Code:

Installing packages and library

```
library('readxl')
library('caTools')
library('lmtest')
```

Setting working directory

```
setwd("D:/great learning/8. FRA/Project")
mydata = read_xlsx("raw-data.xlsx")
mydata.test = read_xlsx("validation_data.xlsx")
View(mydata)
mydata = mydata[-c(22)]
```

Treating NA or missing values

```
mydata = na.omit(mydata)
attach(mydata)
attach(mydata.test)
```

Creating Default variable

```
Default = ifelse(`Networth Next Year`>0,0,1)
```

summary(as.factor(Default))

Default.rate = 243/(3298+243)

names(mydata)

Exploratory data analysis

summary(mydata)

str(mydata)

boxplot(`PAT as % of net worth`, horizontal = TRUE, col = "Red", main = "boxplot for PAT as % of Net worth")

boxplot(`Quick ratio (times)`, horizontal = TRUE, col = "Yellow", main = "boxplot for PAT as % of Quick ratio")

boxplot(`Current ratio (times)`, horizontal = TRUE, col = "Blue", main = "boxplot for PAT as % of Current ratio")

boxplot(`PBT as % of total income`, horizontal = TRUE, col = "Brown", main = "boxplot for PBT as % of total income")

boxplot`Cash profit as % of total income`, horizontal = TRUE, col = "Orange", main = "boxplot for Cash profit as % of total income")

Checking Multi Colinearity

library(corrplot)

mydata_1 = mydata[c(3,4,5,7,17,25,27,32,33,34)] ## Taking size variables

mydata_1

mydata_corr = cor(mydata_1)

corrplot(mydata_corr, method = "number")

```
mydata_2 = mydata[c(8,9,10,11,12,13,14,15,16,18,19,21,26)]          ## Taking profit variables

mydata_2

mydata_corr2 = cor(mydata_2)

corrplot(mydata_corr2, method = "number")


mydata_3 = mydata[c(22,28,29,30,37)]          ## Taking leverage variables

mydata_3

mydata_corr3 = cor(mydata_3)

corrplot(mydata_corr3, method = "number")


mydata_4 = mydata[c(23,24,31,36,38,39,40)]          ## Taking liquidity variables

mydata_4

mydata_corr4 = cor(mydata_4)

corrplot(mydata_corr4, method = "number")
```

Building logistic regression model

```
glm(as.factor(Default)~`PBT as % of total income`, family = binomial)

summary(glm(as.factor(Default)~`Cash to average cost of sales per day`, family = binomial))

Default.model1 = glm(as.factor(Default)~`PBT as % of total income`, family = binomial)

summary(Default.model1$fitted.values)

plot(as.factor(Default.model1$y), Default.model1$fitted.values)
```

```
summary(glm(as.factor(Default)~`Cash profit as % of total income`+`Total income`+`current ratio`+`Debt to equity ratio`, family = binomial))

Default.model2 = glm(as.factor(Default)~`Cash profit as % of total income`+`Total income`+`current ratio`+`Debt to equity ratio`, family = binomial)

plot(as.factor(Default.model2$y), Default.model2$fitted.values)

Default.prediction2 = ifelse(Default.model2$fitted.values>0.064,1,0)

table(Default.model2$y,Default.prediction2)


summary(glm(as.factor(Default)~`Cash profit as % of total income`+`Total income`+`TOL/TNW`+`Cash to average cost of sales per day`, family = binomial))

Default.model3 = glm(as.factor(Default)~`Cash profit as % of total income`+`Total income`+`TOL/TNW`+`Cash to average cost of sales per day`, family = binomial)

plot(as.factor(Default.model3$y), Default.model3$fitted.values)

Default.prediction3 = ifelse(Default.model3$fitted.values>0.066,1,0)

table(Default.model3$y,Default.prediction3)


mydata.validate = predict(Default.model3, data = mydata.test, type = "response")

Default.pred = ifelse(mydata.validate>0.07,1,0)

table(Default.model3$y, Default.pred)
```