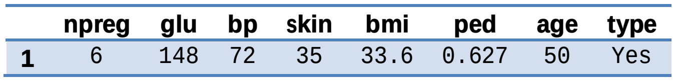
The objective of this report is to use the training data sets (in both form) and develop tree models to predict the presence of diabetes on Pima Indians on test data set to obtain error rate and do cross-validation analysis to choose parameters for best prediction tree. The data in this study is Diabetes in Pima Indians Data, which is divided into two sets, training & test data sets. The test data set is called pima.te (a 332 by 8 R data frame style, plus row 1 being the header) and the training set comes into two forms, pima.tr which has no missing values (a 200 complete observation data set); and pima.tr2 where we additionally have observations with missing values on some attributes (200 complete observations and 100 incomplete observations data set. The data were collected by the US National Institute of Diabetes and Kidney Diseases.

Descriptions of each woman's attributes are as follow:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. Body mass index (weight in kg/(height in m)^2)
6. Diabetes pedigree function
7. Age (years)
8. Response variable: Diabetes (Yes or No)

Exploratory Analysis

First by putting all the data set together (full data set) just to see how the data is presented, we can see that the data has attributes for each subject, and the response is weather the subject has diabetes or not. For example, a female subject with an age of 50, has been pregnant 6 times, with a glucose level of 148, blood pressure of 72, skin fold thickness of 35, BMI of 33.6 and ped 0.627, shows a positive response of having diabetes (see table below).



	npreg	glu	bp	skin	bmi	ped	age	type
1	6	148	72	35	33.6	0.627	50	Yes

By creating a frequency bar plot for the response type in our data (Fig 1); we can see that we have almost twice the number of cases without diabetes than with diabetes. Fig 2 shows box plots for each attribute by response to look at the variability within each group. It also appears that some at the attributes are show different behavior when you have diabetes or not; for example, glucose concentrations are significantly higher in diabetic subjects. Other attributes are not heavily affected by diabetes, such as skin thickness or blood pressure.

Table 1, shows how many observations are missing each attribute. We can see that most of the missing observations are due to skin thickness attribute. This might be important if we find this attribute to be relevant when predicting diabetes on pima Indians.

Running a correlation matrix for all attributes (see Table2), we see that age, as expected, is positive correlated with times pregnant, and BMI is also positive correlated with skin thickness, which was also expected. Blood pressure and BMI are also positive correlated. On Fig 3 we can see that most attributes are positively inter-correlated to some extent.

Tree model classification Analysis

The purpose of using a tree model is to predict if the subject will have diabetes or not given several input variables; it's a simple and intuitive approach to use for predictions. For example, using tree shown in Fig 4 to classify diabetes for subject previously presented (the women subject with the age of 50, number of pregnancy of 6, glucose level of 148, blood pressure of 72, skin 35, bmi of 33.6, and ped 0.627,), the route to follow is glu< 124 (No), ped<0.31(No) and bmi<29 (No), and this fall under response Yes, which means is predicted to have diabetes.

In this study we'll use both training sets to grow our classification trees using recursive partitions, and test it on the test set to get error rates. We also pruned our trees based on estimates given by using recursive partitions on our training set and measure their predictions effectiveness on the test set. Finally we used 10-fold cross validation on cost-complexity pruning using the test set to identify best parameters to prune tree while minimizing error rate.

Training with Complete Observations – Results

When using pima.train (complete observations) to develop a tree model (using rpart) we get the tree shown in Fig 4. Fig 5 shows this tree with impurity percentage on each of its 8 response nodes. Table 3 shows attributes importance to use as predictors, this is also seen in Fig 6. Glucose concentration seems to be the most important attribute for diabetes classification. Using paired attributes classification plots (Fig 7); we see that get low error rates when using glu+bmi, glu+age, glu+ped of around 19%.

Using this tree as to check within-classifier accuracy on pima.train, we get results shown in Table 5 and Table 6. Internal accuracy is about 85%. When using this tree as a predictor on the test set (pima.test) we get results as shown in Table 7 and Table 8. We see now that the accuracy was around 73% and lower Sensitivity and Specificity when compared to prima.train.

Table 4 and Fig 8 show the cross-validated relative error when growing the tree using pima.train; here we see that the error was low when tree size had 5 response nodes and a complexity parameter (Cp) of 0.75. Using this value to prune the tree, we obtained tree as shown in Fig 9, with only 5 response nodes. Node impurity (seen in Fig 10) was somehow maintained after pruning and within-classifier accuracy on pima.train looks also similar (Table 9 and Table 10). With this pruned tree we obtained slightly better results on the test set as seen in Table 11 and Table 12, accuracy is now 75% compared to 73% before pruning.

Training with Incomplete Observations - Results

When using pima.train2 (incomplete observations) to develop a tree model (using rpart) we get the tree shown in Fig 11. Fig 12 shows this tree with impurity percentage on each of its 10 response nodes. We can see the tree grew in size in compare to tree developed with pima.train. Table 13 shows attribute importance to use as predictors, also seen in Fig 13. Again, glucose concentration seems to be the most important attribute for diabetes classification but now followed by bmi, age and oed. Using paired attributes classification plots (Fig 14) on these importante variables we get error rates 22%.

Using this tree as to check within-classifier accuracy on pima.train2, we get results shown in Table 15 and Table 16. Internal accuracy is about 83%. When using this tree as a predictor on the test set (pima.test) we get results as shown in Table 17 and Table 18. We see now that the accuracy was around 76% and lower Sensitivity and Specificity when compared to prima.train2, but better results than pima.train.

Table 14 and Fig 15 show the cross-validated relative error when growing the tree using pima.train2; here we see that the error was low when tree size had 3 response nodes and a complexity parameter (Cp) of 0.707. Using this value to prune the tree, we obtained tree as shown in Fig 16, with only 3 response nodes. As size of tree was significantly reduced, node impurity (seen in Fig 17) was around 20% for each node and within-classifier accuracy on pima.train2 was lowered due to Specificity type-2 error (Table 19 and Table 20). With this pruned tree we again obtained better results on the test set as seen in Table 21 and Table 22, accuracy is now 77% compared to 76% before pruning and Specificity increased 4 percent-points.

Cost-Complexity Pruning Cross-Validation - Results

Now we interested in cross-validating our trees using a 10-fold x-val techniques using trees developed under both training sets. These trees will be cross-validated using the test set with different complexity parameters (Cp) on each fold to identify best Cp to reduce error rate.

Growing the tree using pima.train (Fig 18) we see a tree with high complexity (~40 nodes). Doing our 10 fold cross-validation analysis on the test set, we obtained plot shown in Fig 19. Our range of Cp values are from 0-0.2 in 0.01 increment. Here we see that Cp=0.04 yield the best results in x-val relative error. With this known Cp value we prune our full tree to obtain tree shown in Fig 20; which is the same exact tree as in Fig 10, thus the prediction results on the test set with this tree are the same as shown in Table Table 11 and Table 12.

Growing the tree using pima.train2 (Fig 21) we see a tree with even higher complexity (~60 nodes). Doing our 10 fold cross-validation analysis on the test set, we obtained plot shown in Fig 22. Here we see that any Cp between 0.04 and 0.11 will yield the best results in x-val relative error. Using Cp=0.11 (highest Cp is always preferable), we prune our full tree to obtain tree shown in Fig 23; which again is the same exact tree as in Fig 17, thus the prediction results on the test set with this tree are the same as shown in Table Table 21 and Table 22.

Concluding Remarks

In this Pima Indians dataset we were able to predict presence of diabetes with a good accuracy on trees developed from both training sets. By using 10-fold cross validation of error-rate pruning (cost-complexity pruning) on the test set, we were able to identify the best complexity parameters values which a good predictor tree can be grown. Pima.train2 cross-validated prune tree was the best predictor tree in terms of accuracy, sensitivity and specificity.

APPENDIX

1. Tables

Table1. ***Missing Values***

Variable	# Missing Values
npreg	0
glu	0
bp	13
skin	98
bmi	3
ped	0

Table2. ***Correlation Matrix***

Variables	npreg	glu	bp	skin	bmi	ped	age
Npreg	1.00000	0.12301	0.223732	0.09657	0.01027	-0.00374	0.61327
Glu	-	1.00000	0.221389	0.22799	0.24117	0.16202	0.26293
Bp	-	-	1.00000	0.22567	0.31106	0.00233	0.35845
Skin	-	-	-	1.00000	0.64765	0.11740	0.16279
Bmi	-	-	-	-	1.00000	0.14517	0.04992
Ped	-	-	-	-	-	1.00000	0.04945
age	-	-	-	-	-	-	1.00000

Table3. **pima.train Variable Importance**

glu	26.94491
age	9.256648
bmi	8.186226
bp	7.880943
ped	7.524073
npreg	5.852371
skin	5.765217

Table4. **pima.train CP**

	CP	nsplit	rel error	xerror	xstd
1	0.220588	0	1	1	0.098518
2	0.161765	1	0.779412	0.941176	0.097014
3	0.073529	2	0.617647	0.852941	0.09437
4	0.058824	3	0.544118	0.867647	0.094844
5	0.014706	4	0.485294	0.75	0.090647
6	0.01	7	0.441176	0.794118	0.092331

Table5. Pima.train Train Set Prediction Matrix

	No	Yes
No	121	11
Yes	19	49

Table6. Pima.train Training Results

	Accuracy	Sensitivity	Specificity
Results	0.85	0.864286	0.816667

Table7. Pima.train Prediction Matrix on Test set

	No	Yes
No	182	41
Yes	48	61

Table8. Test set results with pima.train

	Accuracy	Sensitivity	Specificity
Results	0.731928	0.791304	0.598039

Table9. Pruned pima.train Train set Prediction Matrix

	No	Yes
No	123	9
Yes	24	44

Table10. Pruned pima.train Training Results

	Accuracy	Sensitivity	Specificity
Results	0.835	0.836735	0.830189

Table11. Pruned Pima.train Prediction Matrix on Test set

	No	Yes
No	193	30
Yes	51	58

Table12. Test results with Pruned pima.train

	Accuracy	Sensitivity	Specificity
Results	0.756024	0.790984	0.659091

Table13. **pima.train2 Variable Importance**

glu	33.09516
bmi	20.59517
age	11.36768
ped	7.455971
npreg	5.003432
bp	2.530867

Table14. **pima.train2 CP Table**

	CP	nsplit	rel error	xerror	xstd
1	0.235849	0	1	1	0.078107
2	0.132075	1	0.764150943	0.877358	0.075572
3	0.025157	2	0.632075472	0.707547	0.070755
4	0.015723	5	0.556603774	0.783019	0.073097
5	0.01	9	0.471698113	0.764151	0.072543

Table15. Pima.train2 Train Set Prediction Matrix

	No	**Yes**
No	177	17
Yes	33	73

Table16. Pima.train2 Training Results

	Accuracy	**Sensitivity**	**Specificity**
Results	0.833333	0.842857	0.811111111

Table17. Pima.train2 Prediction Matrix on Test set

	No	**Yes**
No	186	37
Yes	43	66

Table18. Test set results with pima.train2

	Accuracy	**Sensitivity**	**Specificity**
Results	0.759036	0.812227	0.640776699

Table19. Pruned pima.train2 Train set Prediction Matrix

	No	**Yes**
No	166	28
Yes	39	67

Table20. Pruned pima.train2 Training Results

	Accuracy	**Sensitivity**	**Specificity**
Results	0.776667	0.809756	0.705263

Table21. Pruned Pima.train2 Prediction Matrix on Test set

	No	**Yes**
No	194	29
Yes	47	62

Table22. Test results with Pruned pima.train2

	Accuracy	**Sensitivity**	**Specificity**
Results	0.771084	0.804979	0.681318681

2. Plots

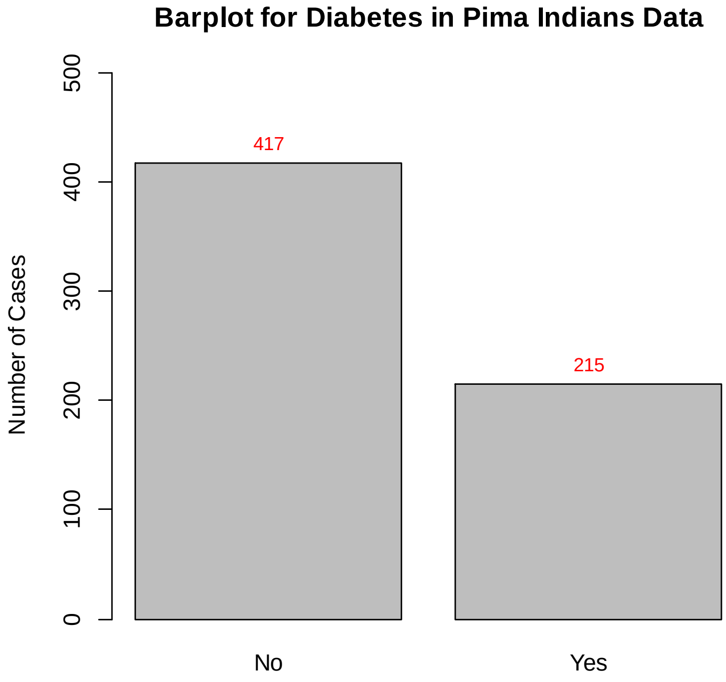


Fig 1

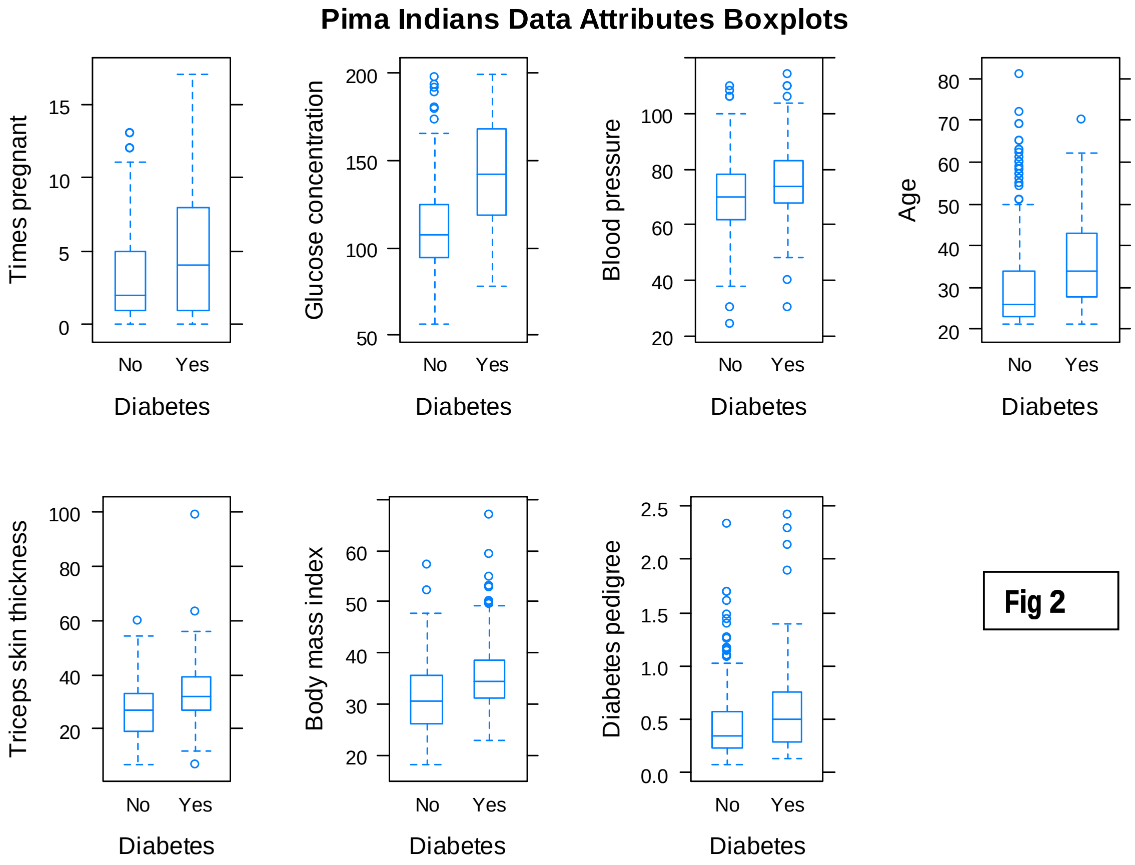


Fig 2

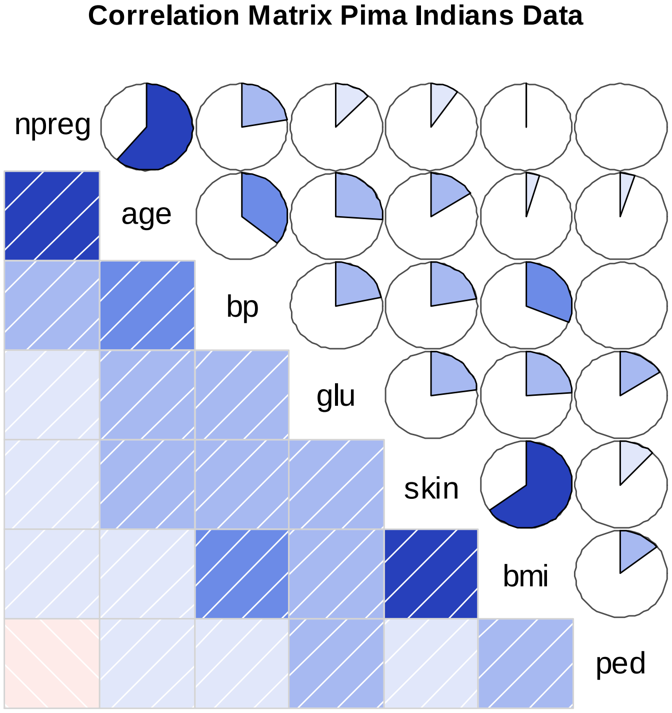

Correlation Matrix Pima Indians Data
npreg
age
bp
glu
skin
bmi
ped

Fig 3

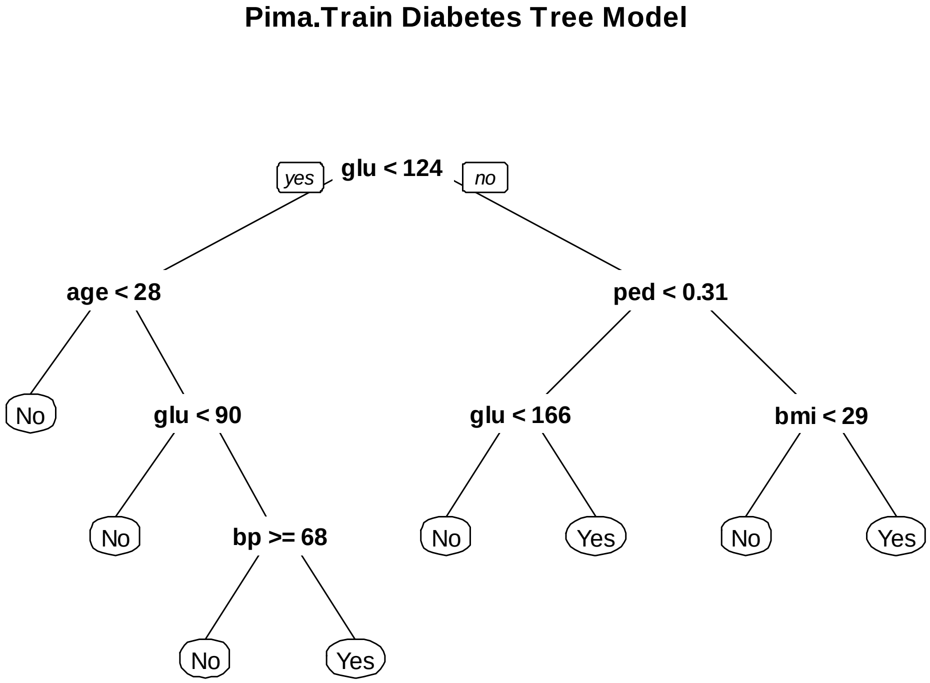

Pima.Train Diabetes Tree Model
yes
glu < 124
no
age < 28
ped < 0.31
No
glu < 90
glu < 166
bmi < 29
No
bp >= 68
No
Yes
No
Yes
No
Yes

Fig 4

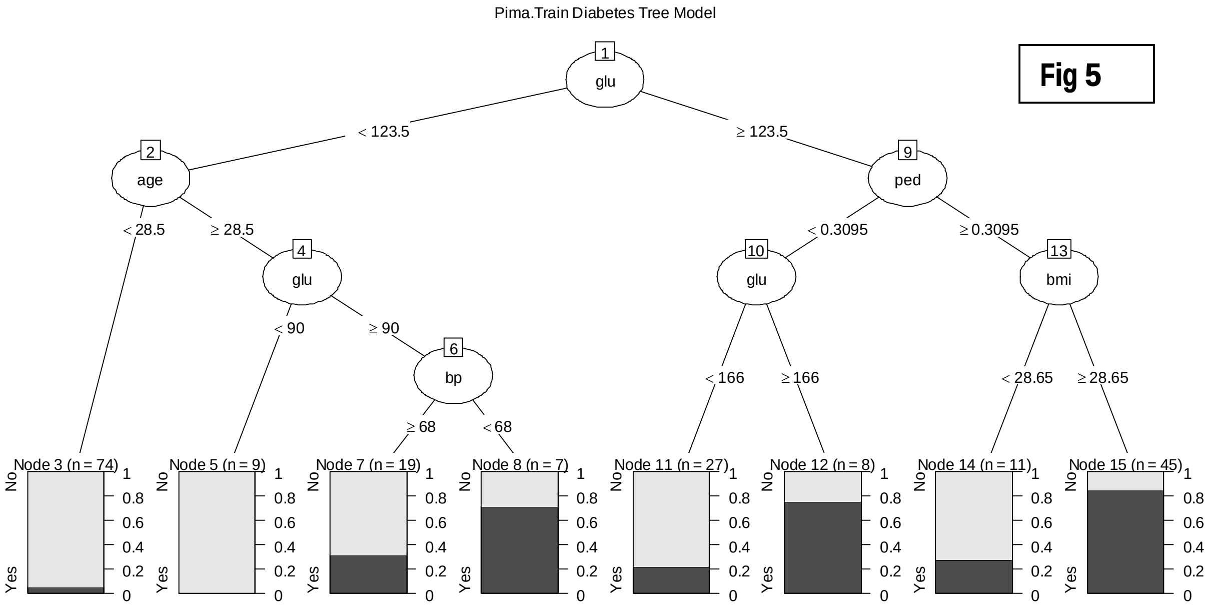

Pima.Train Diabetes Tree Model

Fig 5

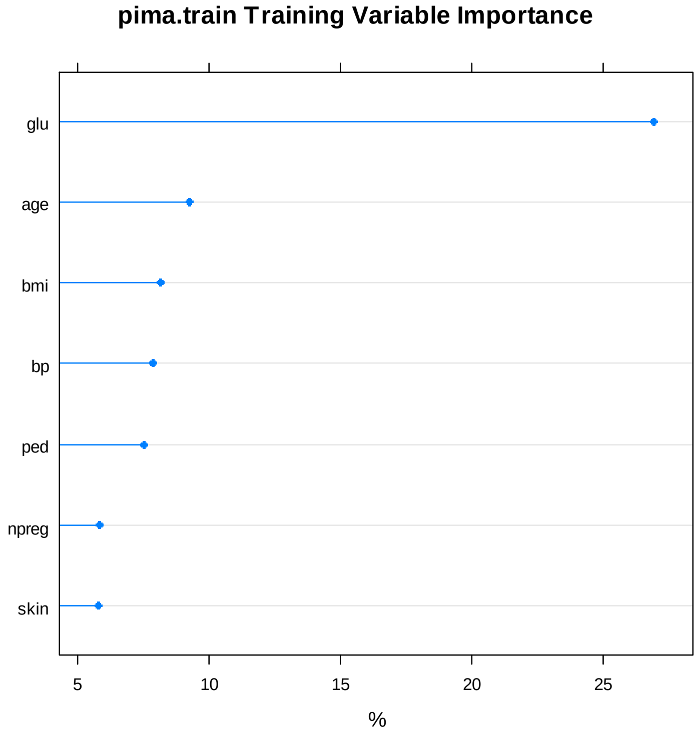

pima.train Training Variable Importance

Fig 6

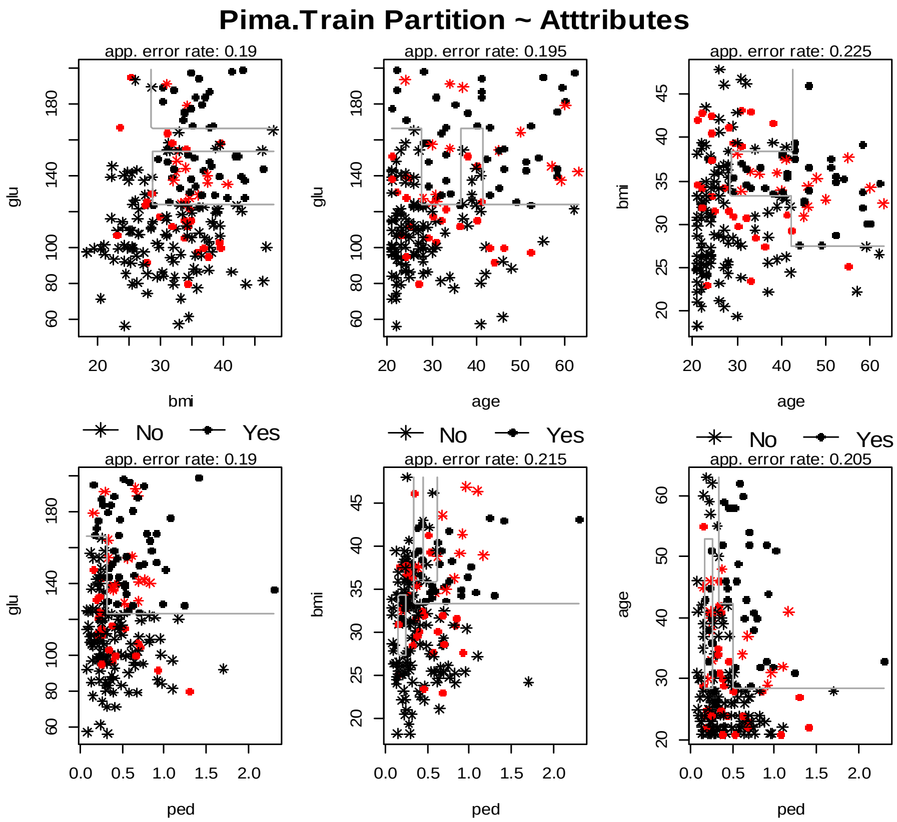

Pima.Train Partition ~ Atttributes
app. error rate: 0.19
app. error rate: 0.195
app. error rate: 0.225
app. error rate: 0.19
app. error rate: 0.215
app. error rate: 0.205
glu
bmi
age
ped
No
Yes
60 80 100 140 180
20 30 40
20 30 40 50 60
20 25 30 35 40 45
0.0 0.5 1.0 1.5 2.0

Fig 7

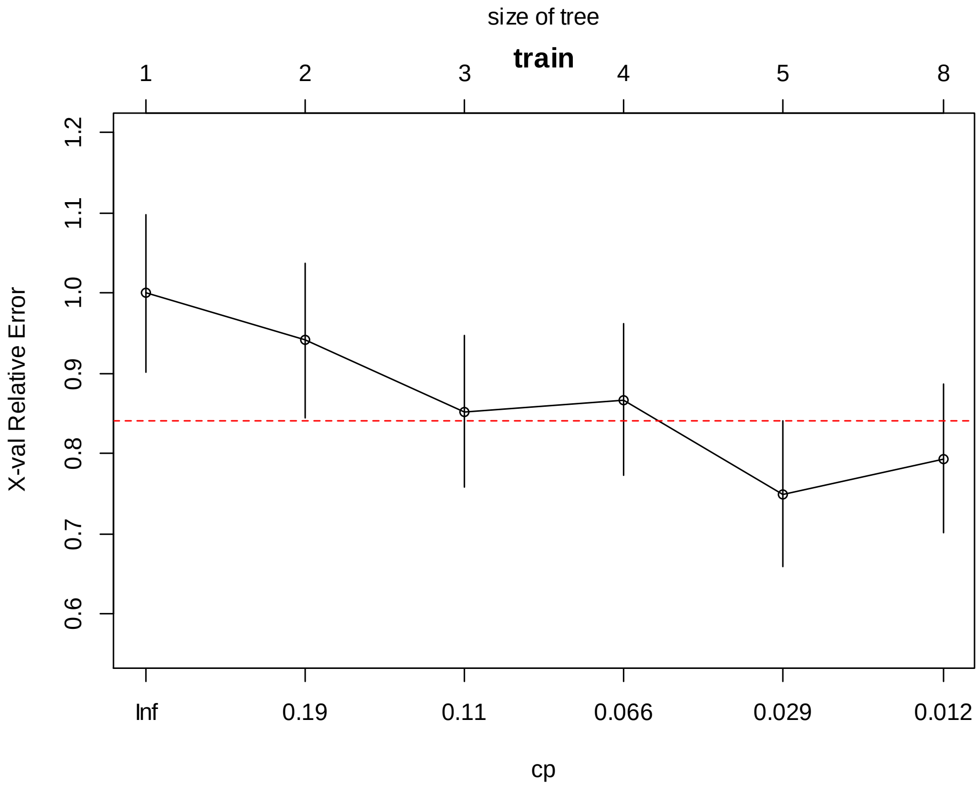

size of tree
train
1 2 3 4 5 8
X-val Relative Error
0.6 0.7 0.8 0.9 1.0 1.1 1.2
Inf 0.19 0.11 0.066 0.029 0.012
cp

Fig 8

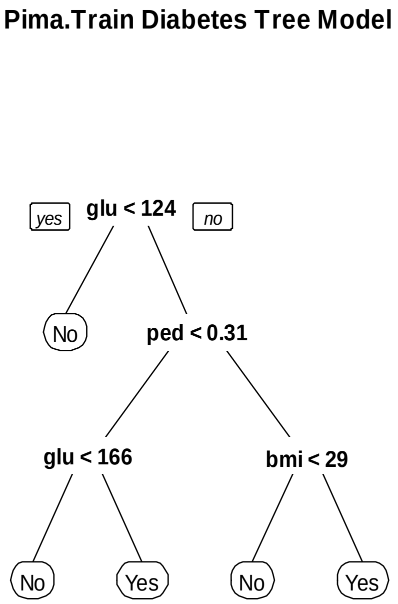

Pima.Train Diabetes Tree Model
yes
glu < 124
no
No
ped < 0.31
glu < 166
bmi < 29
No
Yes
No
Yes

Fig 9

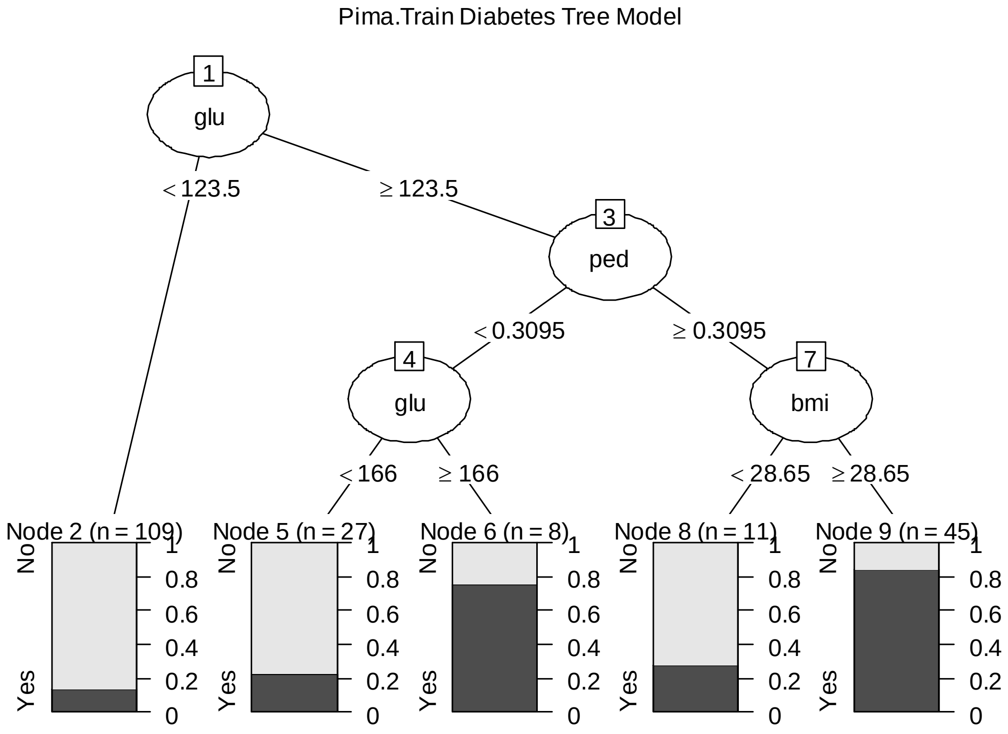

Pima.Train Diabetes Tree Model
1
glu
< 123.5
≥ 123.5
3
ped
< 0.3095
≥ 0.3095
4
glu
7
bmi
< 166
≥ 166
< 28.65
≥ 28.65
Node 2 (n = 109)
Node 5 (n = 27)
Node 6 (n = 8)
Node 8 (n = 11)
Node 9 (n = 45)
No
Yes
1
0.8
0.6
0.4
0.2
0

Fig 10

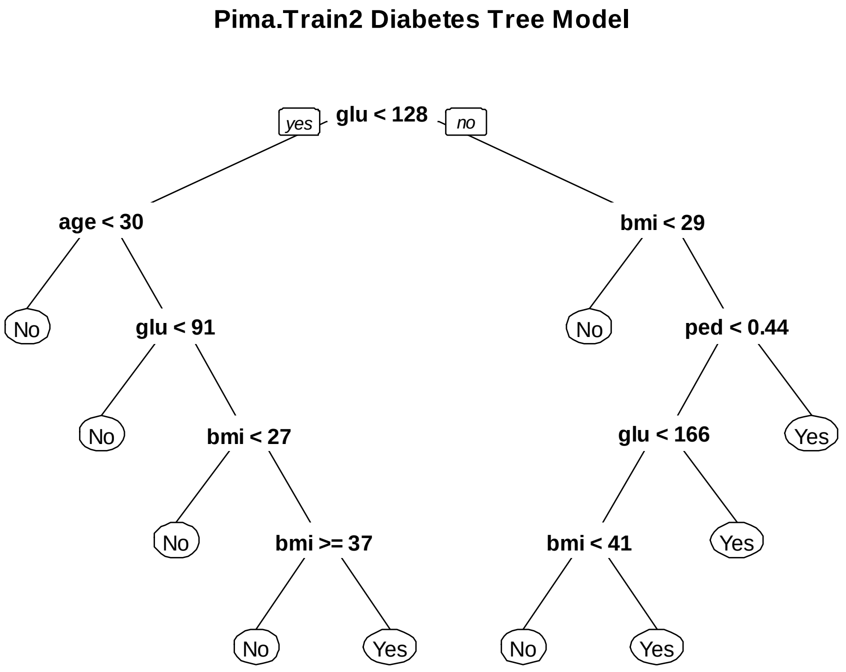

Pima.Train2 Diabetes Tree Model
yes
glu < 128
no
age < 30
bmi < 29
No
glu < 91
No
ped < 0.44
No
bmi < 27
glu < 166
Yes
No
bmi >= 37
bmi < 41
Yes
No
Yes
No
Yes

Fig 11

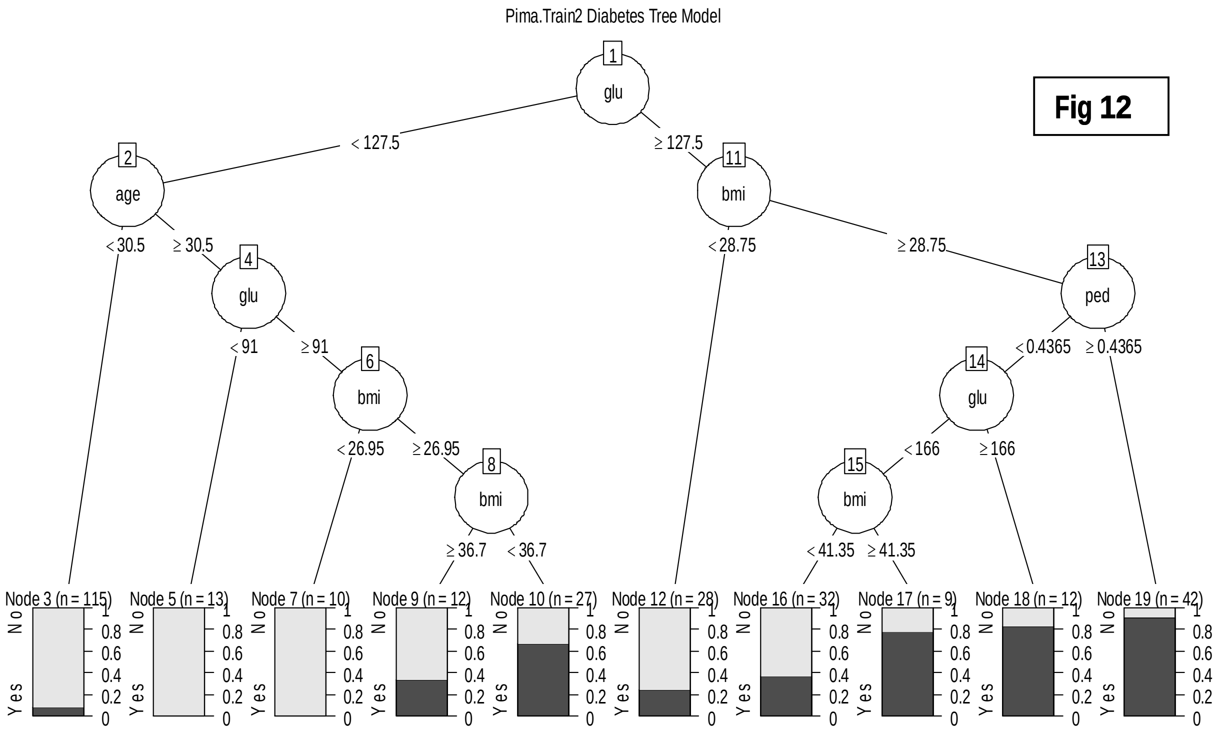

Pima.Train2 Diabetes Tree Model
1
glu
< 127.5
≥ 127.5
2
age
11
bmi
< 30.5
≥ 30.5
4
glu
< 28.75
≥ 28.75
13
ped
< 91
≥ 91
6
bmi
< 0.4365
≥ 0.4365
14
glu
< 26.95
≥ 26.95
8
bmi
< 166
≥ 166
15
bmi
≥ 36.7
< 36.7
< 41.35
≥ 41.35
Node 3 (n = 115)
Node 5 (n = 13)
Node 7 (n = 10)
Node 9 (n = 12)
Node 10 (n = 27)
Node 12 (n = 28)
Node 16 (n = 32)
Node 17 (n = 9)
Node 18 (n = 12)
Node 19 (n = 42)
No
Yes
1
0.8
0.6
0.4
0.2
0

Fig 12

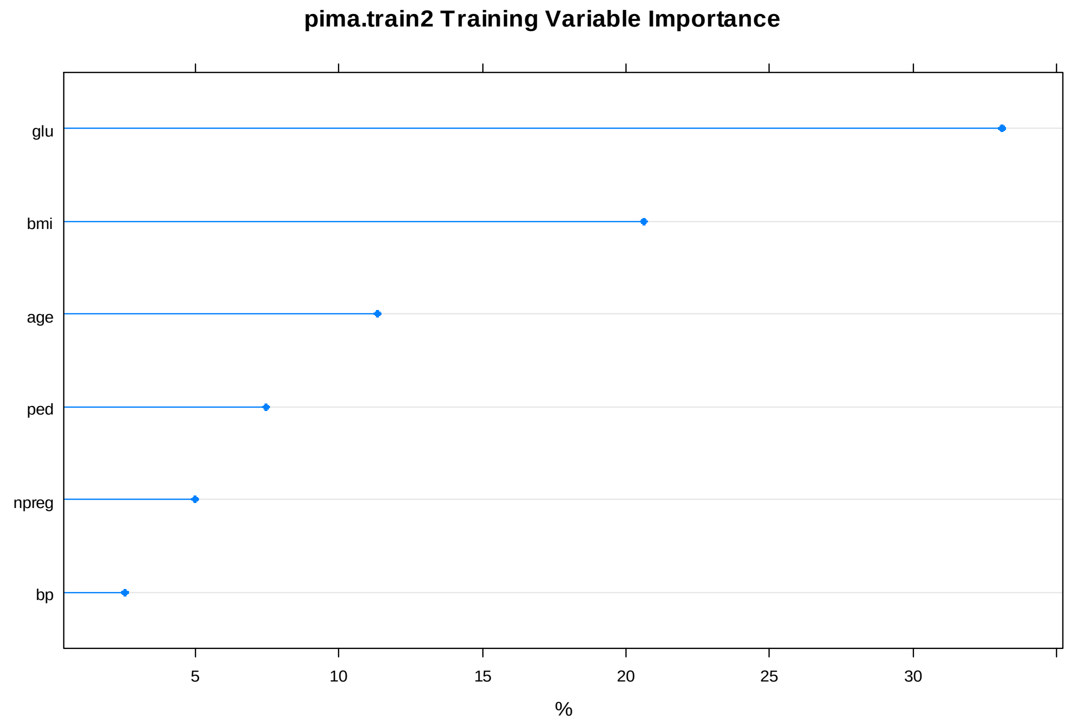

pima.train2 Training Variable Importance
glu
bmi
age
ped
npreg
bp
5
10
15
20
25
30
%

Fig 13

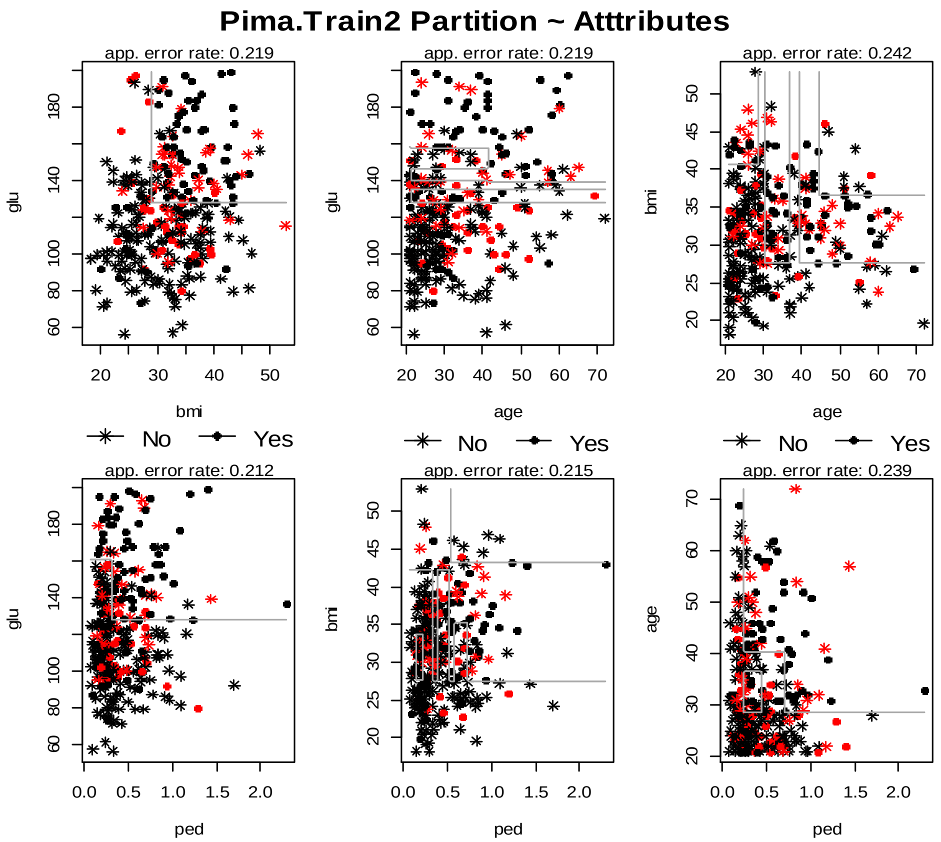

Pima.Train2 Partition ~ Atttributes
app. error rate: 0.219
app. error rate: 0.219
app. error rate: 0.242
glu
bmi
glu
age
bmi
age
No
Yes
app. error rate: 0.212
app. error rate: 0.215
app. error rate: 0.239
glu
ped
bmi
ped
age
ped

Fig 14

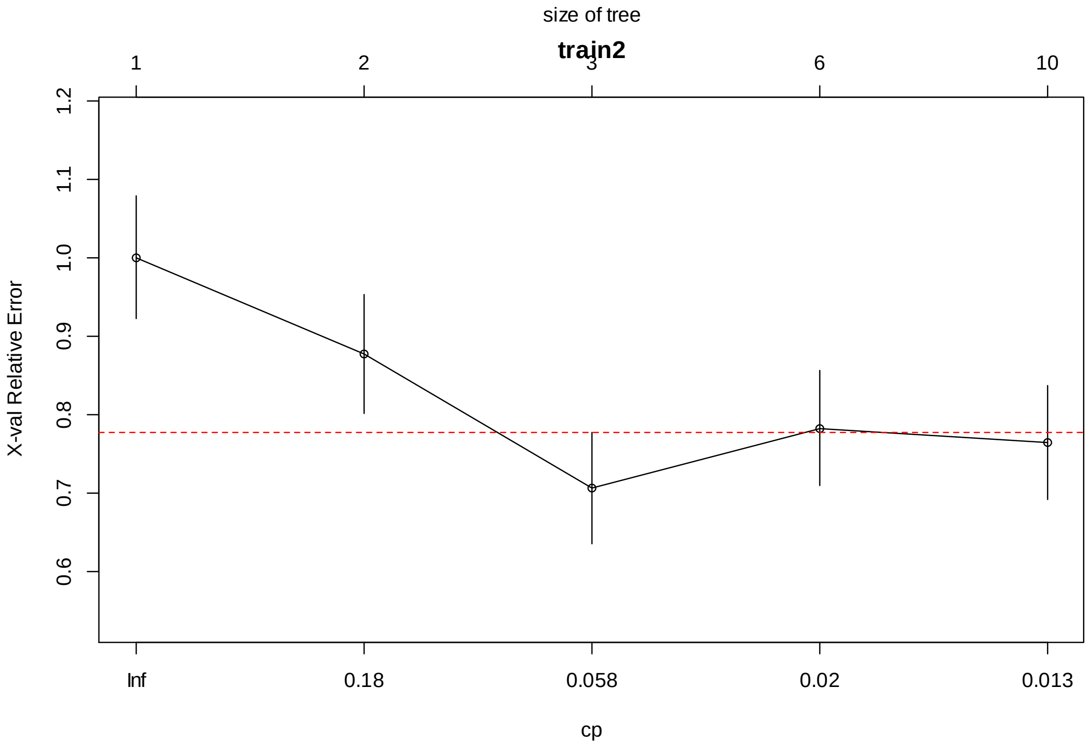


Fig 15

Pima.Train2 Diabetes Pruned Tree Mode

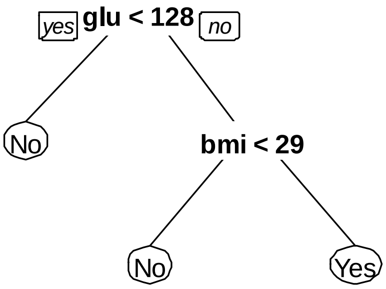


Fig 16

Pima.Train2 Diabetes Pruned Tree Mode

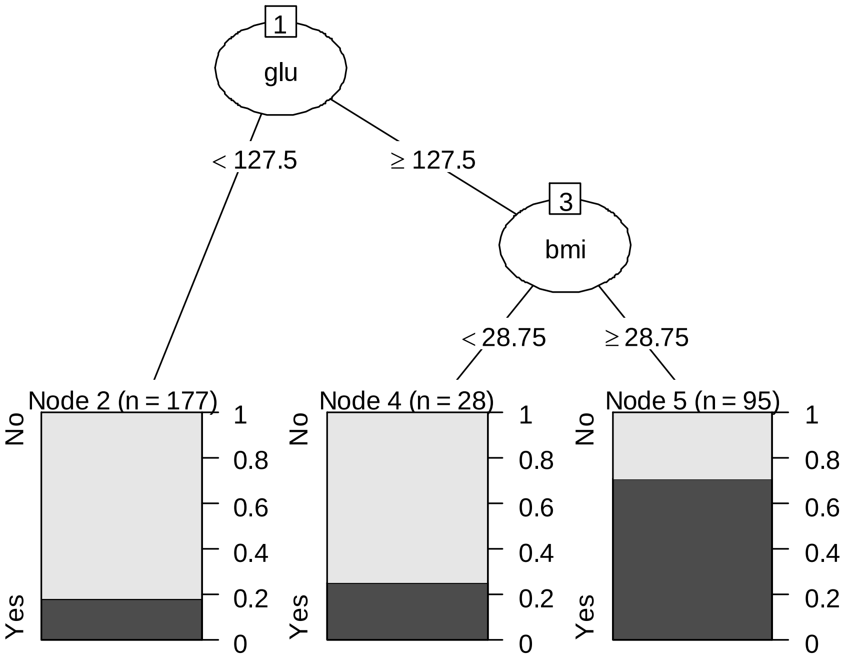
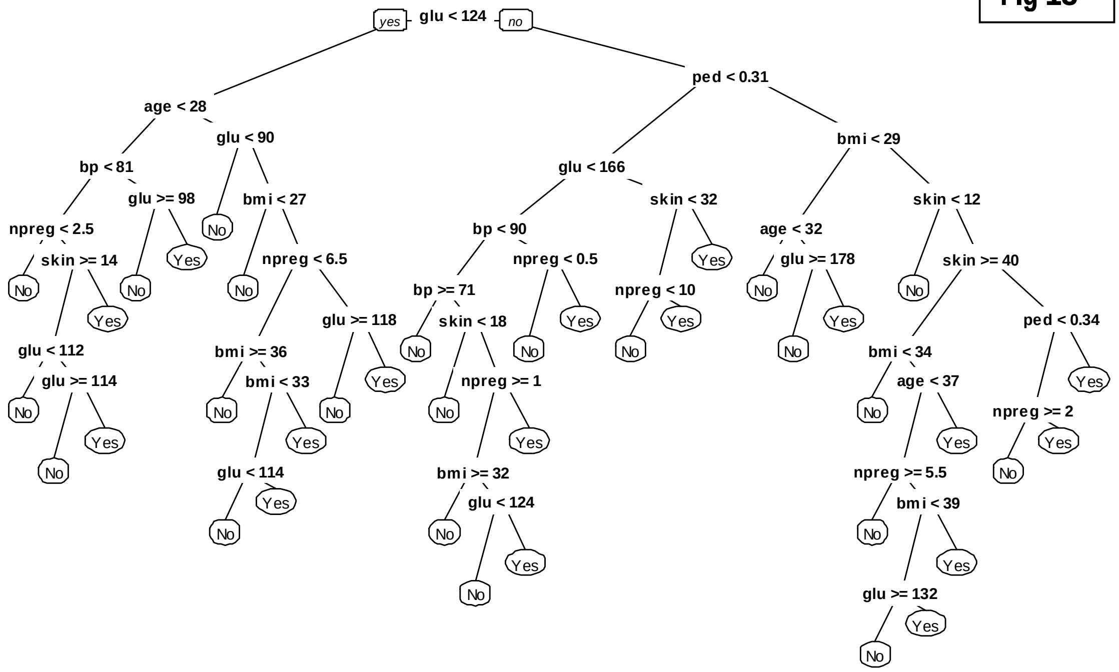


Fig 17

Pima.Train Diabetes Full Size Tree Model

Fig 18



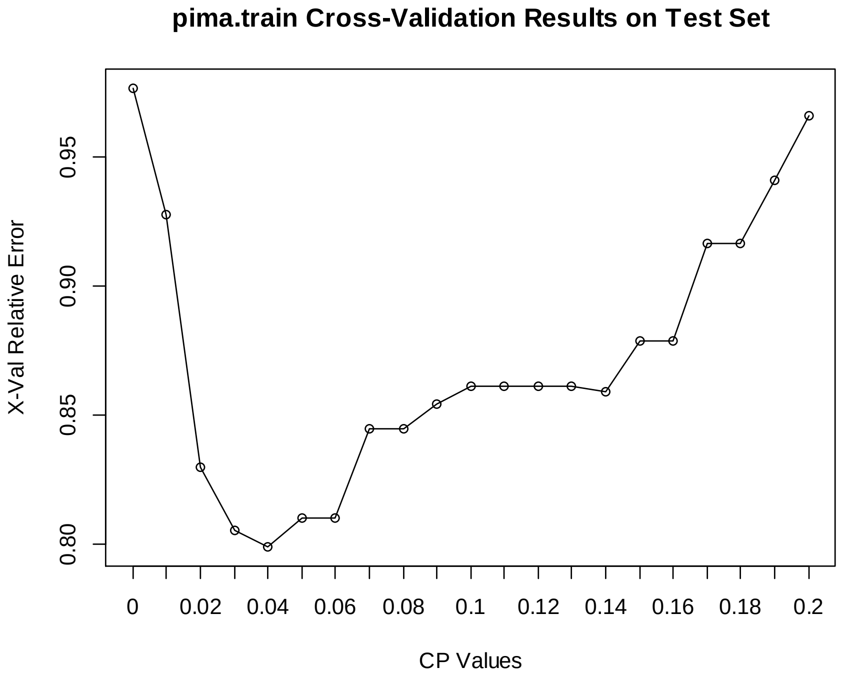

pima.train Cross-Validation Results on Test Set
X-Val Relative Error
0.80
0.85
0.90
0.95
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
CP Values

Fig 19

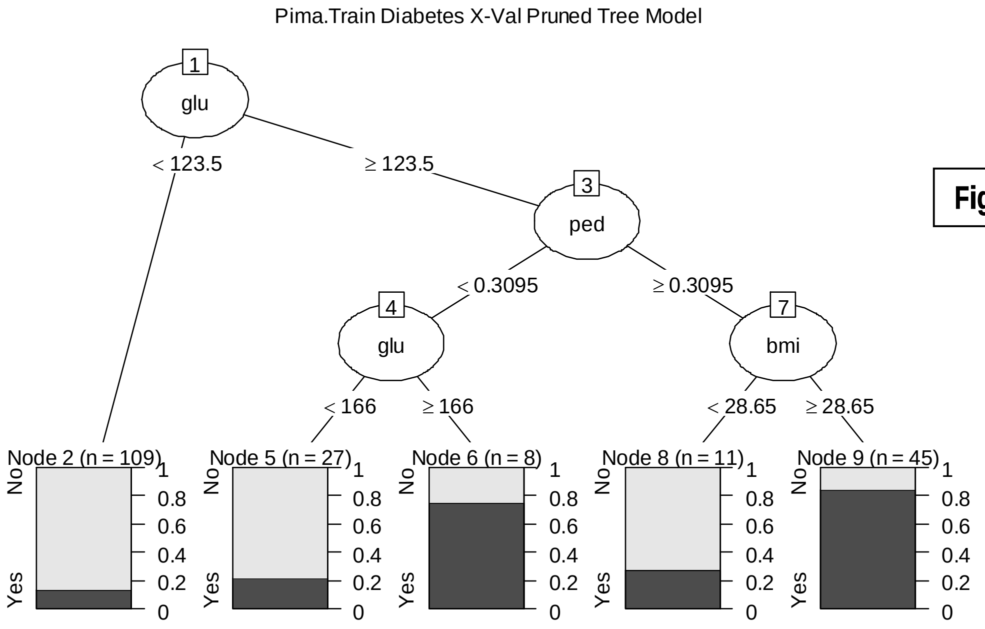

Pima.Train Diabetes X-Val Pruned Tree Model
1
glu
< 123.5
≥ 123.5
3
ped
< 0.3095
≥ 0.3095
4
glu
7
bmi
< 166
≥ 166
< 28.65
≥ 28.65
Node 2 (n = 109)
Node 5 (n = 27)
Node 6 (n = 8)
Node 8 (n = 11)
Node 9 (n = 45)
No
Yes
1
0.8
0.6
0.4
0.2
0

Fig 20

Pima.Train2 Diabetes Full Size Tree Model

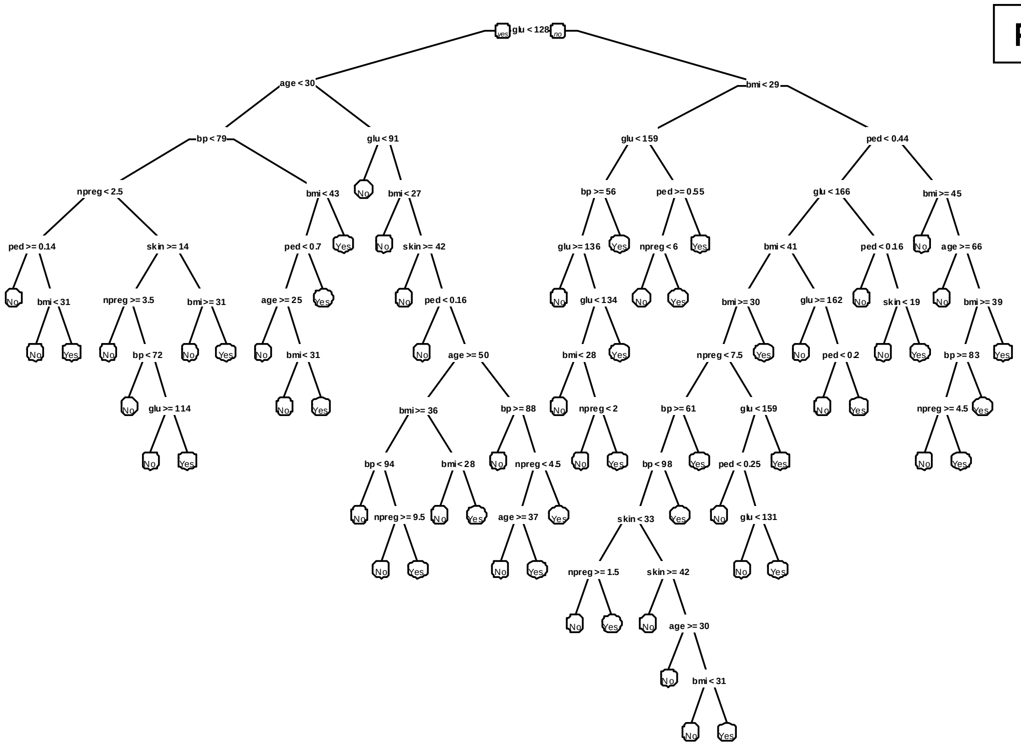


Fig 21

pima.train2 Cross-Validation Results on Test Set

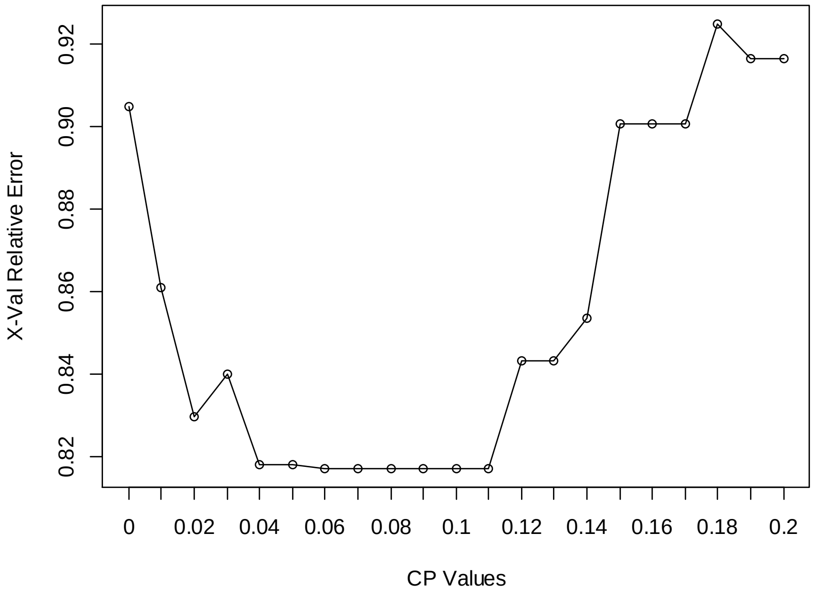


Fig 22

Pima.Train2 Diabetes X-Val Pruned Tree Model

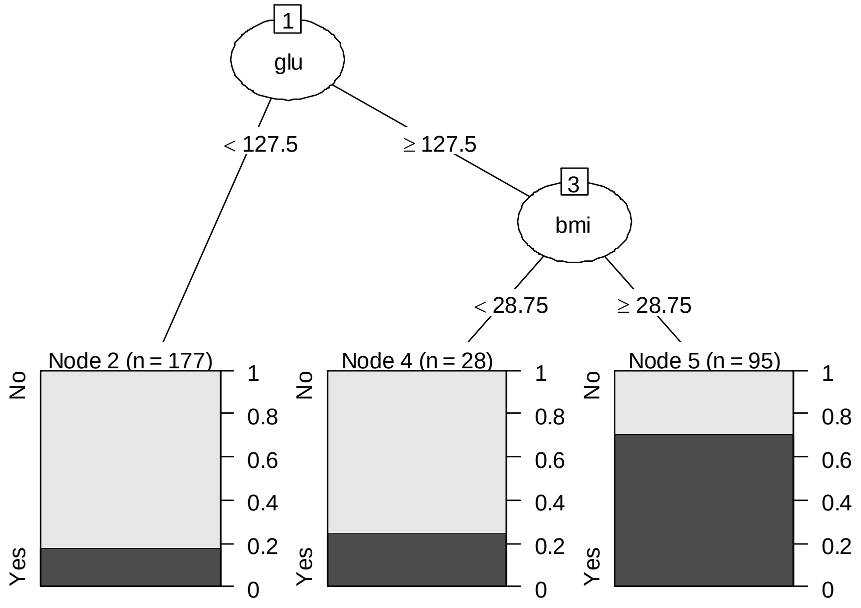


Fig 23

R Code

```
require(latticeExtra)
require(corrgram)
require(rpart)
require(rpart.plot)
require(partykit)
require(caret)
require(klaR)
require(pROC)
require(e1071)

## READ DATA
pima.train <-
  read.table("http://www.stats.ox.ac.uk/pub/PRNN/pima.tr", header=T)
pima.train2 <-
  read.table("http://www.stats.ox.ac.uk/pub/PRNN/pima.tr2", header=T)
pima.test <-
  read.table("http://www.stats.ox.ac.uk/pub/PRNN/pima.te", header=T)

## Attribute Information:
## 1. Number of times pregnant
## 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
## 3. Diastolic blood pressure (mm Hg)
## 4. Triceps skin fold thickness (mm)
## 5. Body mass index (weight in kg/(height in m)^2)
## 6. Diabetes pedigree function
## 7. Age (years)
## 8. Response variable: Diabetes (Yes or No)


#################################################################
###################### Exploratory Analysis #####################

# Merge Data and Finding Duplicates
pima.full <- rbind(pima.test,pima.train2)
which(duplicated(pima.full))

#Diabetes Frequency Barplot
ylim <- c(0, 1.2*max(as.numeric(table(pima.full[8]))))
xx <- barplot(table(pima.full[8]), width = 0.85, ylim = ylim, ylab ="Number of Cases")
text(x =xx, y = as.numeric(table(pima.full[8])), label = as.numeric(table(pima.full[8])),
     pos = 3, cex = 0.8, col = "red")
title("Barplot for Diabetes in Pima Indians Data")

#Attributes Boxplot
plot.new()
title(main = "\nPima Indians Data Attributes Boxplots", outer=T)

plot(bwplot(npreg ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Times pregnant"),
     split = c(1, 1, 4, 2), newpage = FALSE)

plot(bwplot(glu ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Glucose concentration"),
     split = c(2, 1, 4, 2), newpage = FALSE)

plot(bwplot(bp ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Blood pressure"),
     split = c(3, 1, 4, 2), newpage = FALSE)

plot(bwplot(skin ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Triceps skin thickness"),
     split = c(1, 2, 4, 2), newpage = FALSE)

plot(bwplot(bmi ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Body mass index"),
     split = c(2, 2, 4, 2), newpage = FALSE)
```

```
plot(bwplot(ped ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Diabetes pedigree"),
     split = c(3, 2, 4, 2), newpage = FALSE)

plot(bwplot(age ~ type, data = pima.full, pch="|",
            xlab = "Diabetes", ylab="Age"),
     split = c(4, 1, 4, 2), newpage = FALSE)

#Correlation Tests
(cormatrix <- cor(pima.full[1:7],pima.full[1:7], use="p"))
corrgram(pima.full, order=TRUE, lower.panel=panel.shade,
         upper.panel=panel.pie, text.panel=panel.txt,
         main ="Correlation Matrix Pima Indians Data")

cat("Correlation Matrix\n",file = "exploratory.csv")
cat( "\t", colnames(cormatrix), "\n", file = "exploratory.csv", sep = ",",append=T)
write.table(cormatrix, file = "exploratory.csv", sep = ",",append=T,
            qmethod = "double",col.names = F)

#Missing Values
(train2miss <- colSums(is.na(pima.full[1:6])))

cat("\n,", "Missing Values\n", file = "exploratory.csv", append = T)
write.table(train2miss, file = "exploratory.csv", sep = ",", col.names = F,
            qmethod = "double", append=T)


###################################################################
########Training with Complete Observations (pima.train)###########
set.seed(12345)
rp1 <- rpart(type ~ ., data=pima.train, method='class',
             control=NA)
rpart.plot(rp1, main = "Pima.Train Diabetes Tree Model")  #Better Plot
plot(as.party(rp1), main="Pima.Train Diabetes Tree Model") #Informative Plot


######Training pima.train Results
srp1 <- summary(rp1)
srp1$variable.importance
srp1$cptable

cat("pima.train Variable Importance\n", file = "pima.train.csv")
write.table(srp1$variable.importance, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


#Training pima.train Plots
dotplot(sort(srp1$variable.importance, decreasing = F), type=c("p","h"),xlab="%",
        main ="pima.train Training Variable Importance")

partimat(type ~ glu+bmi+age+ped, data=pima.train, method="rpart",
         imageplot=F, gs=c(pch=8,pch=16)[unclass(pima.train$type)],
         main="Pima.Train Partition ~ Atttributes")

legend(locator(1), xpd = TRUE, horiz = T, lwd=1, pch = c(8,16), bty='n', text.width=0.2,
       cex=0.9, legend=c("No","Yes"))


#CP Table
cat("\n", "pima.train CP Table\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(srp1$cptable), "\n", file = "pima.train.csv", sep = ",", append=T)
write.table(srp1$cptable, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

plotcp(rp1, col="red", lty=2, main="train")

crp1 <- confusionMatrix(pima.train[["type"]], predict(rp1, pima.train, type="class"))
crp1$table
rp1res <-
(cbind(crp1$overall["Accuracy"],cbind(sensitivity(crp1$table),specificity(crp1$table))))
colnames(rp1res) = c("Accuracy","Sensitivity","Specificity")
```

```
rownames(rp1res) = "Results"
print(rp1res)

cat("\n", "pima.train Prediction Matrix\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(crp1$table), "\n", file = "pima.train.csv", sep = ",", append=T)
write.table(crp1$table, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "pima.train Training Results\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(rp1res), "\n", file = "pima.train.csv", sep = ",", append=T)
write.table(rp1res, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


###Predicting test with complete observations (pima.train)
test1 <- confusionMatrix(pima.test[["type"]], predict(rp1, pima.test, type="class"))
test1$table
test1res <-
(cbind(test1$overall["Accuracy"],cbind(sensitivity(test1$table),specificity(test1$table))))
colnames(test1res) = c("Accuracy","Sensitivity","Specificity")
rownames(test1res) = "Results"
print(test1res)

cat("\n", "Prediction Matrix Test using pima.train\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(test1$table), "\n", file = "pima.train.csv", sep = ",", append=T)
write.table(test1$table, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with pima.train\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(test1res), "\n", file = "pima.train.csv", sep = ",", append=T)
write.table(test1res, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


##Pruning Training tree
pru1 <- prune(rp1, cp=rp1$cptable[which.min(rp1$cptable[,"xerror"]),"CP"])
rpart.plot(pru1, main = "Pima.Train Diabetes Pruned Tree Model")  #Better Plot
plot(as.party(pru1), main="Pima.Train Diabetes Pruned Tree Model") #Informative Plot


###Training set with pruned complete observations (pruned pima.train)
cprurp1 <- confusionMatrix(pima.train[["type"]], predict(pru1, pima.train, type="class"))
cprurp1$table
prurp1res <-
(cbind(cprurp1$overall["Accuracy"],cbind(sensitivity(cprurp1$table),specificity(cprurp1$table))))
colnames(prurp1res) = c("Accuracy","Sensitivity","Specificity")
rownames(prurp1res) = "Results"
print(prurp1res)

cat("\n", "Pruned pima.train Prediction Matrix\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(cprurp1$table), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(cprurp1$table, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Pruned pima.train Training Results\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(prurp1res), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(prurp1res, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


###Test set with pruned complete observations (pruned pima.train)
testpr1 <- confusionMatrix(pima.test[["type"]], predict(pru1, pima.test, type="class"))
testpr1$table
testpr1res <-
(cbind(testpr1$overall["Accuracy"],cbind(sensitivity(testpr1$table),specificity(testpr1$table))))
colnames(testpr1res) = c("Accuracy","Sensitivity","Specificity")
rownames(testpr1res) = "Results"
print(testpr1res)
```

```
cat("\n", "Prediction Matrix Test using Pruned pima.train\n", file = "pima.train.csv", append =
T)
cat( "\t", colnames(testpr1$table), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(testpr1$table, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with Pruned pima.train\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(testpr1res), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(testpr1res, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)



###################################################################
########Training with Incomplete Observations (pima.train2)########

set.seed(12345)
rp2 <- rpart(type ~ ., data=pima.train2, method='class', na.action = na.rpart,
             control=NA)
rpart.plot(rp2, main = "Pima.Train2 Diabetes Tree Model")  #Better Plot
plot(as.party(rp2), main="Pima.Train2 Diabetes Tree Model") #Informative Plot


#Training pima.train2 Results
srp2 <- summary(rp2)
srp2$variable.importance
srp2$cptable

cat("pima.train2 Variable Importance\n", file = "pima.train2.csv")
write.table(srp2$variable.importance, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


#Training pima.train2 Plots
dotplot(sort(srp2$variable.importance, decreasing = F), type=c("p","h"),xlab="%",
        main ="pima.train2 Training Variable Importance")

partimat(type ~ glu+bmi+age+ped, data=pima.train2, method="rpart",
         imageplot=F, gs=c(pch=8,pch=16)[unclass(pima.train2$type)],
         main="Pima.Train2 Partition ~ Atttributes")

legend(locator(1), xpd = TRUE, horiz = T, lwd=1, pch = c(8,16), bty='n', text.width=0.2,
       cex=0.9, legend=c("No","Yes"))


#CP Tables
cat("\n", "pima.train2 CP Table\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(srp2$cptable), "\n", file = "pima.train2.csv", sep = ",", append=T)
write.table(srp2$cptable, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

plotcp(rp2, col="red", lty=2, main="train2")

crp2 <- confusionMatrix(pima.train2[["type"]], predict(rp2, pima.train2, type="class"))
crp2$table
rp2res <-
(cbind(crp2$overall["Accuracy"],cbind(sensitivity(crp2$table),specificity(crp2$table))))
colnames(rp2res) = c("Accuracy","Sensitivity","Specificity")
rownames(rp2res) = "Results"
print(rp2res)

cat("\n", "pima.train2 Prediction Matrix\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(crp2$table), "\n", file = "pima.train2.csv", sep = ",", append=T)
write.table(crp2$table, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "pima.train2 training Results\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(rp2res), "\n", file = "pima.train2.csv", sep = ",", append=T)
write.table(rp2res, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)
```

```
###Predicting test with incomplete observations (pima.train2)
test2 <- confusionMatrix(pima.test[["type"]], predict(rp2, pima.test, type="class"))
test2$table
test2res <-
(cbind(test2$overall["Accuracy"],cbind(sensitivity(test2$table),specificity(test2$table))))
colnames(test2res) = c("Accuracy","Sensitivity","Specificity")
rownames(test2res) = "Results"
print(test2res)

cat("\n", "Prediction Matrix Test using pima.train2\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(test2$table), "\n", file = "pima.train2.csv", sep = ",", append=T)
write.table(test2$table, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with pima.train2\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(test2res), "\n", file = "pima.train2.csv", sep = ",", append=T)
write.table(test2res, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


##Pruning Test tree2
pru2 <- prune(rp2, cp=rp2$cptable[which.min(rp2$cptable[,"xerror"]),"CP"])
rpart.plot(pru2, main = "Pima.Train2 Diabetes Pruned Tree Mode")  #Better Plot
plot(as.party(pru2), main="Pima.Train2 Diabetes Pruned Tree Mode") #Informative Plot


###Training set with pruned complete observations (pruned pima.train2)
cprurp2 <- confusionMatrix(pima.train2[["type"]], predict(pru2, pima.train2, type="class"))
cprurp2$table
prurp2res <-
(cbind(cprurp2$overall["Accuracy"],cbind(sensitivity(cprurp2$table),specificity(cprurp2$table))))
colnames(prurp2res) = c("Accuracy","Sensitivity","Specificity")
rownames(prurp2res) = "Results"
print(prurp2res)

cat("\n", "Pruned pima.train2 Prediction Matrix\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(cprurp2$table), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(cprurp2$table, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Pruned pima.train2 Training Results\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(prurp2res), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(prurp2res, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)


###Test set with pruned incomplete observations (pruned pima.train2)
testpr2 <- confusionMatrix(pima.test[["type"]], predict(pru2, pima.test, type="class"))
testpr2$table
testpr2res <-
(cbind(testpr2$overall["Accuracy"],cbind(sensitivity(testpr2$table),specificity(testpr2$table))))
colnames(testpr2res) = c("Accuracy","Sensitivity","Specificity")
rownames(testpr2res) = "Results"
print(testpr2res)

cat("\n", "Prediction Matrix Test using Pruned pima.train2\n", file = "pima.train2.csv", append =
T)
cat( "\t", colnames(testpr2$table), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(testpr2$table, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with Pruned pima.train2\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(testpr2res), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(testpr2res, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)
```

```
##################################################################
################## Test Cross Validation ##########################
## Using pima.train

##Create a full tree from train
set.seed(12345)
rcontrol <- rpart.control(minsplit = 2, cp=0, xval = 10)
full1 <- rpart(type ~ ., data=pima.train, method='class',
               control=rcontrol)
rpart.plot(full1, main = "Pima.Train Diabetes Full Size Tree Model")  #Better Plot

##10-Fold X-Val on CP

set.seed(12345)
cvf1 <- createFolds(pima.train$type, k=10)
control <- rpart.control(minsplit = 2, cp=0, xval = 10)
cvres1<-matrix(0,21,10)

for (i in 1:10)
{
data1<-pima.train[-cvf1[[i]],]
set.seed(12345)
cvrp1 <- rpart(type ~., data=data1, method="class", control=rcontrol)
cvrp1$cptable
z=0
for (j in 0:20/100)
{
pr1<- prune(cvrp1, cp=j)
x <- nrow(pr1$cptable)
y <- (pr1$cptable)[x,4]
z=z+1
cvres1[z,i] <- round(y,4)
}
}

colnames(cvres1)= c(1:10)
rownames(cvres1)= c(0:20/100)
plot(rowMeans(cvres1),  type="o", xaxt='n', xlab= "CP Values",
     ylab = "X-Val Relative Error", main = "pima.train Cross-Validation Results on Test Set")
axis(1,at=c(1:21), labels=rownames(cvres1))


#Pruning pima.train full tree
(min1 <- max((which(rowMeans(cvres1)== min(rowMeans(cvres1)))-1)/100))
fullpru1 <- prune(full1, cp=min1)
plot(as.party(fullpru1), main="Pima.Train Diabetes X-Val Pruned Tree Model")


#Predicting Pruned pima.train on Test Set
finalpru1 <- confusionMatrix(pima.test[["type"]], predict(fullpru1, pima.test, type="class"))
finalpru1$table
finalpru1res <-
(cbind(finalpru1$overall["Accuracy"],cbind(sensitivity(finalpru1$table),specificity(finalpru1$tab
le))))
colnames(finalpru1res) = c("Accuracy","Sensitivity","Specificity")
rownames(finalpru1res) = "Results"
print(finalpru1res)

cat("\n", "Prediction Matrix Test using X-Val Pruned Tree\n", file = "pima.train.csv", append =
T)
cat( "\t", colnames(finalpru1$table), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(finalpru1$table, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with X-Val Pruned tree\n", file = "pima.train.csv", append = T)
cat( "\t", colnames(finalpru1res), "\n", file = "pima.train.csv", sep = ",",append=T)
write.table(finalpru1res, file = "pima.train.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)
```

```
######## Using pima.train2
##Create a full tree from train2
set.seed(12345)
rcontrol <- rpart.control(minsplit = 2, cp=0, xval = 10)
full2 <- rpart(type ~ ., data=pima.train2, method='class',
               control=rcontrol)
rpart.plot(full2, main = "Pima.Train2 Diabetes Full Size Tree Model")  #Better Plot

##10-Fold X-Val on CP
set.seed(12345)
cvf2 <- createFolds(pima.train2$type, k=10)
control <- rpart.control(minsplit = 2, cp=0, xval = 10)
cvres2<-matrix(0,21,10)

for (i in 1:10)
{
  data2<-pima.train2[-cvf2[[i]],]
  set.seed(12345)
  cvrp2 <- rpart(type ~., data=data2, method="class", control=rcontrol)
  cvrp2$cptable
  z=0
  for (j in 0:20/100)
  {
    pr2<- prune(cvrp2, cp=j)
    x <- nrow(pr2$cptable)
    y <- (pr2$cptable)[x,4]
    z=z+1
    cvres2[z,i] <- round(y,4)
  }
}

colnames(cvres2)= c(1:10)
rownames(cvres2)= c(0:20/100)
plot(rowMeans(cvres2),  type="o", xaxt='n', xlab= "CP Values",
     ylab = "X-Val Relative Error", main = "pima.train2 Cross-Validation Results on Test Set")
axis(1,at=c(1:21), labels=rownames(cvres2))


#Pruning pima.train2 full tree
(min2 <- max((which(rowMeans(cvres2)== min(rowMeans(cvres2)))-1)/100))
fullpru2 <- prune(full2, cp=min2)
plot(as.party(fullpru2), main="Pima.Train2 Diabetes X-Val Pruned Tree Model")


#Predicting Pruned pima.train2 on Test Set
finalpru2 <- confusionMatrix(pima.test[["type"]], predict(fullpru2, pima.test, type="class"))
finalpru2$table
finalpru2res <-
(cbind(finalpru2$overall["Accuracy"],cbind(sensitivity(finalpru2$table),specificity(finalpru2$tab
le))))
colnames(finalpru2res) = c("Accuracy","Sensitivity","Specificity")
rownames(finalpru2res) = "Results"
print(finalpru2res)

cat("\n", "Prediction Matrix Test using X-Val Pruned Tree 2\n", file = "pima.train2.csv", append
= T)
cat( "\t", colnames(finalpru2$table), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(finalpru2$table, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)

cat("\n", "Test results with X-Val Pruned tree 2\n", file = "pima.train2.csv", append = T)
cat( "\t", colnames(finalpru2res), "\n", file = "pima.train2.csv", sep = ",",append=T)
write.table(finalpru2res, file = "pima.train2.csv", sep = ",",
            qmethod = "double", append = T, col.names=F)
```