#LDA QDA->RDA.

install.packages("MASS")

require(MASS)

usedata=cbind(age,wcmatrix1,fnlwgt,edumatrix,edunum,msmatrix,ocmatrix1,rsmatrix,rcmatrix,sexmatrix,capitalgain,capitalloss,hoursperweek,ncmatrix1,salary)

dim(usedata)

#1 stands for <=50K, 2 stands for >50

#

#traindata=data first

#30162/2=15081

View(traindata)

#outlier

require("outliers")

chisq.out.test(fnlwgt, variance=var(fnlwgt), opposite = FALSE)

#OK, no outlier nor missing data nor "?"

df=as.data.frame(usedata)

View(df)

traindf<-df[1:15081,]

table(traindf$salary)

#

# Probability for the label '>50K' : 23.93% / 24.78% (without unknowns)

|# Probability for the label '<=50K' : 76.07% / 75.22% (without unknowns)

z <- lda(salary ~ ., df, prior = c(0.7522,0.2478),subset=c(1:15081))

# We HAVE TO DELETE COL 4 AND 79 SINCE WE CREATE MODEL FROM TRAINING DATA！

usedata=usedata[,-4]

usedata=usedata[,-78]

dim(usedata)

df=as.data.frame(usedata)

traindf<-df[1:15081,]

testdf<-df[15082:30162,]

#\*\*\*\*LDA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#1 stands for <=50K, 2 stands for >50

lda <- lda(salary ~ ., df, prior = c(0.7522,0.2478),subset=c(1:15081))

lda.predict=predict(lda, testdf)$class

truetest=testdf[,104]

lda.predict=as.numeric(lda.predict)

lda.mis=abs(truetest-lda.predict)

lda.misrate=sum(lda.mis)/length(truetest)

#Here is the test error rate!

lda.misrate

#[1] 0.1608647

#How about trainning error?

lda.predict.train=predict(lda, traindf)$class

truetrain=traindf[,104]

lda.predict.train=as.numeric(lda.predict.train)

lda.trainmis=abs(truetrain-lda.predict.train)

lda.misrate.train=sum(lda.trainmis)/length(truetrain)

lda.misrate.train

#[1] 0.1648432

#\*\*\*\*QDA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

usedata=cbind(age,wcmatrix1,fnlwgt,edumatrix,edunum,msmatrix,ocmatrix1,rsmatrix,rcmatrix,sexmatrix,capitalgain,capitalloss,hoursperweek,ncmatrix1,salary)

usedata=usedata[,-4]

usedata=usedata[,-78]

df=as.data.frame(usedata)

traindf<-df[1:15081,]

testdf<-df[15082:30162,]

trainsalary<-traindf[,104]

testsalary<-testdf[,104]

# Then for QDA we need to delete the response variable from dataframe!

# Still error!

# Need to jitter the data to avoid exact multicolinearity

train.qda<- traindf[,-104]

test.qda<-testdf[,-104]

set.seed(1)

train.qda <- apply(train.qda, 2, jitter)

train.qda<-as.data.frame(train.qda)

qda <- qda(trainsalary ~ .,train.qda, prior = c(0.7522,0.2478))

qda.predict=predict(qda, test.qda)$class

truetest=testsalary

qda.predict=as.numeric(qda.predict)

qda.mis=abs(truetest-qda.predict)

qda.misrate=sum(qda.mis)/length(truetest)

#Test train error rate.

qda.misrate

#[1] 0.2227306

#Error may comes from general error, jitter error, and data shape, etc.

#Then train error rate

qda.predict.train=predict(qda, train.qda)$class

truetrain=trainsalary

qda.predict.train=as.numeric(qda.predict.train)

qda.trainmis=abs(truetrain-qda.predict.train)

qda.mistrain=sum(qda.trainmis)/length(truetrain)

#Here train error rate.

qda.mistrain

#[1] 0.1692859

#\*\*\*\*RDA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#In the klaR package’s rda function,

#λ represents the amount of weight given to the pooled covariance matrix.

#Because I have already looked at λ=1 (LDA) and λ=0 (QDA)

#I will use cross-validation for range of λ values between these extremes.

#Just like QDA, RDA will have problems with exactly multicolinear columns,

#so we will reuse the jittered data."test.qda"

train.qda<- traindf[,-104]

test.qda<-testdf[,-104]

train.qda <- apply(train.qda, 2, jitter)

train.qda<-as.data.frame(train.qda)

#change the data colname!

test.rda=data.frame(test.qda)

colnames(test.qda)=names(test.rda)

test.rda=test.qda

train.rda=data.frame(train.qda)

colnames(train.qda)=names(train.rda)

train.rda=train.qda

# We must do this to change the data colname!

#####################Now RDA METHOD

library("klaR")

truetest=testsalary

lambda <- seq(0, 1, 0.1)

#lambda=0<-QDA; lambda=1<-LDA

rda <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =lambda,gamma=0)

rda.trainerror=sum(abs(truetrain-(as.numeric(predict(rda, train.rda)$class))))/length(truetrain)

rda.trainerror

##############Now you gonna see something really stupid!

#alpha=1

rda10 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =1,gamma=0)

rda.trainerror10=sum(abs(truetrain-(as.numeric(predict(rda10, train.rda)$class))))/length(truetrain)

#alpha=0.9

rda9 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.9,gamma=0)

rda.trainerror9=sum(abs(truetrain-(as.numeric(predict(rda9, train.rda)$class))))/length(truetrain)

#alpha=0.8

rda8 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.8,gamma=0)

rda.trainerror8=sum(abs(truetrain-(as.numeric(predict(rda8, train.rda)$class))))/length(truetrain)

#alpha=0.7

rda7 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.7,gamma=0)

rda.trainerror7=sum(abs(truetrain-(as.numeric(predict(rda7, train.rda)$class))))/length(truetrain)

#alpha=0.6

rda6 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.6,gamma=0)

rda.trainerror6=sum(abs(truetrain-(as.numeric(predict(rda6, train.rda)$class))))/length(truetrain)

#alpha=0.5

rda5 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.5,gamma=0)

rda.trainerror5=sum(abs(truetrain-(as.numeric(predict(rda5, train.rda)$class))))/length(truetrain)

#alpha=0.4

rda4 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.4,gamma=0)

rda.trainerror4=sum(abs(truetrain-(as.numeric(predict(rda4, train.rda)$class))))/length(truetrain)

#alpha=0.3

rda3 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.3,gamma=0)

rda.trainerror3=sum(abs(truetrain-(as.numeric(predict(rda3, train.rda)$class))))/length(truetrain)

#alpha=0.2

rda2 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.2,gamma=0)

rda.trainerror2=sum(abs(truetrain-(as.numeric(predict(rda2, train.rda)$class))))/length(truetrain)

#alpha=0.1

rda1 <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.1,gamma=0)

rda.trainerror1=sum(abs(truetrain-(as.numeric(predict(rda1, train.rda)$class))))/length(truetrain)

rda.trainerror1

rda.trainerror2

rda.trainerror3

rda.trainerror4

rda.trainerror5

rda.trainerror6

rda.trainerror7

rda.trainerror8

rda.trainerror9

rda.trainerror10

rda <- rda(trainsalary ~ .,data=train.rda, prior = c(0.7522,0.2478),lambda =0.9,gamma=0)

rda.testerror=sum(abs(truetest-(as.numeric(predict(rda, test.rda)$class))))/length(truetest)

rda.testerror

lambda1 <- lambda[which.min(rda.trainerror)]

lambda1