> bankdata<-read.csv("D:\\Assignment\_\_Rstudio\\Multilinear\\bank-full.csv",sep = ';')

> library(caret)

> set.seed(3456)

> # Indicate the classification variable

> bankdata$y=factor(bankdata$y)

>

> # Use glm function, with y as dependent variable and all others as independent variables (IVs), written

> # in the usual manner with '+' sign.

> logitModel=glm(y~poutcome+previous+pdays+campaign+duration+month+day+contact+loan+housing+balance+default+education+marital+job+age, data=bankdata,family=binomial(logit))

> # As all variables besides, y, are IVs, we could have used a shorter form (with dot), as:

> logitModel=glm(y~., data=bankdata,family=binomial(logit))

> # Get model summary

> summary(logitModel)

Call:

glm(formula = y ~ ., family = binomial(logit), data = bankdata)

Deviance Residuals:

Min 1Q Median 3Q Max

-5.7286 -0.3744 -0.2530 -0.1502 3.4288

Coefficients:

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

(Intercept) -2.536e+00 1.837e-01 -13.803 < 2e-16 \*\*\*

age 1.127e-04 2.205e-03 0.051 0.959233

jobblue-collar -3.099e-01 7.267e-02 -4.264 2.01e-05 \*\*\*

jobentrepreneur -3.571e-01 1.256e-01 -2.844 0.004455 \*\*

jobhousemaid -5.040e-01 1.365e-01 -3.693 0.000221 \*\*\*

jobmanagement -1.653e-01 7.329e-02 -2.255 0.024130 \*

jobretired 2.524e-01 9.722e-02 2.596 0.009436 \*\*

jobself-employed -2.983e-01 1.120e-01 -2.664 0.007726 \*\*

jobservices -2.238e-01 8.406e-02 -2.662 0.007763 \*\*

jobstudent 3.821e-01 1.090e-01 3.505 0.000457 \*\*\*

jobtechnician -1.760e-01 6.893e-02 -2.554 0.010664 \*

jobunemployed -1.767e-01 1.116e-01 -1.583 0.113456

jobunknown -3.133e-01 2.335e-01 -1.342 0.179656

maritalmarried -1.795e-01 5.891e-02 -3.046 0.002318 \*\*

maritalsingle 9.250e-02 6.726e-02 1.375 0.169066

educationsecondary 1.835e-01 6.479e-02 2.833 0.004618 \*\*

educationtertiary 3.789e-01 7.532e-02 5.031 4.88e-07 \*\*\*

educationunknown 2.505e-01 1.039e-01 2.411 0.015915 \*

defaultyes -1.668e-02 1.628e-01 -0.102 0.918407

balance 1.283e-05 5.148e-06 2.493 0.012651 \*

housingyes -6.754e-01 4.387e-02 -15.395 < 2e-16 \*\*\*

loanyes -4.254e-01 5.999e-02 -7.091 1.33e-12 \*\*\*

contacttelephone -1.634e-01 7.519e-02 -2.173 0.029784 \*

contactunknown -1.623e+00 7.317e-02 -22.184 < 2e-16 \*\*\*

day 9.969e-03 2.497e-03 3.993 6.53e-05 \*\*\*

monthaug -6.939e-01 7.847e-02 -8.842 < 2e-16 \*\*\*

monthdec 6.911e-01 1.767e-01 3.912 9.17e-05 \*\*\*

monthfeb -1.473e-01 8.941e-02 -1.648 0.099427 .

monthjan -1.262e+00 1.217e-01 -10.367 < 2e-16 \*\*\*

monthjul -8.308e-01 7.740e-02 -10.733 < 2e-16 \*\*\*

monthjun 4.536e-01 9.367e-02 4.843 1.28e-06 \*\*\*

monthmar 1.590e+00 1.199e-01 13.265 < 2e-16 \*\*\*

monthmay -3.991e-01 7.229e-02 -5.521 3.36e-08 \*\*\*

monthnov -8.734e-01 8.441e-02 -10.347 < 2e-16 \*\*\*

monthoct 8.814e-01 1.080e-01 8.159 3.37e-16 \*\*\*

monthsep 8.741e-01 1.195e-01 7.314 2.58e-13 \*\*\*

duration 4.194e-03 6.453e-05 64.986 < 2e-16 \*\*\*

campaign -9.078e-02 1.014e-02 -8.955 < 2e-16 \*\*\*

pdays -1.027e-04 3.061e-04 -0.335 0.737268

previous 1.015e-02 6.503e-03 1.561 0.118476

poutcomeother 2.035e-01 8.986e-02 2.265 0.023543 \*

poutcomesuccess 2.291e+00 8.235e-02 27.821 < 2e-16 \*\*\*

poutcomeunknown -9.179e-02 9.347e-02 -0.982 0.326093

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 32631 on 45210 degrees of freedom

Residual deviance: 21562 on 45168 degrees of freedom

AIC: 21648

Number of Fisher Scoring iterations: 6

> # Confusion matrix table

> prob <- predict(logitModel,type=c("response"),bankdata)

> View(prob)

> confusion<-table(prob>0.5,bankdata$y)

> confusion

no yes

FALSE 38940 3456

TRUE 982 1833

> # Model Accuracy

> Accuracy<-sum(diag(confusion)/sum(confusion))

> Accuracy#70.52

[1] 0.901838

> 1-Accuracy

[1] 0.09816195

> sum(confusion[cbind(2:1, 1:2)])/sum(confusion)

[1] 0.09816195

> ?diag

> # ROC Curve

> #install.packages("ROCR")

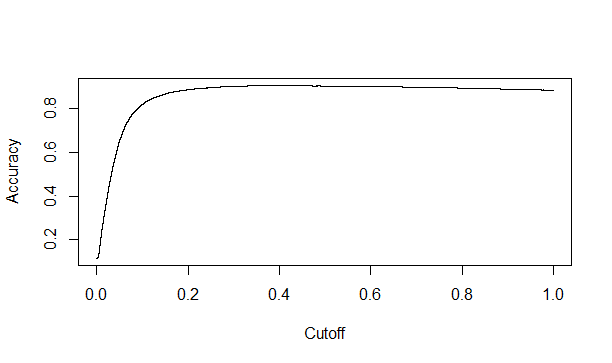
> #install.packages("ROCR", dependencies = T)

> library(ROCR)

> rocrpred<-prediction(prob,bankdata$y)

> eval <- performance(rocrpred,"acc")

> plot(eval)

> 

max <- which.max(slot(eval,"y.values")[[1]])

> acc <- slot(eval,"y.values")[[1]][max]

> cut <- slot(eval,"x.values")[[1]][max]

> cut

41712

0.3945801

> print(c(Accuracy=acc, Cutoff=cut))

Accuracy Cutoff.41712

0.9039835 0.3945801

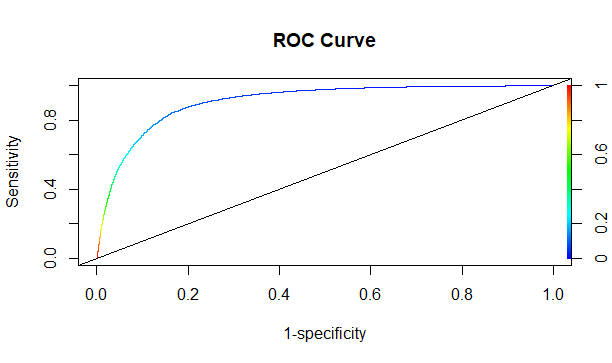
#Area under the curve"

> rocrperf<-performance(rocrpred,'tpr','fpr')

> #plot(rocrperf,colorize=T,text.adj=c(0.5,NA))

> plot(rocrperf,colorize=T,main="ROC Curve",xlab="1-specificity",ylab="Sensitivity")

> ?plot

 > abline(a=0,b=1)

> auc <- performance(rocrpred,"auc")

> auc <- unlist(slot(auc,"y.values"))

> auc <- round(auc,4)

> auc

[1] 0.9079

> ##Alternative

> pred\_values <- NULL

> pred\_values

NULL

> yes\_no <- NULL

> for (i in 1:45211){

+ pred\_values[i] <- ifelse(prob[i]>=0.5,1,0)

+ yes\_no[i] <- ifelse(prob[i]>=0.5,"yes","no")

+ }

>

> bankdata[,"prob"] <- prob

> View(prob)

> View(bankdata)

> bankdata[,"pred\_values"] <- pred\_values

> View(pred\_values)

> View(bankdata)

> bankdata[,"yes\_no"] <- yes\_no

> View(bankdata)

>

> View(bankdata[,c(1,2,3, 4, 5, 6,7,8,9)])

>

> # Accuracy

> acc <- table(bankdata$y,pred\_values)

> acc

pred\_values

0 1

no 38940 982

yes 3456 1833

> Accuracy<-sum(diag(acc)/sum(acc))

> Accuracy # 70.62

[1] 0.901838

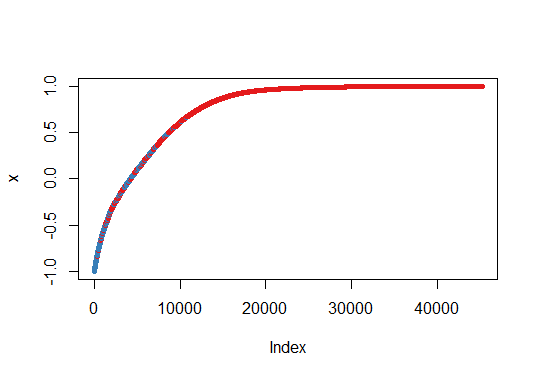
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Feature selection\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

install.packages("randomForest")

library(randomForest)

bankdata.rf <- randomForest(y ~ ., bankdata, keep.forest=FALSE)

plot(margin(bankdata.rf))



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Boruta Methos for Feature selection\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> library(Boruta)

> boruta\_output <- Boruta(y ~ ., data=na.omit(bankdata), doTrace=2)

1. run of importance source...

2. run of importance source...

3. run of importance source...

4. run of importance source...

5. run of importance source...

6. run of importance source...

7. run of importance source...

8. run of importance source...

9. run of importance source...

10. run of importance source...

11. run of importance source...

After 11 iterations, +4.2 mins:

confirmed 18 attributes: age, balance, campaign, contact, day and 13 more;

still have 1 attribute left.

12. run of importance source...

13. run of importance source...

14. run of importance source...

15. run of importance source...

After 15 iterations, +5.6 mins:

confirmed 1 attribute: default;

no more attributes left.

> boruta\_signif <- names(boruta\_output$finalDecision[boruta\_output$finalDecision %in% c("Confirmed", "Tentative")])

> print(boruta\_signif)

[1] "age" "job" "marital" "education"

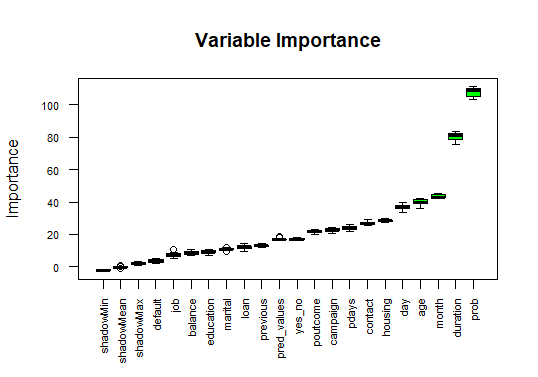
[5] "default" "balance" "housing" "loan"

[9] "contact" "day" "month" "duration"

[13] "campaign" "pdays" "previous" "poutcome"

[17] "prob" "pred\_values" "yes\_no"

> plot(boruta\_output, cex.axis=.7, las=2, xlab="", main="Variable Importance") # plot variable importance



Blue boxplots correspond to minimal, average and maximum Z score of a shadow attribute. Red, yellow and green boxplots represent Z scores of rejected, tentative and confirmed attributes respectively.

finalModel=glm(y~prob+duration+month+age, data=bankdata,family=binomial(logit))

> # As all variables besides, y, are IVs, we could have used a shorter form (with dot), as:

> #finalModel=glm(y~., data=bankdata,family=binomial(logit))

> # Get model summary

> summary(finalModel)

Call:

glm(formula = y ~ prob + duration + month + age, family = binomial(logit),

data = bankdata)

Deviance Residuals:

Min 1Q Median 3Q Max

-3.8457 -0.3607 -0.2982 -0.2593 2.6829

Coefficients:

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

(Intercept) -2.991e+00 8.631e-02 -34.656 < 2e-16 \*\*\*

prob 4.944e+00 1.128e-01 43.842 < 2e-16 \*\*\*

duration 1.187e-03 8.506e-05 13.953 < 2e-16 \*\*\*

monthaug -2.081e-01 7.345e-02 -2.834 0.00460 \*\*

monthdec 2.438e-01 1.808e-01 1.348 0.17761

monthfeb -1.225e-02 8.401e-02 -0.146 0.88407

monthjan -3.279e-01 1.178e-01 -2.782 0.00540 \*\*

monthjul -4.517e-01 7.565e-02 -5.971 2.36e-09 \*\*\*

monthjun -3.592e-01 7.800e-02 -4.605 4.13e-06 \*\*\*

monthmar 3.605e-01 1.253e-01 2.878 0.00401 \*\*

monthmay -6.345e-01 6.956e-02 -9.122 < 2e-16 \*\*\*

monthnov -3.280e-01 8.373e-02 -3.918 8.94e-05 \*\*\*

monthoct 3.014e-01 1.115e-01 2.702 0.00689 \*\*

monthsep 2.729e-01 1.220e-01 2.236 0.02534 \*

age -1.150e-03 1.546e-03 -0.743 0.45723

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 32631 on 45210 degrees of freedom

Residual deviance: 22912 on 45196 degrees of freedom

AIC: 22942

Number of Fisher Scoring iterations: 5

> # Confusion matrix table

> prob <- predict(finalModel,type=c("response"),bankdata)

> View(prob)

> confusion<-table(prob>0.5,bankdata$y)

> confusion

no yes

FALSE 38963 3519

TRUE 959 1770

> # Model Accuracy

> Accuracy<-sum(diag(confusion)/sum(confusion))

> Accuracy#70.52

[1] 0.9009533

> 1-Accuracy

[1] 0.09904669

> sum(confusion[cbind(2:1, 1:2)])/sum(confusion)

[1] 0.09904669

> ?diag

> # Model Accuracy

> Accuracy<-sum(diag(confusion)/sum(confusion))

> Accuracy#70.52

[1] 0.9009533

> 1-Accuracy

[1] 0.09904669

> sum(confusion[cbind(2:1, 1:2)])/sum(confusion)

[1] 0.09904669

> ?diag

> # ROC Curve

> #install.packages("ROCR")

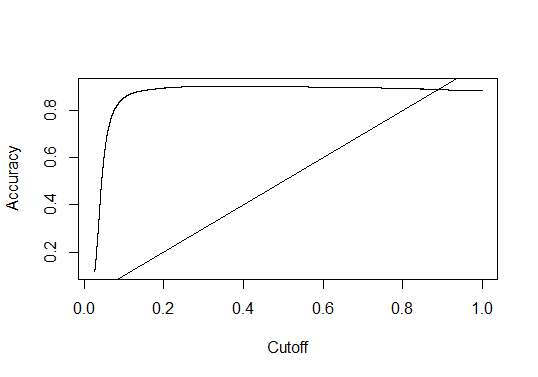
> #install.packages("ROCR", dependencies = T)

> library(ROCR)

> rocrpred<-prediction(prob,bankdata$y)

> eval <- performance(rocrpred,"acc")

> plot(eval)

‘

>

>

> max <- which.max(slot(eval,"y.values")[[1]])

> acc <- slot(eval,"y.values")[[1]][max]

> cut <- slot(eval,"x.values")[[1]][max]

> cut

40447

0.3345077

> print(c(Accuracy=acc, Cutoff=cut))

Accuracy Cutoff.40447

0.9033642 0.3345077

>

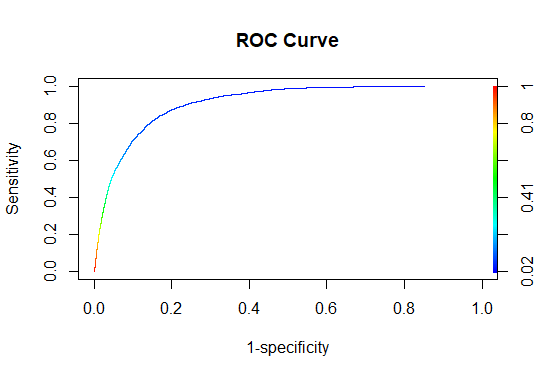
> #Area under the curve"

> rocrperf<-performance(rocrpred,'tpr','fpr')

> #plot(rocrperf,colorize=T,text.adj=c(0.5,NA))

> plot(rocrperf,colorize=T,main="ROC Curve",xlab="1-specificity",ylab="Sensitivity")

> ?plot



> abline(a=0,b=1)

> auc <- performance(rocrpred,"auc")

> auc <- unlist(slot(auc,"y.values"))

> auc <- round(auc,4)

> auc

[1] 0.9085