Contraceptive Method Choice

Yiming Chen 2017/5/2

SUMMARY:

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. It is posted on UCI data mining websites. The samples are married women. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

It has 9 predict attributes and 1 response attribute. most of the predict attributes are defined by categories(e.g. from 1 to 4, 1 is low, 4 is high), and it is a classification case. No missing values.

I address several questions: which method could have a best predict accuracy?

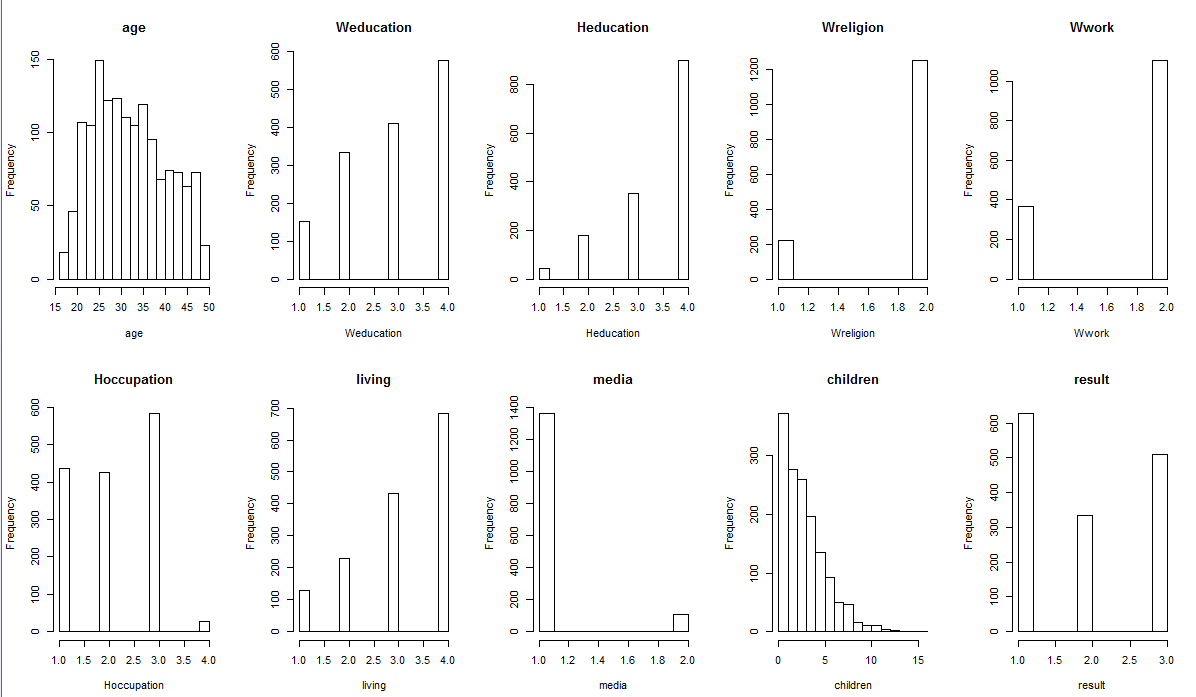
what is the variable importance? Could we combine 3 class categories of response values into two classes to improve prediction performance?

The main results are: when we want to detect "no use, long-term and short-term methods", random forest is the best model. But we could see the result that short term and long term is very hard to split. when we want to detect" no use and use", boosting is the best model. And the predict accuracy is enhanced. Then, if we see the classification plot, we know less children and elder than 35 is the main reasons lead to "no use". it means when people go through middle age, they start to think about raise a baby and their marriage maybe get stable.

INTRODUCTION:

Most of attributes are classified by factor. Women's religion, work, and media use are unbalanced and they only have two categories. Women’s education, living standard, husband's education and husband's occupation is relatively scattered and they have four categories. Age and children are both integers, so they have a bigger range for values. The result has three categories(no use/short term/long term). Proportion is :no use: short: long=42.7%: 22.6%: 34.7%.

Because the data is collected from a survey, it might have subjective errors, such as survey design limitation, small coverage, or someone refuses to participate. The following graphs are the distribution of attributes.



METHODS:

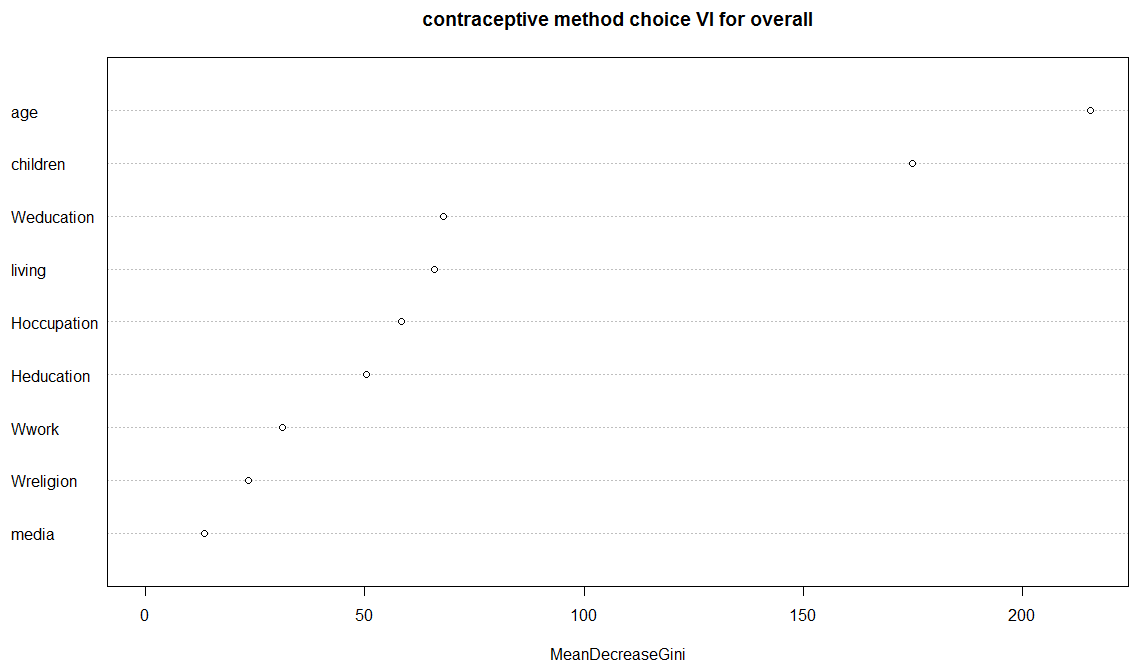
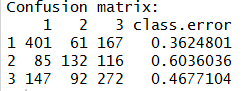
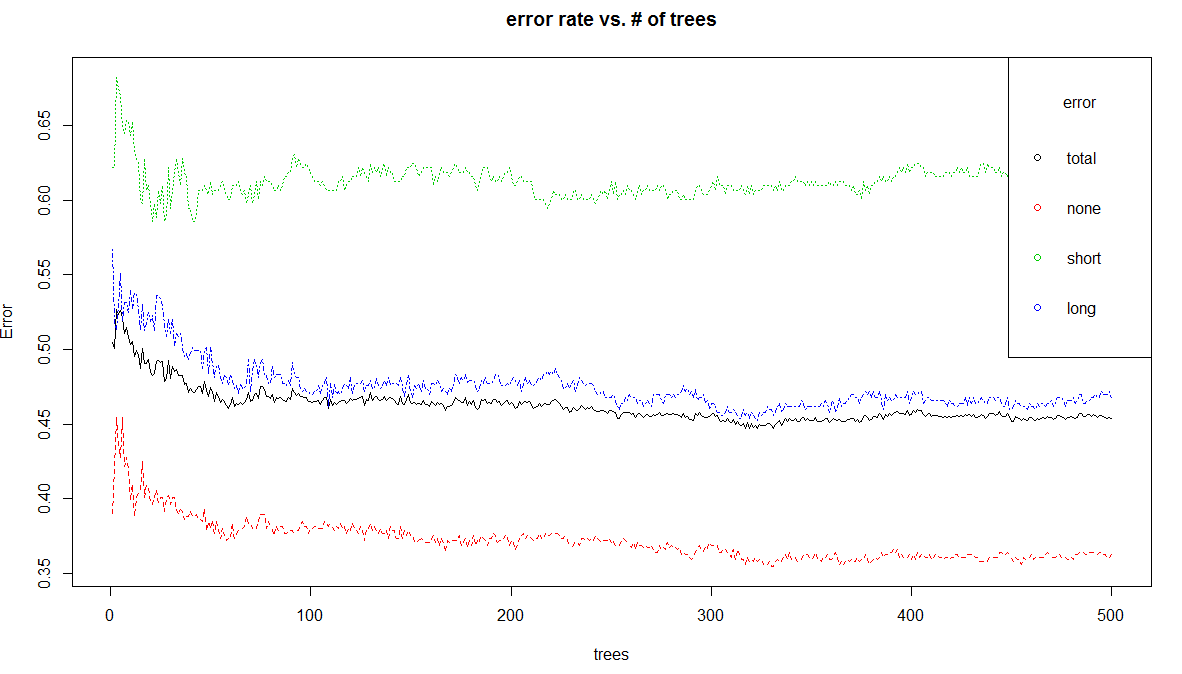
The experimenter designed this survey and created several questions as attributes related to the response variable. The definition of long term includes Copper Intrauterine Device, Hormonal Intrauterine system, Birth Control Implant, Hormonal Injection, Sterilization, etc. The definition of short term includes The Pill, Male Condom, Contraceptive Patch, Vaginal Ring, etc.

I use R studio to build analytic models and to run them to get results. Based on them, I want to improve prediction accuracy and choose one model to use for the new observations. Also I could give the classification rules to the pharmacy or medical companies to increase awareness about who needs short term and long term contraceptives, or none at all. Before I apply the data into the model, I make a conversion, and specify them with three categories by use vs. no use; long term vs. short term; no use vs. long term vs. short term.

I use random forest ,boosting and classification tree as candidate models.

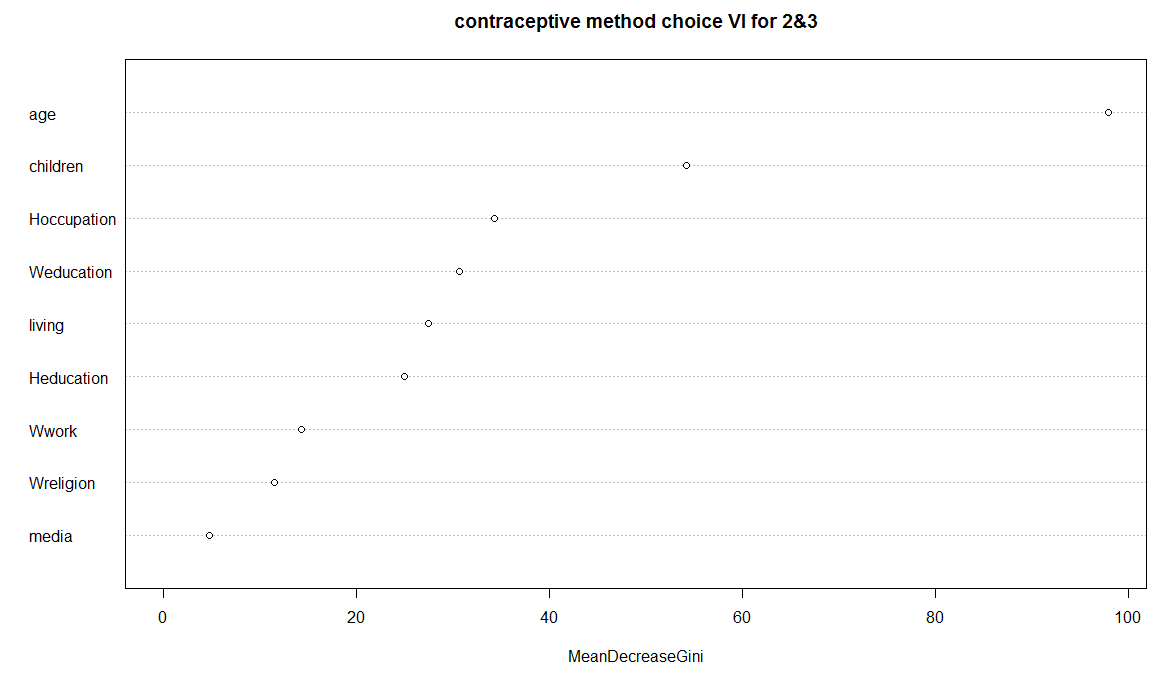
RESULTS:

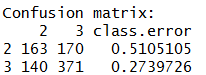
**Random Forest**

 OOB estimate of error rate: 45.35%

First, I exploit random forest on no use vs. long term vs. short term.

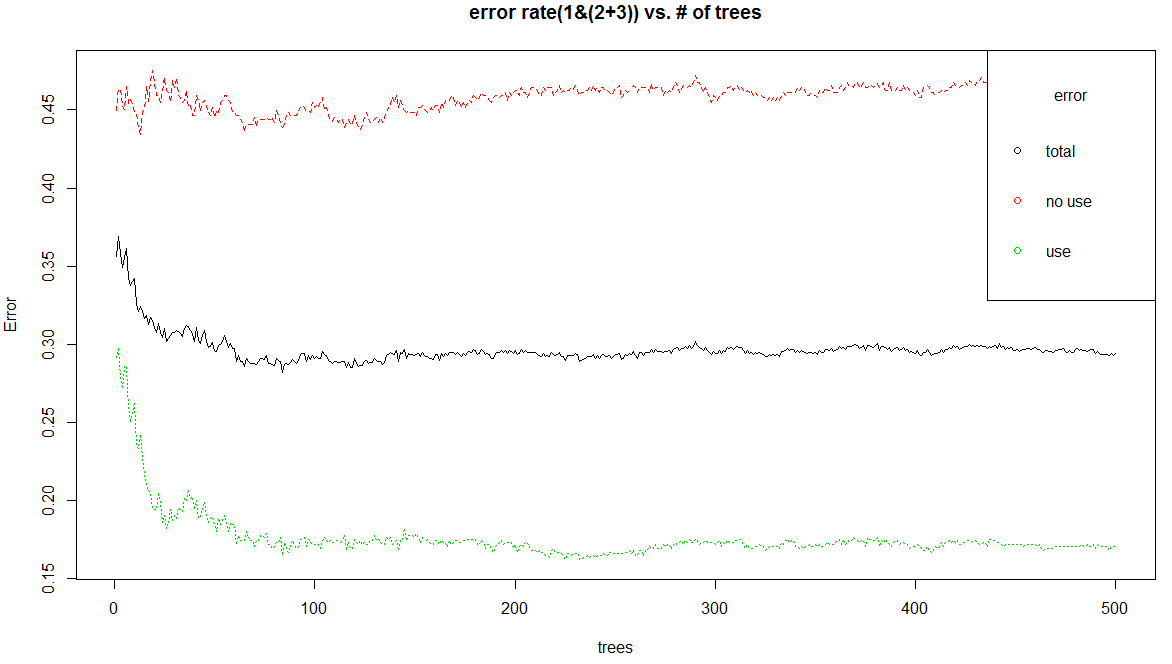
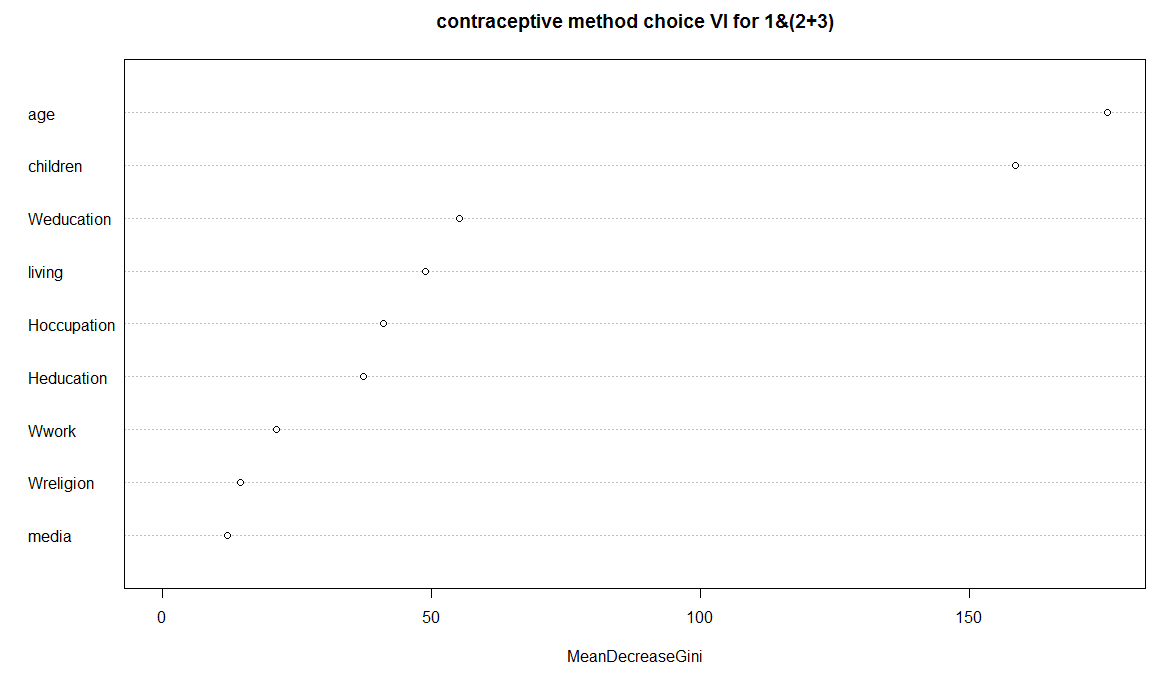
In the outcome, “No use” is easy to detect because of its significantly lower error rate.“Short term” and “long term” have a higher error rate, so only use these two categories to do another random forest. Here I use decreasing node impurity to run variable importance code since it is a classification case.

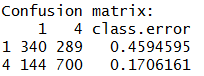
 

OOB estimate of error rate: 36.73%

Through the results, "short term" error rate is still very high. I start to think about the reason. Single class error rate may be related with this class proportion in the data set. The short term error rate is less accurate because a smaller sample was surveyed than long term. To lower the overall error rate of the data, the prediction accuracy is better with a larger sample size. The variable importance order changes. It is hard to get good prediction accuracy in short term, Maybe it also because when woman use short term contraceptives,

it mostly happens randomly without any plan. The attributes designed here is not the main reason to determine whether they could lead to use a short term product. Then I apply random forest into use vs. no use dataset.

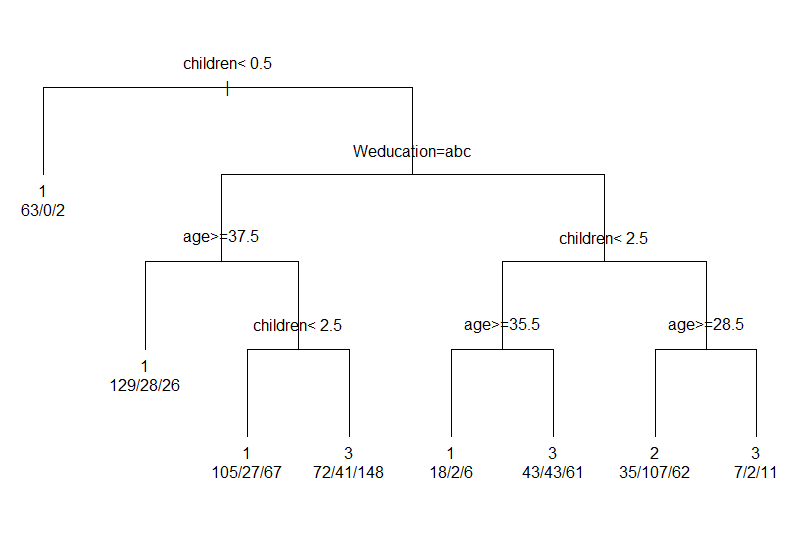


OOB estimate of error rate: 29.4%

To compare this outcome, the overall error rate goes down obviously. And to see the difference in the variable importance, no use& long term& short term is same with use & no use: it means the algorithm that classifies observations is the same. If you don't care about exact short term or long term, combining use and no use will increase estimate accuracy.

**CLASSIFICATION TREE** **(10-fold cross validation):**

I stratified sampling: 75% training data, 25% test data. Each class in training and test data has original proportion.(42.7%: 22.6%: 34.7%)

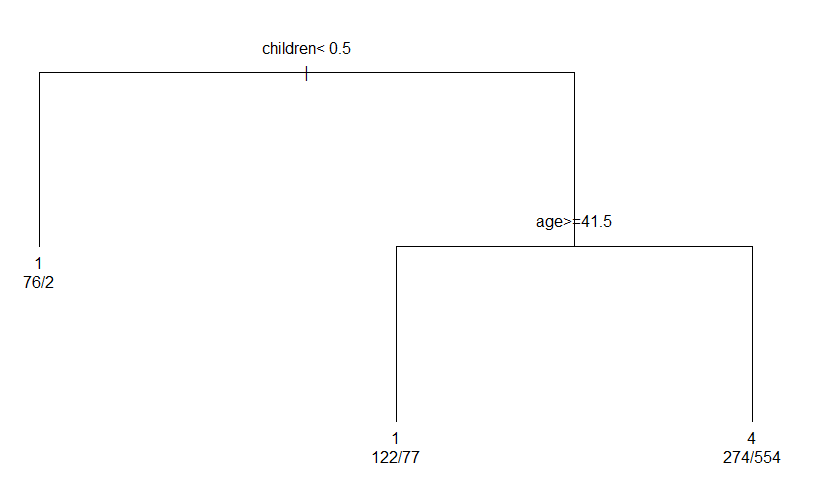


For no use& long term& short term:

Test error rate in rpart :0.4375

Test error rate in randomforest:

0.4320652.



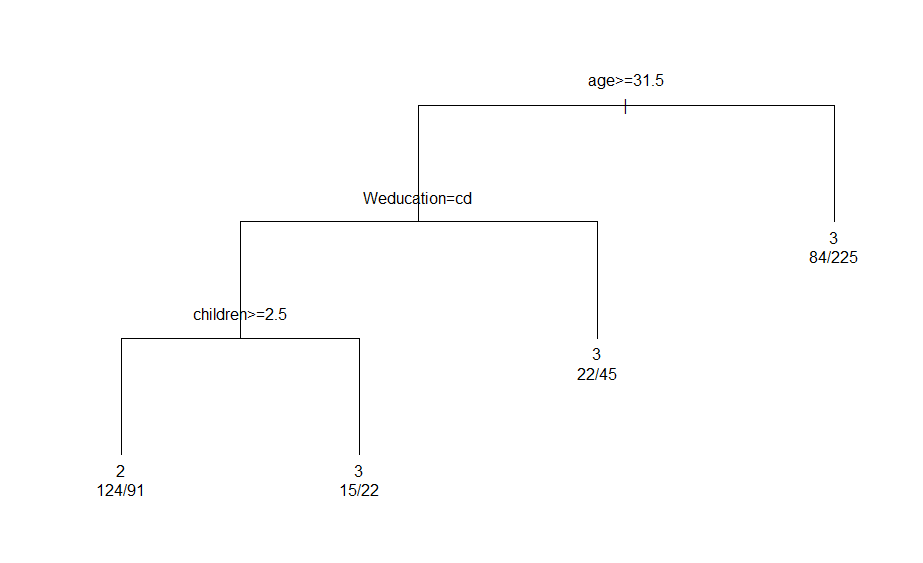
For use& no use:

Test error rate in rpart:

0.3641304

Test error rate in randomforest:

0.3179348



For long term& short term:

Test error rate in rpart:

0.3101852

Test error rate in randomforest:

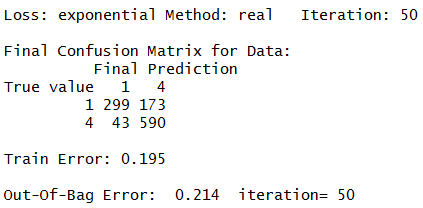
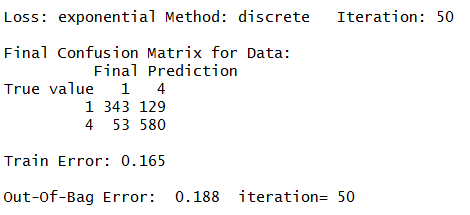
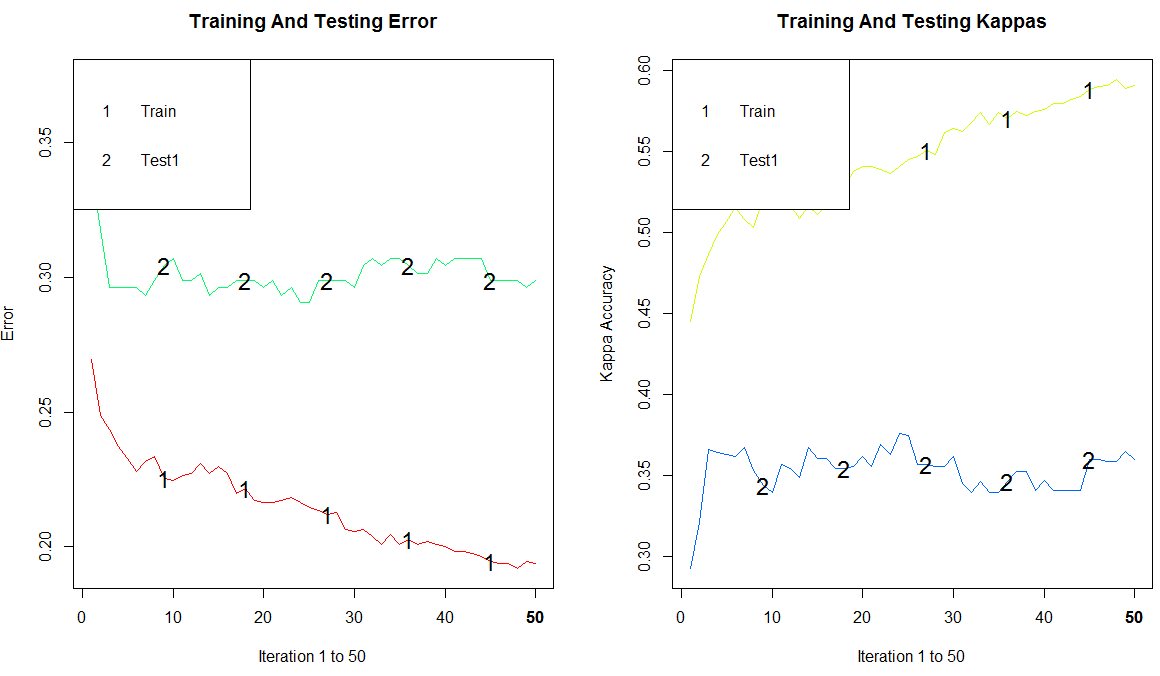
0.2916667

through these three result, we will choose random forest as best model to make a prediction due to low error rate. These three classification trees show different rules and I will make a summary in later conclusion. But the limitation of the tree is, every time when you run the code you will got a different tree splitting result and corresponding a different error rate. It is not fair just rely on one tree but it is more interpretable than ensemble model.

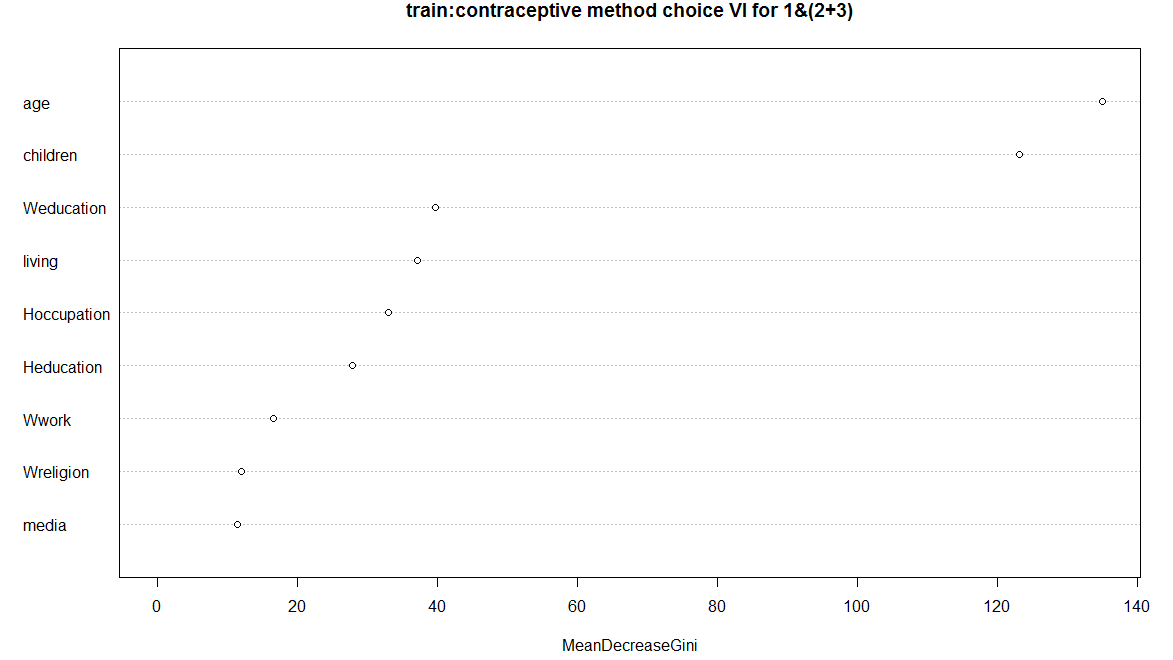
**BOOSTING:**

Because boosting method could only apply for two categories response variable.

I use this model into use& no use data and also stratified sampling. I use real adaboost and discrete adaboost.

Test error rate in discrete adaboost:0.2907609.It is better than real adaboost. We see discrete model is better because the real adaboost can be numerically unstable.



The left side is for random forest(varImpPlot). They have different function to present variable importance and the result has significant difference. But only through these two plots we cannot see the underlying influence on them.

**DISCUSSION:**

In random forest, the advantage is : Good performance on unbalanced data. Not sensitive to multiple collinearity. De-correlating the tree. Don't care about the outliers. The weakness is : variable selection bias problem. The attributes that have more levels tend to affect model severely.

So, to solve this problem, I try to use RandomForestRSC to eliminates variable selection bias, and use minimal depth to calculate variable importance. The overall error rate is 47.45% .Use “rfsrc” function, it doesn’t help with overall error rate compared with regular RF even I choose more trees， I choose 5000 trees in "rfsrc" and 500 trees in regular model. Maybe because it doesn’t has severely variable selection bias in this case. The variable importance in minimal depth is children->age->women's education->husband's occupation. To compare this with the above result, we choose the overlap part. We know children, age, women's education, husband's occupation is the most common variables used in split.

**CONCLUSION:**

Boosting is the best predict model for use& no use. Random forest is the best predict model for no use& long term& short term. Ensemble models have better prediction accuracy if you don't care about the interpretable splitting rules. Otherwise, to see more detail information use classification tree. OOB values are different from predict error rate. Variable importance sequence changed a lot from model to model.

Here I only summarize the splitting rules that are under higher accuracy rates. So from the classification trees , we conclude that less children and older than 35 will tend to “no use”. They want to get pregnant. It means people don’t want to raise baby when they are young. When comparing short term and long term,

we assume short term contraceptives are used because women still want a chance to have a baby in the near future. The rule is greater than 32, woman’s education is high(may indicate high salary and economic independence), children greater than 3, they tend to use short term. It means when women have economic independence and they already have more children, it indicates they might be able to raise more children based on this condition.

REFERANCE:

1.dataset and variable definitions are from UCI repository.

http://archive.ics.uci.edu/ml/

**Appendix:**

**First, install all the packages needed.**

install.packages("caret")

library(caret)

install.packages("randomForest")

library(randomForest)

library(rpart)

library("ggplot2")

install.packages("party")

library(party)

install.packages("randomForestSRC")

library(randomForestSRC)

install.packages("ada")

library(ada)

install.packages("arules")

**Before do the histograms, transform the data to the numerical type.**

cmc$result<-as.numeric(cmc$result)

cmc$Weducation<-as.numeric(cmc$Weducation)

cmc$Heducation<-as.numeric(cmc$Heducation)

cmc$Wreligion<-as.numeric(cmc$Wreligion)

cmc$Wwork<-as.numeric(cmc$Wwork)

cmc$Hoccupation<-as.numeric(cmc$Hoccupation)

cmc$living<-as.numeric(cmc$living)

cmc$media<-as.numeric(cmc$media)

cmc$age<-as.numeric(cmc$age)

hist(cmc$age,xlab="age",main="age")

hist(as.numeric(cmc$Weducation),xlab="Weducation",main="Weducation")

hist(as.numeric(cmc$Heducation),xlab="Heducation",main="Heducation")

hist(as.numeric(cmc$Wreligion),xlab="Wreligion",main="Wreligion")

hist(as.numeric(cmc$Wwork),xlab="Wwork",main="Wwork")

hist(as.numeric(cmc$Hoccupation),xlab="Hoccupation",main="Hoccupation")

hist(as.numeric(cmc$living),xlab="living",main="living")

hist(as.numeric(cmc$media),xlab="media",main="media")

hist(cmc$children,xlab="children",main="children")

hist(as.numeric(cmc$result),xlab="result",main="result")

**Random forest:**

fit.a<-randomForest(cmc$result~., data=cmc,ntree=500,keep.forest=TRUE,importance=TRUE)

varImp(fit.a)

varImpPlot(fit.a,type=2,main ="contraceptive method choice VI for overall")

plot(fit.a, main="error rate vs. # of trees")

legend("topright",c("total","none","short","long"),col=c("1","2","3","4"),pch=1,title="error")

**only use data by long term& short term, also transform the response variable as numeric type at first:**

cmc$result<-as.numeric(cmc$result)

cmm<-cmc[cmc$result==2|cmc$result==3,]

cmm$result<-as.factor(cmm$result)

fit.b<-randomForest(cmm$result~., data=cmm,ntree=500,keep.forest=TRUE,importance=TRUE)

varImp(fit.b,type=2)

varImpPlot(fit.b,type=2,main ="contraceptive method choice VI for 2&3")

plot(fit.b, main="error rate(2&3) vs. # of trees")

legend("topright",c("total","short","long"),col=c("1","2","3"),pch=1,title="error")

**define data by use& no use, also transform the response variable as numeric type at first:**

cmb<-cmc

cmb$result[which(cmb$result=="2")]<-4

cmb$result[which(cmb$result=="3")]<-4

cmb$result<-as.factor(cmb$result)

fit.c<-randomForest(cmb$result~., data=cmb,ntree=500,keep.forest=TRUE,importance=TRUE)

varImp(fit.c,type=2)

varImpPlot(fit.c,type=2,main ="contraceptive method choice VI for 1&(2+3)")

plot(fit.c, main="error rate(1&(2+3)) vs. # of trees")

legend("topright",c("total","no use","use"),col=c("1","2","3"),pch=1,title="error")

**do RandomForestSRC function, it needs to define a data frame at first:**

cmc$result<-as.factor(cmc$result)

abc<-data.frame(x1=cmc$age,x2=cmc$Weducation,x3=cmc$Heducation,x4=cmc$children,x5=cmc$Wreligion,x6=cmc$Wwork,x7=cmc$Hoccupation,x8=cmc$living,x9=cmc$media,y=cmc$result)

rfsrc.a<-rfsrc(y~x1+x2+x3+x4+x5+x6+x7+x8+x9,data=abc,ntree=5000,OOB=TRUE,coerce.factor =abc$y)

var.select(y~.,type=2,data=abc,ntree=1000,outcome.target=abc$y)

**stratify sampling, for long term& short term& no use :**

set1<-cmc[cmc$result=="1",]

set2<-cmc[cmc$result=="2",]

set3<-cmc[cmc$result=="3",]

training1<-sample(1:629,472)

test1<-(1:629)[-training1]

training2<-sample(1:333,250)

test2<-(1:333)[-training2]

training3<-sample(1:511,383)

test3<-(1:511)[-training3]

train<-rbind(set1[training1,],set2[training2,],set3[training3,])

test<-rbind(set1[test1,], set2[test2,],set3[test3,])

**do classification tree and calculate the predict error rate:**

my.control <- rpart.control(cp=0, xval=10)

fit1<- rpart(train$result ~ ., data=train, method="class",control=my.control)

printcp(fit1)

tree5<-prune(fit1,cp=0.008)

plot(tree5,uniform=T, margin=0.2)

text(tree5,use.n=T)

pred5<-predict(tree5,newdata=test,type="class")

ta<-table(test$result,pred5)

error5<-ta[1,2]+ta[2,1]+ta[1,3]+ta[2,3]+ta[3,1]+ta[3,2]

error.rate.a<-error5/length(test$result)

**then to compare with random forest prediction error rate:**

rfa<-randomForest(train$result~., data=train)

rfa.pred<-predict(rfa,test)

tb<-table(test$result,rfa.pred)

errortb<-tb[1,2]+tb[2,1]+tb[1,3]+tb[2,3]+tb[3,1]+tb[3,2]

error.rate.b<-errortb/length(test$result)

**stratify sampling, for use& no use :**

set4<-cmb[cmb$result=="1",]

set5<-cmb[cmb$result=="4",]

training4<-sample(1:629,472)

test4<-(1:629)[-training4]

training5<-sample(1:844,633)

test5<-(1:844)[-training5]

train1<-rbind(set4[training4,],set5[training5,])

test1<-rbind(set4[test4,], set5[test5,])

**and then, use two ways of ada boost, count the prediction error rate:**

adam<-ada(train1$result~.,data=train1,iter=50,loss="e",type="real", control=my.control)

print(adam)

adan<-ada(train1$result~.,data=train1,iter=50,loss="e",type="discrete", control=my.control)

print(adan)

adann<-addtest(x=adan, test.x=test1[,-10], test.y=test1[,10])

summary(adann,n.iter=50)

plot(adann,kappa=T, test=T)

pred<-predict(adan,test1[,-10])

tf<-table(test1$result,pred)

errortf<-tf[1,2]+tf[2,1]

error.rate.f<-errortf/length(test1$result)

varplot(adan,type = "none")

**do the same thing above to compare the prediction error rate with classification tree and random forest.**

**then, stratify sampling, for long term& short term only :**

set6<-cmm[cmm$result=="2",]

set7<-cmm[cmm$result=="3",]

training6<-sample(1:333,245)

test6<-(1:333)[-training6]

training7<-sample(1:511,383)

test7<-(1:511)[-training7]

train2<-rbind(set6[training6,],set7[training7,])

test2<-rbind(set6[test6,], set7[test7,])

t**hen, repeat the following steps to calculate the prediction error rate and compare it with random forest and classification tree.**