Bank\_loan

venukiran sankar

23/09/2020

data\_new=read.csv("C:/Users/venuk/OneDrive/Desktop/projects/Project 2\_Logistic\_R (1)/bankloan.csv")  
View(data\_new)  
str(data\_new)

## 'data.frame': 700 obs. of 9 variables:  
## $ age : int 41 27 40 41 24 41 39 43 24 36 ...  
## $ ed : int 3 1 1 1 2 2 1 1 1 1 ...  
## $ employ : int 17 10 15 15 2 5 20 12 3 0 ...  
## $ address : int 12 6 14 14 0 5 9 11 4 13 ...  
## $ income : int 176 31 55 120 28 25 67 38 19 25 ...  
## $ debtinc : num 9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...  
## $ creddebt: num 11.36 1.36 0.86 2.66 1.79 ...  
## $ othdebt : num 5.01 4 2.17 0.82 3.06 ...  
## $ default : int 1 0 0 0 1 0 0 0 1 0 ...

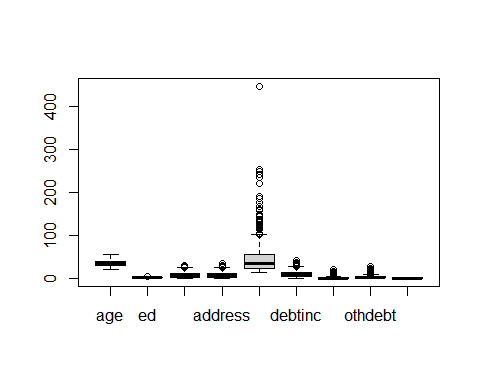
summary(data\_new)

## age ed employ address   
## Min. :20.00 Min. :1.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.:29.00 1st Qu.:1.000 1st Qu.: 3.000 1st Qu.: 3.000   
## Median :34.00 Median :1.000 Median : 7.000 Median : 7.000   
## Mean :34.86 Mean :1.723 Mean : 8.389 Mean : 8.279   
## 3rd Qu.:40.00 3rd Qu.:2.000 3rd Qu.:12.000 3rd Qu.:12.000   
## Max. :56.00 Max. :5.000 Max. :31.000 Max. :34.000   
## income debtinc creddebt othdebt   
## Min. : 14.0 Min. : 0.40 Min. : 0.010 Min. : 0.050   
## 1st Qu.: 24.0 1st Qu.: 5.00 1st Qu.: 0.370 1st Qu.: 1.048   
## Median : 34.0 Median : 8.60 Median : 0.855 Median : 1.985   
## Mean : 45.6 Mean :10.26 Mean : 1.553 Mean : 3.058   
## 3rd Qu.: 55.0 3rd Qu.:14.12 3rd Qu.: 1.905 3rd Qu.: 3.928   
## Max. :446.0 Max. :41.30 Max. :20.560 Max. :27.030   
## default   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.2614   
## 3rd Qu.:1.0000   
## Max. :1.0000

## Including Plots

You can also embed plots, for example:

boxplot(data\_new)



sum(is.na(data\_new))

## [1] 0

names(data\_new)

## [1] "age" "ed" "employ" "address" "income" "debtinc" "creddebt"  
## [8] "othdebt" "default"

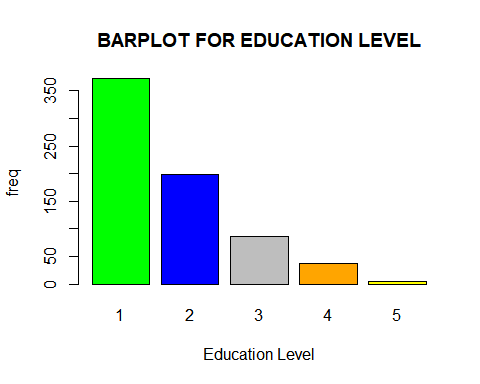
table(data\_new$age)

##   
## 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45   
## 2 10 12 18 24 20 23 28 37 44 24 36 25 25 33 27 29 22 23 37 27 34 16 17 9 16   
## 46 47 48 49 50 51 52 53 54 55 56   
## 13 16 15 4 8 6 7 6 4 2 1

ed\_data=table(data\_new$ed)  
  
ed\_data

##   
## 1 2 3 4 5   
## 372 198 87 38 5

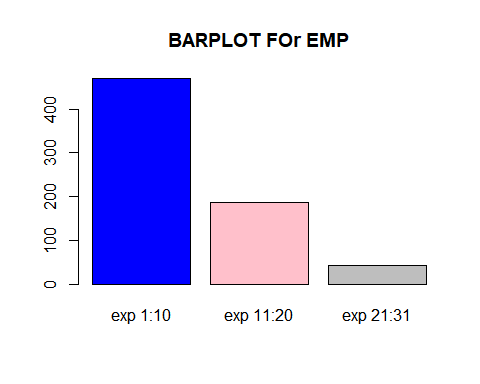
barplot(ed\_data,xlab = "Education Level",ylab = "freq",main = "BARPLOT FOR EDUCATION LEVEL",col = c("green","blue","grey","orange","yellow"))



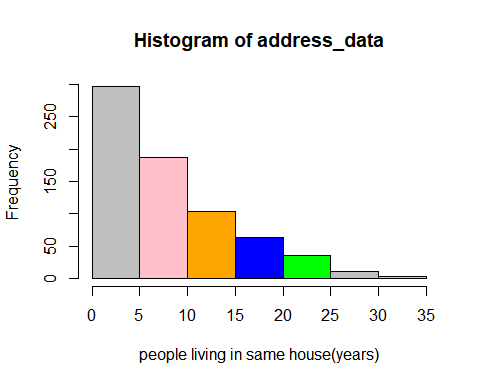
emp\_data=table(data\_new$employ)  
emp\_data

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 62 49 44 42 47 36 46 38 31 45 30 26 30 27 14 19 25 12 17 12 5 8 13 5 4 3   
## 26 27 28 29 30 31   
## 1 2 1 1 2 3

exp\_1\_10=470  
exp\_11\_20=187  
exp\_21\_31=43  
  
exP\_data=barplot(c(exp\_1\_10,exp\_11\_20,exp\_21\_31),names.arg = c("exp 1:10","exp 11:20","exp 21:31"),main="BARPLOT FOr EMP",col = c("blue","pink","grey"))



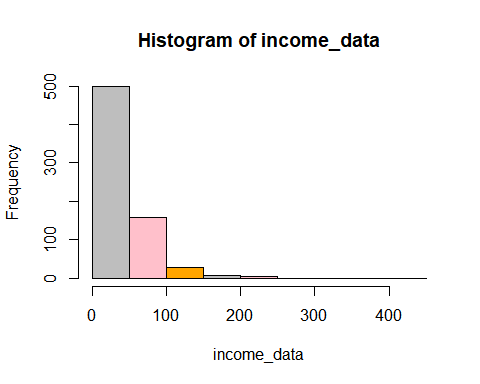
address\_data=data\_new$address  
hist(address\_data,col = c("grey","pink","orange","blue","green"),xlab = "people living in same house(years)")



table(address\_data)

## address\_data  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25   
## 50 57 59 48 49 34 43 33 40 39 32 27 20 18 22 16 18 17 9 13 7 9 7 9 3 7   
## 26 27 29 31 34   
## 7 3 1 2 1

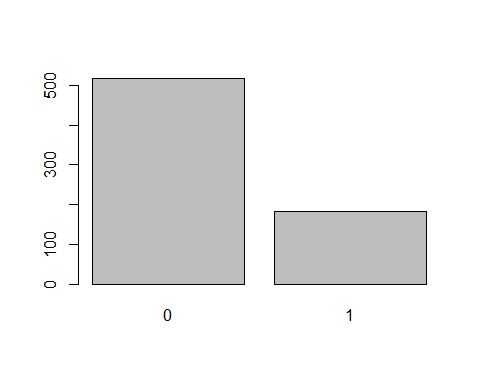
income\_data=data\_new$income  
hist(income\_data,col = c("grey","pink","orange"))



balanced\_data=table(data\_new$default)  
balanced\_data

##   
## 0 1   
## 517 183

barplot(balanced\_data)



set.seed(100)  
sample\_data=sample(2,nrow(data\_new),replace = T,prob = c(.7,.3))  
train\_data=data\_new[sample\_data==1,]  
test\_data=data\_new[sample\_data==2,]  
dim(train\_data)

## [1] 483 9

dim(test\_data)

## [1] 217 9

model=glm(default~.,data=train\_data,family = "binomial")  
summary(model)

##   
## Call:  
## glm(formula = default ~ ., family = "binomial", data = train\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0361 -0.6196 -0.2928 0.1542 2.7002   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.643623 0.788855 -0.816 0.414560   
## age 0.017658 0.023080 0.765 0.444224   
## ed 0.050559 0.152279 0.332 0.739877   
## employ -0.271496 0.041427 -6.554 5.61e-11 \*\*\*  
## address -0.106703 0.029741 -3.588 0.000334 \*\*\*  
## income -0.014932 0.009429 -1.584 0.113292   
## debtinc 0.033514 0.038578 0.869 0.384990   
## creddebt 0.801185 0.151836 5.277 1.32e-07 \*\*\*  
## othdebt 0.093144 0.101517 0.918 0.358867   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 550.24 on 482 degrees of freedom  
## Residual deviance: 372.37 on 474 degrees of freedom  
## AIC: 390.37  
##   
## Number of Fisher Scoring iterations: 6

model\_1=glm(default~employ+address+creddebt,data=train\_data,family = "binomial")  
summary(model\_1)

##   
## Call:  
## glm(formula = default ~ employ + address + creddebt, family = "binomial",   
## data = train\_data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.92764 -0.62778 -0.31200 0.07352 2.71216   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.02347 0.21944 0.107 0.915   
## employ -0.28779 0.03597 -8.000 1.24e-15 \*\*\*  
## address -0.09416 0.02377 -3.961 7.46e-05 \*\*\*  
## creddebt 0.86595 0.10537 8.218 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 550.24 on 482 degrees of freedom  
## Residual deviance: 383.82 on 479 degrees of freedom  
## AIC: 391.82  
##   
## Number of Fisher Scoring iterations: 6

predicted\_data\_train=predict(model\_1,train\_data,type="response")  
head(predicted\_data\_train)

## 1 2 3 4 5 6   
## 0.978923219 0.096065101 0.007637767 0.035288720 0.730655133 0.175290819

table(train\_data$default,predicted\_data\_train>=0.4)

##   
## FALSE TRUE  
## 0 307 52  
## 1 46 78

acc\_train\_data=(307+78)/(307+52+46+78)  
acc\_train\_data

## [1] 0.7971014

predicted\_data=predict(model\_1,test\_data,type="response")  
head(predicted\_data)

## 7 12 15 22 24 27   
## 0.036854247 0.286753648 0.010810386 0.001769079 0.076036148 0.012127213

table(test\_data$default,predicted\_data>=0.4)

##   
## FALSE TRUE  
## 0 136 22  
## 1 30 29

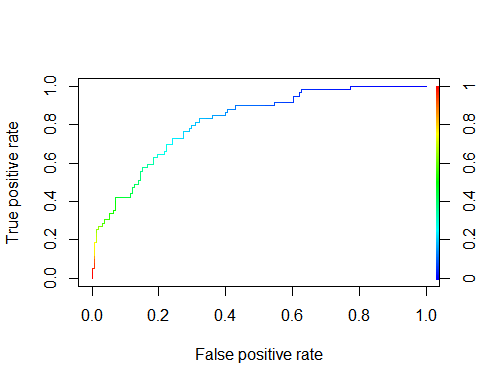
accu\_test\_data=(128+37)/(128+30+37+22)  
accu\_test\_data

## [1] 0.7603687

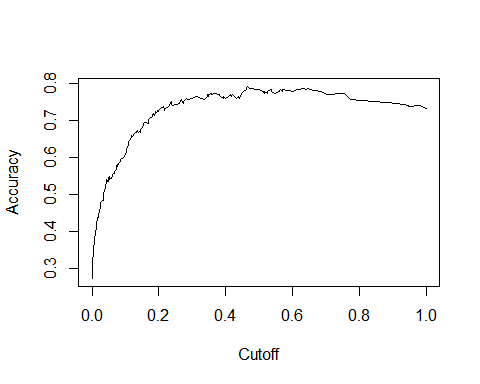
library(ROCR)

## Warning: package 'ROCR' was built under R version 4.0.2

roc\_vector=prediction(predicted\_data,test\_data$default)  
roc\_data=performance(roc\_vector,"tpr","fpr")  
plot(roc\_data,colorize=T)



accuracy\_data\_curve=performance(roc\_vector,"acc")  
plot(accuracy\_data\_curve)



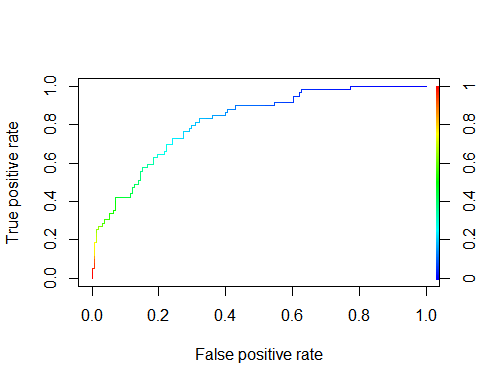
auc\_data=performance(roc\_vector,"auc")  
auc\_data

## A performance instance  
## 'Area under the ROC curve'

auc\_value=unlist(slot(auc\_data,"y.values"))  
auc\_value

## [1] 0.8163484

plot(roc\_data,colorize=T)



library(ROSE)

## Warning: package 'ROSE' was built under R version 4.0.2

## Loaded ROSE 0.0-3

dim(data\_new)

## [1] 700 9

table(data\_new$default)

##   
## 0 1   
## 517 183

data\_balanced\_over = ovun.sample(default ~ ., data = data\_new, method = "over",N =1034)$data  
class(data\_balanced\_over)

## [1] "data.frame"

dim(data\_balanced\_over)

## [1] 1034 9

table(data\_balanced\_over$default)

##   
## 0 1   
## 517 517

set.seed(100)  
sample\_data\_balanced=sample(2,nrow(data\_balanced\_over),replace = T,prob = c(.7,.3))  
train\_data\_balanced=data\_balanced\_over[sample\_data\_balanced==1,]  
dim(train\_data\_balanced)

## [1] 704 9

test\_data\_balanced=data\_balanced\_over[sample\_data\_balanced==2,]  
dim(test\_data\_balanced)

## [1] 330 9

balanced\_glm=glm(default~.,data=train\_data\_balanced,family = "binomial")  
summary(balanced\_glm)

##   
## Call:  
## glm(formula = default ~ ., family = "binomial", data = train\_data\_balanced)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.67081 -0.73291 -0.06595 0.77469 2.83304   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.528949 0.555084 -0.953 0.34063   
## age 0.048368 0.016173 2.991 0.00278 \*\*   
## ed 0.127489 0.115731 1.102 0.27064   
## employ -0.261923 0.030915 -8.472 < 2e-16 \*\*\*  
## address -0.124857 0.021666 -5.763 8.27e-09 \*\*\*  
## income -0.018701 0.009739 -1.920 0.05483 .   
## debtinc 0.036830 0.030683 1.200 0.23001   
## creddebt 0.679552 0.108751 6.249 4.14e-10 \*\*\*  
## othdebt 0.136740 0.077150 1.772 0.07633 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 974.99 on 703 degrees of freedom  
## Residual deviance: 630.36 on 695 degrees of freedom  
## AIC: 648.36  
##   
## Number of Fisher Scoring iterations: 6

balanced\_model=glm(default~employ+address+income+creddebt,data=train\_data\_balanced,family = "binomial")  
summary(balanced\_model)

##   
## Call:  
## glm(formula = default ~ employ + address + income + creddebt,   
## family = "binomial", data = train\_data\_balanced)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7650 -0.8096 -0.1021 0.8008 2.4079   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.197827 0.197332 6.070 1.28e-09 \*\*\*  
## employ -0.245109 0.026150 -9.373 < 2e-16 \*\*\*  
## address -0.085381 0.017669 -4.832 1.35e-06 \*\*\*  
## income -0.009164 0.005586 -1.640 0.101   
## creddebt 0.840206 0.086951 9.663 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 974.99 on 703 degrees of freedom  
## Residual deviance: 667.12 on 699 degrees of freedom  
## AIC: 677.12  
##   
## Number of Fisher Scoring iterations: 5

predicted\_data\_balanced\_test=predict(balanced\_model,test\_data\_balanced,type="response")  
head(predicted\_data\_balanced\_test)

## 7 12 15 22 24 27   
## 0.899759687 0.036162684 0.320951122 0.001484516 0.610927570 0.616953144

length(predicted\_data\_balanced\_test)

## [1] 330

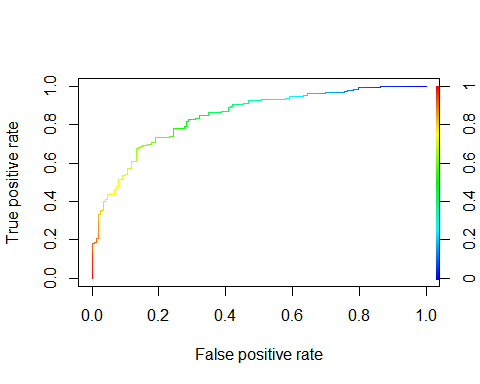
table(test\_data\_balanced$default,predicted\_data\_balanced\_test>=0.5)

##   
## FALSE TRUE  
## 0 110 42  
## 1 38 140

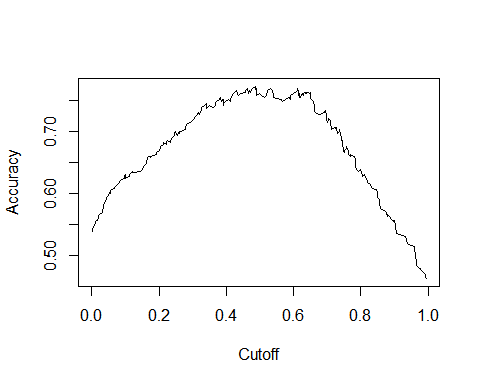
acc\_data\_test\_balance=(110+140)/(110+140+38+42)  
acc\_data\_test\_balance

## [1] 0.7575758

roc\_vector1\_test\_balance=prediction(predicted\_data\_balanced\_test,test\_data\_balanced$default)  
roc\_data\_data\_balance\_test=performance(roc\_vector1\_test\_balance,"tpr","fpr")  
accuracy\_\_test\_balance=performance(roc\_vector1\_test\_balance,"acc")  
  
plot(roc\_data\_data\_balance\_test,colorize=T)



plot(accuracy\_\_test\_balance)



auc\_data1\_balance\_test=performance(roc\_vector1\_test\_balance,"auc")  
auc\_data1\_balance\_test

## A performance instance  
## 'Area under the ROC curve'

auc\_value\_test\_value=unlist(slot(auc\_data1\_balance\_test,"y.values"))  
auc\_value\_test\_value

## [1] 0.8446555

```

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.