Case Study: Bank Defaulter

bankloan<-read.csv(file.choose(),sep = ",",header=TRUE)

head(bankloan)

str(bankloan)

OP:

$ SN : int 1 2 3 4 5 6 7 8 9 10 ...

$ AGE : int 3 1 2 3 1 3 2 3 1 2 ...

$ EMPLOY : int 17 10 15 15 2 5 20 12 3 0 ...

$ ADDRESS : int 12 6 14 14 0 5 9 11 4 13 ...

$ 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 ...

$ DEFAULTER: int 1 0 0 0 1 0 0 0 1 0 ...

## Convert age into factor ie categorical variable so that in future it can be used to categorize

bankloan$AGE<-factor(bankloan$AGE)

OP:

'data.frame': 700 obs. of 8 variables:

$ SN : int 1 2 3 4 5 6 7 8 9 10 ...

$ AGE : Factor w/ 3 levels "1","2","3": 3 1 2 3 1 3 2 3 1 2 ...

$ EMPLOY : int 17 10 15 15 2 5 20 12 3 0 ...

$ ADDRESS : int 12 6 14 14 0 5 9 11 4 13 ...

$ 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 ...

$ DEFAULTER: int 1 0 0 0 1 0 0 0 1 0 ...

##Create model

riskmodel<-glm(DEFAULTER~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT+OTHDEBT,family = binomial,data = bankloan)

summary(riskmodel)

OP:

> riskmodel<-glm(DEFAULTER~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT+OTHDEBT,family = binomial,data = bankloan)

> summary(riskmodel)

Call:

glm(formula = DEFAULTER ~ AGE + EMPLOY + ADDRESS + DEBTINC +

CREDDEBT + OTHDEBT, family = binomial, data = bankloan)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.3495 -0.6601 -0.2974 0.2509 2.8583

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.78821 0.26407 -2.985 0.00284 \*\*

AGE2 0.25202 0.26651 0.946 0.34433

AGE3 0.62707 0.36056 1.739 0.08201 .

EMPLOY -0.26172 0.03188 -8.211 < 2e-16 \*\*\*

ADDRESS -0.09964 0.02234 -4.459 8.22e-06 \*\*\*

DEBTINC 0.08506 0.02212 3.845 0.00012 \*\*\*

CREDDEBT 0.56336 0.08877 6.347 2.20e-10 \*\*\*

OTHDEBT 0.02315 0.05709 0.405 0.68517

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 804.36 on 699 degrees of freedom

Residual deviance: 553.41 on 692 degrees of freedom

AIC: 569.41

Number of Fisher Scoring iterations: 6

This shows us the significant variables.So after removing non significant variables,we create refined model

riskmodel1<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family=binomial,data = bankloan)

summary(riskmodel1)

OP:

> riskmodel1<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family=binomial,data = bankloan)

> summary(riskmodel1)

Call:

glm(formula = DEFAULTER ~ EMPLOY + ADDRESS + DEBTINC + CREDDEBT,

family = binomial, data = bankloan)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.4483 -0.6396 -0.3108 0.2583 2.8496

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -0.79107 0.25154 -3.145 0.00166 \*\*

EMPLOY -0.24258 0.02806 -8.646 < 2e-16 \*\*\*

ADDRESS -0.08122 0.01960 -4.144 3.41e-05 \*\*\*

DEBTINC 0.08827 0.01854 4.760 1.93e-06 \*\*\*

CREDDEBT 0.57290 0.08725 6.566 5.17e-11 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 804.36 on 699 degrees of freedom

Residual deviance: 556.74 on 695 degrees of freedom

AIC: 566.74

Number of Fisher Scoring iterations: 6

**Comparing this above model with Null Model**

##Create NUll model

nullmodel<-glm(DEFAULTER~1,family = binomial,data=bankloan)

summary(nullmodel)

OP:

> nullmodel<-glm(DEFAULTER~1,family = binomial,data=bankloan)

> summary(nullmodel)

Call:

glm(formula = DEFAULTER ~ 1, family = binomial, data = bankloan)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.7785 -0.7785 -0.7785 1.6380 1.6380

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.03856 0.08602 -12.07 <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: 804.36 on 699 degrees of freedom

Residual deviance: 804.36 on 699 degrees of freedom

AIC: 806.36

Number of Fisher Scoring iterations: 4

AIC Value of NULL model is much more than that of our model.AIC values indicates errors. So our model is significant.To prove we this also have ANOVA test

anova(nullmodel,riskmodel1,test="Chisq")

#Anova method proves. Here p value<0.05 shows that our model is more significant with respect to null model

OP:

Analysis of Deviance Table

Model 1: DEFAULTER ~ 1

Model 2: DEFAULTER ~ EMPLOY + ADDRESS + DEBTINC + CREDDEBT

Resid. Df Resid. Dev Df Deviance Pr(>Chi)

1 699 804.36

2 695 556.74 4 247.62 < 2.2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Since value of p is very less than 0.05,we reject Ho, i.e. there is no significant difference between null model and riskmodel1(w.r.t. Variance).Anova tests only whether the variance of 2 models is equal or not.To find out which is better model, we refer to AIC Value.

FIT THE MODEL

#Fitted used in MLR gives continuous value.Here fitted function gives probability

bankloan$predprob<-round(fitted(riskmodel1),2)

head(bankloan)

OP:

> bankloan$predprob<-round(fitted(riskmodel1),2)

> head(bankloan)

SN AGE EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT DEFAULTER predprob

1 1 3 17 12 9.3 11.36 5.01 1 0.81

2 2 1 10 6 17.3 1.36 4.00 0 0.20

3 3 2 15 14 5.5 0.86 2.17 0 0.01

4 4 3 15 14 2.9 2.66 0.82 0 0.02

5 5 1 2 0 17.3 1.79 3.06 1 0.78

6 6 3 5 5 10.2 0.39 2.16 0 0.22

Predprob finds probabilities from the equation.

If p>0.5 Y=1

If p<0.5 Y=0

Misclassification Rate

install.packages("gmodels")

library(gmodels)

table(bankloan$DEFAULTER,fitted(riskmodel)>0.5)

no\_of\_rows<-dim(bankloan)

no\_of\_rows

misclassification\_rate=((41+88)/700)\*100

misclassification\_rate

OP

> library(gmodels)

Warning message:

package ‘gmodels’ was built under R version 3.4.4

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.5)

FALSE TRUE

0 476 41

1 88 95

> no\_of\_rows<-dim(bankloan)

> no\_of\_rows

[1] 700 9

> misclassification\_rate=(41+88)/700\*100

> misclassification\_rate

[1] 18.42857

table(bankloan$DEFAULTER,fitted(riskmodel)>0.1)

table(bankloan$DEFAULTER,fitted(riskmodel)>0.2)

table(bankloan$DEFAULTER,fitted(riskmodel)>0.3)

table(bankloan$DEFAULTER,fitted(riskmodel)>0.4)

table(bankloan$DEFAULTER,fitted(riskmodel)>0.5)

missClassificationRate\_of\_0.4<-((68+67)/700)\*100

missClassificationRate\_of\_0.4

OP:

> coef(riskmodel1)

(Intercept) EMPLOY ADDRESS DEBTINC CREDDEBT

-0.79107079 -0.24258492 -0.08122146 0.08826530 0.57289682

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.1)

FALSE TRUE

0 248 269

1 12 171

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.2)

FALSE TRUE

0 347 170

1 25 158

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.3)

FALSE TRUE

0 407 110

1 44 139

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.4)

FALSE TRUE

0 449 68

1 67 116

> table(bankloan$DEFAULTER,fitted(riskmodel)>0.5)

FALSE TRUE

0 476 41

1 88 95

> missClassificationRate\_of\_0.4<-((68+67)/700)\*100

> missClassificationRate\_of\_0.4

[1] 19.28571

head(bankloan$predprob)

pred<-prediction(bankloan$predprob,bankloan$DEFAULTER)

perf<-performance(pred,"tpr","fpr")

plot(perf)

abline(0,1)

OP:

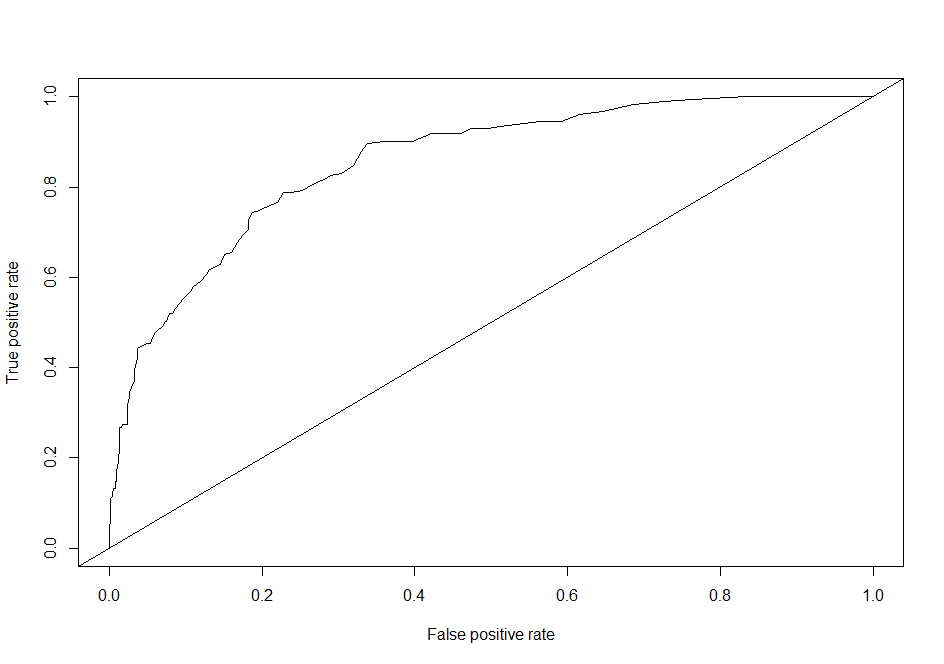
> head(bankloan$predprob)

[1] 0.81 0.20 0.01 0.02 0.78 0.22

> pred<-prediction(bankloan$predprob,bankloan$DEFAULTER)

> perf<-performance(pred,"tpr","fpr")

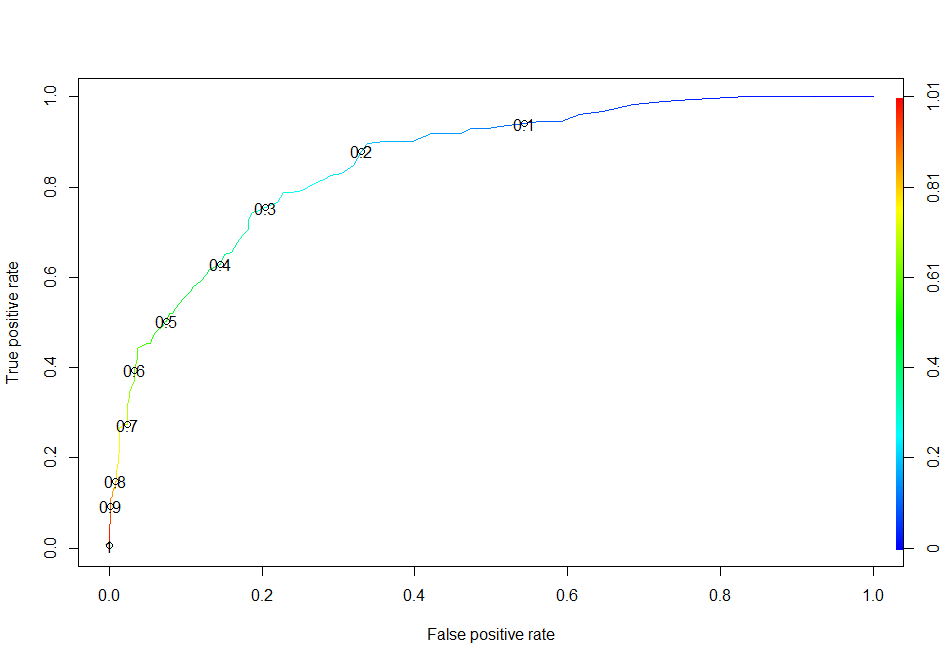
> plot(perf)

> abline(0,1) 

## To understand better use this plot/curve

plot(perf,colorize=T,print.cutoffs.at=seq(0.1,by=0.1))

OP:



### Co-efficient is expressed in the form of odds (log). So take exponential of that

##These values are relatvie(under influence of other variables) All are relative effects

coef(riskmodel)

exp(coef(riskmodel))

abs((exp(coef(riskmodel))-1)\*100)

OP:

|  |
| --- |
| > coef(riskmodel)  (Intercept) AGE2 AGE3 EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT  -0.78821163 0.25202282 0.62706544 -0.26172484 -0.09964341 0.08506325 0.56336319 0.02314748  > exp(coef(riskmodel))  (Intercept) AGE2 AGE3 EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT  0.4546572 1.2866254 1.8721087 0.7697228 0.9051601 1.0887859 1.7565703 1.0234175  > abs((exp(coef(riskmodel))-1)\*100)  (Intercept) AGE2 AGE3 EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT  54.534283 28.662540 87.210870 23.027721 9.483987 8.878593 75.657026 2.341746 |
|  |
| |  | | --- | | > | |

coef(riskmodel1)

exp(coef(riskmodel1))

abs((exp(coef(riskmodel1))-1)\*100)

#abs values are easy to answer

OP:

> coef(riskmodel1)

(Intercept) EMPLOY ADDRESS DEBTINC CREDDEBT

-0.79107079 -0.24258492 -0.08122146 0.08826530 0.57289682

> exp(coef(riskmodel1))

(Intercept) EMPLOY ADDRESS DEBTINC CREDDEBT

0.4533591 0.7845971 0.9219895 1.0922779 1.7733968

> abs((exp(coef(riskmodel1))-1)\*100)

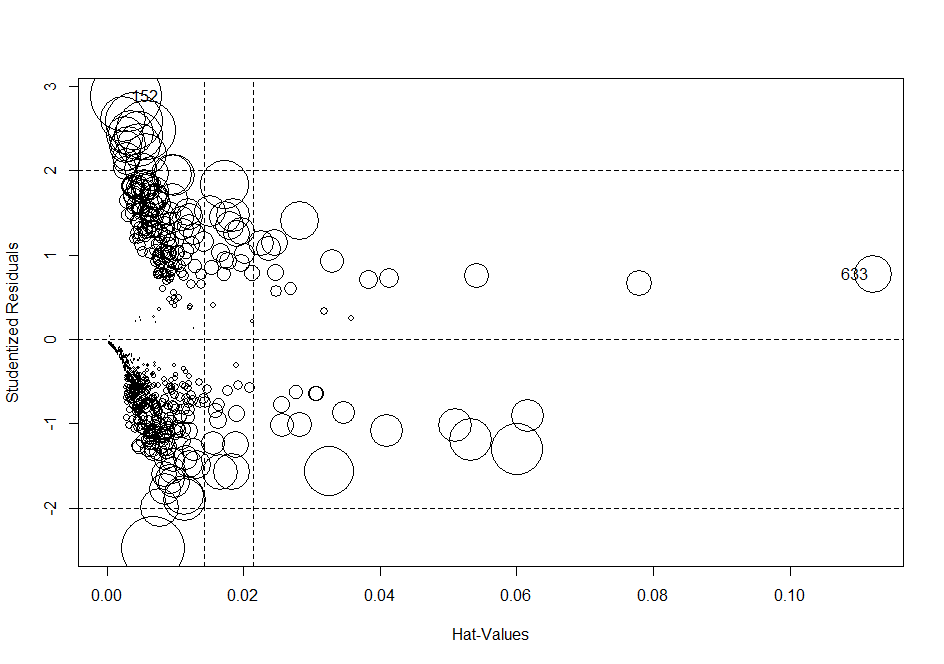
(Intercept) EMPLOY ADDRESS DEBTINC CREDDEBT

54.664092 21.540288 7.801051 9.227786 77.339683

sign indicates it will reduce. If number of years at a particular addresses increases,the percentage that the person will not default reduces by 7.8%

library(car)

influencePlot(riskmodel1)



To find out vif

vif(riskmodel1)

OP:

> vif(riskmodel1)

EMPLOY ADDRESS DEBTINC CREDDEBT

1.807288 1.131470 1.328375 2.335715

As Vif of all values is less than 5 there is no multicolinearity.

**To choose variables:**

step(riskmodel,scope = list(lower=nullmodel,upper=riskmodel),direction = "backward")

OP:

> step(riskmodel,scope = list(lower=nullmodel,upper=riskmodel),direction = "backward")

Start: AIC=569.41

DEFAULTER ~ AGE + EMPLOY + ADDRESS + DEBTINC + CREDDEBT + OTHDEBT

Df Deviance AIC

- OTHDEBT 1 553.57 567.57

- AGE 2 556.42 568.42

<none> 553.41 569.41

- DEBTINC 1 568.64 582.64

- ADDRESS 1 574.99 588.99

- CREDDEBT 1 611.64 625.64

- EMPLOY 1 651.68 665.68

Step: AIC=567.57

DEFAULTER ~ AGE + EMPLOY + ADDRESS + DEBTINC + CREDDEBT

Df Deviance AIC

- AGE 2 556.74 566.74

<none> 553.57 567.57

- ADDRESS 1 574.99 586.99

- DEBTINC 1 577.58 589.58

- CREDDEBT 1 617.49 629.49

- EMPLOY 1 659.19 671.19

Step: AIC=566.74

DEFAULTER ~ EMPLOY + ADDRESS + DEBTINC + CREDDEBT

Df Deviance AIC

<none> 556.74 566.74

- ADDRESS 1 575.64 583.64

- DEBTINC 1 580.02 588.02

- CREDDEBT 1 622.35 630.35

- EMPLOY 1 667.22 675.22

Call: glm(formula = DEFAULTER ~ EMPLOY + ADDRESS + DEBTINC + CREDDEBT,

family = binomial, data = bankloan)

Coefficients:

(Intercept) EMPLOY ADDRESS DEBTINC CREDDEBT

-0.79107 -0.24258 -0.08122 0.08827 0.57290

Degrees of Freedom: 699 Total (i.e. Null); 695 Residual

Null Deviance: 804.4

Residual Deviance: 556.7 AIC: 566.7

**Validations**

**1.Hold Out Validation**

**library(caret)**

**index<-createDataPartition(bankloan$DEFAULTER,P=0.7,list=F)**

head(index)

dim(index)

**OP:**

> head(index)

Resample1

[1,] 1

[2,] 3

[3,] 4

[4,] 5

[5,] 6

[6,] 7

> dim(index)

[1] 800 1

Warning message

traindata1<-bankloan[index,]

testdata1<-bankloan[-index,]

dim(traindata1)

dim(testdata1)

head(traindata1)

OP:

> traindata1<-bankloan[index,]

> testdata1<-bankloan[-index,]

> dim(traindata1)

[1] 800 9

> dim(testdata1)

[1] 140 9

> head(traindata1)

SN AGE EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT DEFAULTER predprob

1 1 3 17 12 9.3 11.36 5.01 1 0.81

3 3 2 15 14 5.5 0.86 2.17 0 0.01

4 4 3 15 14 2.9 2.66 0.82 0 0.02

5 5 1 2 0 17.3 1.79 3.06 1 0.78

6 6 3 5 5 10.2 0.39 2.16 0 0.22

7 7 2 20 9 30.6 3.83 16.67 0 0.19

##Like if else we have a substitute for it function:cut

riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family = binomial,data=traindata1)

traindata1$predprob<-predict(riskmodel,traindata1,type='response')

head(traindata1)

OP:

> riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family = binomial,data=traindata1)

Warning messages:

1: In doTryCatch(return(expr), name, parentenv, handler) :

display list redraw incomplete

2: In doTryCatch(return(expr), name, parentenv, handler) :

invalid graphics state

3: In doTryCatch(return(expr), name, parentenv, handler) :

invalid graphics state

**Operations on traindata1**

> traindata1$predprob<-predict(riskmodel,traindata1,type='response')

> traindata1$Predicted\_DefalutersY<-ifelse(traindata1$predprob>0.3,1,0)

> head(traindata1)

> head(traindata1)

SN AGE EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT DEFAULTER predprob

1 1 3 17 12 9.3 11.36 5.01 1 0.733004696

3 3 2 15 14 5.5 0.86 2.17 0 0.007440155

4 4 3 15 14 2.9 2.66 0.82 0 0.014814899

5 5 1 2 0 17.3 1.79 3.06 1 0.805317703

6 6 3 5 5 10.2 0.39 2.16 0 0.206477083

7 7 2 20 9 30.6 3.83 16.67 0 0.233605541

Predicted\_DefalutersY

1 1

3 0

4 0

5 1

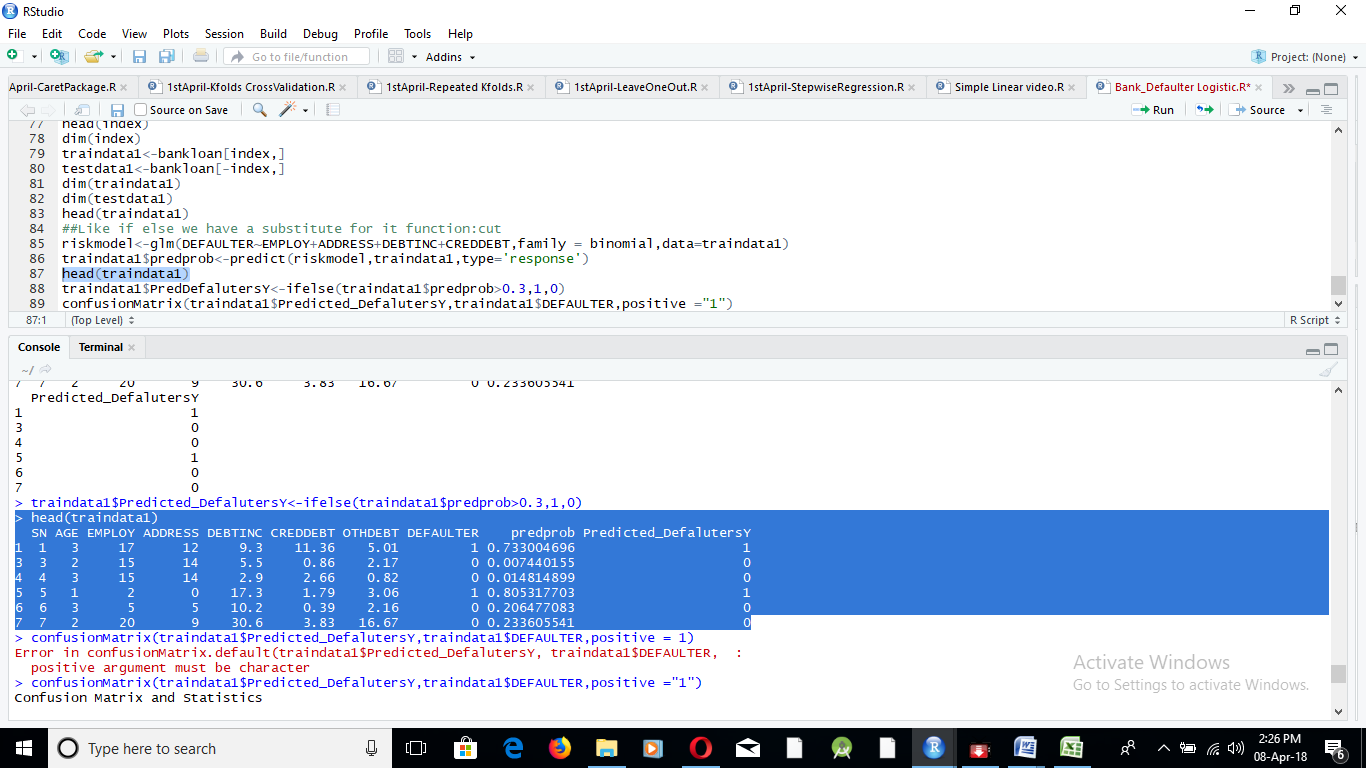
6 0

7 0

traindata1$PredDefalutersY<-ifelse(traindata1$predprob>0.3,1,0)

head(traindata1)

OP:



ConfusionMatrix(traindata1$Predicted\_DefalutersY,traindata1$DEFAULTER,positive ="1")

OP:

Here positive=”1” is written when defaulter is indicated by 1 and non defaulter is indicated by 0.For us defaulter is the positive event.If it is other way round (i.e.defaulter is represented by 0 and non defaulter by 1 ),then positive=0.

> confusionMatrix(traindata1$Predicted\_DefalutersY,traindata1$DEFAULTER,positive ="1")

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 346 32

1 74 108

Accuracy : 0.8107

95% CI : (0.7758, 0.8423)

No Information Rate : 0.75

P-Value [Acc > NIR] : 0.0003899

Kappa : 0.5411

Mcnemar's Test P-Value : 6.826e-05

Sensitivity : 0.7714

Specificity : 0.8238

Pos Pred Value : 0.5934

Neg Pred Value : 0.9153

Prevalence : 0.2500

Detection Rate : 0.1929

Detection Prevalence : 0.3250

Balanced Accuracy : 0.7976

'Positive' Class : 1

Operation on Test Data

testdata1$predprob<-predict(riskmodel2,testdata1,type = 'response')

head(testdata1)

OP

> testdata1$predprob<-predict(riskmodel2,testdata1,type = 'response')

> head(testdata1)

SN AGE EMPLOY ADDRESS DEBTINC CREDDEBT OTHDEBT DEFAULTER predprob

2 2 1 10 6 17.3 1.36 4.00 0 0.20967557

8 8 3 12 11 3.6 0.13 1.24 0 0.01095389

12 12 1 4 0 5.2 0.25 0.94 0 0.21472595

22 22 3 17 21 5.6 0.59 1.82 0 0.00212538

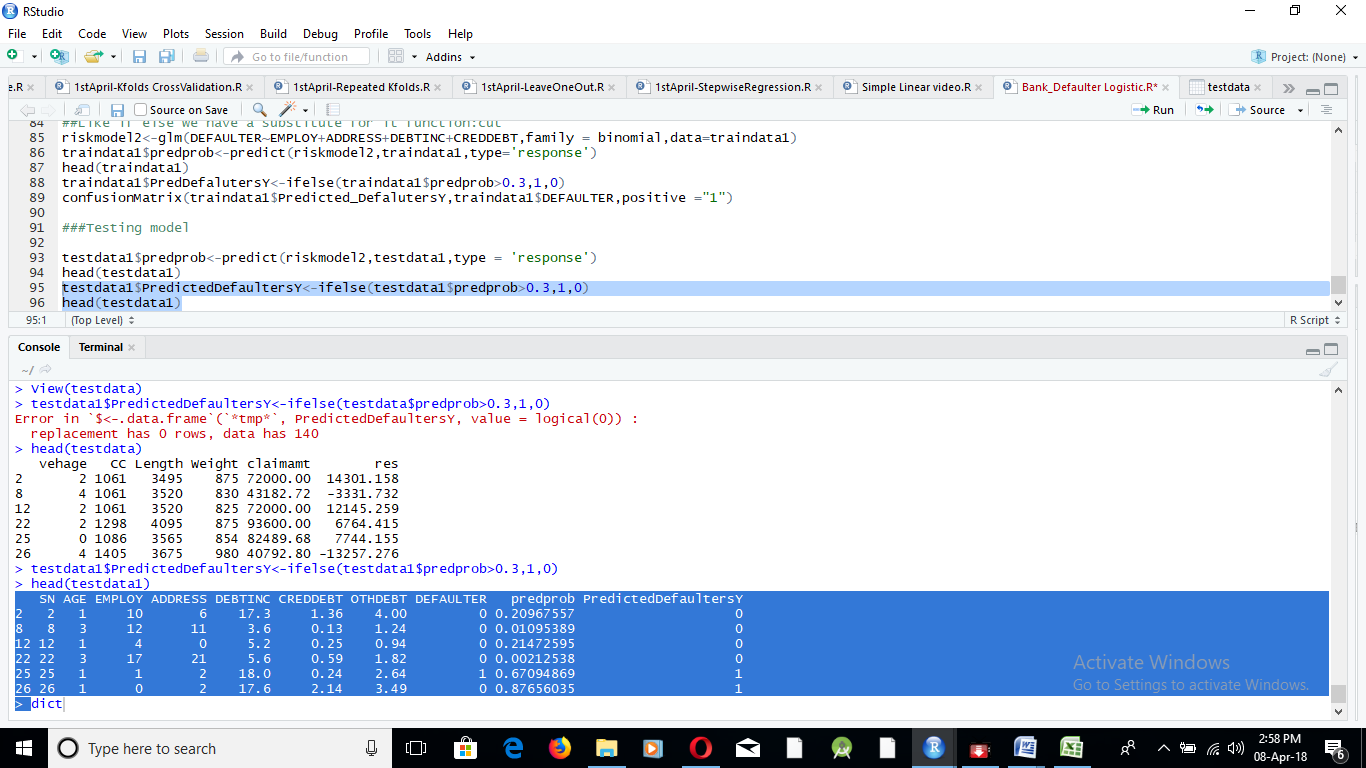
25 25 1 1 2 18.0 0.24 2.64 1 0.67094869

26 26 1 0 2 17.6 2.14 3.49 0 0.87656035

testdata1$PredictedDefaultersY<-ifelse(testdata1$predprob>0.3,1,0)

head(testdata1)

OP:



confusionMatrix(testdata1$PredictedDefaultersY,testdata1$DEFAULTER)

OP:

Reference

Prediction 0 1

0 72 17

1 25 26

Accuracy : 0.7

95% CI : (0.6168, 0.7745)

No Information Rate : 0.6929

P-Value [Acc > NIR] : 0.4682

Kappa : 0.3298

Mcnemar's Test P-Value : 0.2801

Sensitivity : 0.7423

Specificity : 0.6047

Pos Pred Value : 0.8090

Neg Pred Value : 0.5098

Prevalence : 0.6929

Detection Rate : 0.5143

Detection Prevalence : 0.6357

Balanced Accuracy : 0.6735

'Positive' Class : 0

**ROC CURVE FOR TRAIN AND TEST DATA**

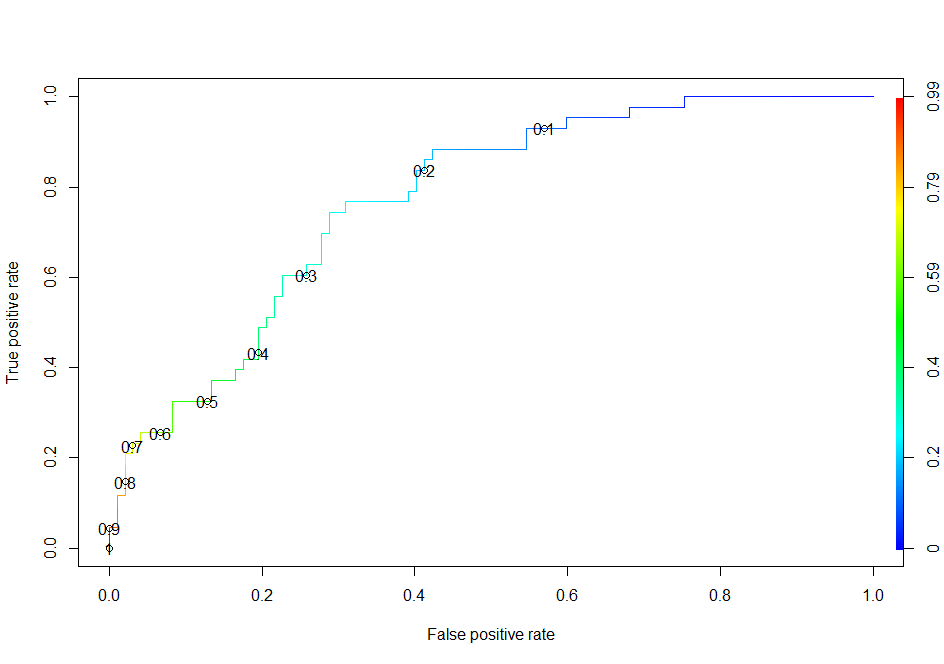
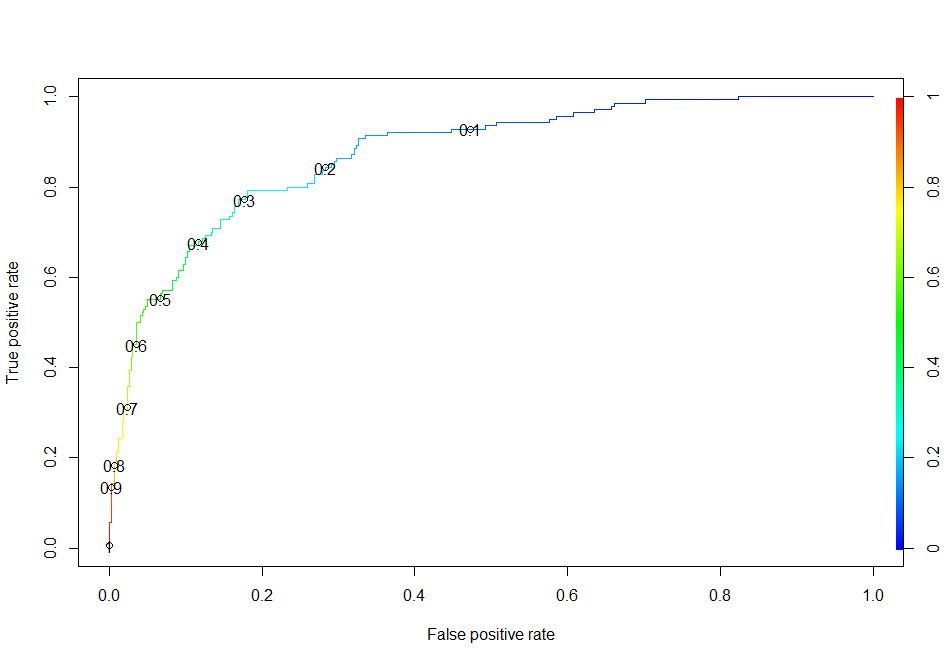
###ROC CURVE on Train data

pred\_train<-prediction(traindata1$predprob,traindata1$DEFAULTER)

perf\_train<-performance(pred\_train,"tpr","fpr")

plot(perf\_train,colorize=T,print.cutoffs.at=seq(0.1,by=0.1))

Plot



###ROC CURVE on Test Data

pred\_test<-prediction(testdata1$predprob,testdata1$DEFAULTER)

perf\_test<-performance(pred\_test,"tpr","fpr")

plot(perf\_test,colorize=T,print.cutoffs.at=seq(0.1,by=0.1))

**Accuracy of the model**

auc<-performance(pred\_train,"auc")

auc@y.values

OP:

> auc@y.values

[[1]]

[1] 0.8741327

auc<-performance(pred\_test,"auc")

[auc@y.values](mailto:auc@y.values)

OP:

> auc@y.values

[[1]]

[1] 0.7707984