library("GGally")

## Loading required package: ggplot2

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library("DAAG")

## Loading required package: lattice

library(tree)  
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

set.seed(1234)

#Question 10.3 #1. Using the GermanCredit data set germancredit.txt, use logistic #regression to find a good predictive model for whether credit applicants are good credit risks or #not. Show your model (factors used and their coefficients), the software output, and the quality #of fit.

credit<- read.table("german.data")  
head(credit)

## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17  
## 1 A11 6 A34 A43 1169 A65 A75 4 A93 A101 4 A121 67 A143 A152 2 A173  
## 2 A12 48 A32 A43 5951 A61 A73 2 A92 A101 2 A121 22 A143 A152 1 A173  
## 3 A14 12 A34 A46 2096 A61 A74 2 A93 A101 3 A121 49 A143 A152 1 A172  
## 4 A11 42 A32 A42 7882 A61 A74 2 A93 A103 4 A122 45 A143 A153 1 A173  
## 5 A11 24 A33 A40 4870 A61 A73 3 A93 A101 4 A124 53 A143 A153 2 A173  
## 6 A14 36 A32 A46 9055 A65 A73 2 A93 A101 4 A124 35 A143 A153 1 A172  
## V18 V19 V20 V21  
## 1 1 A192 A201 1  
## 2 1 A191 A201 2  
## 3 2 A191 A201 1  
## 4 2 A191 A201 1  
## 5 2 A191 A201 2  
## 6 2 A192 A201 1

str(credit)

## 'data.frame': 1000 obs. of 21 variables:  
## $ V1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...  
## $ V2 : int 6 48 12 42 24 36 24 36 12 30 ...  
## $ V3 : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...  
## $ V4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...  
## $ V5 : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ V6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...  
## $ V7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...  
## $ V8 : int 4 2 2 2 3 2 3 2 2 4 ...  
## $ V9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...  
## $ V10: Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...  
## $ V11: int 4 2 3 4 4 4 4 2 4 2 ...  
## $ V12: Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...  
## $ V13: int 67 22 49 45 53 35 53 35 61 28 ...  
## $ V14: Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ V15: Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...  
## $ V16: int 2 1 1 1 2 1 1 1 1 2 ...  
## $ V17: Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...  
## $ V18: int 1 1 2 2 2 2 1 1 1 1 ...  
## $ V19: Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...  
## $ V20: Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...  
## $ V21: int 1 2 1 1 2 1 1 1 1 2 ...

#accordingly to the description, we found that V21 is the response. 1 means good, 2 means bad. #Recode the V21 to be a 0/1 variable, instead of 1/2

credit$V21[credit$V21==1]<-0  
credit$V21[credit$V21==2]<-1

# Divide the data into training and test datasets.

trainno <- sample(1:nrow(credit), size = round(nrow(credit)\*0.7), replace = FALSE)  
train <- credit[trainno,]  
test<- credit[-trainno,]

#Fit the logistic model

log<-glm(V21~.,data=train,family=binomial(link="logit"))  
summary(log)

##   
## Call:  
## glm(formula = V21 ~ ., family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9894 -0.6316 -0.2844 0.5607 2.7712   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.145e-01 1.412e+00 0.293 0.769176   
## V1A12 -5.691e-01 2.826e-01 -2.014 0.043993 \*   
## V1A13 -9.997e-01 4.345e-01 -2.301 0.021396 \*   
## V1A14 -1.981e+00 3.044e-01 -6.509 7.57e-11 \*\*\*  
## V2 2.384e-02 1.146e-02 2.081 0.037438 \*   
## V3A31 1.183e+00 7.156e-01 1.653 0.098312 .   
## V3A32 -8.881e-02 5.584e-01 -0.159 0.873638   
## V3A33 -5.834e-01 6.099e-01 -0.957 0.338774   
## V3A34 -1.323e+00 5.695e-01 -2.322 0.020207 \*   
## V4A41 -1.779e+00 4.929e-01 -3.610 0.000307 \*\*\*  
## V4A410 6.396e-02 9.810e-01 0.065 0.948015   
## V4A42 -6.925e-01 3.418e-01 -2.026 0.042774 \*   
## V4A43 -9.077e-01 3.209e-01 -2.829 0.004669 \*\*   
## V4A44 -6.715e-01 9.159e-01 -0.733 0.463487   
## V4A45 -1.905e-01 6.888e-01 -0.277 0.782109   
## V4A46 -2.280e-01 5.014e-01 -0.455 0.649233   
## V4A48 -1.157e+00 1.351e+00 -0.857 0.391482   
## V4A49 -8.954e-02 3.941e-01 -0.227 0.820265   
## V5 1.789e-04 5.568e-05 3.213 0.001313 \*\*   
## V6A62 -1.459e-01 3.608e-01 -0.405 0.685816   
## V6A63 -5.666e-02 4.604e-01 -0.123 0.902048   
## V6A64 -7.304e-01 5.976e-01 -1.222 0.221658   
## V6A65 -1.294e+00 3.428e-01 -3.773 0.000161 \*\*\*  
## V7A72 -5.221e-01 5.568e-01 -0.938 0.348351   
## V7A73 -6.229e-01 5.344e-01 -1.166 0.243777   
## V7A74 -1.373e+00 5.814e-01 -2.361 0.018235 \*   
## V7A75 -7.513e-01 5.330e-01 -1.409 0.158691   
## V8 3.515e-01 1.146e-01 3.068 0.002153 \*\*   
## V9A92 -1.056e+00 4.911e-01 -2.150 0.031521 \*   
## V9A93 -1.287e+00 4.808e-01 -2.677 0.007431 \*\*   
## V9A94 -9.030e-01 5.949e-01 -1.518 0.129011   
## V10A102 5.670e-01 4.892e-01 1.159 0.246489   
## V10A103 -1.771e+00 6.244e-01 -2.837 0.004557 \*\*   
## V11 3.038e-02 1.111e-01 0.273 0.784591   
## V12A122 5.665e-01 3.361e-01 1.686 0.091891 .   
## V12A123 3.049e-01 3.107e-01 0.981 0.326462   
## V12A124 1.126e+00 5.160e-01 2.182 0.029089 \*   
## V13 -1.561e-02 1.175e-02 -1.329 0.183844   
## V14A142 -6.678e-01 5.323e-01 -1.255 0.209627   
## V14A143 -6.982e-01 2.960e-01 -2.359 0.018342 \*   
## V15A152 -7.042e-01 2.928e-01 -2.405 0.016165 \*   
## V15A153 -8.727e-01 5.865e-01 -1.488 0.136744   
## V16 4.489e-01 2.321e-01 1.934 0.053112 .   
## V17A172 1.107e+00 8.815e-01 1.256 0.209218   
## V17A173 1.198e+00 8.462e-01 1.416 0.156874   
## V17A174 1.201e+00 8.478e-01 1.417 0.156550   
## V18 6.828e-02 3.218e-01 0.212 0.831962   
## V19A192 -6.701e-01 2.641e-01 -2.538 0.011160 \*   
## V20A202 -1.520e+00 8.447e-01 -1.799 0.071946 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 853.51 on 699 degrees of freedom  
## Residual deviance: 571.94 on 651 degrees of freedom  
## AIC: 669.94  
##   
## Number of Fisher Scoring iterations: 6

#keep the significant preditors under p-value=0.1,for the categorical predictors, keep them if any of the categories are significant. Then re-fit the model

log2<-glm(V21~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V12+V14+V16+V19+V20,data=train,family=binomial(link="logit"))  
summary(log2)

##   
## Call:  
## glm(formula = V21 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 +   
## V10 + V12 + V14 + V16 + V19 + V20, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1214 -0.6483 -0.2913 0.5920 2.7799   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.851e-01 1.040e+00 0.274 0.784106   
## V1A12 -6.247e-01 2.768e-01 -2.257 0.024021 \*   
## V1A13 -1.148e+00 4.261e-01 -2.694 0.007055 \*\*   
## V1A14 -2.013e+00 2.981e-01 -6.753 1.45e-11 \*\*\*  
## V2 2.335e-02 1.112e-02 2.100 0.035753 \*   
## V3A31 9.932e-01 6.935e-01 1.432 0.152091   
## V3A32 -1.948e-01 5.429e-01 -0.359 0.719776   
## V3A33 -6.234e-01 5.984e-01 -1.042 0.297499   
## V3A34 -1.456e+00 5.545e-01 -2.626 0.008634 \*\*   
## V4A41 -1.647e+00 4.714e-01 -3.493 0.000479 \*\*\*  
## V4A410 -1.807e-01 9.518e-01 -0.190 0.849462   
## V4A42 -5.854e-01 3.360e-01 -1.742 0.081439 .   
## V4A43 -8.615e-01 3.126e-01 -2.756 0.005845 \*\*   
## V4A44 -8.063e-01 9.435e-01 -0.855 0.392761   
## V4A45 -4.838e-01 6.711e-01 -0.721 0.470968   
## V4A46 -1.709e-01 4.955e-01 -0.345 0.730189   
## V4A48 -1.028e+00 1.286e+00 -0.799 0.424009   
## V4A49 -1.522e-01 3.865e-01 -0.394 0.693758   
## V5 1.751e-04 5.289e-05 3.310 0.000932 \*\*\*  
## V6A62 2.042e-02 3.463e-01 0.059 0.952985   
## V6A63 -7.969e-02 4.498e-01 -0.177 0.859365   
## V6A64 -7.470e-01 5.898e-01 -1.266 0.205378   
## V6A65 -1.246e+00 3.349e-01 -3.720 0.000199 \*\*\*  
## V7A72 -5.074e-02 4.710e-01 -0.108 0.914201   
## V7A73 -1.768e-01 4.423e-01 -0.400 0.689397   
## V7A74 -8.837e-01 4.950e-01 -1.785 0.074243 .   
## V7A75 -4.078e-01 4.537e-01 -0.899 0.368781   
## V8 3.368e-01 1.103e-01 3.054 0.002257 \*\*   
## V9A92 -8.548e-01 4.785e-01 -1.786 0.074028 .   
## V9A93 -1.233e+00 4.701e-01 -2.624 0.008700 \*\*   
## V9A94 -6.741e-01 5.798e-01 -1.163 0.244924   
## V10A102 6.601e-01 4.876e-01 1.354 0.175818   
## V10A103 -1.715e+00 5.998e-01 -2.859 0.004253 \*\*   
## V12A122 5.097e-01 3.257e-01 1.565 0.117623   
## V12A123 3.091e-01 2.987e-01 1.035 0.300678   
## V12A124 8.195e-01 3.813e-01 2.149 0.031622 \*   
## V14A142 -7.085e-01 5.220e-01 -1.357 0.174661   
## V14A143 -6.281e-01 2.891e-01 -2.173 0.029792 \*   
## V16 4.308e-01 2.229e-01 1.933 0.053242 .   
## V19A192 -6.423e-01 2.416e-01 -2.658 0.007859 \*\*   
## V20A202 -1.539e+00 8.367e-01 -1.839 0.065913 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 853.51 on 699 degrees of freedom  
## Residual deviance: 584.77 on 659 degrees of freedom  
## AIC: 666.77  
##   
## Number of Fisher Scoring iterations: 5

#For the categorical variables, not all the levels are significant. So create a binary (0/1) variable for each of them: 0 for not significant, 1 for significant

train$V1A12[train$V1=="A12"]<-1  
train$V1A12[train$V1!="A12"]<-0  
  
train$V1A13[train$V1=="A13"]<-1  
train$V1A13[train$V1!="A13"]<-0  
  
train$V1A14[train$V1=="A14"]<-1  
train$V1A14[train$V1!="A14"]<-0  
  
train$V3A34[train$V1=="A34"]<-1  
train$V3A34[train$V1!="A34"]<-0  
  
train$V4A41[train$V1=="A41"]<-1  
train$V4A41[train$V1!="A41"]<-0  
  
train$V4A42[train$V1=="A42"]<-1  
train$V4A42[train$V1!="A42"]<-0  
  
train$V4A43[train$V1=="A43"]<-1  
train$V4A43[train$V1!="A43"]<-0  
  
train$V6A65[train$V1=="A65"]<-1  
train$V6A65[train$V1!="A65"]<-0  
  
train$V7A74[train$V1=="A74"]<-1  
train$V7A74[train$V1!="A74"]<-0  
  
train$V9A92[train$V1=="A92"]<-1  
train$V9A92[train$V1!="A92"]<-0  
  
train$V9A93[train$V1=="A93"]<-1  
train$V9A93[train$V1!="A93"]<-0  
  
train$V10A103[train$V1=="A103"]<-1  
train$V10A103[train$V1!="A103"]<-0  
  
train$V12A124[train$V1=="A124"]<-1  
train$V12A124[train$V1!="A124"]<-0  
  
train$V14A143[train$V1=="A143"]<-1  
train$V14A143[train$V1!="A143"]<-0  
  
train$V19A192[train$V1=="A192"]<-1  
train$V19A192[train$V1!="A192"]<-0  
  
train$V20A202[train$V1=="A202"]<-1  
train$V20A202[train$V1!="A202"]<-0

#re-fit the model with these significant variables

log3<-glm(V21~V1A12+V1A13+V1A14+V2+V3A34+V4A41+V4A42+V4A43+V5+V6A65+V7A74+V8+V9A92+V9A93+V10A103+V12A124+V14A143+V16+V19A192+V20A202,data=train,family=binomial(link="logit"))  
summary(log3)

##   
## Call:  
## glm(formula = V21 ~ V1A12 + V1A13 + V1A14 + V2 + V3A34 + V4A41 +   
## V4A42 + V4A43 + V5 + V6A65 + V7A74 + V8 + V9A92 + V9A93 +   
## V10A103 + V12A124 + V14A143 + V16 + V19A192 + V20A202, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9255 -0.8605 -0.4281 0.9381 2.4280   
##   
## Coefficients: (13 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.173e+00 4.258e-01 -2.755 0.00586 \*\*   
## V1A12 -5.140e-01 2.220e-01 -2.316 0.02058 \*   
## V1A13 -1.166e+00 3.830e-01 -3.045 0.00233 \*\*   
## V1A14 -2.313e+00 2.540e-01 -9.107 < 2e-16 \*\*\*  
## V2 2.538e-02 9.340e-03 2.718 0.00657 \*\*   
## V3A34 NA NA NA NA   
## V4A41 NA NA NA NA   
## V4A42 NA NA NA NA   
## V4A43 NA NA NA NA   
## V5 9.641e-05 4.164e-05 2.316 0.02058 \*   
## V6A65 NA NA NA NA   
## V7A74 NA NA NA NA   
## V8 1.633e-01 9.208e-02 1.773 0.07620 .   
## V9A92 NA NA NA NA   
## V9A93 NA NA NA NA   
## V10A103 NA NA NA NA   
## V12A124 NA NA NA NA   
## V14A143 NA NA NA NA   
## V16 -7.674e-02 1.574e-01 -0.488 0.62581   
## V19A192 NA NA NA NA   
## V20A202 NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 853.51 on 699 degrees of freedom  
## Residual deviance: 702.82 on 692 degrees of freedom  
## AIC: 718.82  
##   
## Number of Fisher Scoring iterations: 5

#only keep the significant terms

log4<-glm(V21~V1A12+V1A13+V1A14+V2+V5+V8,data=train,family=binomial(link="logit"))  
summary(log4)

##   
## Call:  
## glm(formula = V21 ~ V1A12 + V1A13 + V1A14 + V2 + V5 + V8, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9446 -0.8595 -0.4272 0.9275 2.4128   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.285e+00 3.596e-01 -3.574 0.000352 \*\*\*  
## V1A12 -5.099e-01 2.218e-01 -2.299 0.021477 \*   
## V1A13 -1.155e+00 3.820e-01 -3.023 0.002503 \*\*   
## V1A14 -2.316e+00 2.539e-01 -9.123 < 2e-16 \*\*\*  
## V2 2.545e-02 9.339e-03 2.725 0.006436 \*\*   
## V5 9.637e-05 4.162e-05 2.316 0.020582 \*   
## V8 1.633e-01 9.207e-02 1.774 0.076030 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 853.51 on 699 degrees of freedom  
## Residual deviance: 703.06 on 693 degrees of freedom  
## AIC: 717.06  
##   
## Number of Fisher Scoring iterations: 5

#Now every term are significant, this is the final model

#Add the remained binary variables to the test dataset

test$V1A12[test$V1=="A12"]<-1  
test$V1A12[test$V1!="A12"]<-0  
  
test$V1A13[test$V1=="A13"]<-1  
test$V1A13[test$V1!="A13"]<-0  
  
test$V1A14[test$V1=="A14"]<-1  
test$V1A14[test$V1!="A14"]<-0

#validate the model using the test dataset

yhatlog<-predict(log4,test,type = "response")  
head(yhatlog)

## 5 8 9 16 18 20   
## 0.5707649 0.5292916 0.0644921 0.5256126 0.6417318 0.1024695

#round the yhatlog to be 0/1 variabls

y<- as.integer(yhatlog > 0.5)  
head(y)

## [1] 1 1 0 1 1 0

t <- table(y,test$V21)  
t

##   
## y 0 1  
## 0 182 58  
## 1 27 33

correct<-(182+33)/300  
correct

## [1] 0.7166667

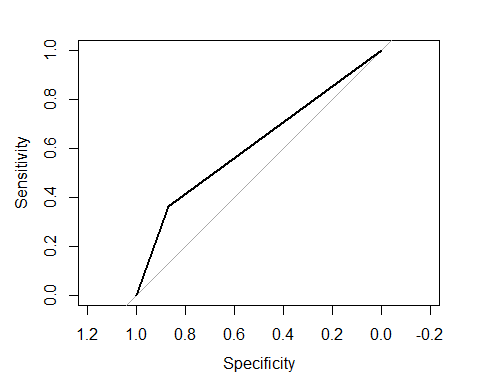
roc<-roc(test$V21,y)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

# Plot the ROC curve

plot(roc)



roc

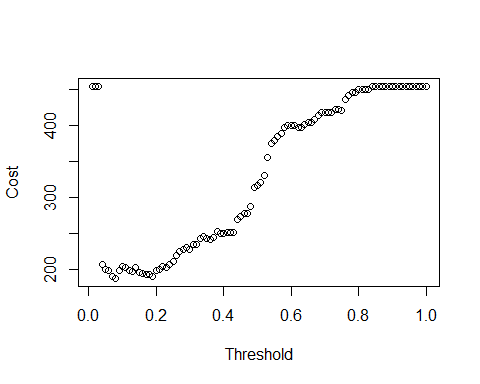
##   
## Call:  
## roc.default(response = test$V21, predictor = y)  
##   
## Data: y in 209 controls (test$V21 0) < 91 cases (test$V21 1).  
## Area under the curve: 0.6167

#The model I developed is: log(p/(1-p))=-1.285e+00-5.099e-01*V1A12-1.155e+00*V1A13-2.316e+00*V1A14+2.545e-02*V2+9.637e-05*V5+1.633e-01*V8 The accuracy rate is 71.67%, AIC is 717.06,and AUC is 61.67%,which means the model will correctly classify the samples 61.67% of the times.

#2. Because the model gives a result between 0 and 1, it requires setting a threshold probability to separate between ��good�� and ��bad�� answers. In this data set, they estimate that incorrectly identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad. Determine a good threshold probability based on your model.

Calulating loss for tthe cost for thresholds ranging from 0.01 to 1.

cost <- c()  
for(i in 1:100){  
 y.hat<- as.integer(yhatlog > (i/100)) #0.01-100  
   
 table<-as.matrix(table(y.hat,test$V21))  
   
 if(nrow(table)>1) { cst1 <- table[2,1] } else { cst1 <- 0 }  
 if(ncol(table)>1) { cst2 <- table[1,2] } else { cst2 <- 0 }  
 cost <- c(cost, cst1+cst2\*5)  
}  
  
plot(c(1:100)/100,cost,xlab = "Threshold",ylab = "Cost")



which.min(cost)

## [1] 8

cost

## [1] 455 455 455 207 200 198 190 187 198 204 202 198 197 202 195 194 192  
## [18] 192 189 198 200 204 202 207 211 219 224 227 230 227 234 235 242 245  
## [35] 242 241 244 252 249 249 251 251 251 269 274 277 278 287 314 317 321  
## [52] 330 356 375 379 385 389 397 401 401 400 398 397 402 405 405 409 414  
## [69] 419 419 419 418 423 423 422 437 442 446 446 451 451 450 450 455 455  
## [86] 455 455 455 455 455 455 455 455 455 455 455 455 455 455 455

#when threshold=0.08, we have minimum cost 187.