Deutsche Bank Credit Score Case

Takahiro Yamada

11/17/2021

Deutche Bank Credit Score Analysis

The credit review, issue or not issue the credit card, is a theme in this case study of Deutsche Bank. Based on the customer data base which contains 1000 observations and 31 variables,, figuring out the scoring model using and evaluating logistic regression model.

Data cleaing

The data sheet does not contain missing value however, contained “X” in the purpose variable instead. Omitted the rows which contain “X” as variables.

```
#################################################################
##### Libraries
#################################################################
#install.packages("ggplot2")
#install.packages("plotly")
#install.packages("caret")
#install.packages("ROCR")
#install.packages("rpart")
#install.packages("rpart.plot")
library(ggplot2)
library(plotly)
library(caret)
library(ROCR)
library(rpart)
library(rpart.plot)



#################################################################
##### Data Massage
#################################################################
library(readxl)
my_germ <- read_excel("/Users/takahiroyamada/Desktop/MBAN/20211012 Data scien
ce R/class deuchebank/german credit card.xls")

any(is.na(my_germ))

## [1] FALSE

table(my_germ$purpose)
```

```
##
##   0   1   2   3   4   5   6   8   9   X
## 234 103 181 280  12  22  50   9  97  12

#omit rows with X
which(my_germ$purpose == "X")

##  [1]  73  84 106 288 311 375 432 443 595 666 819 916

my_germ[which(my_germ$purpose == "X"),]

## # A tibble: 12 × 21
##    checking duration history purpose amount savings employed installp mari
tal
##       <dbl>    <dbl>   <dbl> <chr>    <dbl>   <dbl>    <dbl>    <dbl>   <d
bl>
##  1        1        8       4 X         1164       1        5        3
  3
##  2        1       24       2 X         1755       1        5        4
  2
##  3        2       24       4 X        11938       1        3        2
  3
##  4        2       48       3 X         7582       2        1        2
  3
##  5        2       48       2 X         5381       5        1        3
  3
##  6        2       60       1 X        14782       2        5        3
  2
##  7        2       24       2 X        11328       1        3        2
  3
##  8        2       20       3 X         2629       1        3        2
  3
##  9        1       24       1 X         1358       5        5        4
  3
## 10        4       24       4 X         6314       1        1        4
  3
## 11        1       36       2 X        15857       1        1        2
  1
## 12        2       48       0 X        18424       1        3        1
  2
## # … with 12 more variables: coapp <dbl>, resident <dbl>, property <dbl>,
## #   age <dbl>, other <dbl>, housing <dbl>, existcr <dbl>, job <dbl>,
## #   depends <dbl>, telephon <dbl>, foreign <dbl>, good_bad <chr>

my_germ <- as.data.frame(my_germ)

my_germ[my_germ=="X"] <- NA
my_germ <- my_germ[-which(is.na(my_germ$purpose)),]
colSums(is.na(my_germ))
```

```
## checking duration  history  purpose   amount  savings employed installp
##        0        0        0        0        0        0        0        0
##  marital    coapp resident property      age    other  housing  existcr
##        0        0        0        0        0        0        0        0
##      job  depends telephon  foreign good_bad
##        0        0        0        0        0
```

```
#replace good with 1, bad with 0
my_germ$binary <- gsub("good","1",my_germ$good_bad)
my_germ$binary <- gsub("bad","0",my_germ$binary)
my_germ$binary <- as.numeric(my_germ$binary)
```

Visual Analysis

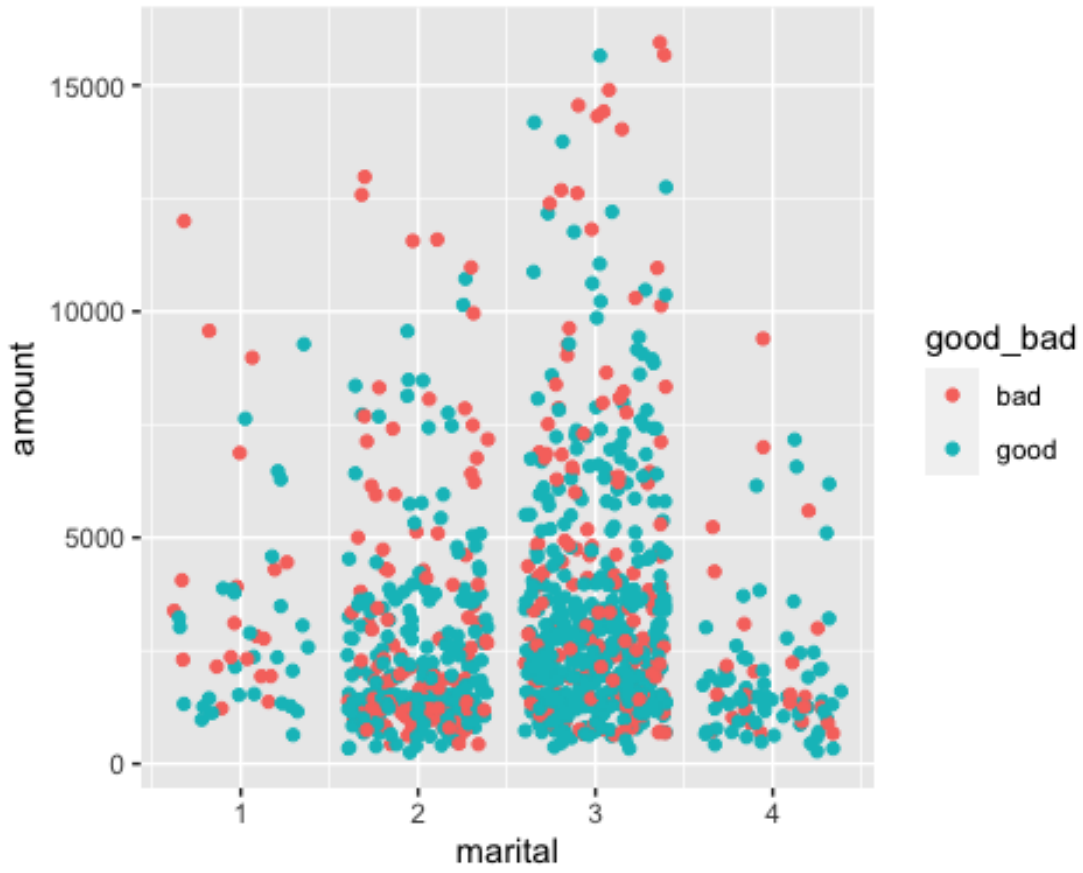
Checked the relationship of "amount" with three variables, "marital", "depends" and "duration" using different color for good/bad customer calcification which is made by Deutsche Bank.

Amount x Marital: most of customers are concentrated in status 2 and 3 / lower 5000 amount area, and good/bad looks equally distributed.

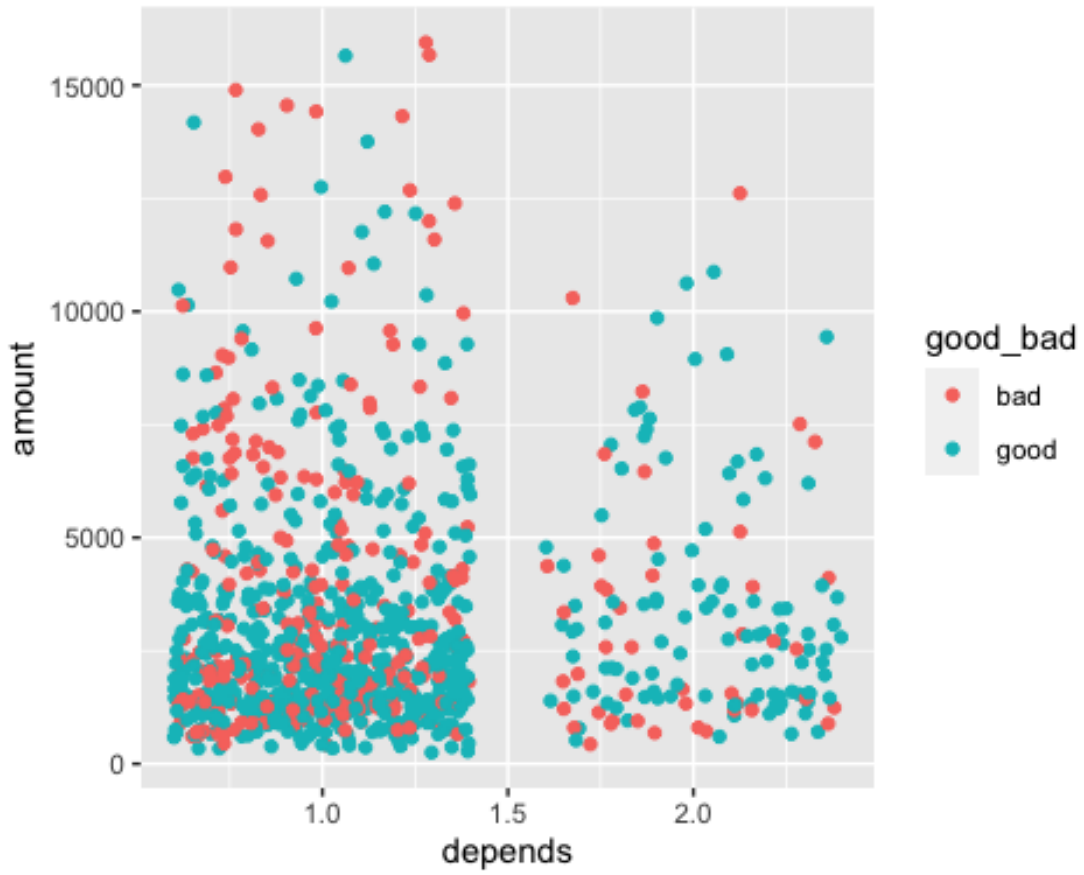
Amount x Depends: most of customers are counted as 1.0 /lower 5000 amount area. The good/bad is equally distributed.

Amount x Duration: good customer seems concentrated around duration 1-2 / lower 5000 amount area, and bad customer looks distributes to the longer duration side and relatively higher amount. The smooth liner shows almost same propotion.

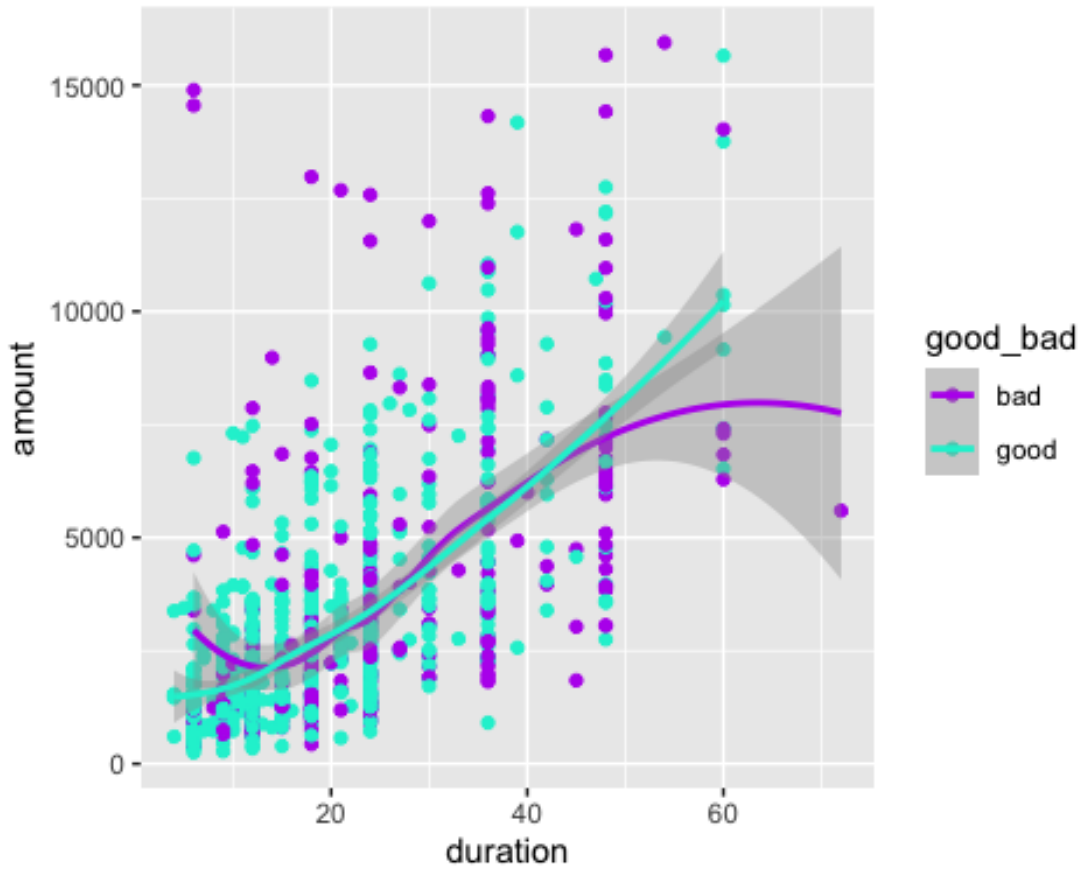
```
#################################################################
#####Visualization
#################################################################
ggplot(data=my_germ, aes(x=marital, y=amount, color=good_bad)) +
  geom_jitter()
```




```
ggplot(data=my_germ, aes(x=depends, y=amount, color=good_bad)) +
  geom_jitter()
```




```
ggplot(data=my_germ, aes(x=duration, y=amount, color=good_bad)) +
  geom_point()+geom_smooth()+scale_color_manual(values=c("#B81CEE","#0AF0D4")
)
```



Standardization & Normalization

To see the positioning of each customer's age and amount, created t-score UDF and created new variable age_standard and amount_standard. Furthermore, re-scale the all numeric variable range 0 to 1 for coming unit less regression calculation.

```
#####################################################################
#####Standardization, Normalization, Sampling and Classification
#####################################################################
#re scale data to have a mean of 0 and a standard deviation of 1
#UDF Z-Score
standard <- function(var1) {
  my_standard <- (var1-mean(var1))/sd(var1)
  return(my_standard)
} #closing the standard variable

my_germ$age_standard <- standard(var1=my_germ$age)
summary(my_germ$age_standard)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.4504 -0.7469 -0.2192  0.0000  0.5722  3.4742
```

```
my_germ$amount_standard <- standard(var1=my_germ$amount)
summary(my_germ$amount_standard)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0924 -0.6815 -0.3352  0.0000  0.2731  4.6975

#UDF T-score
standard <- function(var1) {
  my_standard <- (var1-mean(var1))/sd(var1)*10+50
  return(my_standard)
} #closing the standard variable

my_germ$age_standard <- standard(var1=my_germ$age)
summary(my_germ$age_standard)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   35.50   42.53   47.81   50.00   55.72   84.74

my_germ$amount_standard <- standard(var1=my_germ$amount)
summary(my_germ$amount_standard)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   39.08   43.19   46.65   50.00   52.73   96.97

###Normalization - Re-scale the values into a range of 0 and 1
normal <- function(var1){
  my_normal <- (var1-min(var1))/(max(var1)-min(var1))
  return(my_normal)
} #closing the normal UDF
my_germ$checking_norm <- normal(var1=my_germ$checking)
my_germ$duration_norm <- normal(var1=my_germ$duration)
my_germ$amount_norm <- normal(var1=my_germ$amount)
my_germ$employed_norm <- normal(var1=my_germ$employed)
my_germ$installp_norm <- normal(var1=my_germ$installp)
my_germ$age_norm <- normal(var1=my_germ$age)
my_germ$existcr_norm <- normal(var1=my_germ$existcr)
my_germ$telephon_norm <- normal(var1=my_germ$telephon)
```

Classification with Logistic Regression

Created the regression mode by setting the “binary” variable which is converted form “good_bad” as objective variable and other numerical variables as an explained variables. Through ruining the regression model test, reduced the insignificant variables in terms of p-value. And also checked the regression models with unit and unitless variables. As a consequence, figure out that “checking”, “duration”, “age” and “installp”, have significant relationship with good_bad variable. Checking has positive stronger impact for good_bad variable. Duration has negative stronger impact for good_bad variable. Age has positive normal impact for good_bad variable. Install negative weak impact for good_bad variable.

```
###############################################################
#####Classification with Logistic Regression
```

```
#################################################################
#creating training and testing data sets by random sampling
train_index <- sample(1:nrow(my_germ),size=0.8*nrow(my_germ))
germ_train <- my_germ[train_index,]
germ_test <- my_germ[-train_index,]

#linear prediction
my_linear <- lm(amount~age, data=germ_train)
summary(my_linear)

##
## Call:
## lm(formula = amount ~ age, data = germ_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3049.9 -1872.4  -912.2   901.8 12506.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2986.325    313.858   9.515   <2e-16 ***
## age            7.648      8.423   0.908    0.364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2676 on 788 degrees of freedom
## Multiple R-squared:  0.001045,   Adjusted R-squared:  -0.0002226
## F-statistic: 0.8244 on 1 and 788 DF,  p-value: 0.3642

#Logistic Regression
my_logit <- glm(binary~checking + duration + age + telephon + amount + saving
s
                + installp + coapp, data=germ_train, family = "binomial")
summary(my_logit)

##
## Call:
## glm(formula = binary ~ checking + duration + age + telephon +
##     amount + savings + installp + coapp, family = "binomial",
##     data = germ_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5620  -0.9643   0.5065   0.7979   1.7998
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.110e+00  5.400e-01  -2.055   0.0399 *
## checking     6.090e-01  7.418e-02   8.210  < 2e-16 ***
## duration    -1.930e-02  9.054e-03  -2.131   0.0331 *
```

```
## age          1.544e-02  7.875e-03   1.961   0.0499 *
## telephon     3.061e-01  1.887e-01   1.622   0.1048
## amount      -1.040e-04  4.388e-05  -2.370   0.0178 *
## savings      2.494e-01  6.299e-02   3.960  7.5e-05 ***
## installp    -1.963e-01  8.652e-02  -2.269   0.0232 *
## coapp        3.538e-01  1.969e-01   1.797   0.0724 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 971.85  on 789  degrees of freedom
## Residual deviance: 818.68  on 781  degrees of freedom
## AIC: 836.68
##
## Number of Fisher Scoring iterations: 4

#remove telephon and amount to improve the analysis
my_logit_better <- glm(binary~checking + duration + age + savings
                       + installp + coapp, data=germ_train, family = "binomia
l")
summary(my_logit_better)

##
## Call:
## glm(formula = binary ~ checking + duration + age + savings +
##     installp + coapp, family = "binomial", data = germ_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6119  -0.9849   0.5133   0.7884   1.7925
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.011056   0.494188  -2.046   0.0408 *
## checking     0.614160   0.073855   8.316  < 2e-16 ***
## duration    -0.031833   0.007008  -4.542 5.57e-06 ***
## age          0.015200   0.007717   1.970   0.0489 *
## savings      0.245904   0.062378   3.942 8.08e-05 ***
## installp    -0.113387   0.077139  -1.470   0.1416
## coapp        0.354576   0.196208   1.807   0.0707 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 971.85  on 789  degrees of freedom
## Residual deviance: 825.40  on 783  degrees of freedom
## AIC: 839.4
```

```
##
## Number of Fisher Scoring iterations: 4

#designing logistic regression after normalization of the data
my_logit_norm <- glm(binary~checking_norm+duration_norm+age_norm+installp_nor
m, data=germ_train, family = "binomial")
summary(my_logit_norm)

##
## Call:
## glm(formula = binary ~ checking_norm + duration_norm + age_norm +
##     installp_norm, family = "binomial", data = germ_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3217  -1.0415   0.5137   0.8476   1.6762
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.4280     0.2534   1.689   0.0911 .
## checking_norm   1.9684     0.2166   9.087  < 2e-16 ***
## duration_norm  -2.0785     0.4656  -4.465 8.02e-06 ***
## age_norm        0.8705     0.4260   2.044   0.0410 *
## installp_norm  -0.3151     0.2284  -1.380   0.1677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 971.85  on 789  degrees of freedom
## Residual deviance: 844.61  on 785  degrees of freedom
## AIC: 854.61
##
## Number of Fisher Scoring iterations: 4
```

Confusion Matrix and AUC/ROC Analysis

Make sure how the unit less regression models is accurate using Confusion matrix. Confusion matrix for model with training data shows about 74% accuracy. The model with testing data shows about 78% accuracy. The unitless regression model would be reliable since it shows similar accuracy between two different data sets. The following AUC ROC model showing good sing which the curve is not intercepting the diagonal line form TPR:FPR = 0:0 point to TPR:FPR = 1:1 point. This means the true positive number of observations is exceeding false positive number.

```
#################################################################
#####Confusion Matrix, Decision Tree, Type GINI and comparing different model
s
#################################################################
#for training data
```

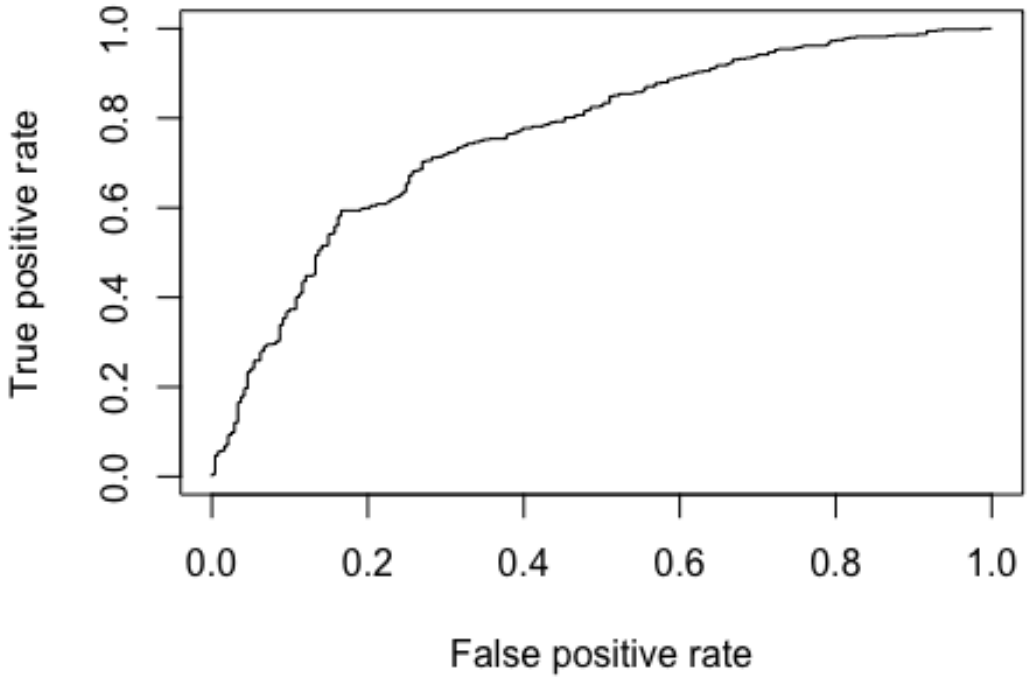
```r
my_prediction_training <- predict(my_logit,germ_train, type="response")
cnf_mtrx_train <- confusionMatrix(data=as.factor(as.numeric(my_prediction_tra
ining > 0.5)),
                                  reference=as.factor(as.numeric(germ_trai
n$binary))) %>%
  print()
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 102  66
##          1 139 483
##
##                Accuracy : 0.7405
##                  95% CI : (0.7084, 0.7708)
##     No Information Rate : 0.6949
##     P-Value [Acc > NIR] : 0.002724
##
##                   Kappa : 0.3312
##
##  Mcnemar's Test P-Value : 4.938e-07
##
##             Sensitivity : 0.4232
##             Specificity : 0.8798
##          Pos Pred Value : 0.6071
##          Neg Pred Value : 0.7765
##              Prevalence : 0.3051
##          Detection Rate : 0.1291
##    Detection Prevalence : 0.2127
##       Balanced Accuracy : 0.6515
##
##        'Positive' Class : 0
##
```

```r
#for testing data
my_prediction_testing <- predict(my_logit,germ_test, type="response")
cnf_mtrx_test <-confusionMatrix(data=as.factor(as.numeric(my_prediction_testi
ng > 0.5)),
                                reference=as.factor(as.numeric(germ_test$b
inary))) %>%
  print()
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  22  11
##          1  32 133
##
```

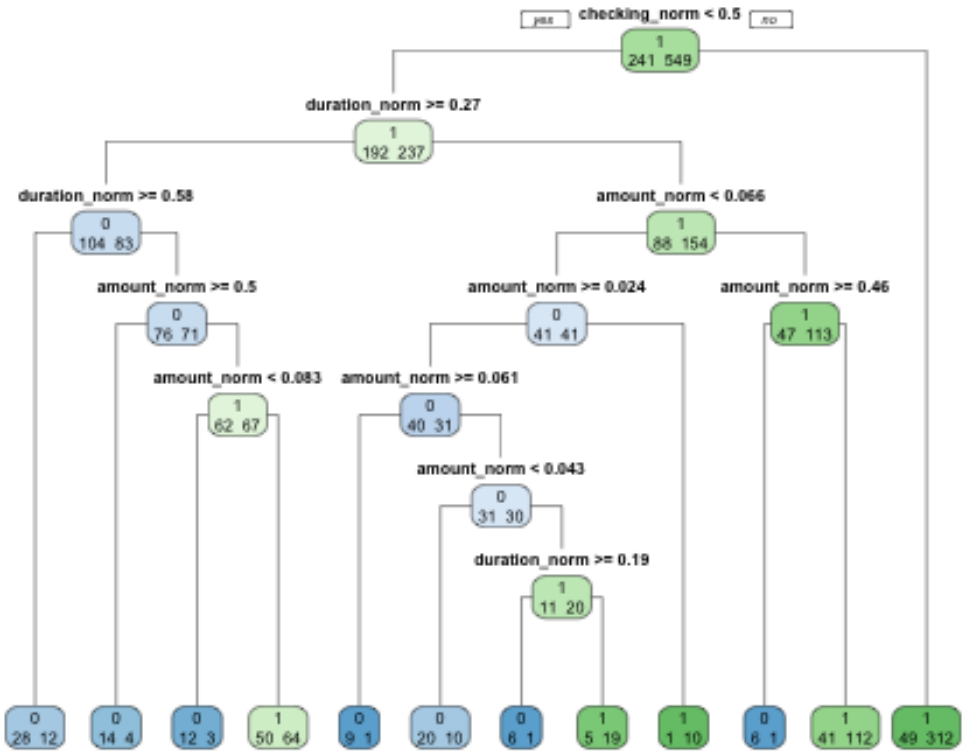
```
##                Accuracy : 0.7828
##                  95% CI : (0.7188, 0.8381)
##     No Information Rate : 0.7273
##     P-Value [Acc > NIR] : 0.044528
##
##                   Kappa : 0.3768
##
##  Mcnemar's Test P-Value : 0.002289
##
##             Sensitivity : 0.4074
##             Specificity : 0.9236
##          Pos Pred Value : 0.6667
##          Neg Pred Value : 0.8061
##              Prevalence : 0.2727
##          Detection Rate : 0.1111
##    Detection Prevalence : 0.1667
##       Balanced Accuracy : 0.6655
##
##        'Positive' Class : 0
##

#AUC ROC framework
my_prediction <- my_prediction_training
pred_val_logit <- prediction(my_prediction, germ_train$binary)
perf_logit <- performance(pred_val_logit,"tpr","fpr")
plot(perf_logit)
```



Decision Tree as a competitive model

Also create the decision tree for comparing with the unitless regression if these different two models will create the similar consequence. Decision tree clearly shows that the "checking", "duration", "amount" and "age" are the significant factor which decide the customer status good or bad. These key variables are almost identical with the unitless regression model. In the tree visual, 1 represents "good" and 0 represents "bad".

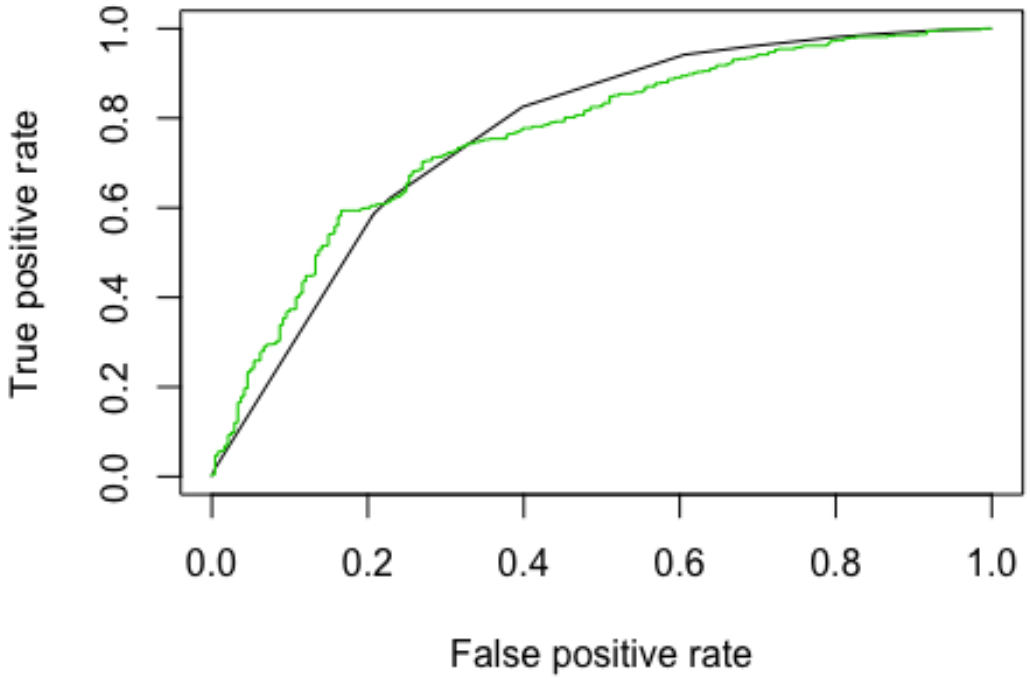
```
#Challenger Decision Tree for my_Germ
my_tree <- rpart(binary~checking_norm+duration_norm+age_norm+amount_norm+inst
allp_norm, data=germ_train, method = "class",cp=0.017)
rpart.plot(my_tree, extra=1, type=1)
```



Compare regression model and decision tree model

To check the similarity of regression model and decision tree, created the AOC/ROC curves to check how duplicate both curves are. Decision tree AOC/ROC curve is described by black line and regression model is by green line. They have quite similar behavior therefore, it is possible to say that both models are equally accurate. Since the green curve, regression model is more left sided compared with decision tree model, decision tree is more accrate than decision tree.

```
#####################################################################
#####Comparing model performance
#####################################################################
my_tree_predict_testt<-predict(my_tree, germ_test,type="prob")
my_tree_predict_train<-predict(my_tree, germ_train,type="prob")
my_tree_prediction<-prediction(my_tree_predict_train[,2], germ_train$binary)

my_tree_performance<-performance(my_tree_prediction,"tpr","fpr")
plot(my_tree_performance,col="black")
plot(perf_logit,col="green3",add=TRUE)#low false positive rate more important
 than higher rates
```



Scoring based on the regression and decision tree model analysis

Through the regression and decision tree analysis, it is figured out that the “checking”, “duration”, “amount” and “age” have significant impact for the customer classification, good or bad. And also it is figured out that the regression model is more accurate than decision tree. Therefore, adopted regression model and weighted “checking” x 50, “duration” = 20, “installp” x 20 and “age” x 40. The installp and duration weighting is lower because duration show significant p-value and larger negative beta and installp show larger p-value and weaker negative beta. After creating scoring model, put the label “outstanding” and “not outstanding” for the individual customer which score is lower than the score mean.

```
#################################################################
#####Credit score formula
#################################################################
#Scoring based on normalized logistic regression
my_germ$score <- c()
for (i in 1:nrow(my_germ)) {
  my_germ$score[i] <- 20*my_germ$duration_norm[i]+
    50*my_germ$checking_norm[i]+
    40*my_germ$age_norm[i]+
    20*my_germ$installp_norm[i]
```

```r
} #closing the i loop
summary(my_germ$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.445  35.630  54.811  56.323  77.938 108.501
```

```r
#check for both score and good_bad, if score is below score mean, label custo
mer "outstanding"
my_germ$label <- c()

for (i in 1:nrow(my_germ)) {
  if (my_germ$score[i]<mean(my_germ$score) & my_germ$binary[i] == 1) {
    my_germ$label[i] <- "outstanding"
  } else {
    my_germ$label[i] <- "not outstanding"
  } #closing if statement
} #closing the i loop
table(my_germ$label)
```

```
##
## not outstanding     outstanding
##             685             303
```