Homework 6

**Chinki**

**May 8, 2017**

**1.Logistic Regression analysis of the challenger data**

**Step 1: Collecting the data**

we will use data donated to the UCI Machine Learning Data Repository. The United States space shuttle Challenger were killed when a rocket booster failed, causing a catastrophic disintegration. Data contains distress\_ct, temperature, field\_check\_pressure & flight\_num.

#Reading the dataset  
launch <- read.csv("http://www.sci.csueastbay.edu/~esuess/classes/Statistics\_6620/Presentations/ml10/challenger.csv")

**Step 2: Exploring and preparing the data**

Getting structure of the dataset

# examine the launch data  
str(launch)

## 'data.frame': 23 obs. of 4 variables:  
## $ distress\_ct : int 0 1 0 0 0 0 0 0 1 1 ...  
## $ temperature : int 66 70 69 68 67 72 73 70 57 63 ...  
## $ field\_check\_pressure: int 50 50 50 50 50 50 100 100 200 200 ...  
## $ flight\_num : int 1 2 3 4 5 6 7 8 9 10 ...

All the foure variables are int type.

# First recode the distress\_ct variable into 0 and 1, making 1 to represent at least  
launch$distress\_ct = ifelse(launch$distress\_ct<1,0,1)  
launch$distress\_ct

## [1] 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1

# Creating random sampling  
indx = sample(1:nrow(launch), as.integer(0.9\*nrow(launch)))  
indx

## [1] 6 14 11 5 18 21 2 23 10 7 22 16 19 3 8 20 1 13 12 15

#Creating training & test data set  
launch\_train = launch[indx,]  
launch\_test = launch[-indx,]

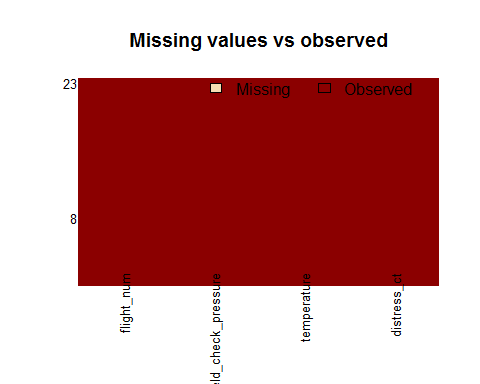
#Creating labels  
launch\_train\_labels = launch[indx,1]  
launch\_test\_labels = launch[-indx,1]

# Check if there are any missing values  
  
library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.4, built: 2015-12-05)  
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

missmap(launch, main = "Missing values vs observed")



# number of missing values in each column  
sapply(launch,function(x) sum(is.na(x)))

## distress\_ct temperature field\_check\_pressure   
## 0 0 0   
## flight\_num   
## 0

# number of unique values in each column  
sapply(launch, function(x) length(unique(x)))

## distress\_ct temperature field\_check\_pressure   
## 2 16 3   
## flight\_num   
## 23

**Step 3: Training a model on the data**

Applying Regression model to all variables.

# fit the logistic regression model, with all predictor variables  
model <- glm(distress\_ct ~.,family=binomial(link='logit'),data=launch\_train)  
model

##   
## Call: glm(formula = distress\_ct ~ ., family = binomial(link = "logit"),   
## data = launch\_train)  
##   
## Coefficients:  
## (Intercept) temperature field\_check\_pressure   
## 12.430550 -0.212933 0.010425   
## flight\_num   
## -0.006711   
##   
## Degrees of Freedom: 19 Total (i.e. Null); 16 Residual  
## Null Deviance: 24.43   
## Residual Deviance: 17.34 AIC: 25.34

AIC is 24.33, to improve model performance we will look for lower AIC than 24.33.

#Getting summary of model  
summary(model)

##   
## Call:  
## glm(formula = distress\_ct ~ ., family = binomial(link = "logit"),   
## data = launch\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2468 -0.6225 -0.4704 0.3441 2.0484   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 12.430550 8.261321 1.505 0.1324   
## temperature -0.212933 0.118254 -1.801 0.0718 .  
## field\_check\_pressure 0.010425 0.017158 0.608 0.5435   
## flight\_num -0.006711 0.179234 -0.037 0.9701   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 24.435 on 19 degrees of freedom  
## Residual deviance: 17.338 on 16 degrees of freedom  
## AIC: 25.338  
##   
## Number of Fisher Scoring iterations: 5

None of parameter is significant.

**Step 4: Evaluating model performance**

#ANOVA of the model  
anova(model, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: distress\_ct  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 19 24.435   
## temperature 1 5.8273 18 18.607 0.01578 \*  
## field\_check\_pressure 1 1.2675 17 17.340 0.26023   
## flight\_num 1 0.0014 16 17.338 0.97020   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Temperature is significant in the model.

# drop the insignificant predictors, alpha = 0.10  
model <- glm(distress\_ct ~ temperature,family=binomial(link='logit'),data=launch\_train)  
model

##   
## Call: glm(formula = distress\_ct ~ temperature, family = binomial(link = "logit"),   
## data = launch\_train)  
##   
## Coefficients:  
## (Intercept) temperature   
## 13.5701 -0.2086   
##   
## Degrees of Freedom: 19 Total (i.e. Null); 18 Residual  
## Null Deviance: 24.43   
## Residual Deviance: 18.61 AIC: 22.61

#Creating summary of the model  
summary(model)

##   
## Call:  
## glm(formula = distress\_ct ~ temperature, family = binomial(link = "logit"),   
## data = launch\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.0954 -0.7994 -0.4410 0.4563 2.0936   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 13.5701 7.5377 1.800 0.0718 .  
## temperature -0.2086 0.1095 -1.906 0.0567 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 24.435 on 19 degrees of freedom  
## Residual deviance: 18.607 on 18 degrees of freedom  
## AIC: 22.607  
##   
## Number of Fisher Scoring iterations: 5

Temperature is significant.

#ANOVA to check model performance  
anova(model, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: distress\_ct  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 19 24.435   
## temperature 1 5.8273 18 18.607 0.01578 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# check Accuracy  
fitted.results <- predict(model,newdata=launch\_test,type='response')  
fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results != launch\_test$distress\_ct)  
print(paste('Accuracy',1-misClasificError))

## [1] "Accuracy 1"

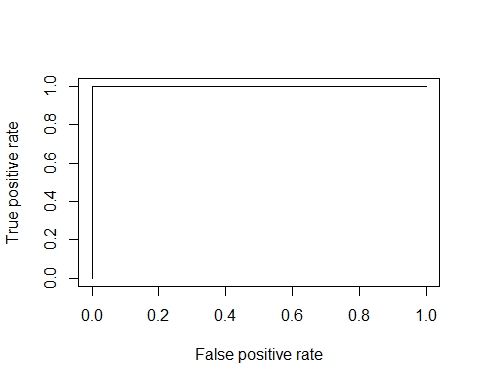
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

p <- predict(model, newdata=launch\_test, type="response")  
pr <- prediction(p,launch\_test$distress\_ct)  
prf <- performance(pr, measure = "tpr", x.measure = "fpr")  
plot(prf)



auc <- performance(pr, measure = "auc")  
auc <- auc@y.values[[1]]  
auc

## [1] 1

As I am getting score 0.5 which means No discrimination.

1. **Logistic Regression analysis of the credit data**

**Step 1: Collecting the data**

source of the data <http://archive.ics.uci.edu/ml.The> dataset contains information on loans obtained from a credit agency in Germany.The credit dataset includes 1,000 examples on loans, plus a set of numeric and nominal features indicating the characteristics of the loan and the loan applicant. A class variable indicates whether the loan went into default.

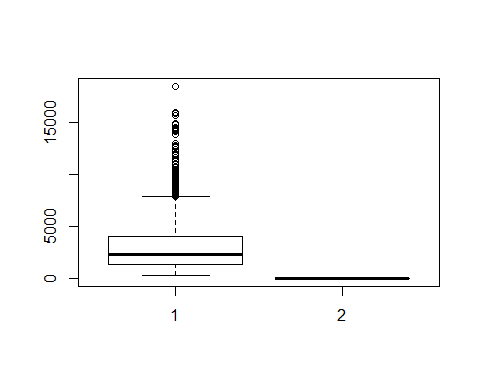
**Step 2: Exploring and preparing the data**

#Reading the data  
credit <- read.csv("credit.csv")

#Examin structure of the data  
str(credit)

## 'data.frame': 1000 obs. of 17 variables:  
## $ checking\_balance : Factor w/ 4 levels "< 0 DM","> 200 DM",..: 1 3 4 1 1 4 4 3 4 3 ...  
## $ months\_loan\_duration: int 6 48 12 42 24 36 24 36 12 30 ...  
## $ credit\_history : Factor w/ 5 levels "critical","good",..: 1 2 1 2 4 2 2 2 2 1 ...  
## $ purpose : Factor w/ 6 levels "business","car",..: 5 5 4 5 2 4 5 2 5 2 ...  
## $ amount : int 1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...  
## $ savings\_balance : Factor w/ 5 levels "< 100 DM","> 1000 DM",..: 5 1 1 1 1 5 4 1 2 1 ...  
## $ employment\_duration : Factor w/ 5 levels "< 1 year","> 7 years",..: 2 3 4 4 3 3 2 3 4 5 ...  
## $ percent\_of\_income : int 4 2 2 2 3 2 3 2 2 4 ...  
## $ years\_at\_residence : int 4 2 3 4 4 4 4 2 4 2 ...  
## $ age : int 67 22 49 45 53 35 53 35 61 28 ...  
## $ other\_credit : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ housing : Factor w/ 3 levels "other","own",..: 2 2 2 1 1 1 2 3 2 2 ...  
## $ existing\_loans\_count: int 2 1 1 1 2 1 1 1 1 2 ...  
## $ job : Factor w/ 4 levels "management","skilled",..: 2 2 4 2 2 4 2 1 4 1 ...  
## $ dependents : int 1 1 2 2 2 2 1 1 1 1 ...  
## $ phone : Factor w/ 2 levels "no","yes": 2 1 1 1 1 2 1 2 1 1 ...  
## $ default : Factor w/ 2 levels "no","yes": 1 2 1 1 2 1 1 1 1 2 ...

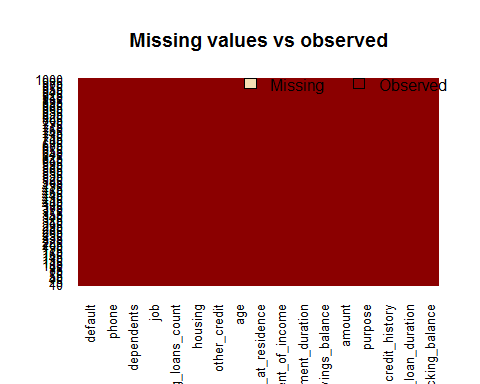
#Boxplot of loan duration & amout  
boxplot(credit$months\_loan\_duration+credit$amount,credit$default)



#Training & test dataset ups  
index=sample(1:nrow(credit),as.integer(0.9\*nrow(credit)))

#Creating training & test data  
training=credit[index,]  
test=credit[-index,]  
  
#Creating levels for training & test data  
training\_labels=credit[index,17]  
test\_labels=credit[-index,17]

#Checking for missing values   
library(Amelia)  
missmap(credit, main = "Missing values vs observed")



# number of missing values in each column  
sapply(credit,function(x) sum(is.na(x)))

## checking\_balance months\_loan\_duration credit\_history   
## 0 0 0   
## purpose amount savings\_balance   
## 0 0 0   
## employment\_duration percent\_of\_income years\_at\_residence   
## 0 0 0   
## age other\_credit housing   
## 0 0 0   
## existing\_loans\_count job dependents   
## 0 0 0   
## phone default   
## 0 0

# number of unique values in each column  
sapply(credit, function(x) length(unique(x)))

## checking\_balance months\_loan\_duration credit\_history   
## 4 33 5   
## purpose amount savings\_balance   
## 6 921 5   
## employment\_duration percent\_of\_income years\_at\_residence   
## 5 4 4   
## age other\_credit housing   
## 53 3 3   
## existing\_loans\_count job dependents   
## 4 4 2   
## phone default   
## 2 2

# fit the logistic regression model, with all predictor variables  
model=glm(default~. ,family = binomial(link='logit'),data=training)  
model

##   
## Call: glm(formula = default ~ ., family = binomial(link = "logit"),   
## data = training)  
##   
## Coefficients:  
## (Intercept) checking\_balance> 200 DM   
## -1.0958049 -0.8406957   
## checking\_balance1 - 200 DM checking\_balanceunknown   
## -0.4563032 -1.7485868   
## months\_loan\_duration credit\_historygood   
## 0.0274463 0.9512670   
## credit\_historyperfect credit\_historypoor   
## 1.5132722 0.7754412   
## credit\_historyvery good purposecar   
## 1.5305203 0.2715367   
## purposecar0 purposeeducation   
## -0.7238229 0.4610626   
## purposefurniture/appliances purposerenovations   
## -0.1912292 0.4825717   
## amount savings\_balance> 1000 DM   
## 0.0001035 -0.7435116   
## savings\_balance100 - 500 DM savings\_balance500 - 1000 DM   
## -0.0689413 -0.3944426   
## savings\_balanceunknown employment\_duration> 7 years   
## -0.6633038 -0.5757302   
## employment\_duration1 - 4 years employment\_duration4 - 7 years   
## -0.2849664 -0.9746603   
## employment\_durationunemployed percent\_of\_income   
## -0.1692971 0.2169001   
## years\_at\_residence age   
## -0.0103225 -0.0125848   
## other\_creditnone other\_creditstore   
## -0.6299381 -0.3876587   
## housingown housingrent   
## -0.1553603 0.2364133   
## existing\_loans\_count jobskilled   
## 0.3334848 -0.1015442   
## jobunemployed jobunskilled   
## -0.5122173 -0.2622086   
## dependents phoneyes   
## -0.0556588 -0.3303436   
##   
## Degrees of Freedom: 899 Total (i.e. Null); 864 Residual  
## Null Deviance: 1086   
## Residual Deviance: 848.6 AIC: 920.6

#Summary of regression model  
summary(model)

##   
## Call:  
## glm(formula = default ~ ., family = binomial(link = "logit"),   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1425 -0.7605 -0.4090 0.8005 2.7066   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.0958049 0.9422424 -1.163 0.244839   
## checking\_balance> 200 DM -0.8406957 0.3716788 -2.262 0.023704 \*   
## checking\_balance1 - 200 DM -0.4563032 0.2169273 -2.103 0.035423 \*   
## checking\_balanceunknown -1.7485868 0.2370507 -7.376 1.63e-13 \*\*\*  
## months\_loan\_duration 0.0274463 0.0093620 2.932 0.003371 \*\*   
## credit\_historygood 0.9512670 0.2703072 3.519 0.000433 \*\*\*  
## credit\_historyperfect 1.5132721 0.4507568 3.357 0.000787 \*\*\*  
## credit\_historypoor 0.7754412 0.3354398 2.312 0.020793 \*   
## credit\_historyvery good 1.5305203 0.4380870 3.494 0.000476 \*\*\*  
## purposecar 0.2715367 0.3290342 0.825 0.409228   
## purposecar0 -0.7238229 0.7778061 -0.931 0.352063   
## purposeeducation 0.4610626 0.4581461 1.006 0.314240   
## purposefurniture/appliances -0.1912292 0.3236521 -0.591 0.554622   
## purposerenovations 0.4825717 0.5814478 0.830 0.406568   
## amount 0.0001035 0.0000426 2.429 0.015147 \*   
## savings\_balance> 1000 DM -0.7435116 0.5059447 -1.470 0.141683   
## savings\_balance100 - 500 DM -0.0689413 0.2848400 -0.242 0.808753   
## savings\_balance500 - 1000 DM -0.3944426 0.4369490 -0.903 0.366675   
## savings\_balanceunknown -0.6633038 0.2568516 -2.582 0.009810 \*\*   
## employment\_duration> 7 years -0.5757302 0.3015555 -1.909 0.056236 .   
## employment\_duration1 - 4 years -0.2849664 0.2426601 -1.174 0.240257   
## employment\_duration4 - 7 years -0.9746603 0.3071069 -3.174 0.001505 \*\*   
## employment\_durationunemployed -0.1692971 0.4327697 -0.391 0.695654   
## percent\_of\_income 0.2169001 0.0873547 2.483 0.013029 \*   
## years\_at\_residence -0.0103225 0.0871839 -0.118 0.905751   
## age -0.0125848 0.0092814 -1.356 0.175125   
## other\_creditnone -0.6299381 0.2449884 -2.571 0.010132 \*   
## other\_creditstore -0.3876587 0.4240180 -0.914 0.360585   
## housingown -0.1553603 0.3005551 -0.517 0.605218   
## housingrent 0.2364133 0.3454862 0.684 0.493791   
## existing\_loans\_count 0.3334848 0.1950019 1.710 0.087236 .   
## jobskilled -0.1015442 0.2854258 -0.356 0.722017   
## jobunemployed -0.5122173 0.6525939 -0.785 0.432515   
## jobunskilled -0.2622086 0.3488545 -0.752 0.452275   
## dependents -0.0556587 0.2440766 -0.228 0.819617   
## phoneyes -0.3303436 0.2050906 -1.611 0.107241   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1085.66 on 899 degrees of freedom  
## Residual deviance: 848.59 on 864 degrees of freedom  
## AIC: 920.59  
##   
## Number of Fisher Scoring iterations: 5

#Evaluvationg model performance  
anova(model, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: default  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 899 1085.66   
## checking\_balance 3 114.800 896 970.86 < 2.2e-16 \*\*\*  
## months\_loan\_duration 1 37.936 895 932.92 7.310e-10 \*\*\*  
## credit\_history 4 28.332 891 904.59 1.068e-05 \*\*\*  
## purpose 5 6.449 886 898.14 0.26492   
## amount 1 2.067 885 896.07 0.15051   
## savings\_balance 4 10.381 881 885.69 0.03448 \*   
## employment\_duration 4 12.566 877 873.13 0.01360 \*   
## percent\_of\_income 1 5.853 876 867.28 0.01555 \*   
## years\_at\_residence 1 0.014 875 867.26 0.90529   
## age 1 2.793 874 864.47 0.09467 .   
## other\_credit 2 7.098 872 857.37 0.02875 \*   
## housing 2 2.998 870 854.37 0.22332   
## existing\_loans\_count 1 2.542 869 851.83 0.11084   
## job 3 0.577 866 851.25 0.90170   
## dependents 1 0.036 865 851.22 0.84861   
## phone 1 2.625 864 848.59 0.10518   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Droping insignificant predictor  
model1=glm(default~checking\_balance+months\_loan\_duration+credit\_history+savings\_balance+employment\_duration+percent\_of\_income+age,family = binomial(link = 'logit'),data=training)  
model1

##   
## Call: glm(formula = default ~ checking\_balance + months\_loan\_duration +   
## credit\_history + savings\_balance + employment\_duration +   
## percent\_of\_income + age, family = binomial(link = "logit"),   
## data = training)  
##   
## Coefficients:  
## (Intercept) checking\_balance> 200 DM   
## -1.080534 -1.020661   
## checking\_balance1 - 200 DM checking\_balanceunknown   
## -0.457750 -1.732459   
## months\_loan\_duration credit\_historygood   
## 0.039777 0.641227   
## credit\_historyperfect credit\_historypoor   
## 1.547142 0.702438   
## credit\_historyvery good savings\_balance> 1000 DM   
## 1.494873 -0.663026   
## savings\_balance100 - 500 DM savings\_balance500 - 1000 DM   
## -0.018385 -0.567252   
## savings\_balanceunknown employment\_duration> 7 years   
## -0.566156 -0.502043   
## employment\_duration1 - 4 years employment\_duration4 - 7 years   
## -0.291079 -0.909395   
## employment\_durationunemployed percent\_of\_income   
## -0.162527 0.112400   
## age   
## -0.009475   
##   
## Degrees of Freedom: 899 Total (i.e. Null); 881 Residual  
## Null Deviance: 1086   
## Residual Deviance: 879.8 AIC: 917.8

#summary of model1  
summary(model1)

##   
## Call:  
## glm(formula = default ~ checking\_balance + months\_loan\_duration +   
## credit\_history + savings\_balance + employment\_duration +   
## percent\_of\_income + age, family = binomial(link = "logit"),   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7513 -0.7586 -0.4494 0.8696 2.5707   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.080534 0.472744 -2.286 0.022274 \*   
## checking\_balance> 200 DM -1.020661 0.360969 -2.828 0.004691 \*\*   
## checking\_balance1 - 200 DM -0.457750 0.207494 -2.206 0.027378 \*   
## checking\_balanceunknown -1.732459 0.227729 -7.608 2.79e-14 \*\*\*  
## months\_loan\_duration 0.039777 0.007018 5.668 1.45e-08 \*\*\*  
## credit\_historygood 0.641227 0.218098 2.940 0.003281 \*\*   
## credit\_historyperfect 1.547142 0.425052 3.640 0.000273 \*\*\*  
## credit\_historypoor 0.702438 0.320381 2.193 0.028343 \*   
## credit\_historyvery good 1.494873 0.388149 3.851 0.000117 \*\*\*  
## savings\_balance> 1000 DM -0.663026 0.489521 -1.354 0.175596   
## savings\_balance100 - 500 DM -0.018385 0.275577 -0.067 0.946810   
## savings\_balance500 - 1000 DM -0.567252 0.431508 -1.315 0.188651   
## savings\_balanceunknown -0.566156 0.245287 -2.308 0.020991 \*   
## employment\_duration> 7 years -0.502043 0.280403 -1.790 0.073384 .   
## employment\_duration1 - 4 years -0.291079 0.235080 -1.238 0.215638   
## employment\_duration4 - 7 years -0.909395 0.295409 -3.078 0.002081 \*\*   
## employment\_durationunemployed -0.162527 0.371989 -0.437 0.662174   
## percent\_of\_income 0.112400 0.076307 1.473 0.140751   
## age -0.009475 0.008464 -1.119 0.262927   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1085.7 on 899 degrees of freedom  
## Residual deviance: 879.8 on 881 degrees of freedom  
## AIC: 917.8  
##   
## Number of Fisher Scoring iterations: 5

#Evaluvating model performance  
anova(model1, test="Chisq")

## Analysis of Deviance Table  
##   
## Model: binomial, link: logit  
##   
## Response: default  
##   
## Terms added sequentially (first to last)  
##   
##   
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)   
## NULL 899 1085.66   
## checking\_balance 3 114.800 896 970.86 < 2.2e-16 \*\*\*  
## months\_loan\_duration 1 37.936 895 932.92 7.310e-10 \*\*\*  
## credit\_history 4 28.332 891 904.59 1.068e-05 \*\*\*  
## savings\_balance 4 9.391 887 895.20 0.05203 .   
## employment\_duration 4 11.990 883 883.21 0.01743 \*   
## percent\_of\_income 1 2.141 882 881.07 0.14344   
## age 1 1.269 881 879.80 0.25993   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Age is not significant so we can drop age as well.

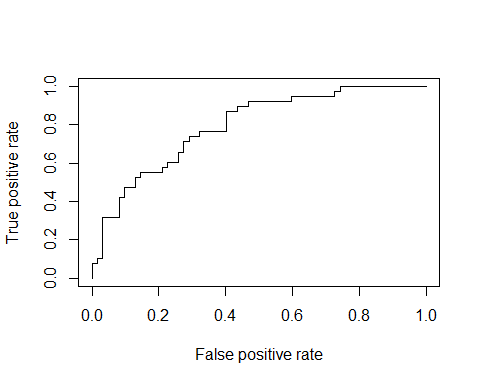
#Checking Accuracy  
fitted.results <- predict(model1,newdata=test,type='response')  
fitted.results <- ifelse(fitted.results > 0.5,1,0)

misClasificError <- mean(fitted.results != test$default)  
print(paste('Accuracy',1-misClasificError))

## [1] "Accuracy 0"

As data is very small so test data does not contains both 0 & 1 values.

#Roc curve  
  
library(ROCR)  
p <- predict(model, newdata=test, type="response")  
pr <- prediction(p, test$default)  
prf <- performance(pr, measure = "tpr", x.measure = "fpr")  
plot(prf)



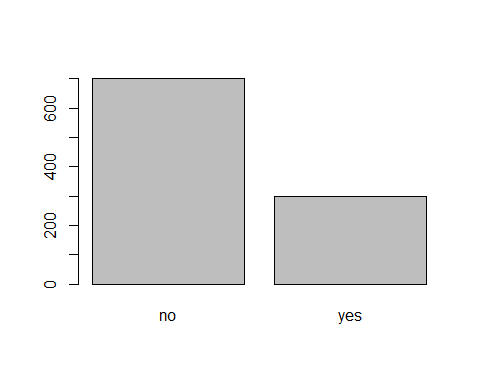
#Getting AUC values  
auc <- performance(pr, measure = "auc")  
auc <- auc@y.values[[1]]  
auc

## [1] 0.7924448

As auc value is 0.78 it is acceptable.

Classification Trees

# the distribution of defaults  
plot(credit$default)



# regression tree using rpart  
library(rpart)  
m.rpart <- rpart(default ~ ., data = training)

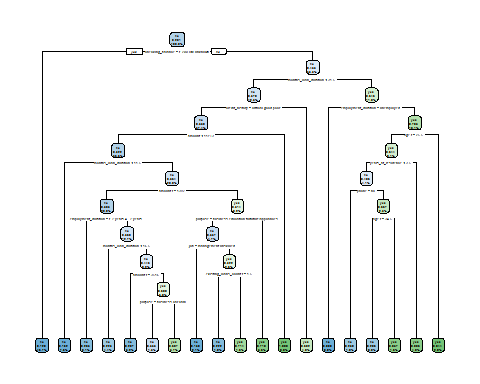
# get basic information about the tree  
m.rpart

## n= 900   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 900 262 no (0.70888889 0.29111111)   
## 2) checking\_balance=> 200 DM,unknown 420 54 no (0.87142857 0.12857143) \*  
## 3) checking\_balance=< 0 DM,1 - 200 DM 480 208 no (0.56666667 0.43333333)   
## 6) months\_loan\_duration< 31.5 381 144 no (0.62204724 0.37795276)   
## 12) credit\_history=critical,good,poor 337 114 no (0.66172107 0.33827893)   
## 24) amount< 10975.5 329 106 no (0.67781155 0.32218845)   
## 48) months\_loan\_duration< 11.5 68 11 no (0.83823529 0.16176471) \*  
## 49) months\_loan\_duration>=11.5 261 95 no (0.63601533 0.36398467)   
## 98) amount>=1381.5 187 57 no (0.69518717 0.30481283)   
## 196) employment\_duration=> 7 years,4 - 7 years 73 15 no (0.79452055 0.20547945) \*  
## 197) employment\_duration=< 1 year,1 - 4 years,unemployed 114 42 no (0.63157895 0.36842105)   
## 394) months\_loan\_duration< 16.5 40 9 no (0.77500000 0.22500000) \*  
## 395) months\_loan\_duration>=16.5 74 33 no (0.55405405 0.44594595)   
## 790) amount>=3515.5 29 6 no (0.79310345 0.20689655) \*  
## 791) amount< 3515.5 45 18 yes (0.40000000 0.60000000)   
## 1582) purpose=business,car,car0 12 4 no (0.66666667 0.33333333) \*  
## 1583) purpose=education,furniture/appliances 33 10 yes (0.30303030 0.69696970) \*  
## 99) amount< 1381.5 74 36 yes (0.48648649 0.51351351)   
## 198) purpose=business,education,furniture/appliances 42 15 no (0.64285714 0.35714286)   
## 396) job=management,unskilled 19 3 no (0.84210526 0.15789474) \*  
## 397) job=skilled,unemployed 23 11 yes (0.47826087 0.52173913)   
## 794) existing\_loans\_count>=1.5 9 2 no (0.77777778 0.22222222) \*  
## 795) existing\_loans\_count< 1.5 14 4 yes (0.28571429 0.71428571) \*  
## 199) purpose=car,renovations 32 9 yes (0.28125000 0.71875000) \*  
## 25) amount>=10975.5 8 0 yes (0.00000000 1.00000000) \*  
## 13) credit\_history=perfect,very good 44 14 yes (0.31818182 0.68181818) \*  
## 7) months\_loan\_duration>=31.5 99 35 yes (0.35353535 0.64646465)   
## 14) employment\_duration=unemployed 8 0 no (1.00000000 0.00000000) \*  
## 15) employment\_duration=< 1 year,> 7 years,1 - 4 years,4 - 7 years 91 27 yes (0.29670330 0.70329670)   
## 30) age>=25.5 73 26 yes (0.35616438 0.64383562)   
## 60) years\_at\_residence< 3.5 37 18 no (0.51351351 0.48648649)   
## 120) phone=no 16 4 no (0.75000000 0.25000000) \*  
## 121) phone=yes 21 7 yes (0.33333333 0.66666667)   
## 242) age>=34.5 7 2 no (0.71428571 0.28571429) \*  
## 243) age< 34.5 14 2 yes (0.14285714 0.85714286) \*  
## 61) years\_at\_residence>=3.5 36 7 yes (0.19444444 0.80555556) \*  
## 31) age< 25.5 18 1 yes (0.05555556 0.94444444) \*

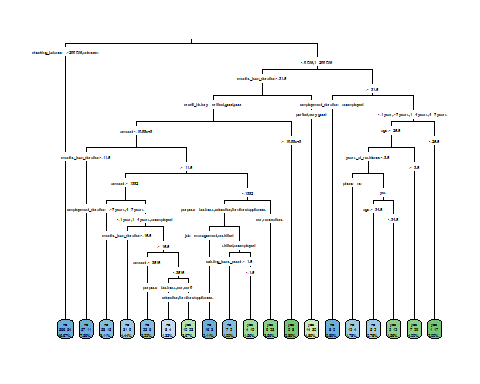
# get more detailed information about the tree  
summary(m.rpart)

## Call:  
## rpart(formula = default ~ ., data = training)  
## n= 900   
##   
## CP nsplit rel error xerror xstd  
## 1 0.05534351 0 1.0000000 1.0000000 0.05201618  
## 2 0.03053435 3 0.8282443 0.9312977 0.05090076  
## 3 0.01781170 5 0.7671756 0.8854962 0.05008523  
## 4 0.01145038 8 0.7137405 0.8969466 0.05029470  
## 5 0.01017812 14 0.6412214 0.9045802 0.05043226  
## 6 0.01000000 18 0.5992366 0.9007634 0.05036368  
##   
## Variable importance  
## checking\_balance amount months\_loan\_duration   
## 24 15 13   
## credit\_history age employment\_duration   
## 9 8 8   
## purpose savings\_balance job   
## 7 4 3   
## years\_at\_residence phone housing   
## 3 2 2   
## existing\_loans\_count   
## 2   
##   
## Node number 1: 900 observations, complexity param=0.05534351  
## predicted class=no expected loss=0.2911111 P(node) =1  
## class counts: 638 262  
## probabilities: 0.709 0.291   
## left son=2 (420 obs) right son=3 (480 obs)  
## Primary splits:  
## checking\_balance splits as RLRL, improve=41.610160, (0 missing)  
## credit\_history splits as LLRLR, improve=15.426680, (0 missing)  
## amount < 3913.5 to the left, improve=13.794140, (0 missing)  
## months\_loan\_duration < 34.5 to the left, improve=13.243030, (0 missing)  
## savings\_balance splits as RLRLL, improve= 9.554261, (0 missing)  
## Surrogate splits:  
## savings\_balance splits as RLRLL, agree=0.604, adj=0.152, (0 split)  
## credit\_history splits as LRRRR, agree=0.591, adj=0.124, (0 split)  
## age < 30.5 to the right, agree=0.564, adj=0.067, (0 split)  
## employment\_duration splits as RLRRR, agree=0.553, adj=0.043, (0 split)  
## months\_loan\_duration < 10.5 to the left, agree=0.550, adj=0.036, (0 split)  
##   
## Node number 2: 420 observations  
## predicted class=no expected loss=0.1285714 P(node) =0.4666667  
## class counts: 366 54  
## probabilities: 0.871 0.129   
##   
## Node number 3: 480 observations, complexity param=0.05534351  
## predicted class=no expected loss=0.4333333 P(node) =0.5333333  
## class counts: 272 208  
## probabilities: 0.567 0.433   
## left son=6 (381 obs) right son=7 (99 obs)  
## Primary splits:  
## months\_loan\_duration < 31.5 to the left, improve=11.331200, (0 missing)  
## amount < 8079 to the left, improve= 9.340311, (0 missing)  
## credit\_history splits as LLRLR, improve= 9.101618, (0 missing)  
## savings\_balance splits as RLRLL, improve= 5.939649, (0 missing)  
## housing splits as RLR, improve= 3.273356, (0 missing)  
## Surrogate splits:  
## amount < 6648 to the left, agree=0.831, adj=0.182, (0 split)  
##   
## Node number 6: 381 observations, complexity param=0.05534351  
## predicted class=no expected loss=0.3779528 P(node) =0.4233333  
## class counts: 237 144  
## probabilities: 0.622 0.378   
## left son=12 (337 obs) right son=13 (44 obs)  
## Primary splits:  
## credit\_history splits as LLRLR, improve=9.186294, (0 missing)  
## amount < 10975.5 to the left, improve=6.323869, (0 missing)  
## months\_loan\_duration < 11.5 to the left, improve=4.274574, (0 missing)  
## employment\_duration splits as RLLLR, improve=3.336725, (0 missing)  
## purpose splits as LRRRLR, improve=3.127277, (0 missing)  
##   
## Node number 7: 99 observations, complexity param=0.03053435  
## predicted class=yes expected loss=0.3535354 P(node) =0.11  
## class counts: 35 64  
## probabilities: 0.354 0.646   
## left son=14 (8 obs) right son=15 (91 obs)  
## Primary splits:  
## employment\_duration splits as RRRRL, improve=7.274503, (0 missing)  
## age < 25.5 to the right, improve=3.906846, (0 missing)  
## savings\_balance splits as RRRLL, improve=2.729836, (0 missing)  
## job splits as LRLR, improve=1.675060, (0 missing)  
## years\_at\_residence < 3.5 to the left, improve=1.510484, (0 missing)  
## Surrogate splits:  
## purpose splits as RRLRRR, agree=0.929, adj=0.125, (0 split)  
## job splits as RRLR, agree=0.929, adj=0.125, (0 split)  
##   
## Node number 12: 337 observations, complexity param=0.03053435  
## predicted class=no expected loss=0.3382789 P(node) =0.3744444  
## class counts: 223 114  
## probabilities: 0.662 0.338   
## left son=24 (329 obs) right son=25 (8 obs)  
## Primary splits:  
## amount < 10975.5 to the left, improve=7.176355, (0 missing)  
## months\_loan\_duration < 11.5 to the left, improve=4.112960, (0 missing)  
## credit\_history splits as LR-R-, improve=4.048679, (0 missing)  
## employment\_duration splits as RLRLR, improve=3.402219, (0 missing)  
## purpose splits as LRRRRR, improve=2.573173, (0 missing)  
##   
## Node number 13: 44 observations  
## predicted class=yes expected loss=0.3181818 P(node) =0.04888889  
## class counts: 14 30  
## probabilities: 0.318 0.682   
##   
## Node number 14: 8 observations  
## predicted class=no expected loss=0 P(node) =0.008888889  
## class counts: 8 0  
## probabilities: 1.000 0.000   
##   
## Node number 15: 91 observations, complexity param=0.01017812  
## predicted class=yes expected loss=0.2967033 P(node) =0.1011111  
## class counts: 27 64  
## probabilities: 0.297 0.703   
## left son=30 (73 obs) right son=31 (18 obs)  
## Primary splits:  
## age < 25.5 to the right, improve=2.609681, (0 missing)  
## savings\_balance splits as RRRLL, improve=2.271271, (0 missing)  
## years\_at\_residence < 3.5 to the left, improve=2.248815, (0 missing)  
## months\_loan\_duration < 43.5 to the left, improve=1.334885, (0 missing)  
## dependents < 1.5 to the right, improve=1.264190, (0 missing)  
##   
## Node number 24: 329 observations, complexity param=0.0178117  
## predicted class=no expected loss=0.3221884 P(node) =0.3655556  
## class counts: 223 106  
## probabilities: 0.678 0.322   
## left son=48 (68 obs) right son=49 (261 obs)  
## Primary splits:  
## months\_loan\_duration < 11.5 to the left, improve=4.411960, (0 missing)  
## credit\_history splits as LR-R-, improve=3.610395, (0 missing)  
## employment\_duration splits as RLRLR, improve=3.057396, (0 missing)  
## age < 33.5 to the right, improve=2.832780, (0 missing)  
## purpose splits as LRLRRR, improve=2.830600, (0 missing)  
## Surrogate splits:  
## amount < 527.5 to the left, agree=0.815, adj=0.103, (0 split)  
## age < 65.5 to the right, agree=0.799, adj=0.029, (0 split)  
##   
## Node number 25: 8 observations  
## predicted class=yes expected loss=0 P(node) =0.008888889  
## class counts: 0 8  
## probabilities: 0.000 1.000   
##   
## Node number 30: 73 observations, complexity param=0.01017812  
## predicted class=yes expected loss=0.3561644 P(node) =0.08111111  
## class counts: 26 47  
## probabilities: 0.356 0.644   
## left son=60 (37 obs) right son=61 (36 obs)  
## Primary splits:  
## years\_at\_residence < 3.5 to the left, improve=3.715188, (0 missing)  
## savings\_balance splits as RRRLL, improve=2.033704, (0 missing)  
## employment\_duration splits as LRLL-, improve=1.715816, (0 missing)  
## percent\_of\_income < 3.5 to the left, improve=1.601074, (0 missing)  
## age < 46.5 to the left, improve=1.576304, (0 missing)  
## Surrogate splits:  
## age < 33.5 to the left, agree=0.740, adj=0.472, (0 split)  
## housing splits as RLR, agree=0.726, adj=0.444, (0 split)  
## employment\_duration splits as LRLL-, agree=0.671, adj=0.333, (0 split)  
## purpose splits as RRLLLL, agree=0.644, adj=0.278, (0 split)  
## amount < 3347.5 to the right, agree=0.616, adj=0.222, (0 split)  
##   
## Node number 31: 18 observations  
## predicted class=yes expected loss=0.05555556 P(node) =0.02  
## class counts: 1 17  
## probabilities: 0.056 0.944   
##   
## Node number 48: 68 observations  
## predicted class=no expected loss=0.1617647 P(node) =0.07555556  
## class counts: 57 11  
## probabilities: 0.838 0.162   
##   
## Node number 49: 261 observations, complexity param=0.0178117  
## predicted class=no expected loss=0.3639847 P(node) =0.29  
## class counts: 166 95  
## probabilities: 0.636 0.364   
## left son=98 (187 obs) right son=99 (74 obs)  
## Primary splits:  
## amount < 1381.5 to the right, improve=4.618602, (0 missing)  
## credit\_history splits as LR-R-, improve=3.337861, (0 missing)  
## purpose splits as LRLRRR, improve=3.019072, (0 missing)  
## phone splits as RL, improve=2.601976, (0 missing)  
## employment\_duration splits as RLLLR, improve=2.342206, (0 missing)  
## Surrogate splits:  
## months\_loan\_duration < 12.5 to the right, agree=0.736, adj=0.068, (0 split)  
## purpose splits as LLLRLL, agree=0.720, adj=0.014, (0 split)  
## age < 20.5 to the right, agree=0.720, adj=0.014, (0 split)  
## existing\_loans\_count < 3.5 to the left, agree=0.720, adj=0.014, (0 split)  
##   
## Node number 60: 37 observations, complexity param=0.01017812  
## predicted class=no expected loss=0.4864865 P(node) =0.04111111  
## class counts: 19 18  
## probabilities: 0.514 0.486   
## left son=120 (16 obs) right son=121 (21 obs)  
## Primary splits:  
## phone splits as LR, improve=3.153153, (0 missing)  
## savings\_balance splits as RRLLL, improve=2.249449, (0 missing)  
## purpose splits as LLRLRR, improve=1.644381, (0 missing)  
## amount < 4180 to the left, improve=1.430542, (0 missing)  
## age < 34.5 to the right, improve=1.430542, (0 missing)  
## Surrogate splits:  
## amount < 3757.5 to the left, agree=0.730, adj=0.375, (0 split)  
## job splits as RLLL, agree=0.676, adj=0.250, (0 split)  
## months\_loan\_duration < 37.5 to the left, agree=0.649, adj=0.188, (0 split)  
## purpose splits as LLRRRL, agree=0.649, adj=0.188, (0 split)  
## dependents < 1.5 to the right, agree=0.649, adj=0.188, (0 split)  
##   
## Node number 61: 36 observations  
## predicted class=yes expected loss=0.1944444 P(node) =0.04  
## class counts: 7 29  
## probabilities: 0.194 0.806   
##   
## Node number 98: 187 observations, complexity param=0.01145038  
## predicted class=no expected loss=0.3048128 P(node) =0.2077778  
## class counts: 130 57  
## probabilities: 0.695 0.305   
## left son=196 (73 obs) right son=197 (114 obs)  
## Primary splits:  
## employment\_duration splits as RLRLR, improve=2.363089, (0 missing)  
## purpose splits as LRLRRL, improve=1.971337, (0 missing)  
## amount < 1813.5 to the left, improve=1.911747, (0 missing)  
## months\_loan\_duration < 27.5 to the left, improve=1.706427, (0 missing)  
## savings\_balance splits as RLLLL, improve=1.516043, (0 missing)  
## Surrogate splits:  
## age < 34.5 to the right, agree=0.679, adj=0.178, (0 split)  
## phone splits as RL, agree=0.647, adj=0.096, (0 split)  
## credit\_history splits as LR-R-, agree=0.642, adj=0.082, (0 split)  
## existing\_loans\_count < 2.5 to the right, agree=0.636, adj=0.068, (0 split)  
## amount < 1428.5 to the left, agree=0.626, adj=0.041, (0 split)  
##   
## Node number 99: 74 observations, complexity param=0.0178117  
## predicted class=yes expected loss=0.4864865 P(node) =0.08222222  
## class counts: 36 38  
## probabilities: 0.486 0.514   
## left son=198 (42 obs) right son=199 (32 obs)  
## Primary splits:  
## purpose splits as LR-LLR, improve=4.749759, (0 missing)  
## months\_loan\_duration < 22.5 to the left, improve=2.929962, (0 missing)  
## phone splits as RL, improve=2.843103, (0 missing)  
## other\_credit splits as RLR, improve=2.558687, (0 missing)  
## existing\_loans\_count < 1.5 to the right, improve=2.389057, (0 missing)  
## Surrogate splits:  
## amount < 1163 to the left, agree=0.662, adj=0.219, (0 split)  
## months\_loan\_duration < 19.5 to the left, agree=0.649, adj=0.188, (0 split)  
## years\_at\_residence < 2.5 to the right, agree=0.622, adj=0.125, (0 split)  
## age < 40.5 to the left, agree=0.622, adj=0.125, (0 split)  
## housing splits as RLL, agree=0.608, adj=0.094, (0 split)  
##   
## Node number 120: 16 observations  
## predicted class=no expected loss=0.25 P(node) =0.01777778  
## class counts: 12 4  
## probabilities: 0.750 0.250   
##   
## Node number 121: 21 observations, complexity param=0.01017812  
## predicted class=yes expected loss=0.3333333 P(node) =0.02333333  
## class counts: 7 14  
## probabilities: 0.333 0.667   
## left son=242 (7 obs) right son=243 (14 obs)  
## Primary splits:  
## age < 34.5 to the right, improve=3.0476190, (0 missing)  
## years\_at\_residence < 1.5 to the left, improve=1.1904760, (0 missing)  
## purpose splits as RLRLR-, improve=0.7179487, (0 missing)  
## credit\_history splits as RLRR-, improve=0.3888889, (0 missing)  
## employment\_duration splits as LRRL-, improve=0.3888889, (0 missing)  
## Surrogate splits:  
## amount < 4174.5 to the left, agree=0.810, adj=0.429, (0 split)  
## purpose splits as RLRRR-, agree=0.762, adj=0.286, (0 split)  
## housing splits as LRR, agree=0.762, adj=0.286, (0 split)  
## employment\_duration splits as RLRR-, agree=0.714, adj=0.143, (0 split)  
## years\_at\_residence < 1.5 to the left, agree=0.714, adj=0.143, (0 split)  
##   
## Node number 196: 73 observations  
## predicted class=no expected loss=0.2054795 P(node) =0.08111111  
## class counts: 58 15  
## probabilities: 0.795 0.205   
##   
## Node number 197: 114 observations, complexity param=0.01145038  
## predicted class=no expected loss=0.3684211 P(node) =0.1266667  
## class counts: 72 42  
## probabilities: 0.632 0.368   
## left son=394 (40 obs) right son=395 (74 obs)  
## Primary splits:  
## months\_loan\_duration < 16.5 to the left, improve=2.535064, (0 missing)  
## amount < 5488.5 to the right, improve=2.493606, (0 missing)  
## purpose splits as LRLRRL, improve=1.624060, (0 missing)  
## age < 33.5 to the right, improve=1.247754, (0 missing)  
## percent\_of\_income < 1.5 to the left, improve=1.136842, (0 missing)  
## Surrogate splits:  
## amount < 1658 to the left, agree=0.746, adj=0.275, (0 split)  
## purpose splits as RRRRRL, agree=0.658, adj=0.025, (0 split)  
## savings\_balance splits as RLRRR, agree=0.658, adj=0.025, (0 split)  
##   
## Node number 198: 42 observations, complexity param=0.01145038  
## predicted class=no expected loss=0.3571429 P(node) =0.04666667  
## class counts: 27 15  
## probabilities: 0.643 0.357   
## left son=396 (19 obs) right son=397 (23 obs)  
## Primary splits:  
## job splits as LRRL, improve=2.754822, (0 missing)  
## other\_credit splits as RLR, improve=2.194805, (0 missing)  
## employment\_duration splits as RLLLR, improve=1.719048, (0 missing)  
## existing\_loans\_count < 1.5 to the right, improve=1.219048, (0 missing)  
## phone splits as RL, improve=1.065126, (0 missing)  
## Surrogate splits:  
## age < 38.5 to the right, agree=0.667, adj=0.263, (0 split)  
## amount < 738.5 to the left, agree=0.643, adj=0.211, (0 split)  
## months\_loan\_duration < 12.5 to the left, agree=0.619, adj=0.158, (0 split)  
## purpose splits as L--RR-, agree=0.595, adj=0.105, (0 split)  
## savings\_balance splits as RLRRR, agree=0.595, adj=0.105, (0 split)  
##   
## Node number 199: 32 observations  
## predicted class=yes expected loss=0.28125 P(node) =0.03555556  
## class counts: 9 23  
## probabilities: 0.281 0.719   
##   
## Node number 242: 7 observations  
## predicted class=no expected loss=0.2857143 P(node) =0.007777778  
## class counts: 5 2  
## probabilities: 0.714 0.286   
##   
## Node number 243: 14 observations  
## predicted class=yes expected loss=0.1428571 P(node) =0.01555556  
## class counts: 2 12  
## probabilities: 0.143 0.857   
##   
## Node number 394: 40 observations  
## predicted class=no expected loss=0.225 P(node) =0.04444444  
## class counts: 31 9  
## probabilities: 0.775 0.225   
##   
## Node number 395: 74 observations, complexity param=0.01145038  
## predicted class=no expected loss=0.4459459 P(node) =0.08222222  
## class counts: 41 33  
## probabilities: 0.554 0.446   
## left son=790 (29 obs) right son=791 (45 obs)  
## Primary splits:  
## amount < 3515.5 to the right, improve=5.450326, (0 missing)  
## purpose splits as LRLRRL, improve=2.767568, (0 missing)  
## phone splits as RL, improve=2.759138, (0 missing)  
## age < 36.5 to the right, improve=2.727050, (0 missing)  
## savings\_balance splits as RLRRL, improve=2.234234, (0 missing)  
## Surrogate splits:  
## age < 34.5 to the right, agree=0.743, adj=0.345, (0 split)  
## purpose splits as LLRLRL, agree=0.716, adj=0.276, (0 split)  
## employment\_duration splits as R-R-L, agree=0.676, adj=0.172, (0 split)  
## job splits as LRLR, agree=0.662, adj=0.138, (0 split)  
## credit\_history splits as LR-R-, agree=0.649, adj=0.103, (0 split)  
##   
## Node number 396: 19 observations  
## predicted class=no expected loss=0.1578947 P(node) =0.02111111  
## class counts: 16 3  
## probabilities: 0.842 0.158   
##   
## Node number 397: 23 observations, complexity param=0.01145038  
## predicted class=yes expected loss=0.4782609 P(node) =0.02555556  
## class counts: 11 12  
## probabilities: 0.478 0.522   
## left son=794 (9 obs) right son=795 (14 obs)  
## Primary splits:  
## existing\_loans\_count < 1.5 to the right, improve=2.652864, (0 missing)  
## months\_loan\_duration < 14 to the left, improve=1.781291, (0 missing)  
## amount < 1173 to the left, improve=1.278261, (0 missing)  
## credit\_history splits as LR-L-, improve=1.121118, (0 missing)  
## years\_at\_residence < 2.5 to the right, improve=0.746118, (0 missing)  
## Surrogate splits:  
## checking\_balance splits as R-L-, agree=0.783, adj=0.444, (0 split)  
## credit\_history splits as LR-R-, agree=0.739, adj=0.333, (0 split)  
## amount < 999.5 to the left, agree=0.652, adj=0.111, (0 split)  
## savings\_balance splits as R-LRR, agree=0.652, adj=0.111, (0 split)  
## employment\_duration splits as LRRRR, agree=0.652, adj=0.111, (0 split)  
##   
## Node number 790: 29 observations  
## predicted class=no expected loss=0.2068966 P(node) =0.03222222  
## class counts: 23 6  
## probabilities: 0.793 0.207   
##   
## Node number 791: 45 observations, complexity param=0.01145038  
## predicted class=yes expected loss=0.4 P(node) =0.05  
## class counts: 18 27  
## probabilities: 0.400 0.600   
## left son=1582 (12 obs) right son=1583 (33 obs)  
## Primary splits:  
## purpose splits as LLLRR-, improve=2.327273, (0 missing)  
## amount < 2284 to the right, improve=1.896296, (0 missing)  
## age < 25.5 to the right, improve=1.896296, (0 missing)  
## savings\_balance splits as RLR-L, improve=1.637594, (0 missing)  
## checking\_balance splits as R-L-, improve=1.049393, (0 missing)  
## Surrogate splits:  
## age < 31.5 to the right, agree=0.778, adj=0.167, (0 split)  
## other\_credit splits as LRR, agree=0.756, adj=0.083, (0 split)  
## dependents < 1.5 to the right, agree=0.756, adj=0.083, (0 split)  
##   
## Node number 794: 9 observations  
## predicted class=no expected loss=0.2222222 P(node) =0.01  
## class counts: 7 2  
## probabilities: 0.778 0.222   
##   
## Node number 795: 14 observations  
## predicted class=yes expected loss=0.2857143 P(node) =0.01555556  
## class counts: 4 10  
## probabilities: 0.286 0.714   
##   
## Node number 1582: 12 observations  
## predicted class=no expected loss=0.3333333 P(node) =0.01333333  
## class counts: 8 4  
## probabilities: 0.667 0.333   
##   
## Node number 1583: 33 observations  
## predicted class=yes expected loss=0.3030303 P(node) =0.03666667  
## class counts: 10 23  
## probabilities: 0.303 0.697

# use the rpart.plot package to create a visualization  
library(rpart.plot)  
# a basic decision tree diagram  
rpart.plot(m.rpart, digits = 3)



# a few adjustments to the diagram  
rpart.plot(m.rpart, digits = 4, fallen.leaves = TRUE, type = 3, extra = 101)



**Step 4: Evaluate model performance**

# generate predictions for the testing dataset  
p.rpart <- predict(m.rpart, test)

# compare the distribution of predicted values vs. actual values  
summary(p.rpart)

## no yes   
## Min. :0.05556 Min. :0.1286   
## 1st Qu.:0.57955 1st Qu.:0.1286   
## Median :0.81638 Median :0.1836   
## Mean :0.68973 Mean :0.3103   
## 3rd Qu.:0.87143 3rd Qu.:0.4205   
## Max. :0.87143 Max. :0.9444

summary(test$default)

## no yes   
## 62 38

# compare the correlation  
cor(p.rpart, as.numeric(test$default))

## [,1]  
## no -0.4029158  
## yes 0.4029158

# function to calculate the mean absolute error  
MAE <- function(actual, predicted) {  
 mean(abs(actual - predicted))   
}  
  
# mean absolute error between predicted and actual values  
MAE(p.rpart, as.numeric(test$default))

## [1] 0.88

# mean absolute error between actual values and mean value  
mean(as.numeric(training$default))

## [1] 1.291111

MAE( 0.86, as.numeric(training$default))

## [1] 0.4311111

**The Random Forest analysis of the credit data-**

**Step 1: Collecting the data**

Source of the data UCI repository for data sciences.

**Step 2: Exploring & Preparing the data**

#reading the data  
credit <- read.csv("http://www.sci.csueastbay.edu/~esuess/classes/Statistics\_6620/Presentations/ml10/credit.csv")  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(lattice)  
library(ggplot2)

**Step 3: Traning a model on the data**

I will use random forest package for random forest.

library(randomForest)

## randomForest 4.6-12

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

##   
## Attaching package: 'randomForest'

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

#Set.seed for getting the same output  
set.seed(300)

**Step 4: Evaluating the model performance**

#Creating random forest  
rf <- randomForest(default ~ ., data = credit)  
rf

##   
## Call:  
## randomForest(formula = default ~ ., data = credit)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 23.8%  
## Confusion matrix:  
## no yes class.error  
## no 640 60 0.08571429  
## yes 178 122 0.59333333

The randomForest() function creates an ensemble of 500 trees & 4 variable at each split.Estimated error rate is 23.8%.

**step 4: Improving model performance**

library(caret)  
ctrl <- trainControl(method = "repeatedcv",  
 number = 10, repeats = 10)

In this example, we are repeating 10 cross fold 10 times repeate.

# tunning a random forest  
grid\_rf <- expand.grid(.mtry = c(2, 4, 8, 16))

we need to create a grid with values of 2, 4, 8, and 16.

set.seed(300)  
m\_rf <- train(default ~ ., data = credit, method = "rf",  
 metric = "Kappa", trControl = ctrl,  
 tuneGrid = grid\_rf)  
m\_rf

## Random Forest   
##   
## 1000 samples  
## 16 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 900, 900, 900, 900, 900, 900, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.7256 0.1311283  
## 4 0.7476 0.2878470  
## 8 0.7519 0.3346061  
## 16 0.7557 0.3618152  
##   
## Kappa was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 16.

**Step 5: Improving model performance**

I am going to train model based on rf & using kappa for accuracy checking. we'll compare that to a boosted tree using 10, 20, 30, and 40 iterations:

# auto-tune a boosted C5.0 decision tree  
grid\_c50 <- expand.grid(.model = "tree",  
 .trials = c(10, 20, 30, 40),  
 .winnow = "FALSE")

set.seed(300)  
library(C50)  
m\_c50 <- train(default ~ ., data = credit, method = "C5.0",  
 metric = "Kappa", trControl = ctrl,  
 tuneGrid = grid\_c50)

## Loading required package: plyr

## Warning in Ops.factor(x$winnow): '!' not meaningful for factors

m\_c50

## C5.0   
##   
## 1000 samples  
## 16 predictor  
## 2 classes: 'no', 'yes'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 900, 900, 900, 900, 900, 900, ...   
## Resampling results across tuning parameters:  
##   
## trials Accuracy Kappa   
## 10 0.7325 0.3215655  
## 20 0.7343 0.3268052  
## 30 0.7381 0.3343137  
## 40 0.7388 0.3335082  
##   
## Tuning parameter 'model' was held constant at a value of tree  
##   
## Tuning parameter 'winnow' was held constant at a value of FALSE  
## Kappa was used to select the optimal model using the largest value.  
## The final values used for the model were trials = 30, model = tree  
## and winnow = FALSE.

With a kappa of about 0.361, the random forest model with mtry = 16 was the winner among these eight models. It was higher than the best C5.0 decision tree, which had a kappa of about 0.334, and slightly higher than the AdaBoost.M1 model with a kappa of about 0.360. Based on these results, we would submit the random forest as our final model.

**The stock Market Data**

The Smarket data,is part of the ISLR library.This data set consists of percentage returns for the S&P 500 stock index over 1,250 days, from the beginning of 2001 until the end of 2005.

library(ISLR)  
names(Smarket)

## [1] "Year" "Lag1" "Lag2" "Lag3" "Lag4" "Lag5"   
## [7] "Volume" "Today" "Direction"

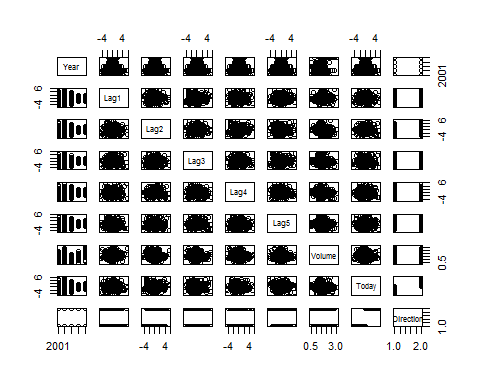
#Dimention of dataset  
dim(Smarket)

## [1] 1250 9

#Getting summary of dataset  
summary(Smarket)

## Year Lag1 Lag2   
## Min. :2001 Min. :-4.922000 Min. :-4.922000   
## 1st Qu.:2002 1st Qu.:-0.639500 1st Qu.:-0.639500   
## Median :2003 Median : 0.039000 Median : 0.039000   
## Mean :2003 Mean : 0.003834 Mean : 0.003919   
## 3rd Qu.:2004 3rd Qu.: 0.596750 3rd Qu.: 0.596750   
## Max. :2005 Max. : 5.733000 Max. : 5.733000   
## Lag3 Lag4 Lag5   
## Min. :-4.922000 Min. :-4.922000 Min. :-4.92200   
## 1st Qu.:-0.640000 1st Qu.:-0.640000 1st Qu.:-0.64000   
## Median : 0.038500 Median : 0.038500 Median : 0.03850   
## Mean : 0.001716 Mean : 0.001636 Mean : 0.00561   
## 3rd Qu.: 0.596750 3rd Qu.: 0.596750 3rd Qu.: 0.59700   
## Max. : 5.733000 Max. : 5.733000 Max. : 5.73300   
## Volume Today Direction   
## Min. :0.3561 Min. :-4.922000 Down:602   
## 1st Qu.:1.2574 1st Qu.:-0.639500 Up :648   
## Median :1.4229 Median : 0.038500   
## Mean :1.4783 Mean : 0.003138   
## 3rd Qu.:1.6417 3rd Qu.: 0.596750   
## Max. :3.1525 Max. : 5.733000

#Scatter plot of all variables of Dataset  
pairs(Smarket)



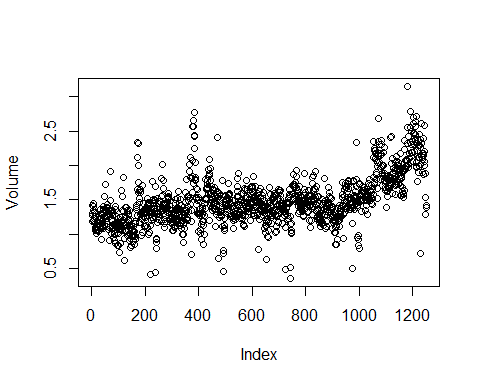
The cor() function produces a matrix that contains all of the pairwise correlations among the predictors in a data set.

#Correlation matrix  
cor(Smarket[,-9])

## Year Lag1 Lag2 Lag3 Lag4  
## Year 1.00000000 0.029699649 0.030596422 0.033194581 0.035688718  
## Lag1 0.02969965 1.000000000 -0.026294328 -0.010803402 -0.002985911  
## Lag2 0.03059642 -0.026294328 1.000000000 -0.025896670 -0.010853533  
## Lag3 0.03319458 -0.010803402 -0.025896670 1.000000000 -0.024051036  
## Lag4 0.03568872 -0.002985911 -0.010853533 -0.024051036 1.000000000  
## Lag5 0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641  
## Volume 0.53900647 0.040909908 -0.043383215 -0.041823686 -0.048414246  
## Today 0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527  
## Lag5 Volume Today  
## Year 0.029787995 0.53900647 0.030095229  
## Lag1 -0.005674606 0.04090991 -0.026155045  
## Lag2 -0.003557949 -0.04338321 -0.010250033  
## Lag3 -0.018808338 -0.04182369 -0.002447647  
## Lag4 -0.027083641 -0.04841425 -0.006899527  
## Lag5 1.000000000 -0.02200231 -0.034860083  
## Volume -0.022002315 1.00000000 0.014591823  
## Today -0.034860083 0.01459182 1.000000000

As one would expect, the correlations between the lag variables and today's returns are close to zero. In other words, there appears to be little correlation between today's returns and previous days' returns. The only substantial correlation is between Year and Volume.

#Scatterplot of volume  
attach(Smarket)  
plot(Volume)



**Logistic Regression**

#Running logistic regression  
glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume , data=Smarket ,family=binomial )  
glm.fit

##   
## Call: glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +   
## Volume, family = binomial, data = Smarket)  
##   
## Coefficients:  
## (Intercept) Lag1 Lag2 Lag3 Lag4   
## -0.126000 -0.073074 -0.042301 0.011085 0.009359   
## Lag5 Volume   
## 0.010313 0.135441   
##   
## Degrees of Freedom: 1249 Total (i.e. Null); 1243 Residual  
## Null Deviance: 1731   
## Residual Deviance: 1728 AIC: 1742

#getting summary  
summary(glm.fit)

##   
## Call:  
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +   
## Volume, family = binomial, data = Smarket)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.446 -1.203 1.065 1.145 1.326   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.126000 0.240736 -0.523 0.601  
## Lag1 -0.073074 0.050167 -1.457 0.145  
## Lag2 -0.042301 0.050086 -0.845 0.398  
## Lag3 0.011085 0.049939 0.222 0.824  
## Lag4 0.009359 0.049974 0.187 0.851  
## Lag5 0.010313 0.049511 0.208 0.835  
## Volume 0.135441 0.158360 0.855 0.392  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1731.2 on 1249 degrees of freedom  
## Residual deviance: 1727.6 on 1243 degrees of freedom  
## AIC: 1741.6  
##   
## Number of Fisher Scoring iterations: 3

The predict() function can be used to predict the probability that the market will go up, given values of the predictors.

glm.probs=predict(glm.fit,type="response")  
glm.probs[1:10]

## 1 2 3 4 5 6 7   
## 0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509   
## 8 9 10   
## 0.5092292 0.5176135 0.4888378

#Getting contrast   
contrasts(Direction)

## Up  
## Down 0  
## Up 1

In order to make a prediction as to whether the market will go up or down on a particular day, we must convert these predicted probabilities into class labels, Up or Down.

glm.pred=rep("Down",1250)  
glm.pred[glm.probs>0.5]="up"

The first command creates a vector of 1,250 Down elements. The second line transforms to Up all of the elements for which the predicted probability of a market increase exceeds 0.5.

#Creating table   
table(glm.pred ,Direction )

## Direction  
## glm.pred Down Up  
## Down 145 141  
## up 457 507

(507+145)/1250

## [1] 0.5216

mean(glm.pred==Direction )

## [1] 0.116

In this case, logistic regression correctly predicted the movement of the market 52.2%ofthetime.

train=(Year<2005)  
Smarket.2005= Smarket [!train ,]  
dim(Smarket.2005)

## [1] 252 9

Direction.2005= Direction [!train]

The object train is a vector of 1,250 elements, corresponding to the observations in our data set.

glm.fit=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data = Smarket,family = binomial,subset=train)  
glm.probs=predict(glm.fit,Smarket.2005,type = "response")

glm.pred=rep("Down",252)  
glm.pred[glm.probs>0.5]="up"  
table(glm.pred,Direction.2005)

## Direction.2005  
## glm.pred Down Up  
## Down 77 97  
## up 34 44

mean(glm.pred==Direction.2005)

## [1] 0.3055556

mean(glm.pred!=Direction.2005)

## [1] 0.6944444

glm.fit=glm(Direction~Lag1+Lag2 ,data=Smarket ,family=binomial, subset=train)   
glm.probs=predict(glm.fit ,Smarket.2005, type="response")   
glm.pred=rep("Down",252)  
glm.pred[glm.probs >.5]="up"  
table(glm.pred ,Direction.2005)

## Direction.2005  
## glm.pred Down Up  
## Down 35 35  
## up 76 106

mean(glm.pred==Direction.2005)

## [1] 0.1388889

106/(106+76)

## [1] 0.5824176

predict (glm.fit ,newdata=data.frame(Lag1=c(1.2,1.5), Lag2=c(1.1,-0.8)),type="response")

## 1 2   
## 0.4791462 0.4960939