**Default Prediction**

Rawdata and validation data save in CSV format.Let us take data into R .

> #default bank prediction

> data<- read.csv("C:\\Projects\\Defaultprediction\\rawdata.csv",header = TRUE)

> validationdata<-read.csv("C:\\Projects\\Defaultprediction\\validationdata.csv",header = TRUE)

Target variable will be created from networth next year column data.Below is the target variable creation.

> data$default=ifelse(data$Networth.Next.Year<=0,1,0)

Have a look into the variables type by str() command.

> str(data)

'data.frame': 3541 obs. of 53 variables:

$ Num : int 3274 327 3434 3199 3164 2800 3267 1750 2293 1070 ...

$ Networth.Next.Year : num -74266 -7591 -3977 -2550 -1734 ...

$ Total.assets : num 112637 20099 33592 6894 15319 ...

$ Net.worth : num 29240 398 1666 782 1029 ...

$ Total.income : num 7330 1239 33935 849 2387 ...

$ Change.in.stock : num NA -6.7 53.6 -18.3 70.3 NA 21.4 7.5 -38.2 36.1 ...

we have 3541 observations and 53 variables.In above only few output pasted.

Summary() function gives minimum,1st quartile,median,mean,3rd quartile,maximum values.Summary() also report missing values in separate row.Below are few output of summary() function.

> summary(data)

Num Networth.Next.Year Total.assets Net.worth Total.income Change.in.stock

Min. : 1 Min. :-74265.6 Min. : 0.1 Min. : 0.0 Min. : 0.0 Min. :-3029.40

1st Qu.: 886 1st Qu.: 31.7 1st Qu.: 91.3 1st Qu.: 31.3 1st Qu.: 106.5 1st Qu.: -1.80

Median :1773 Median : 116.3 Median : 309.7 Median : 102.3 Median : 444.9 Median : 1.60

Mean :1772 Mean : 1616.3 Mean : 3443.4 Mean : 1295.9 Mean : 4582.8 Mean : 41.49

3rd Qu.:2658 3rd Qu.: 456.1 3rd Qu.: 1098.7 3rd Qu.: 377.3 3rd Qu.: 1440.9 3rd Qu.: 18.05

Max. :3545 Max. :805773.4 Max. :1176509.2 Max. :613151.6 Max. :2442828.2 Max. :14185.50

NA's :198 NA's :458

Num is identification variable. You should not consider that for analysis. Networth Next year is target variable creation variable. Let us look into Total assets variable. Its minimum is 0.1,midian is 309.7,third quartile is 1098.7 and maximum is 1176509.2.Data for this variable lie within 1100 value but few data are very high compare to 1100.so it indicates few improbable values present .Missing values separately reported .for this variable 198 missing present.we can get %age missing values and number of rows and columns separately like below.

> nrow(data)

[1] 3541

> ncol(data)

[1] 53

> colSums(is.na(data))\*100/nrow(data) # percentage missing values in each variable

Num Networth.Next.Year

0.0000000 0.0000000

Total.assets Net.worth

0.0000000 0.0000000

Total.income Change.in.stock

5.5916408 12.9341994

Total.expenses Profit.after.tax

3.9254448 3.6995199

Reserves.and.funds Deposits..accepted.by.commercial.banks.

2.4004518  **100.0000000**

Deposits..accepted.by.commercial.banks. variable column is fully missing ,No data is present for this column.we should remove this variable from our analysis.We can see the target variable distribution like below.

> prop.table(table(data$default))

0 1

0.93137532 0.06862468

> paste(prop.table(table(data$default))\*100,"%") # default and non default %age

[1] "93.1375317706862 %" "6.86246822931375 %"

> #droping high %age missing values

> drops<-c("Deposits..accepted.by.commercial.banks.","slno")

> data<-data[,!(names(data)%in%drops)]

Out of 3541 observations we have 6.8624% are default and 93.1375% are non default.

let us look into different percentile values of each variable so that we can get little bit more idea of the distribution of variables. Do a judgmental analysis of 95%tile, 96%tile, 97%tile,99%tile and 100%tile values and find a suitable percentile for capping that variable. Suppose 99% is 15 times than 98% then decide 98% for capping.

> for (i in 1:ncol(data))

+ {

+ print(colnames(data)[i])

+ print(quantile(data[i],c(0.01,0.05,0.1,0.2,0.95,0.96,0.97,0.98,0.99,1.0),na.rm = TRUE))

+ }

[1] "Num"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

36.4 178.0 355.0 709.0 3368.0 3403.4 3438.8 3474.2 3509.6 3545.0

[1] "Networth.Next.Year"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

-77.58 -1.40 4.00 20.90 3764.40 4627.90 5786.86 8913.54 25534.10 805773.40

[1] "Total.assets"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

1.70 10.60 25.50 65.70 8452.90 10746.80 13703.08 21439.70 51658.78 1176509.20

[1] "Net.worth"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

0.30 2.90 8.40 22.30 3034.40 3757.00 5008.80 7273.68 20920.76 613151.60

[1] "Total.income"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

0.242 5.700 18.820 71.500 9339.680 10916.916 15104.682 22547.928 43480.724 2442828.200

[1] "Change.in.stock"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

-271.314 -44.190 -16.780 -3.600 171.680 199.644 258.132 370.720 675.766 14185.500

[1] "Total.expenses"

1% 5% 10% 20% 95% 96% 97% 98% 99% 100%

0.200 3.305 15.300 62.940 8769.765 10244.436 13904.000 20642.018 38580.804 2366035.300

Let us do 1% capping for lower values and 99% for higher values and median value replacement for missing values.Before that we should keep these percentile values for future reference.Below are the codes for capping and missing value treatment.

> percentiledata<-data.frame(c("Dummy"),c(0),c(0))

> names(percentiledata)<-c("varname","lowcut","highcut")

> percentiledata

varname lowcut highcut

1 Dummy 0 0

>

> for (i in 1:ncol(data))

+ {

+ x1<-quantile(data[i],0.01,na.rm = TRUE)

+ x2<-quantile(data[i],0.99,na.rm = TRUE)

+ xname<-colnames(data)[i]

+ tempdata<-data.frame(xname,x1,x2 )

+ names(tempdata)<-c("varname","lowcut","highcut")

+ percentiledata<-rbind.data.frame(percentiledata,tempdata)

+ }

>

> percentiledata

varname lowcut highcut

1 Dummy 0.0000 0.000000e+00

1% Num 36.4000 3.509600e+03

1%1 Networth.Next.Year -77.5800 2.553410e+04

1%2 Total.assets 1.7000 5.165878e+04

1%3 Net.worth 0.3000 2.092076e+04

1%4 Total.income 0.2420 4.348072e+04

1%5 Change.in.stock -271.3140 6.757660e+02

> for (i in 1:ncol(data))

+ {

+ xname<-colnames(data)[i]

+ x1<-quantile(data[i],0.01,na.rm = TRUE)

+ x2<-quantile(data[i],0.99,na.rm = TRUE)

+ xmedian<-median(data[,xname],na.rm = TRUE)

+ data[,xname]<-ifelse(data[,xname]<=x1,x1,ifelse(data[,xname]>x2,x2,data[,xname]))

+ data[,xname][is.na(data[,xname])]<-xmedian

+ }

Now create new ratio variables. "Ratio variables" always contain better information than a normal variable.You should create meaning full ratio variables.Below are the creation of few ratio variables.

> # Create new ratio variables

>

> data$PATTI<-data$Profit.after.tax/data$Total.income

> data$PATSFCRP<-data$Profit.after.tax/(data$Shareholders.funds+data$Cumulative.retained.profits)

> data$PATTS<-data$Profit.after.tax/data$Total.assets

> data$TASFCRP<-data$Total.assets/(data$Shareholders.funds+data$Cumulative.retained.profits)

> data$BOSFCRP<-data$Borrowings/(data$Shareholders.funds+data$Cumulative.retained.profits)

> data$BOTA<-data$Borrowings/data$Total.assets

> data$TITA<-data$Total.income/data$Total.assets

> data$TINFA<-data$Total.income/data$Net.fixed.assets

> data$CACLP<- data$Current.assets/data$Current.liabilities...provisions

> data$TLTA<-data$Total.liabilities/data$Total.assets

> data$ROS<-data$PBT/data$Sales

> data$PBTNWC<-data$PBT/data$Net.working.capital

> data$SLTA<-data$Sales/data$Total.assets

>

> # check new ration variable nan and inf values

When you create new variable by diving a variable there may be chance of zero division in new variable as a result "Inf" and "nan" values created in new variable.below are the codes to replace the "nan" and "inf" values by median values.

>

> for (i in 1:ncol(data))

+ {

+ xname<-colnames(data)[i]

+ xmedian<-median(data[,xname],na.rm = TRUE)

+ data[,xname][is.na(data[,xname])]<-xmedian

+ data[,xname][is.infinite(data[,xname])]<-xmedian

+ }

If mean difference of a variable in default and nondefault category are clearly distinguishable then that variable is a good variable for prediction. You can use some techniques like regression technique, variable importance plot, decision tree technique to find out the important variables.Let us plot mean in default and nondefault category of each variable.

> # find the mean between default group and non default group

> require(ggplot2)

Loading required package: ggplot2

Stackoverflow is a great place to get help: http://stackoverflow.com/tags/ggplot2.

Warning message:

package ‘ggplot2’ was built under R version 3.2.5

> for (i in 1:ncol(data))

+ {

+ gdata<-aggregate(data[i], list(company = data$default), mean,na.rm=TRUE) #it will find mean of default and non default

+ print(gdata)

+ a<-colnames(data)[i]

+ barplot(gdata[,a],legend=gdata$company,col=c("darkblue","red"),main=colnames(data)[i],names.arg=c("0(Non-Default)","1(default)"))

+ }

company Num

1 0 1759.240

2 1 1951.757

company Networth.Next.Year

1 0 981.39236

2 1 -20.63778

company Networth.Next.Year

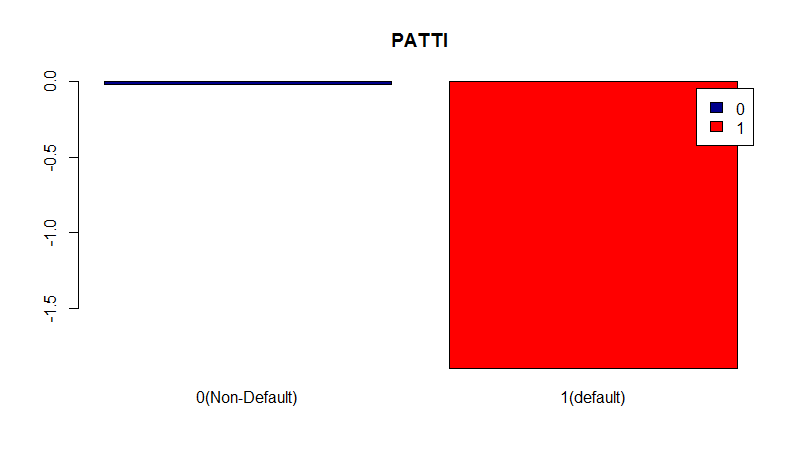
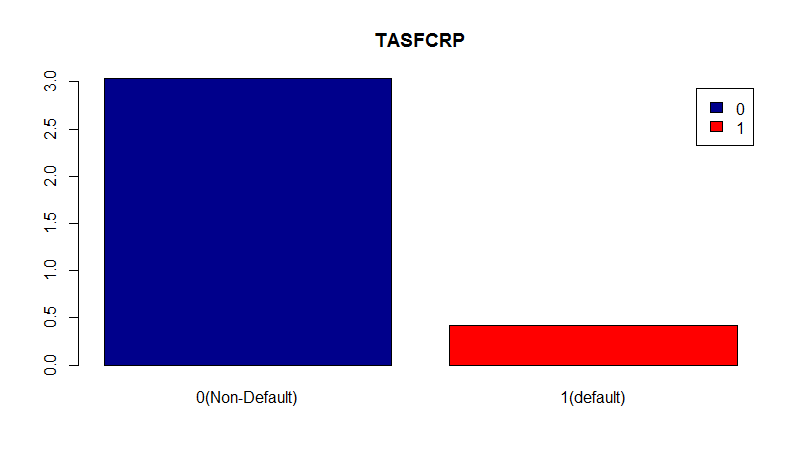
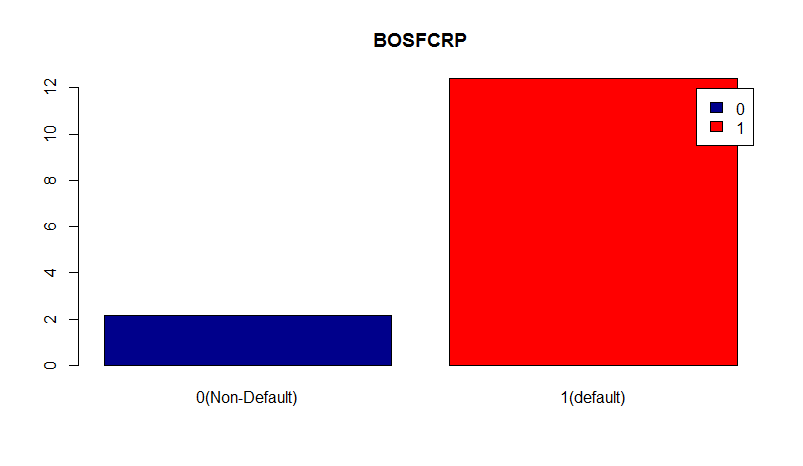
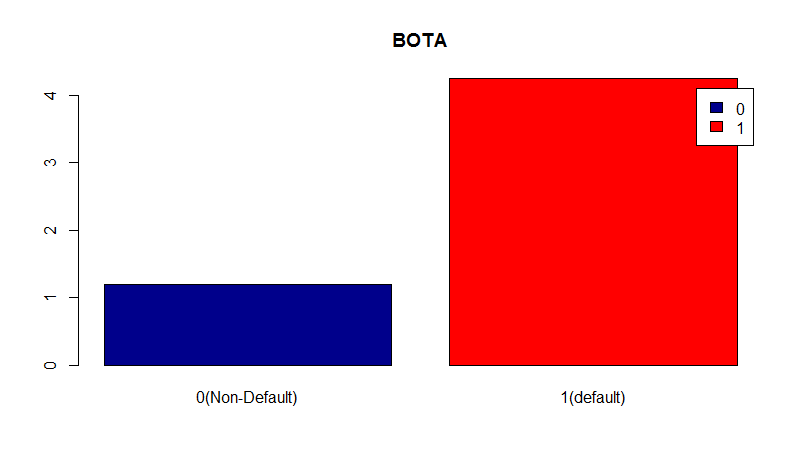
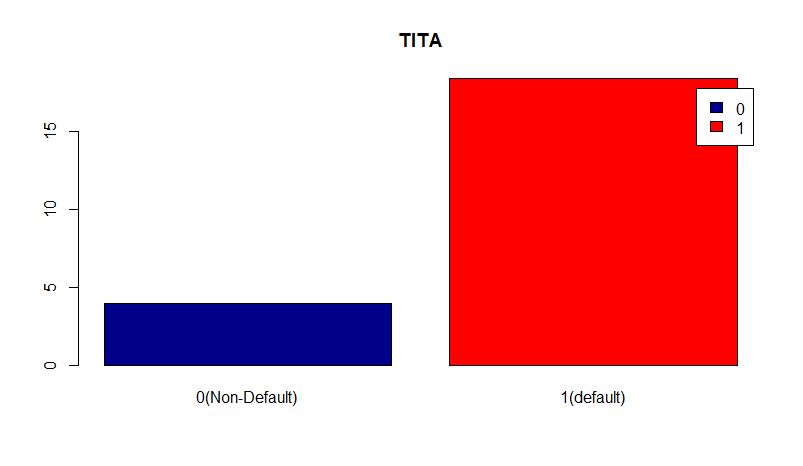
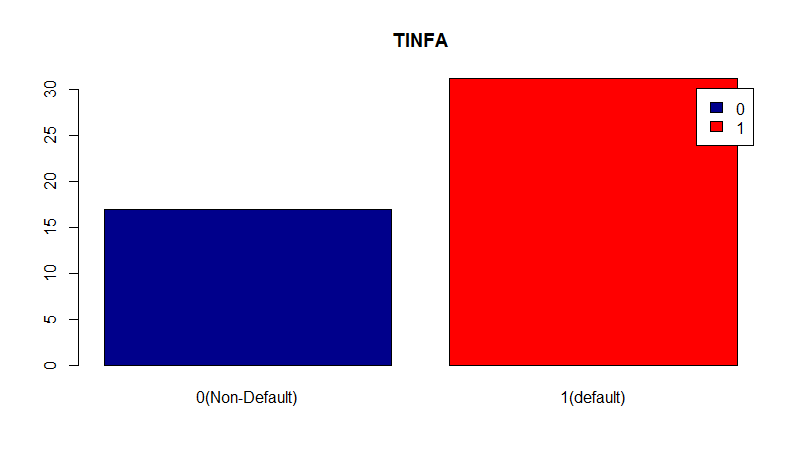
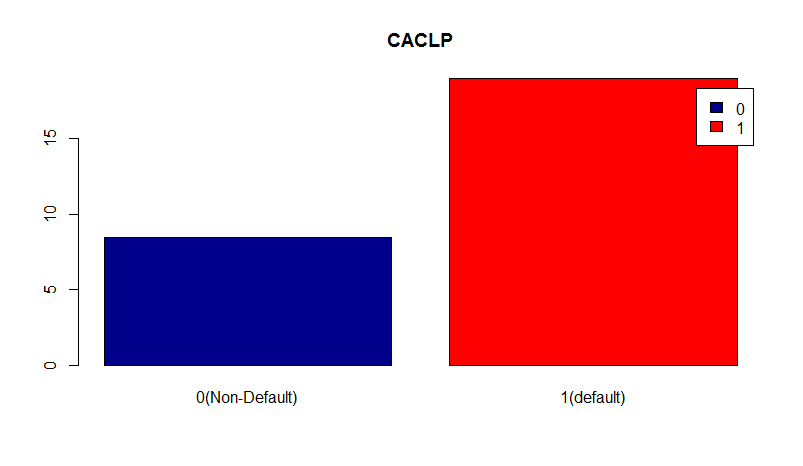
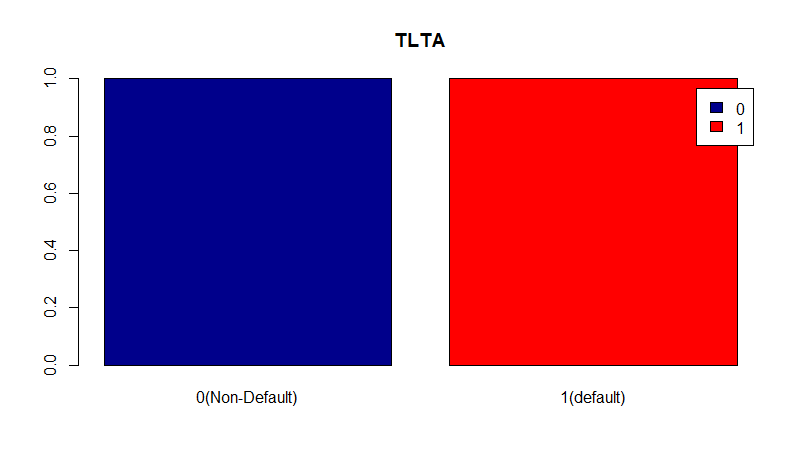
1 0 981.39236

2 1 -20.63778

company Total.assets

1 0 2113.432

2 1 1204.085



Few outputs are pasted in above. Have look into 1st graph and last graph. First graph "TLTA" variable and last graph "PATTI" variable. There is almost no mean difference between default and nondefault group of TLTA variable. So you should ignore the variable for modeling. "PATTI" variable is a good variable. You should take that.

Similar way you should find 15-20 good variables for modeling.

Now divide the data into train and test data and see the default and nondefault %age are maintained in the split.

> set.seed(151)

> sample1=sample.split(data$default,SplitRatio = 0.7)

> traindata=subset(data,sample1==TRUE)

> testdata=subset(data,sample1==FALSE)

> prop.table(table(data$default))

0 1

0.93137532 0.06862468

> prop.table(table(traindata$default))

0 1

0.93142396 0.06857604

> prop.table(table(testdata$default))

0 1

0.93126177 0.06873823

Now let us build logistic model with few selected variables and find the significance of variables in model.

> model1<-glm(default ~ PBDITA.as...of.total.income+PBT.as...of.total.income+PAT.as...of.total.income

+ +Cash.profit.as...of.total.income+Quick.ratio..times.

+ +Current.ratio..times.+Debt.to.equity.ratio..times.+Cash.to.current.liabilities..times.

+ +Cash.to.average.cost.of.sales.per.day+EPS+Adjusted.EPS+PBDITA.as...of.total.income

+ +PAT.as...of.total.income+Cash.profit.as...of.total.income+TOL.TNW

+ +PATTI+PATSFCRP+PATTS+TASFCRP+BOSFCRP+BOTA+TITA+TINFA+CACLP+ROS+PBTNWC+SLTA,

+ data=traindata,family = binomial(link="logit"))

Warning message:

glm.fit: fitted probabilities numerically 0 or 1 occurred

>

> summary(model1)

Call:

glm(formula = default ~ PBDITA.as...of.total.income + PBT.as...of.total.income +

PAT.as...of.total.income + Cash.profit.as...of.total.income +

Quick.ratio..times. + Current.ratio..times. + Debt.to.equity.ratio..times. +

Cash.to.current.liabilities..times. + Cash.to.average.cost.of.sales.per.day +

EPS + Adjusted.EPS + PBDITA.as...of.total.income + PAT.as...of.total.income +

Cash.profit.as...of.total.income + TOL.TNW + PATTI + PATSFCRP +

PATTS + TASFCRP + BOSFCRP + BOTA + TITA + TINFA + CACLP +

ROS + PBTNWC + SLTA, family = binomial(link = "logit"), data = traindata)

Deviance Residuals:

Min 1Q Median 3Q Max

-3.0060 -0.2921 -0.2479 -0.1670 3.7968

Coefficients:

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

(Intercept) -3.122e+00 1.626e-01 -19.198 < 2e-16 \*\*\*

PBDITA.as...of.total.income -1.311e-03 6.852e-03 -0.191 0.8483

PBT.as...of.total.income -1.935e-04 1.729e-02 -0.011 0.9911

PAT.as...of.total.income -3.324e-03 1.780e-02 -0.187 0.8519

Cash.profit.as...of.total.income -1.553e-02 7.394e-03 -2.100 0.0357 \*

Quick.ratio..times. -2.241e-01 1.886e-01 -1.189 0.2346

Current.ratio..times. 1.158e-03 1.057e-01 0.011 0.9913

Debt.to.equity.ratio..times. 1.010e-01 2.458e-02 4.107 4.01e-05 \*\*\*

Cash.to.current.liabilities..times. 1.796e-01 2.801e-01 0.641 0.5214

Cash.to.average.cost.of.sales.per.day 1.114e-03 5.033e-04 2.214 0.0268 \*

EPS -9.359e-02 6.333e-02 -1.478 0.1395

Adjusted.EPS 4.860e-02 6.427e-02 0.756 0.4496

TOL.TNW 2.624e-02 1.662e-02 1.579 0.1144

PATTI -4.807e-02 5.123e-02 -0.938 0.3481

PATSFCRP 1.678e-02 2.214e-02 0.758 0.4484

PATTS -2.827e-01 2.136e-01 -1.323 0.1857

TASFCRP -3.600e-03 3.417e-03 -1.054 0.2920

BOSFCRP 7.082e-04 2.362e-03 0.300 0.7643

BOTA 1.492e-02 1.553e-02 0.960 0.3369

TITA 1.690e-02 1.077e-02 1.570 0.1164

TINFA 6.328e-04 5.704e-04 1.109 0.2673

CACLP -4.969e-06 6.952e-04 -0.007 0.9943

ROS 1.634e-02 1.614e-02 1.012 0.3115

PBTNWC -4.581e-03 2.025e-03 -2.263 0.0236 \*

SLTA -5.509e-03 1.012e-02 -0.544 0.5864

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1239.20 on 2478 degrees of freedom

Residual deviance: 815.73 on 2454 degrees of freedom

AIC: 865.73

Number of Fisher Scoring iterations: 8

From above output we found most of the variables are not significant. Remove one by one variable from the model and rerun the model. Add some new variable. This is iterative and most important task to findout the final significant variables and final model.

In some cases after discussion with client and expertise you find some variable is not significant but still that variable is important to stay in model then you should keep that variable in final model. That is pure subjective and judgmental.

suppose after lot of iteration we found Cash.profit.as...of.total.income,Debt.to.equity.ratio..times.

,EPS,Cash.profit.as...of.total.income,TOL.TNW,PATSFCRP,PBTNWC are the final variables .The final model and prediction on test data are below.

> model2<-glm(default ~ Cash.profit.as...of.total.income+Debt.to.equity.ratio..times.

+ +EPS+Cash.profit.as...of.total.income+TOL.TNW

+ +PATSFCRP+PBTNWC,

+ data=traindata,family = binomial(link="logit"))

Warning message:

glm.fit: fitted probabilities numerically 0 or 1 occurred

> summary(model2)

Call:

glm(formula = default ~ Cash.profit.as...of.total.income + Debt.to.equity.ratio..times. +

EPS + Cash.profit.as...of.total.income + TOL.TNW + PATSFCRP +

PBTNWC, family = binomial(link = "logit"), data = traindata)

Deviance Residuals:

Min 1Q Median 3Q Max

-3.0707 -0.3064 -0.2700 -0.1902 3.7653

Coefficients:

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

(Intercept) -3.071849 0.112085 -27.406 < 2e-16 \*\*\*

Cash.profit.as...of.total.income -0.023013 0.003044 -7.559 4.06e-14 \*\*\*

Debt.to.equity.ratio..times. 0.087123 0.023508 3.706 0.00021 \*\*\*

EPS -0.044872 0.006733 -6.665 2.65e-11 \*\*\*

TOL.TNW 0.034984 0.015710 2.227 0.02596 \*

PATSFCRP 0.049152 0.013909 3.534 0.00041 \*\*\*

PBTNWC -0.004712 0.002077 -2.269 0.02327 \*

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1239.20 on 2478 degrees of freedom

Residual deviance: 857.97 on 2472 degrees of freedom

AIC: 871.97

Number of Fisher Scoring iterations: 8

After final model see the intuition of the variables. Like if your Cash.profit.as...of.total.income increases then default chance decreases so the sign of coefficient should –ve and in final model the coefficient also coming –ve. So this variable is intuitive. If some variable is not institutive then you should remove that variable from your model.

You can find coefficients and some model fit statitics .Predict the model on test and validation data find confusion matrix ,accuracy and some more statitics.

> confint(model2)

Waiting for profiling to be done...

2.5 % 97.5 %

(Intercept) -3.298333028 -2.8584676271

Cash.profit.as...of.total.income -0.029105207 -0.0171017165

Debt.to.equity.ratio..times. 0.041032566 0.1339444732

EPS -0.058459191 -0.0319386201

TOL.TNW 0.003968563 0.0657800111

PATSFCRP 0.025280151 0.0846990457

PBTNWC -0.010175533 -0.0006484594

There were 50 or more warnings (use warnings() to see the first 50)

> coef(model2)

(Intercept) Cash.profit.as...of.total.income Debt.to.equity.ratio..times.

-3.071848832 -0.023013254 0.087122667

EPS TOL.TNW PATSFCRP

-0.044872402 0.034983981 0.049152177

PBTNWC

-0.004711975

> #confint.default(mymodel1)

> wald.test(b=coef(model2),Sigma = vcov(model2),Terms = 2:7)

Wald test:

----------

Chi-squared test:

X2 = 235.5, df = 6, P(> X2) = 0.0

> anova(model2,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 2478 1239.20

Cash.profit.as...of.total.income 1 119.449 2477 1119.75 < 2.2e-16 \*\*\*

Debt.to.equity.ratio..times. 1 169.726 2476 950.03 < 2.2e-16 \*\*\*

EPS 1 66.428 2475 883.60 3.630e-16 \*\*\*

TOL.TNW 1 4.724 2474 878.88 0.02975 \*

PATSFCRP 1 16.016 2473 862.86 6.282e-05 \*\*\*

PBTNWC 1 4.892 2472 857.97 0.02698 \*

---

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

Warning messages:

1: glm.fit: fitted probabilities numerically 0 or 1 occurred

2: glm.fit: fitted probabilities numerically 0 or 1 occurred

3: glm.fit: fitted probabilities numerically 0 or 1 occurred

> exp(coef(model2)) ## odds ratios only

(Intercept) Cash.profit.as...of.total.income Debt.to.equity.ratio..times.

0.04633541 0.97724953 1.09103050

EPS TOL.TNW PATSFCRP

0.95611947 1.03560312 1.05038018

PBTNWC

0.99529911

> exp(cbind(OR = coef(model2), confint(model2))) ## odds ratios and 95% CI

Waiting for profiling to be done...

OR 2.5 % 97.5 %

(Intercept) 0.04633541 0.0369447 0.05735658

Cash.profit.as...of.total.income 0.97724953 0.9713143 0.98304369

Debt.to.equity.ratio..times. 1.09103050 1.0418860 1.14332933

EPS 0.95611947 0.9432167 0.96856603

TOL.TNW 1.03560312 1.0039764 1.06799174

PATSFCRP 1.05038018 1.0256024 1.08838946

PBTNWC 0.99529911 0.9898761 0.99935175

There were 50 or more warnings (use warnings() to see the first 50)

> fitted.testresult2<- predict(model2,newdata=subset(testdata,select=c(15,48,15,54,28,64,38)),type="response")

> fitted.testresult2<-ifelse(fitted.testresult2>0.6,1,0)

> misclassificationerror2<-mean(fitted.testresult2!=testdata$default)

> print(paste("Accuracy : ",1-misclassificationerror2))

[1] "Accuracy : 0.939736346516008"

> library(ROCR)

Loading required package: gplots

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

lowess

Warning message:

package ‘gplots’ was built under R version 3.2.4

> p<- predict(model2,newdata=subset(testdata,select=c(15,48,15,54,28,64,38)),type="response")

> pr<-prediction(p,testdata$default)

> prf<-performance(pr,measure = "tpr",x.measure = "fpr")

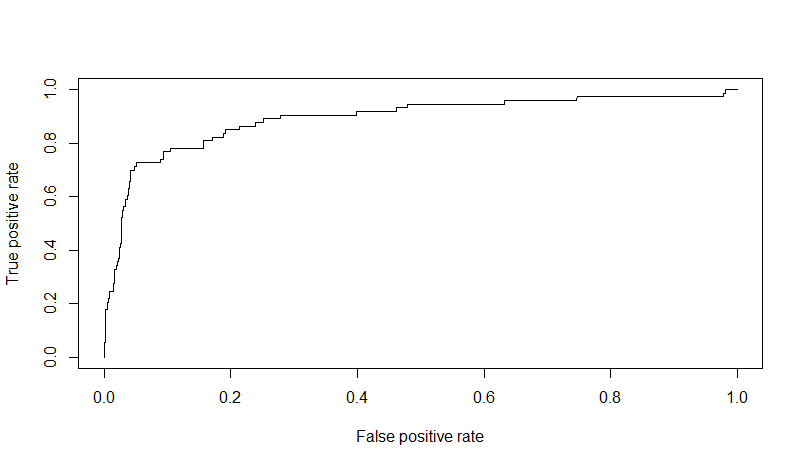
> plot(prf)

> auc<-performance(pr,measure="auc")

> auc<-auc@y.values[[1]]

> auc

[1] 0.8902309



> #confusionMatrix {caret} try

> library(caret)

Loading required package: lattice

Warning message:

package ‘caret’ was built under R version 3.2.5

> confusionMatrix(data=fitted.testresult2, reference=testdata$default)

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 981 56

1 8 17

Accuracy : 0.9397

95% CI : (0.9237, 0.9533)

No Information Rate : 0.9313

P-Value [Acc > NIR] : 0.1509

Kappa : 0.3232

Mcnemar's Test P-Value : 4.228e-09

Sensitivity : 0.9919

Specificity : 0.2329

Pos Pred Value : 0.9460

Neg Pred Value : 0.6800

Prevalence : 0.9313

Detection Rate : 0.9237

Detection Prevalence : 0.9765

Balanced Accuracy : 0.6124

'Positive' Class : 0