WESTERN MICHIGAN UNIVERSITY

STAT 6850

Applied Data Mining

Fall 2014

Instructor: Dr. J.C. Wang

Case Study #2

A classification case study
using German Credit Data

Antonio Giraldi

Saurabh Rajratn Kulkarni

Shoruk Mansour

Joan Martinez

Milton Soto Ferrari

Mustafa Yildiz

Description Summary

The objective of this report is to classify bank clients considering a known set of variables for credit purposes. The outcome (response variable) is either a customer is considered credit worthy (good) or not (bad). Attributes can be given with variables or factor levels, which are used as an indicator for a bank to decide if that customer is worthy to have. For example, Attribute 1 is a qualitative characteristic to indicate the status of existing checking account with the bank; and can have the following levels depending on account balance in DM ("Deutsche Mark" Currency):
A11, represents less than 0 DM
A12, represents between 0 and 200 DM
A13, represents more than 200 DM / salary assignments for at least 1 year
A14, represents no checking account

The german dataset is compromised of 1000 samples. The customers' attributes (Credit History, Age, Status, Job, etc…) accounts as 20 independent variables. The variable called "Risk" is the response variable and our dataset includes 700 "good" responses and 300 "bad" responses.

Exploratory Analysis

Using visualization methods such as frequency tables, histogram and correlation graphs for attribute's categories, we will study the behavior of data and possible detect and eliminate variables that are considered non-significant when performing classification tasks.

- *Frequency graphs*
  These graphs show the proportion and the way the clients are distributed in terms of their attributes. Fig1 represents checking account, credit history, purpose of having the account, and saving account. From here we can see:
  - Usually clients do not have a checking account.
  - Most clients' credit history is distributed between fair and poor.
  - Clients are usually buying furniture and cars.
  - Most clients have [0-100] in their saving account.

  Fig2 represents employment, status, debtor guarantor, and property. Here we can see:

  - Employment duration has high variability.
  - Most clients are male and single.
  - Almost every client has no debtor guarantor.
  - Most clients' properties are distributed somewhat evenly.

  Fig3 represents other installment, housing, job, and phone. We see that

  - Almost every client doesn't have other installments.
  - Most clients own housing.
  - Most clients have skilled jobs.
  - Clients having phone are well distributed.

Fig4 represents installment rate, residence, existent credits, and people liable. We detected that

- Clients usually have high installment rate (4%).
- Most clients have residence more than 3 years.
- Clients have one to two existent credit accounts.
- Most clients have only one people liable.

Fig5 represents if the clients are foreign workers or not. We conclude that:

- Almost every client is a foreign workers

- *Histogram*
  The histogram was applied for the three non categorical variables age (skewed to the right), duration (also skewed to the right), and credit amount (looks exponentially distributed) as shown in Fig6.

- *Correlation Matrix*
  A correlation matrix was run for numerical variables. Some positive correlation was found between Credit Amount and Duration with a r=0.625 as seen in Table1. A Correlogram is shown in Fig7 for other attributes for comparison purposes.

After assessing using exploratory analysis, we found two significant variables that can be potentially removed: Foreign Workers and Debtor Guarantor. We used a 90-10% cutoff for variance to identify these variables. Moscaic plots are shown in Fig8 and Fig9 to visualize the variability of these attributes vs. Risk. Other attributes such as People liable and Other Installment show low variability as seen in mosaic plots in Fig10 and Fig11, but we decided not to remove these as they show a variance ratio above 80-20%.

Classification Analysis

To perform "Risk" classification task in this dataset we'll use the following methods: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes Classifier (NB), Linear Regression of response indicator matrix (LM), and logistic regression (LR/GLM). To apply these, we split the German Credit Data set randomly into train set and test set with approximately 60%/40% proportion of Risk responses. The structure of the original good:bad (70%:30%) credit distribution in both the train set and the test set was considered [(420/180 for good/bad) cases for train set and 400 (280/120 for good/bad) cases for test set].

We ran these methods adjusting the priors to their original proportion (70%/30%) and also based on a costing structure as given in the dataset statement ["It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1)"]. Table2 shows the calculations for original proportion and cost adjusted priors. Our main goal is to compare the errors from these five methods based on the test set and decide the best method that can be used in both priors.

Training Fitting Results

For each method we used a 5 fold cross-validation with 5 iterations on the training set. This is done to estimated methods parameters for best fitting results using both original and cost adjusted priors. The next fitting results are using cost adjusted priors, where in Fig12 we see boxplots of all fitting results in LDA, GLM, NB and QDA. NB shows high sensitivity and LDA show low sensitivity. ROC is significantly different between QDA and LDA/GLM, showing a low receiver operating characteristic. Fig13 is a density plot to show behavior of fitting results, again showing NB as very sensitive, while according to ROC all behave similarly except QDA and LDA/GLM. This is again seen in Fig14 where the differences of each method are plotted; LDA and NB shows significant difference in Sensitivity and Specificity. Fig15 are scatter plots of each metric between all methods; again, LDA and NB differences are clearly shown. For proportional priors w/o cost adjustment we see improvements in GLM, NB and QDA in terms on Sensitivity and Specificity, LDA wasn't affected when changing priors. We can see these results in Fig16, Fig 17, Fig18 and Fig19

For Linear Regression of response indicator matrix (LM), there is not R package for fitting training. Using a custom-made function we were able to get fitting results for this method as shown in Table3. As over-fitting can occur, we do not compare this with the previous models and is also not affected when changing priors.

Methods Classification Results

After cross-validation trainings, we used all models with the Test set to compare the errors from these five methods. Table4 shows all results when considering priors adjusted by cost, while Table5 shows results with proportional priors. These results can be better visualized in Fig20 and Fig21. Fig20 shows performance metrics w/ prior cost adjustment. In contrast to results when fitting the train set, on the Test set LDA has the highest Sensitivity, lowest Specificity and lowest overall cost. While NB now shows the worst Sensitivity and also the highest cost. Fig21shows these metrics w/o adjusting priors for cost, and again LDA and NB are significantly different; where LDA is a better classifier method based on results. Comparing these figures we saw that Logistic Regression improved significantly in terms of Sensitivity and Cost, indicating this method is sensible to prior adjustments.

Concluding Remarks

When considering cost adjusted prior (good=1, bad=5), Linear Discriminant Analysis and Logistic Regression are the best classifying methods for this dataset.

APPENDIX

1. Tables

Table1. ***Credit amount Correlation***

Variables	Credit Amount
Duration	0.6249842
Credit Amount	1.0000000
Installment Rate	-0.2713157
Residence	0.02892632
Age	0.03271642
Credits	0.02079455
People Liable	0.01714215

Table2. ***Priors***

Risk	Cost Adjusted Prior	Proportion Prior
good	0.3182	0.7
bad	0.6818	0.3

Table3. ***Linear Model Fitting Results***

Sensitivity	Specificity	ROC	Accuracy
0.867619	0.4522222	1.5855807	0.743

Table 4. ***Classification Results with Costing Adjustments***

	Sensitivity	Specificity	ROC	ROCNEG	Type 1E	Type 2E	Accuracy	Cost
LDA	0.91	0.45	1.66	0.2	0.55	0.09	0.64	202
QDA	0.83	0.52	1.71	0.33	0.48	0.17	0.71	287
NB	0.72	0.51	1.48	0.54	0.49	0.28	0.7	523
LR	0.85	0.54	1.86	0.27	0.46	0.15	0.73	251
LM	0.78	0.59	1.88	0.38	0.41	0.22	0.74	381

Table 5. ***Classification Results without Costing Adjustments***

Methods	Sensitivity	Specificity	ROC	ROCNEG	Type 1E	Type 2E	Accuracy	Cost
LDA	0.91	0.45	1.66	0.2	0.55	0.09	0.64	202
QDA	0.8	0.56	1.83	0.35	0.44	0.2	0.74	334
NB	0.71	0.53	1.52	0.55	0.47	0.29	0.7	567
LR	0.73	0.53	1.56	0.51	0.47	0.27	0.71	501
LM	0.78	0.59	1.88	0.38	0.41	0.22	0.74	381

2. <u>Plots</u>

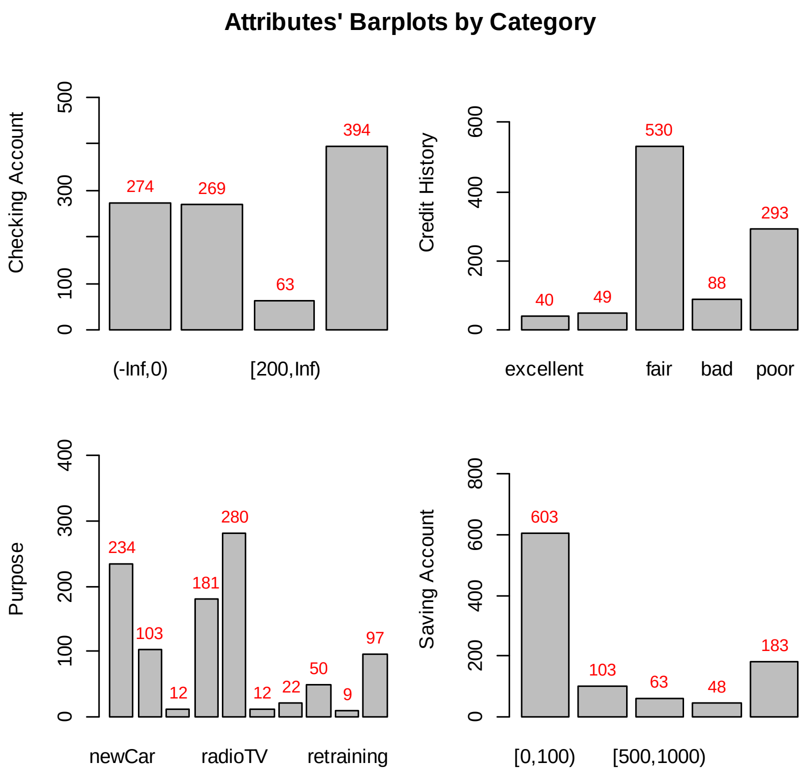


Fig1

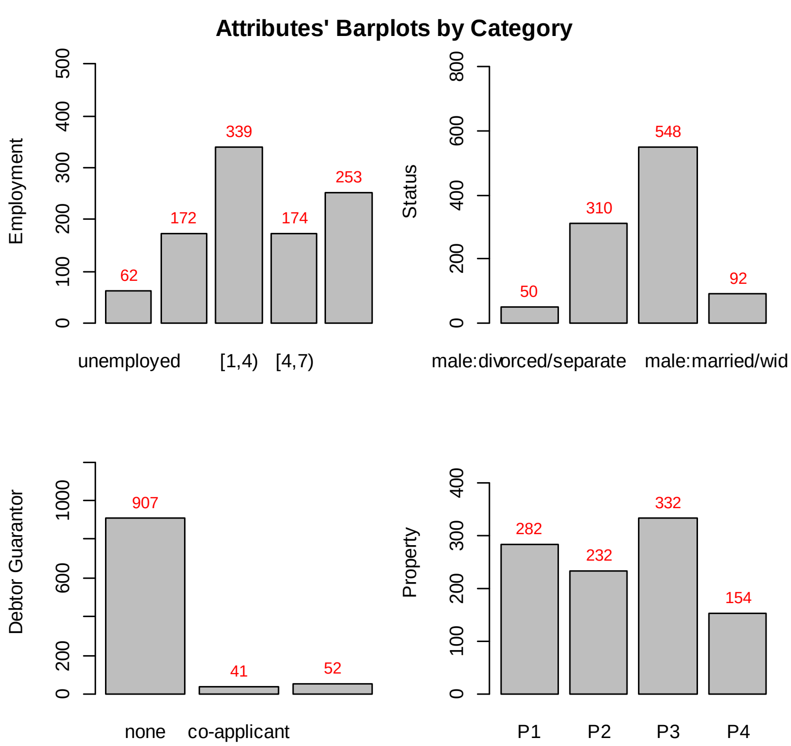


Fig2

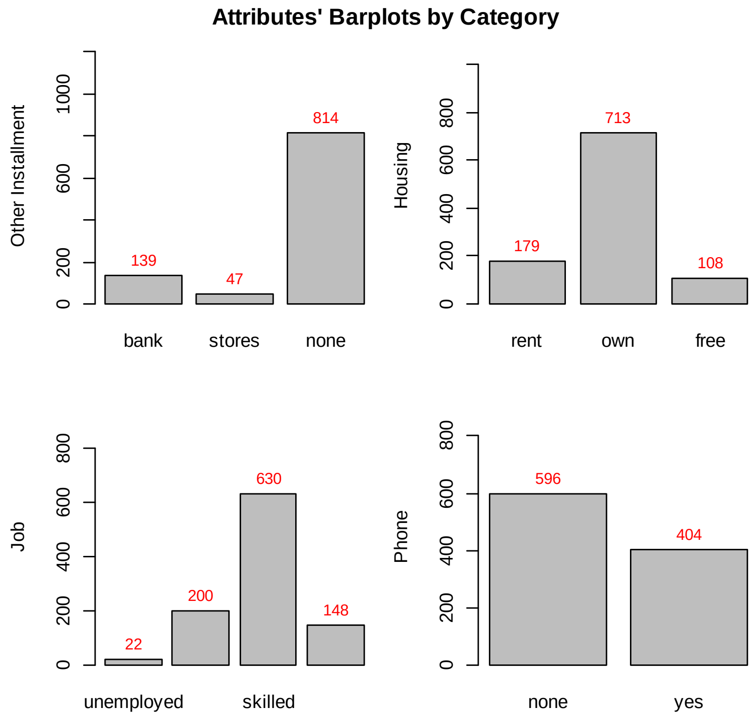

Attributes' Barplots by Category
Other Installment
139
47
814
bank
stores
none
Housing
179
713
108
rent
own
free
Job
22
200
630
148
unemployed
skilled
Phone
596
404
none
yes

Fig3

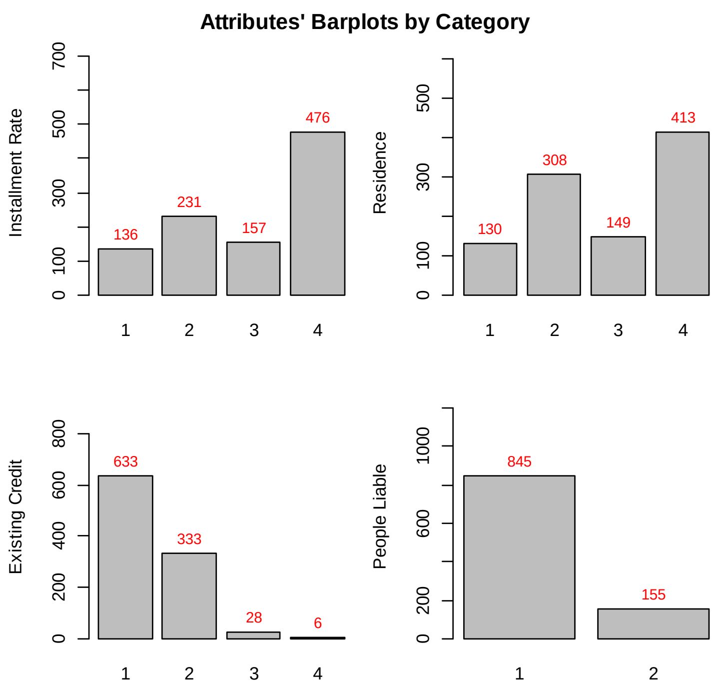

Attributes' Barplots by Category
Installment Rate
136
231
157
476
Residence
130
308
149
413
Existing Credit
633
333
28
6
People Liable
845
155

Fig4

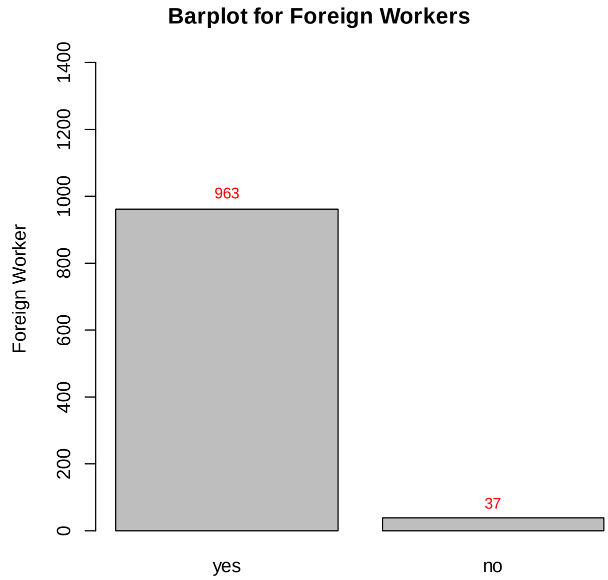

Barplot for Foreign Workers
Foreign Worker
0
200
400
600
800
1000
1200
1400
963
37
yes
no

Fig 5

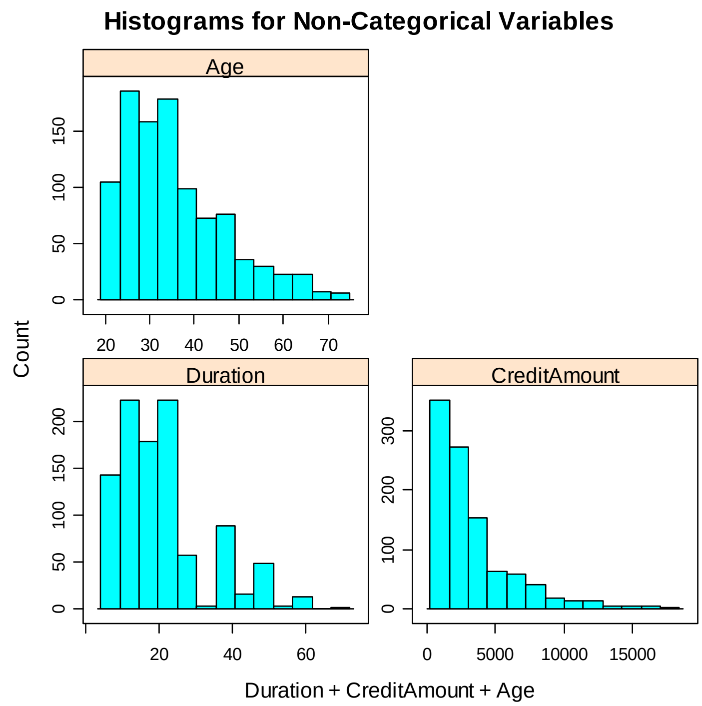

Histograms for Non-Categorical Variables
Age
Duration
CreditAmount
Count
Duration + CreditAmount + Age
0
50
100
150
200
300
20
30
40
50
60
70
5000
10000
15000

Fig 6

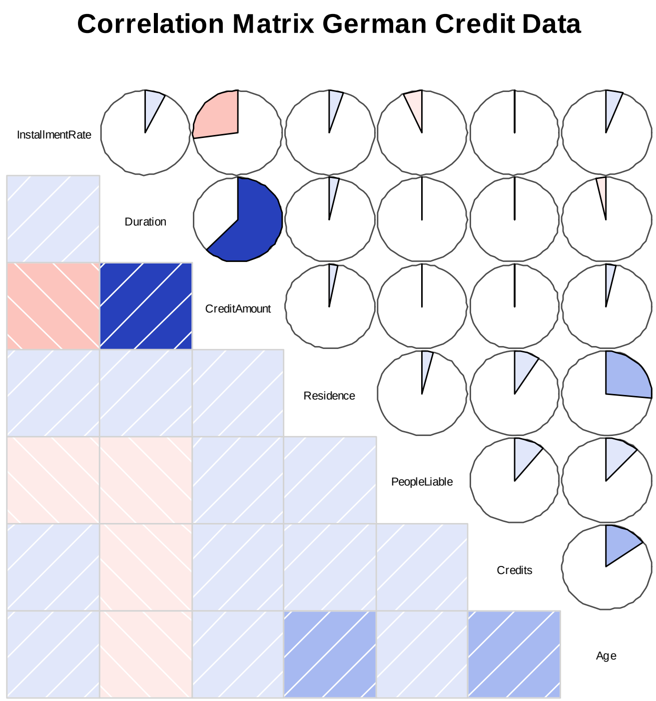

Correlation Matrix German Credit Data
InstallmentRate
Duration
CreditAmount
Residence
PeopleLiable
Credits
Age

Fig 7

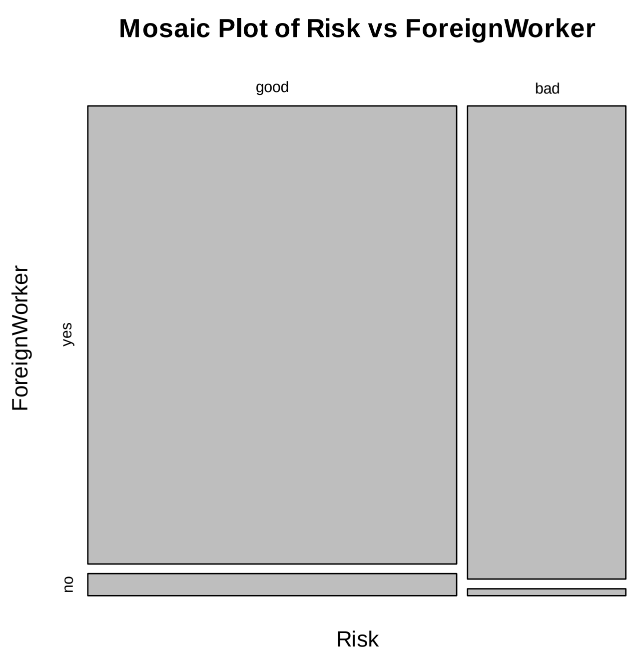

Mosaic Plot of Risk vs ForeignWorker
good
bad
ForeignWorker
yes
no
Risk

Fig 8

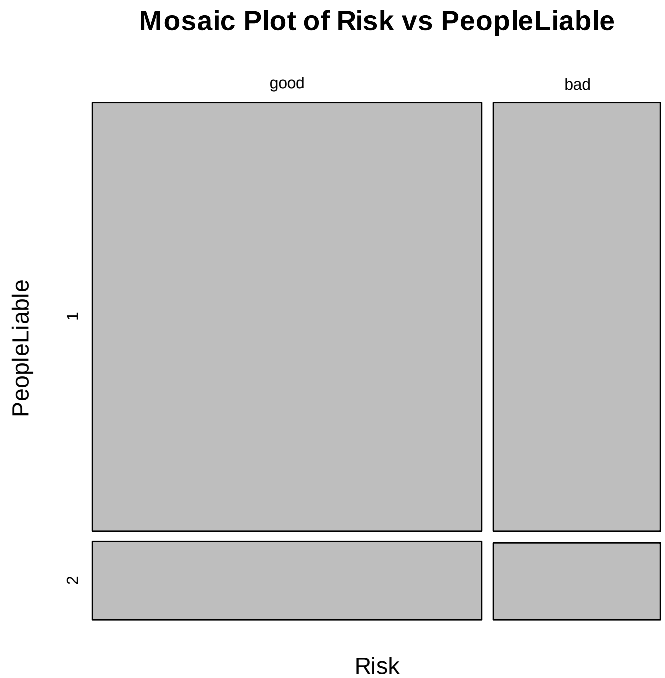


Fig 9

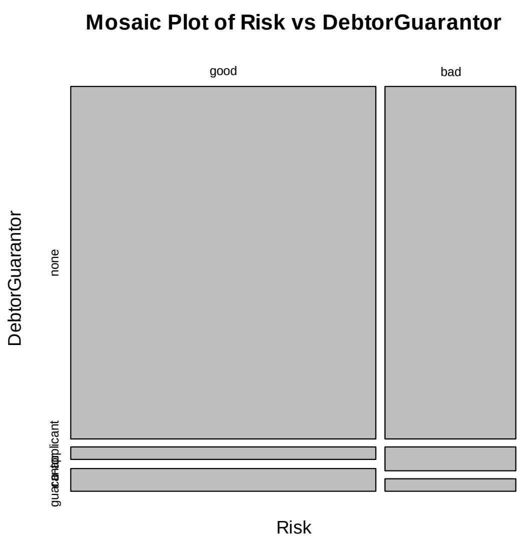


Fig10

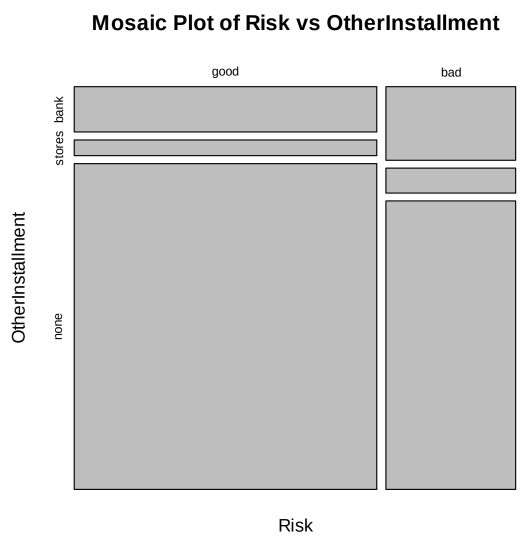
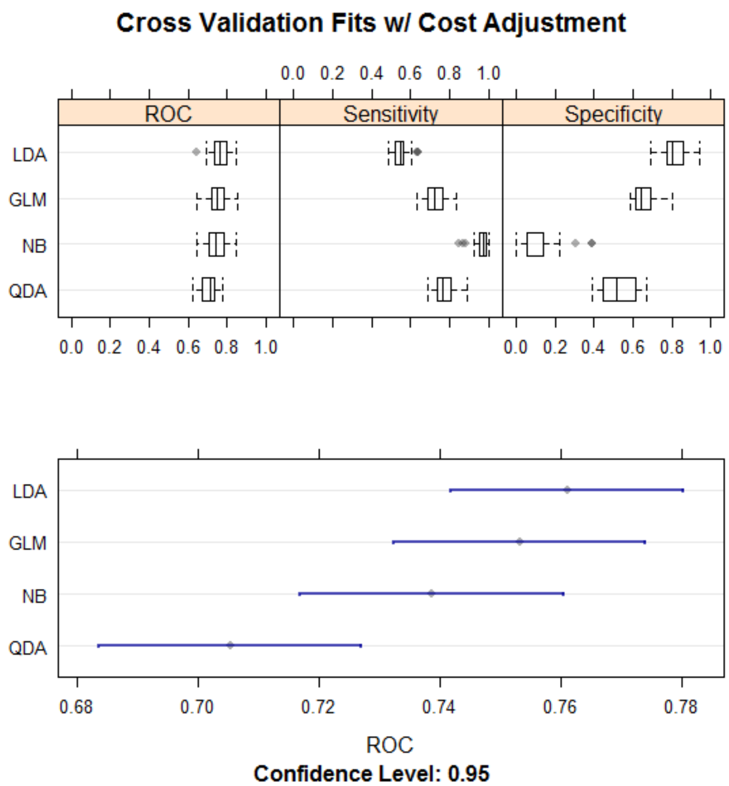
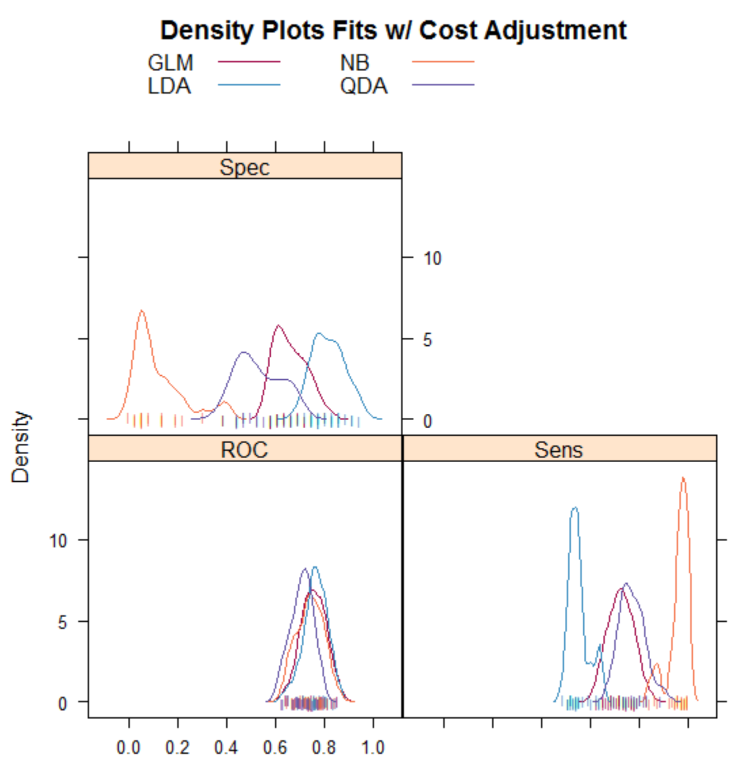


Fig11



Cross Validation Fits w/ Cost Adjustment
ROC
Sensitivity
Specificity
LDA
GLM
NB
QDA
0.0 0.2 0.4 0.6 0.8 1.0
0.68 0.70 0.72 0.74 0.76 0.78
ROC
Confidence Level: 0.95

Fig 12



Density Plots Fits w/ Cost Adjustment
GLM
LDA
NB
QDA
Spec
ROC
Sens
Density
0 5 10
0.0 0.2 0.4 0.6 0.8 1.0

Fig 13

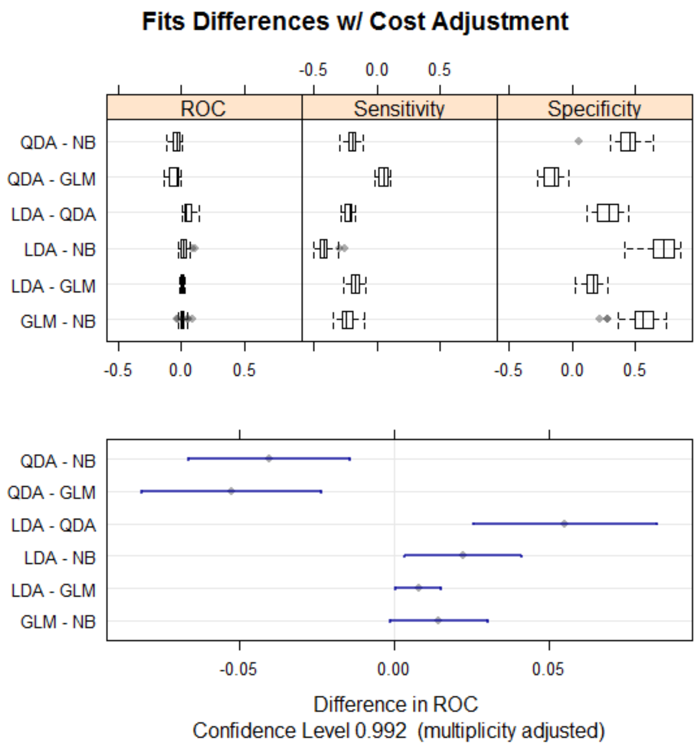
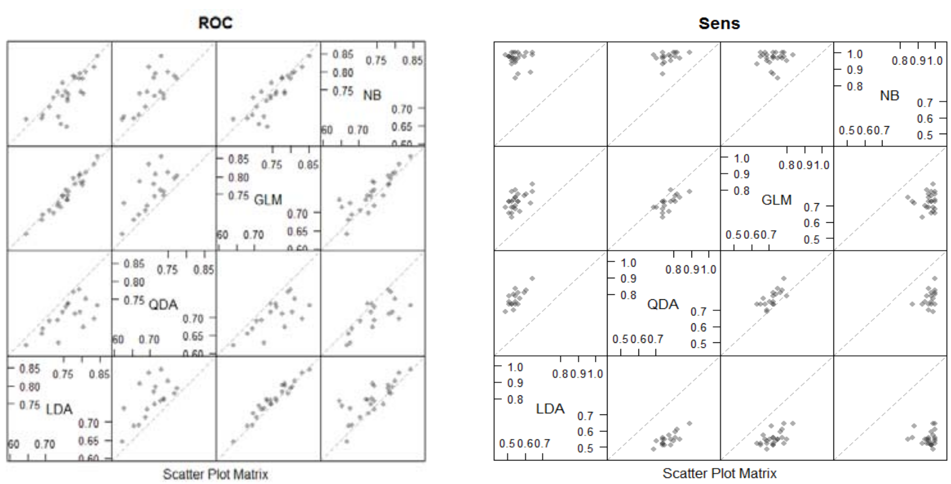

Fits Differences w/ Cost Adjustment
ROC
Sensitivity
Specificity
QDA - NB
QDA - GLM
LDA - QDA
LDA - NB
LDA - GLM
GLM - NB
-0.5
0.0
0.5
-0.05
0.00
0.05
Difference in ROC
Confidence Level 0.992 (multiplicity adjusted)

Fig 14


ROC
NB
GLM
QDA
LDA
Scatter Plot Matrix
Sens
NB
GLM
QDA
LDA
Scatter Plot Matrix

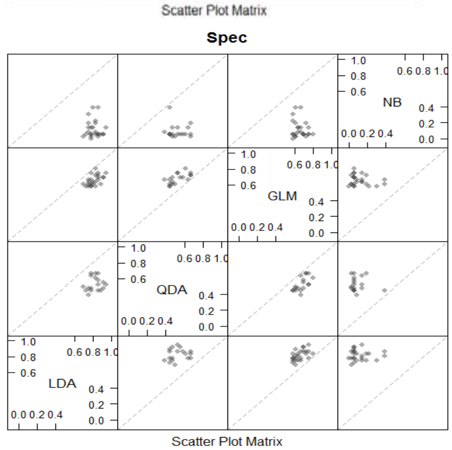

Spec
NB
GLM
QDA
LDA
Scatter Plot Matrix

Fig 15

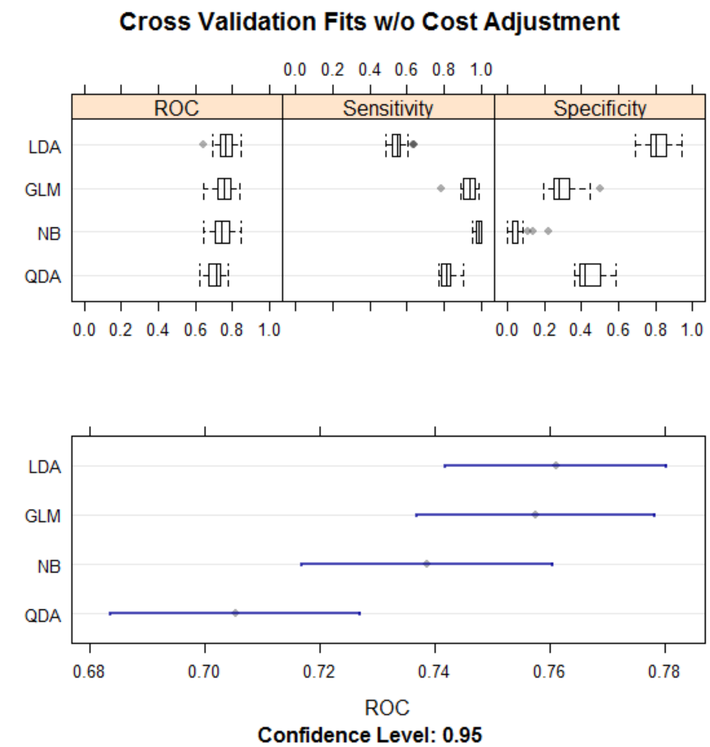

Cross Validation Fits w/o Cost Adjustment
ROC
Sensitivity
Specificity
LDA
GLM
NB
QDA
0.0 0.2 0.4 0.6 0.8 1.0
ROC
Confidence Level: 0.95

Fig 16

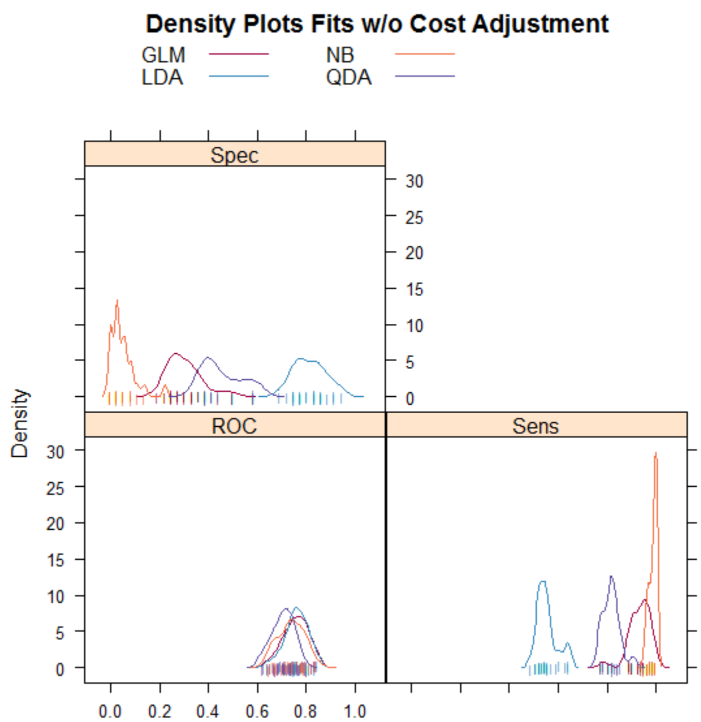

Density Plots Fits w/o Cost Adjustment
GLM
LDA
NB
QDA
Spec
ROC
Sens
Density
0.0 0.2 0.4 0.6 0.8 1.0

Fig 17

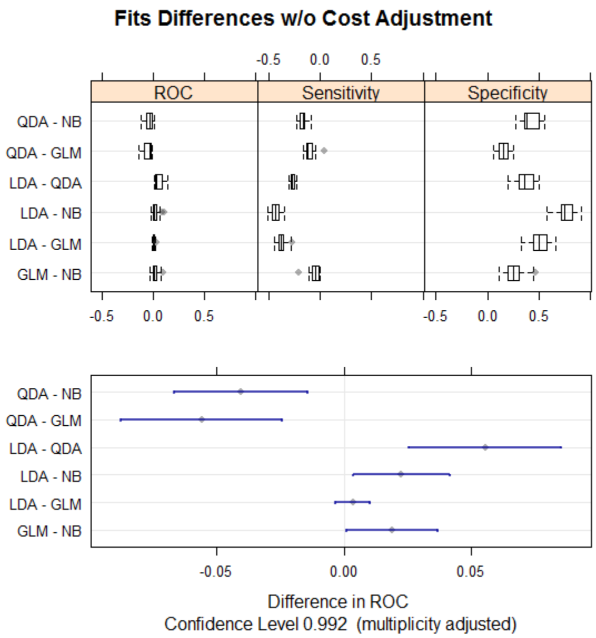

Fits Differences w/o Cost Adjustment
ROC
Sensitivity
Specificity
QDA - NB
QDA - GLM
LDA - QDA
LDA - NB
LDA - GLM
GLM - NB
-0.5
0.0
0.5
-0.05
0.00
0.05
Difference in ROC
Confidence Level 0.992 (multiplicity adjusted)

Fig 18

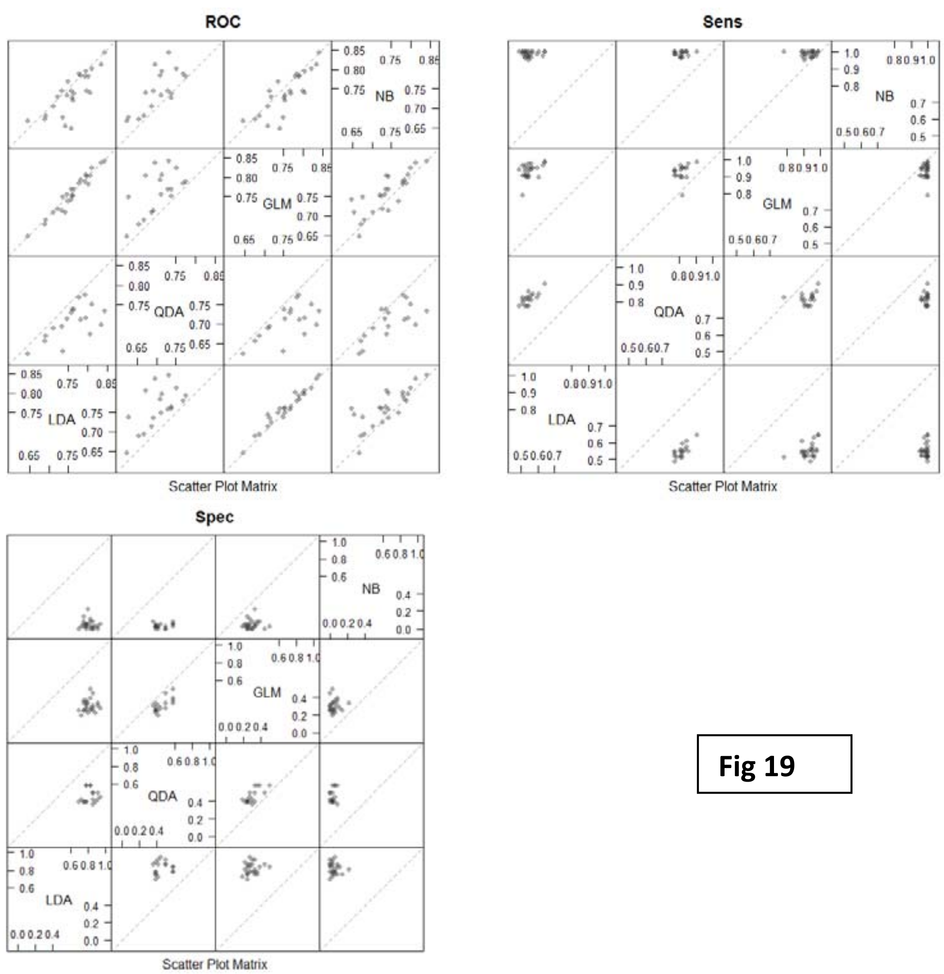

ROC
Sens
Spec
NB
GLM
QDA
LDA
Scatter Plot Matrix
Scatter Plot Matrix
Scatter Plot Matrix

Fig 19

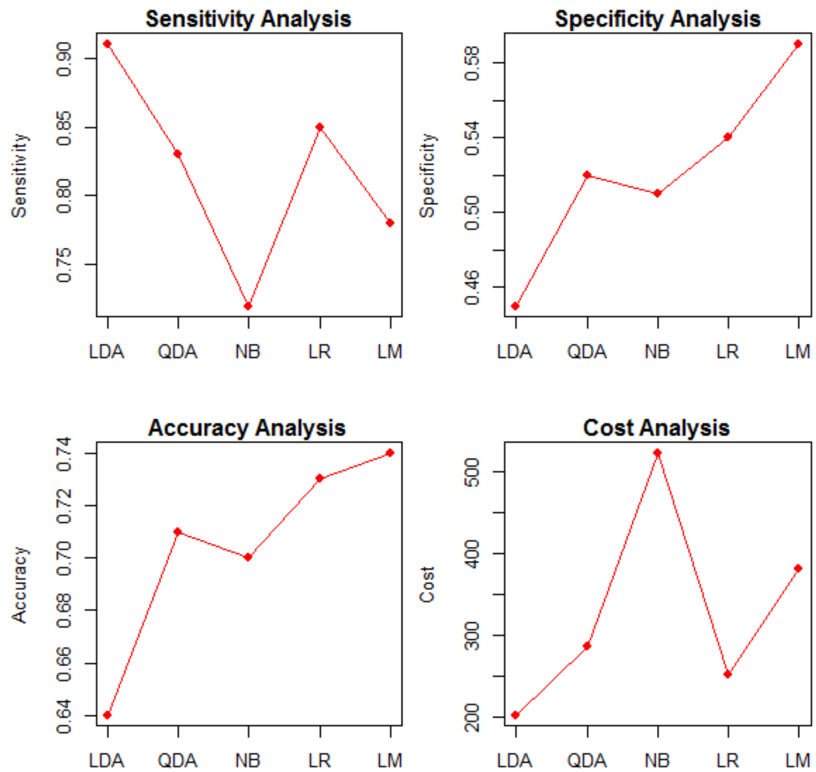

Sensitivity Analysis
Sensitivity
0.75 0.80 0.85 0.90
Specificity Analysis
Specificity
0.46 0.50 0.54 0.58
Accuracy Analysis
Accuracy
0.64 0.66 0.68 0.70 0.72 0.74
Cost Analysis
Cost
200 300 400 500
LDA QDA NB LR LM

Fig 20

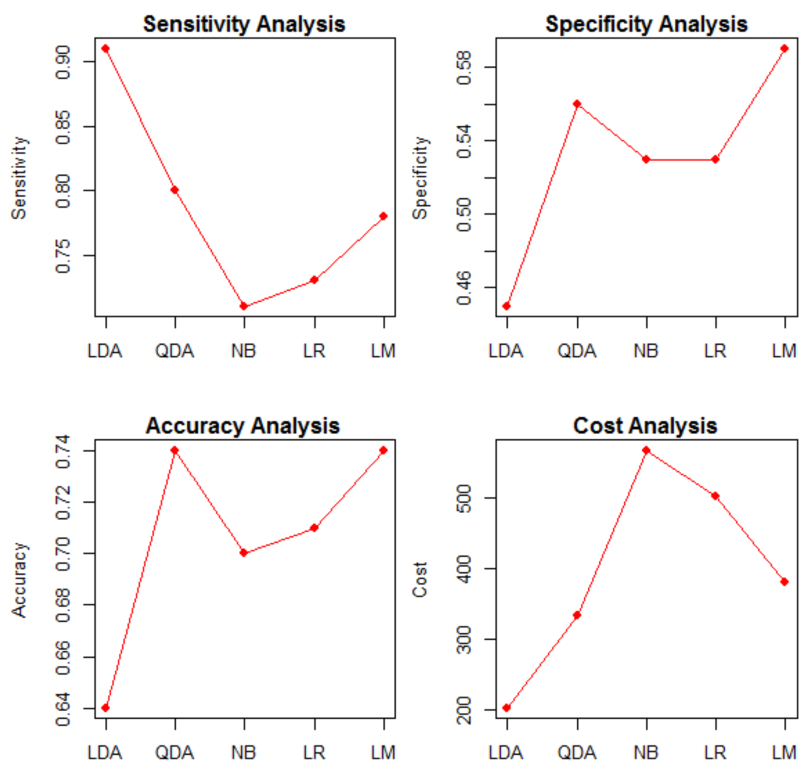

Sensitivity Analysis
Sensitivity
0.75 0.80 0.85 0.90
Specificity Analysis
Specificity
0.46 0.50 0.54 0.58
Accuracy Analysis
Accuracy
0.64 0.66 0.68 0.70 0.72 0.74
Cost Analysis
Cost
200 300 400 500
LDA QDA NB LR LM

Fig 21

3. R Code

```
require(lattice)
require(latticeExtra)
require(corrgram)
require(caret)
require(pROC)
require(klaR)

## DATA SOURCE
uciData <-
  "http://archive.ics.uci.edu/ml/machine-learning-databases"
f <- paste(uciData,"statlog","german","german.data",sep="/")
german <- read.table(f, header=F, as.is=T)
sapply(german, class)

## DATA AND FACTORS LABELING
names(german) <- scan(what="", nmax=21)
CheckingAccount Duration CreditHistory Purpose CreditAmount
SavingAccount Employment InstallmentRate Status DebtorGuarantor
Residence Property Age OtherInstallment Housing
Credits Job PeopleLiable Phone ForeignWorker Risk

german[[1]] <- factor(german[[1]],labels=scan(what='', nmax=4))
(-Inf,0) [0,200) [200,Inf) noAccount

german[[3]] <- factor(german[[3]], labels=scan(what='', nmax=5), ordered=T)
excellent good fair bad poor

german[[4]] <- factor(german[[4]],labels=scan(what='', nmax=10))
newCar usedCar others furniture radioTV appliances
repairs education retraining business

german[[6]] <- factor(german[[6]],labels=scan(what='', nmax=5))
[0,100) [100,500) [500,1000) [1000,Inf) noAccount+unknown

german[[7]] <- factor(german[[7]],labels=scan(what='', nmax=5), ordered=T)
unemployed (0,1) [1,4) [4,7) [7,Inf)

german[[9]] <- factor(german[[9]], labels=scan(what='', nmax=4))
male:divorced/separate female:divorced/separated/married
male:single male:married/widowed

german[[10]] <- factor(german[[10]], labels=scan(what='', nmax=3))
none co-applicant guarantor

german[[12]] <- factor(german[[12]], labels=paste("P",1:4,sep=""))

german[[14]] <- factor(german[[14]], labels=scan(what='', nmax=3))
bank stores none

german[[15]] <- factor(german[[15]], labels=scan(what='', nmax=3))
rent own free

german[[17]] <- factor(german[[17]], labels=scan(what='',nmax=4), ordered=T)
unemployed unskilled skilled management

german[[19]] <- factor(german[[19]], labels=c('none','yes'))

german[[20]] <- factor(german[[20]],labels=c('yes','no'))

german[[21]] <- factor(german[[21]],labels=c('good','bad'))

sapply(german, class)


## EXPLORATORY ANALYSIS
summary(german)
```

```
#Frequency Tables for Categorical Attributes
oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,0.5,0.5), oma=c(0,0,2,0))
ylim <- c(0, 1.5*max(as.numeric(table(german[1]))))
xx <- barplot(table(german[1]), width = 0.85, ylim = ylim, ylab ="Checking Account")
text(x =xx, y = as.numeric(table(german[1])), label = as.numeric(table(german[1])), pos = 3, cex
= 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[3]))))
xx <- barplot(table(german[3]), width = 0.85, ylim = ylim, ylab ="Credit History")
text(x =xx, y = as.numeric(table(german[3])), label = as.numeric(table(german[3])), pos = 3, cex
= 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[4]))))
xx <- barplot(table(german[4]), width = 0.85, ylim = ylim, ylab ="Purpose")
text(x =xx, y = as.numeric(table(german[4])), label = as.numeric(table(german[4])), pos = 3, cex
= 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[6]))))
xx <- barplot(table(german[6]), width = 0.85, ylim = ylim, ylab ="Saving Account")
text(x =xx, y = as.numeric(table(german[6])), label = as.numeric(table(german[6])), pos = 3, cex
= 0.8, col = "red")

title("Attributes' Barplots by Category", outer=TRUE)


oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,0.5,0.5), oma=c(0,0,2,0))
ylim <- c(0, 1.5*max(as.numeric(table(german[7]))))
xx <- barplot(table(german[7]), width = 0.85, ylim = ylim, ylab ="Employment")
text(x =xx, y = as.numeric(table(german[7])), label = as.numeric(table(german[7])), pos = 3, cex
= 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[9]))))
xx <- barplot(table(german[9]), width = 0.85, ylim = ylim, ylab ="Status")
text(x =xx, y = as.numeric(table(german[9])), label = as.numeric(table(german[9])), pos = 3, cex
= 0.8, col = "red")
ylim <- c(0, 1.5*max(as.numeric(table(german[10]))))
xx <- barplot(table(german[10]), width = 0.85, ylim = ylim, ylab ="Debtor Guarantor")
text(x =xx, y = as.numeric(table(german[10])), label = as.numeric(table(german[10])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[12]))))
xx <- barplot(table(german[12]), width = 0.85, ylim = ylim, ylab ="Property")
text(x =xx, y = as.numeric(table(german[12])), label = as.numeric(table(german[12])), pos = 3,
cex = 0.8, col = "red")

title("Attributes' Barplots by Category", outer=TRUE)


oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,0.5,0.5), oma=c(0,0,2,0))
ylim <- c(0, 1.5*max(as.numeric(table(german[14]))))
xx <- barplot(table(german[14]), width = 0.85, ylim = ylim, ylab ="Other Installment")
text(x =xx, y = as.numeric(table(german[14])), label = as.numeric(table(german[14])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[15]))))
xx <- barplot(table(german[15]), width = 0.85, ylim = ylim, ylab ="Housing")
text(x =xx, y = as.numeric(table(german[15])), label = as.numeric(table(german[15])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[17]))))
xx <- barplot(table(german[17]), width = 0.85, ylim = ylim, ylab ="Job")
text(x =xx, y = as.numeric(table(german[17])), label = as.numeric(table(german[17])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[19]))))
xx <- barplot(table(german[19]), width = 0.85, ylim = ylim, ylab ="Phone")
text(x =xx, y = as.numeric(table(german[19])), label = as.numeric(table(german[19])), pos = 3,
cex = 0.8, col = "red")

title("Attributes' Barplots by Category", outer=TRUE)
```

```
oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,0.5,0.5), oma=c(0,0,2,0))
ylim <- c(0, 1.5*max(as.numeric(table(german[8]))))
xx <- barplot(table(german[8]), width = 0.85, ylim = ylim, ylab ="Installment Rate")
text(x =xx, y = as.numeric(table(german[8])), label = as.numeric(table(german[8])), pos = 3, cex
= 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[11]))))
xx <- barplot(table(german[11]), width = 0.85, ylim = ylim, ylab ="Residence")
text(x =xx, y = as.numeric(table(german[11])), label = as.numeric(table(german[11])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[16]))))
xx <- barplot(table(german[16]), width = 0.85, ylim = ylim, ylab ="Existing Credit")
text(x =xx, y = as.numeric(table(german[16])), label = as.numeric(table(german[16])), pos = 3,
cex = 0.8, col = "red")

ylim <- c(0, 1.5*max(as.numeric(table(german[18]))))
xx <- barplot(table(german[18]), width = 0.85, ylim = ylim, ylab ="People Liable")
text(x =xx, y = as.numeric(table(german[18])), label = as.numeric(table(german[18])), pos = 3,
cex = 0.8, col = "red")

title("Attributes' Barplots by Category", outer=TRUE)


oldpar <- par(mfrow=c(1,1), mar=c(4.1,4.1,0.5,0.5), oma=c(0,0,2,0))
ylim <- c(0, 1.5*max(as.numeric(table(german[20]))))
xx <- barplot(table(german[20]), width = 0.85, ylim = ylim, ylab ="Foreign Worker")
text(x =xx, y = as.numeric(table(german[20])), label = as.numeric(table(german[20])), pos = 3,
cex = 0.8, col = "red")

title("Barplot for Foreign Workers", outer=TRUE)

#Histogram for Variables
histogram(~Duration+CreditAmount+Age, data=german,
          type="c", scales=list(relation="free"), breaks=NULL, main ="Histograms for Non-
Categorical Variables")

## Correlation Matrix
( which(sapply(german,function(x)class(x)[1]) == "integer") -> num )
(cormatrix <- cor(german[num],german[num]))
corrgram(german, order=TRUE, lower.panel=panel.shade,
         upper.panel=panel.pie, text.panel=panel.txt, main ="Correlation Matrix German Credit
Data")


#Non-significants
(nonsig <- colnames(german[nearZeroVar(german, freqCut = 90/10)]))

mosaicplot(~ Risk + ForeignWorker, data=german, main="Mosaic Plot of Risk vs ForeignWorker")
mosaicplot(~ Risk + DebtorGuarantor, data=german, main="Mosaic Plot of Risk vs DebtorGuarantor")

#possible Non-significants (80/20 cut)
mosaicplot(~ Risk + OtherInstallment, data=german, main="Mosaic Plot of Risk vs
OtherInstallment")
mosaicplot(~ Risk + PeopleLiable, data=german, main="Mosaic Plot of Risk vs PeopleLiable")


## DATA CLEANING
rcol <- which(names(german)%in%nonsig)
germanclean <- german[-rcol]


##DATA SPLIT

seed <- 12345
set.seed(seed)

trainIndex <- createDataPartition(germanclean$Risk, p = .6, list = FALSE,
                                  times = 1)
head(trainIndex)
```

```
germanTrain <- germanclean[ trainIndex,]
germanTest  <- germanclean[-trainIndex,]

table(germanTrain["Risk"])
table(germanTest["Risk"])

# Cost Structure Missclasification
(prop.table(table(germanclean[['Risk']])) -> prop )
(cost <- c(good=1, bad=5))
(newprior <- cost*prop )
(newprior <- as.vector(newprior/sum(newprior)))
(origprior <- as.vector(prop))

ptab <- cbind(newprior,origprior)
rownames(ptab) <- c("good","bad")
ptab

write.csv(ptab,"priors.csv")

##################### METHODS NEW PRIOR (Cost good = 1, Bad = 5)
results <- matrix(0,5,8)
colnames(results) <- c("Sensitivity","Specificity","ROC","ROCNEG","Type 1E","Type
2E","Accuracy","Cost")
rownames(results) <- c("LDA","QDA","NB","LR","LM")

seed <- as.integer(Sys.Date())

fitControl <- trainControl(method = "repeatedcv",
                           number = 5, repeats = 5,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)

## LDA
set.seed(seed)
ldaFit <- train(Risk ~ ., data=germanTrain, method="lda",
                prior=newprior,
                preProcess=c("center","scale"),
                trControl = fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(ldaFit, germanTest))
tabLDA <- table(germanTest[["Risk"]], predict(ldaFit, germanTest))

results[1,1] <- round(sensitivity(tabLDA), digits=2)
results[1,2] <- round(specificity(tabLDA), digits=2)
results[1,3] <- round(sensitivity(tabLDA) / (1-specificity(tabLDA)), digits=2)
results[1,4] <- round((1-sensitivity(tabLDA))/specificity(tabLDA), digits=2)
results[1,5] <- round(1-specificity(tabLDA), digits=2)
results[1,6] <- round(1-sensitivity(tabLDA), digits=2)
results[1,7] <- round(1 - sum(tabLDA[2:3])/sum(tabLDA), digits=2)
results[1,8] <- (tabLDA[2]*cost[2] + tabLDA[3]*cost[1])

## QDA
set.seed(seed)
qdaFit <- train(Risk ~ ., data=germanTrain, method="qda",
                prior=newprior,
                preProcess=c("center","scale"),
                trControl = fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(qdaFit, germanTest))
tabQDA <- table(germanTest[["Risk"]], predict(qdaFit, germanTest))

results[2,1] <- round(sensitivity(tabQDA), digits=2)
results[2,2] <- round(specificity(tabQDA), digits=2)
results[2,3] <- round(sensitivity(tabQDA) / (1-specificity(tabQDA)), digits=2)
results[2,4] <- round((1-sensitivity(tabQDA))/specificity(tabQDA), digits=2)
results[2,5] <- round(1-specificity(tabQDA), digits=2)
results[2,6] <- round(1-sensitivity(tabQDA), digits=2)
results[2,7] <- round(1 - sum(tabQDA[2:3])/sum(tabQDA), digits=2)
results[2,8] <- (tabQDA[2]*cost[2] + tabQDA[3]*cost[1])
```

```
## Naives Bayes
set.seed(seed)
nbGrid <- expand.grid(fL=c(0,0.5,1), usekernel=c(FALSE,TRUE))
nbFit <- train(Risk ~ ., data=germanTrain, method="nb",
               prior=newprior,
               preProcess=c("center","scale"),
               tuneGrid=nbGrid,
               trControl=fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(nbFit, germanTest))
tabNB <- table(germanTest[["Risk"]], predict(nbFit, germanTest))

results[3,1] <- round(sensitivity(tabNB), digits=2)
results[3,2] <- round(specificity(tabNB), digits=2)
results[3,3] <- round(sensitivity(tabNB) / (1-specificity(tabNB)), digits=2)
results[3,4] <- round((1-sensitivity(tabNB))/specificity(tabNB), digits=2)
results[3,5] <- round(1-specificity(tabNB), digits=2)
results[3,6] <- round(1-sensitivity(tabNB), digits=2)
results[3,7] <- round(1 - sum(tabNB[2:3])/sum(tabNB), digits=2)
results[3,8] <- (tabNB[2]*cost[2] + tabNB[3]*cost[1])

## Logistic Regression
set.seed(seed)
glmFit <- train(Risk ~ ., data=germanTrain, method="glm",
                family=binomial, weights=ifelse(Risk=="good",newprior[1], newprior[2]),
                preProcess=c("center","scale"),
                trControl=fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(glmFit, germanTest))
tabLR <- table(germanTest[["Risk"]], predict(glmFit, germanTest))

results[4,1] <- round(sensitivity(tabLR), digits=2)
results[4,2] <- round(specificity(tabLR), digits=2)
results[4,3] <- round(sensitivity(tabLR) / (1-specificity(tabLR)), digits=2)
results[4,4] <- round((1-sensitivity(tabLR))/specificity(tabLR), digits=2)
results[4,5] <- round(1-specificity(tabLR), digits=2)
results[4,6] <- round(1-sensitivity(tabLR), digits=2)
results[4,7] <- round(1 - sum(tabLR[2:3])/sum(tabLR), digits=2)
results[4,8] <- (tabLR[2]*cost[2] + tabLR[3]*cost[1])

## Linear Model
lm(nnet::class.ind(Risk) ~ ., data=germanTrain) -> m
pred <- predict(m, germanTest)
factor(apply(pred,1,function(x)which.max(x)[1]),
       label=levels(germanTest[['Risk']]))->pred
confusionMatrix(germanTest[['Risk']],pred)
tabLM <- table(germanTest[['Risk']],pred)

results[5,1] <- round(sensitivity(tabLM), digits=2)
results[5,2] <- round(specificity(tabLM), digits=2)
results[5,3] <- round(sensitivity(tabLM) / (1-specificity(tabLM)), digits=2)
results[5,4] <- round((1-sensitivity(tabLM))/specificity(tabLM), digits=2)
results[5,5] <- round(1-specificity(tabLM), digits=2)
results[5,6] <- round(1-sensitivity(tabLM), digits=2)
results[5,7] <- round(1 - sum(tabLM[2:3])/sum(tabLM), digits=2)
results[5,8] <- (tabLM[2]*cost[2] + tabLM[3]*cost[1])

write.csv(results, "resultsnewprio.csv")


#### Cross Validation Training Fitting Results
trellis.par.set(caretTheme())

# Between-Model Performance Analysis
( rs <- resamples(list(LDA=ldaFit, QDA=qdaFit,
                       GLM=glmFit, NB=nbFit)) )
summary(rs)
```

```
#Boxplot for Model Performance Analysis
theme1 <- trellis.par.get()
theme1$plot.symbol$col = rgb(.2, .2, .2, .4)
theme1$plot.symbol$pch = 16
theme1$plot.line$col = rgb(0, 0, 0.6, .7)
theme1$plot.line$lwd <- 2
trellis.par.set(theme1)

bwplot(rs, layout=c(3,1), pch="|", main = "Cross Validation Fits w/ Cost Adjustment",
       panel=function(x,y,...){
         panel.grid(h=-1, v=0)
         panel.bwplot(x, y, ...)
       }) -> p1
dimnames(p1)[[1]][-1] <- c("Sensitivity","Specificity")
dotplot(rs, metric="ROC") -> p2
plot(p1, split = c(1, 1, 1, 2))
plot(p2, split = c(1, 2, 1, 2), newpage = FALSE)

#Density Plot
densityplot(rs,auto.key = list(columns = 3),pch="|", main = "Density Plots Fits w/ Cost
Adjustment")

#Scatter Plots
splom(rs)
splom(rs, metric="Sens")
splom(rs, metric="Spec")


#Models Differences
(dif <- diff(rs) )
summary(dif)
bwplot(dif, layout=c(3,1), pch="|", main = "Fits Differences w/ Cost Adjustment",
       panel=function(x,y,...){
         panel.grid(h=-1, v=0)
         panel.bwplot(x, y, ...)
       }) -> p1
dimnames(p1)[[1]][-1] <- c("Sensitivity","Specificity")
dotplot(dif) -> p2
plot(p1, position=c(0,0.45,1,1))
plot(p2, position=c(0,0,1,0.5),newpage=FALSE)


## LM Cross Validation Training Fitting Results
createMultiFolds(germanTrain[['Risk']], k=10, times=5) -> pt
names(head(pt))

table(germanTrain[['Risk']][pt[[1]]])

a <- vector("list", length=5) -> p
resultlm <- matrix(0,5,4)
for (r in 1:5){
  i <- 10*(r-1)+(1:10)
  a[[r]] <- vector("list", length=10) -> p[[r]]
  for (f in i) {
    lm(nnet::class.ind(Risk) ~ ., data=germanTrain, subset=(1:600)[pt[[f]]]) -> m
    predict(m, germanTrain[-pt[[f]],-20])-> pred
    factor(apply(pred,1,function(x)which.max(x)[1]),
           label=levels(germanTrain[['Risk']]))->p[[r]][[f]]
    a[[r]][[f]] <- germanTrain[['Risk']][-pt[[f]]]
  }
  a[[r]] <- unlist(a[[r]]);  p[[r]] <- unlist(p[[r]])
  print(confusionMatrix(a[[r]],p[[r]]))
  tab <- table(p[[r]],a[[r]])
  resultlm[r,1] <- sensitivity(tab)
  resultlm[r,2] <- specificity(tab)
  resultlm[r,3] <- resultlm[r,1] / (1-resultlm[r,2])
  resultlm[r,4] <- 1 - sum(tab[2:3])/sum(tab)
}
colnames(resultlm) <- c("Sensitivity","Specificity","ROC","Accuracy")
resultlm
colMeans(resultlm)
```

```
################################ METHODS PRIOR (70/30)
results2 <- matrix(0,5,8)
colnames(results2) <- c("Sensitivity","Specificity","ROC","ROCNEG","Type 1E","Type
2E","Accuracy","Cost")
rownames(results2) <- c("LDA","QDA","NB","LR","LM")

## LDA
fitControl <- trainControl(method = "repeatedcv",
                           number = 5, repeats = 5,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)
set.seed(seed)
ldaFit <- train(Risk ~ ., data=germanTrain, method="lda",
                prior=origpior,
                preProcess=c("center","scale"),
                trControl = fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(ldaFit, germanTest))
tabLDA <- table(germanTest[["Risk"]], predict(ldaFit, germanTest))

results2[1,1] <- round(sensitivity(tabLDA), digits=2)
results2[1,2] <- round(specificity(tabLDA), digits=2)
results2[1,3] <- round(sensitivity(tabLDA) / (1-specificity(tabLDA)), digits=2)
results2[1,4] <- round((1-sensitivity(tabLDA))/specificity(tabLDA), digits=2)
results2[1,5] <- round(1-specificity(tabLDA), digits=2)
results2[1,6] <- round(1-sensitivity(tabLDA), digits=2)
results2[1,7] <- round(1 - sum(tabLDA[2:3])/sum(tabLDA), digits=2)
results2[1,8] <- (tabLDA[2]*cost[2] + tabLDA[3]*cost[1])

## QDA
set.seed(seed)
qdaFit <- train(Risk ~ ., data=germanTrain, method="qda",
                prior=origprior,
                preProcess=c("center","scale"),
                trControl = fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(qdaFit, germanTest))
tabQDA <- table(germanTest[["Risk"]], predict(qdaFit, germanTest))

results2[2,1] <- round(sensitivity(tabQDA), digits=2)
results2[2,2] <- round(specificity(tabQDA), digits=2)
results2[2,3] <- round(sensitivity(tabQDA) / (1-specificity(tabQDA)), digits=2)
results2[2,4] <- round((1-sensitivity(tabQDA))/specificity(tabQDA), digits=2)
results2[2,5] <- round(1-specificity(tabQDA), digits=2)
results2[2,6] <- round(1-sensitivity(tabQDA), digits=2)
results2[2,7] <- round(1 - sum(tabQDA[2:3])/sum(tabQDA), digits=2)
results2[2,8] <- (tabQDA[2]*cost[2] + tabQDA[3]*cost[1])

## Naives Bayes
set.seed(seed)
nbGrid <- expand.grid(fL=c(0,0.5,1), usekernel=c(FALSE,TRUE))
nbFit <- train(Risk ~ ., data=germanTrain, method="nb",
               prior=origprior,
               preProcess=c("center","scale"),
               tuneGrid=nbGrid,
               trControl=fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(nbFit, germanTest))
tabNB <- table(germanTest[["Risk"]], predict(nbFit, germanTest))

results2[3,1] <- round(sensitivity(tabNB), digits=2)
results2[3,2] <- round(specificity(tabNB), digits=2)
results2[3,3] <- round(sensitivity(tabNB) / (1-specificity(tabNB)), digits=2)
results2[3,4] <- round((1-sensitivity(tabNB))/specificity(tabNB), digits=2)
results2[3,5] <- round(1-specificity(tabNB), digits=2)
results2[3,6] <- round(1-sensitivity(tabNB), digits=2)
results2[3,7] <- round(1 - sum(tabNB[2:3])/sum(tabNB), digits=2)
results2[3,8] <- (tabNB[2]*cost[2] + tabNB[3]*cost[1])
```

```
## Logistic Regression
set.seed(seed)
glmFit <- train(Risk ~ ., data=germanTrain, method="glm",
                family=binomial, weights=ifelse(Risk=="good",origprior[1],origprior[2]),
                preProcess=c("center","scale"),
                trControl=fitControl, metric="ROC")
confusionMatrix(germanTest[["Risk"]], predict(glmFit, germanTest))
tabLR <- table(germanTest[["Risk"]], predict(glmFit, germanTest))

results2[4,1] <- round(sensitivity(tabLR), digits=2)
results2[4,2] <- round(specificity(tabLR), digits=2)
results2[4,3] <- round(sensitivity(tabLR) / (1-specificity(tabLR)), digits=2)
results2[4,4] <- round((1-sensitivity(tabLR))/specificity(tabLR), digits=2)
results2[4,5] <- round(1-specificity(tabLR), digits=2)
results2[4,6] <- round(1-sensitivity(tabLR), digits=2)
results2[4,7] <- round(1 - sum(tabLR[2:3])/sum(tabLR), digits=2)
results2[4,8] <- (tabLR[2]*cost[2] + tabLR[3]*cost[1])

## Linear Model
lm(nnet::class.ind(Risk) ~ ., data=germanTrain) -> m
pred <- predict(m, germanTest)
factor(apply(pred,1,function(x)which.max(x)[1]),
       label=levels(germanTest[['Risk']]))->pred
confusionMatrix(germanTest[['Risk']],pred)
tabLM <- table(germanTest[['Risk']],pred)

results2[5,1] <- round(sensitivity(tabLM), digits=2)
results2[5,2] <- round(specificity(tabLM), digits=2)
results2[5,3] <- round(sensitivity(tabLM) / (1-specificity(tabLM)), digits=2)
results2[5,4] <- round((1-sensitivity(tabLM))/specificity(tabLM), digits=2)
results2[5,5] <- round(1-specificity(tabLM), digits=2)
results2[5,6] <- round(1-sensitivity(tabLM), digits=2)
results2[5,7] <- round(1 - sum(tabLM[2:3])/sum(tabLM), digits=2)
results2[5,8] <- (tabLM[2]*cost[2] + tabLM[3]*cost[1])

write.csv(results2, "resultsorigprio.csv")

## Cross Validation Training Fitting Results
trellis.par.set(caretTheme())

# Between-Model Performance Analysis
(rs2 <- resamples(list(LDA=ldaFit, QDA=qdaFit,
                       GLM=glmFit, NB=nbFit)) )
summary(rs2)

theme1 <- trellis.par.get()
theme1$plot.symbol$col = rgb(.2, .2, .2, .4)
theme1$plot.symbol$pch = 16
theme1$plot.line$col = rgb(0, 0, 0.6, .7)
theme1$plot.line$lwd <- 2
trellis.par.set(theme1)

bwplot(rs2, layout=c(3,1), pch="|", main = "Cross Validation Fits w/o Cost Adjustment",
       panel=function(x,y,...){
         panel.grid(h=-1, v=0)
         panel.bwplot(x, y, ...)
       }) -> p1
dimnames(p1)[[1]][-1] <- c("Sensitivity","Specificity")
dotplot(rs2, metric="ROC") -> p2
plot(p1, split = c(1, 1, 1, 2))
plot(p2, split = c(1, 2, 1, 2), newpage = FALSE)

#Density Plot
densityplot(rs2,auto.key = list(columns = 3),pch="|", main = "Density Plots Fits w/o Cost
Adjustment")


#Scatter Plots
splom(rs2)
splom(rs2, metric="Sens")
splom(rs2, metric="Spec")
```

```
#Models Differences
(dif <- diff(rs2) )
summary(dif)
bwplot(dif, layout=c(3,1), pch="|", main = "Fits Differences w/o Cost Adjustment",
       panel=function(x,y,...){
         panel.grid(h=-1, v=0)
         panel.bwplot(x, y, ...)
       }) -> p1
dimnames(p1)[[1]][-1] <- c("Sensitivity","Specificity")
dotplot(dif) -> p2
plot(p1, position=c(0,0.45,1,1))
plot(p2, position=c(0,0,1,0.5),newpage=FALSE)


## Table Results
results  # Prior (Cost Weight)
results2 # Prior (70/30)

xx <- rownames(results)

#PLOT NEW PRIOR
oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,1.5,0.5))
plot(results[,1], pch=16, type="o", col="red", ylab=colnames(results)[1],
     xaxt = "n", xlab = "", main = "Sensitivity Analysis")
axis(1, 1:5, labels= xx)
plot(results[,2], pch=16, type="o", col="red", ylab=colnames(results)[2],
     xaxt = "n", xlab = "", main = "Specificity Analysis")
axis(1, 1:5, labels= xx)
plot(results[,7], pch=16, type="o", col="red", ylab=colnames(results)[7],
     xaxt = "n", xlab = "", main = "Accuracy Analysis")
axis(1, 1:5, labels= xx)
plot(results[,8], pch=16, type="o", col="red", ylab=colnames(results)[8],
     xaxt = "n", xlab = "", main = "Cost Analysis")
axis(1, 1:5, labels= xx)


#PLOT ORIG PRIOR
oldpar <- par(mfrow=c(2,2), mar=c(4.1,4.1,1.5,0.5))
plot(results2[,1], pch=16, type="o", col="red", ylab=colnames(results2)[1],
     xaxt = "n", xlab = "", main = "Sensitivity Analysis")
axis(1, 1:5, labels= xx)
plot(results2[,2], pch=16, type="o", col="red", ylab=colnames(results2)[2],
     xaxt = "n", xlab = "", main = "Specificity Analysis")
axis(1, 1:5, labels= xx)
plot(results2[,7], pch=16, type="o", col="red", ylab=colnames(results2)[7],
     xaxt = "n", xlab = "", main = "Accuracy Analysis")
axis(1, 1:5, labels= xx)
plot(results2[,8], pch=16, type="o", col="red", ylab=colnames(results2)[8],
     xaxt = "n", xlab = "", main = "Cost Analysis")
axis(1, 1:5, labels= xx)
```