title: “kdda final”

Group: 12 Names: Javid Bell, Kimani Johnson

id: 620107935, 620013658

understanding the data The dataset used in this project is the German Credit Dataset which can be found at https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29. This dataset contains information on credit applications, where each row represents a credit application and each column represents an attribute of that application. The dataset has 20 attributes in total, which can be divided into two categories: personal attributes and financial attributes. The personal attributes are: Age (numeric): the age of the person applying for credit. Sex (categorical: male, female): the sex of the person applying for credit. Job (numeric): the type of job of the person applying for credit, ranging from 0 (unemployed) to 3 (highly skilled). Housing (categorical: rent, own, free): the type of housing of the person applying for credit. Saving accounts (categorical: little, moderate, quite rich, rich): the amount of savings the person applying for credit has. Checking account (categorical: little, moderate, rich): the amount of money the person applying for credit has in their checking account. Purpose (categorical: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others): the purpose of the credit application. The financial attributes are: Credit amount (numeric): the amount of credit being applied for. Duration (numeric): the duration of the credit in months. Credit history (categorical: no credits taken, all credits paid back duly, existing credits paid back duly till now, delay in paying off in the past, critical account/other credits existing (not at this bank)): the credit history of the person applying for credit. Purpose (categorical: car, furniture/equipment, radio/TV, domestic appliances, repairs, education, business, vacation/others): the purpose of the credit application. Creditability (categorical: good, bad): whether the credit application was approved or not.

#Business Understanding: For this project, we will be using the Statlog (German Credit Data) dataset from the UCI Machine Learning Repository. The dataset contains information on credit applications and whether the applications were approved or denied. The practical use case for this dataset is to build a model that can accurately predict whether a credit application will be approved or denied based on the information provided in the application. A loan applicant is a good or bad credit risk based on their credit history, employment status, income, and other relevant features. We identified a classification problem in which this data could be used to increase a financial institutions profits as well as reduce the risk of loss by identifying people with high and low credit risks. This would allow them to more accurately choose who to lend to and who to avoid lending money to based on the features in the data. This would protect them from making bad loans as well make it easy to identify individuals who are good borrowers. By accurately predicting whether an applicant is a good or bad credit risk, banks can reduce the likelihood of default.

loading and understanding data

```r
options(scipen=999999)

##### getwd()
##### setwd("C:/Users/Lenovo/Documents/COMP6115/Project/Dataset")

###Load dataset into R
data <- read.table("german.data", header=FALSE, sep=" ")

View(data)

######## write.csv(data, "German.csv", row.names=FALSE)

###Add Columns to dataset using names made up from dataset notes
colnames(data) <- c("checking_account_status", "duration", "credit_history",
"purpose", "credit_amount", "savings_account_status", "employment_status",
"installment_rate", "personal_status", "other_debtors", "residence_since",
"property", "age", "other_installment_plans", "housing", "existing_credits",
"job", "num_dependents", "telephone", "foreign_worker", "credit_risk")

str(data)

## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_account_status: chr  "A11" "A12" "A14" "A11" ...
##  $ duration               : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history         : chr  "A34" "A32" "A34" "A32" ...
##  $ purpose                : chr  "A43" "A43" "A46" "A42" ...
##  $ credit_amount          : int  1169 5951 2096 7882 4870 9055 2835 6948
3059 5234 ...
##  $ savings_account_status : chr  "A65" "A61" "A61" "A61" ...
##  $ employment_status      : chr  "A75" "A73" "A74" "A74" ...
##  $ installment_rate       : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status        : chr  "A93" "A92" "A93" "A93" ...
##  $ other_debtors          : chr  "A101" "A101" "A101" "A103" ...
##  $ residence_since        : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property               : chr  "A121" "A121" "A121" "A122" ...
##  $ age                    : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_installment_plans: chr  "A143" "A143" "A143" "A143" ...
##  $ housing                : chr  "A152" "A152" "A152" "A153" ...
##  $ existing_credits       : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                    : chr  "A173" "A173" "A172" "A173" ...
##  $ num_dependents         : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone              : chr  "A192" "A191" "A191" "A191" ...
##  $ foreign_worker         : chr  "A201" "A201" "A201" "A201" ...
##  $ credit_risk            : int  1 2 1 1 2 1 1 1 1 2 ...

sum(is.na(data))

## [1] 0

summary(data)
```

```
##  checking_account_status    duration    credit_history      purpose
##  Length:1000             Min.   : 4.0   Length:1000        Length:1000
##  Class :character        1st Qu.:12.0   Class :character   Class
:character
##  Mode  :character        Median :18.0   Mode  :character   Mode
:character
##                          Mean   :20.9
##                          3rd Qu.:24.0
##                          Max.   :72.0
##  credit_amount   savings_account_status employment_status
installment_rate
##  Min.   :  250   Length:1000            Length:1000        Min.   :1.000
##  1st Qu.: 1366   Class :character       Class :character   1st Qu.:2.000
##  Median : 2320   Mode  :character       Mode  :character   Median :3.000
##  Mean   : 3271                                             Mean   :2.973
##  3rd Qu.: 3972                                             3rd Qu.:4.000
##  Max.   :18424                                             Max.   :4.000
##  personal_status    other_debtors      residence_since   property
##  Length:1000        Length:1000        Min.   :1.000   Length:1000
##  Class :character   Class :character   1st Qu.:2.000   Class :character
##  Mode  :character   Mode  :character   Median :3.000   Mode  :character
##                                        Mean   :2.845
##                                        3rd Qu.:4.000
##                                        Max.   :4.000
##       age        other_installment_plans   housing
existing_credits
##  Min.   :19.00   Length:1000             Length:1000        Min.   :1.000
##  1st Qu.:27.00   Class :character        Class :character   1st Qu.:1.000
##  Median :33.00   Mode  :character        Mode  :character   Median :1.000
##  Mean   :35.55                                              Mean   :1.407
##  3rd Qu.:42.00                                              3rd Qu.:2.000
##  Max.   :75.00                                              Max.   :4.000
##      job            num_dependents   telephone         foreign_worker
##  Length:1000        Min.   :1.000   Length:1000        Length:1000
##  Class :character   1st Qu.:1.000   Class :character   Class :character
##  Mode  :character   Median :1.000   Mode  :character   Mode  :character
##                     Mean   :1.155
##                     3rd Qu.:1.000
##                     Max.   :2.000
##   credit_risk
##  Min.   :1.0
##  1st Qu.:1.0
##  Median :1.0
##  Mean   :1.3
##  3rd Qu.:2.0
##  Max.   :2.0
```

```
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_account_status: chr  "A11" "A12" "A14" "A11" ...
##  $ duration               : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history         : chr  "A34" "A32" "A34" "A32" ...
##  $ purpose                : chr  "A43" "A43" "A46" "A42" ...
##  $ credit_amount          : int  1169 5951 2096 7882 4870 9055 2835 6948
3059 5234 ...
##  $ savings_account_status : chr  "A65" "A61" "A61" "A61" ...
##  $ employment_status      : chr  "A75" "A73" "A74" "A74" ...
##  $ installment_rate       : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status        : chr  "A93" "A92" "A93" "A93" ...
##  $ other_debtors          : chr  "A101" "A101" "A101" "A103" ...
##  $ residence_since        : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property               : chr  "A121" "A121" "A121" "A122" ...
##  $ age                    : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_installment_plans: chr  "A143" "A143" "A143" "A143" ...
##  $ housing                : chr  "A152" "A152" "A152" "A153" ...
##  $ existing_credits       : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                    : chr  "A173" "A173" "A172" "A173" ...
##  $ num_dependents         : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone              : chr  "A192" "A191" "A191" "A191" ...
##  $ foreign_worker         : chr  "A201" "A201" "A201" "A201" ...
##  $ credit_risk            : int  1 2 1 1 2 1 1 1 1 2 ...
```

```
data$credit_risk  <- as.factor(data$credit_risk )
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_account_status: chr  "A11" "A12" "A14" "A11" ...
##  $ duration               : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history         : chr  "A34" "A32" "A34" "A32" ...
##  $ purpose                : chr  "A43" "A43" "A46" "A42" ...
##  $ credit_amount          : int  1169 5951 2096 7882 4870 9055 2835 6948
3059 5234 ...
##  $ savings_account_status : chr  "A65" "A61" "A61" "A61" ...
##  $ employment_status      : chr  "A75" "A73" "A74" "A74" ...
##  $ installment_rate       : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status        : chr  "A93" "A92" "A93" "A93" ...
##  $ other_debtors          : chr  "A101" "A101" "A101" "A103" ...
##  $ residence_since        : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property               : chr  "A121" "A121" "A121" "A122" ...
##  $ age                    : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_installment_plans: chr  "A143" "A143" "A143" "A143" ...
##  $ housing                : chr  "A152" "A152" "A152" "A153" ...
##  $ existing_credits       : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                    : chr  "A173" "A173" "A172" "A173" ...
##  $ num_dependents         : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone              : chr  "A192" "A191" "A191" "A191" ...
##  $ foreign_worker         : chr  "A201" "A201" "A201" "A201" ...
```

```
## $ credit_risk            : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1
2 ...
```

```
data[sapply(data, is.character)] <- lapply(data[sapply(data, is.character)],
as.factor)
str(data)
```

```
## 'data.frame':    1000 obs. of  21 variables:
## $ checking_account_status: Factor w/ 4 levels "A11","A12","A13",..: 1 2 4
1 1 4 4 2 4 2 ...
## $ duration               : int  6 48 12 42 24 36 24 36 12 30 ...
## $ credit_history         : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5
3 4 3 3 3 3 5 ...
## $ purpose                : Factor w/ 10 levels "A40","A41","A410",..: 5 5
8 4 1 8 4 2 5 1 ...
## $ credit_amount          : int  1169 5951 2096 7882 4870 9055 2835 6948
3059 5234 ...
## $ savings_account_status : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1
1 1 5 3 1 4 1 ...
## $ employment_status      : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4
4 3 3 5 3 4 1 ...
## $ installment_rate       : int  4 2 2 2 3 2 3 2 2 4 ...
## $ personal_status        : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3
3 3 3 3 3 1 4 ...
## $ other_debtors          : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1
1 1 1 1 1 ...
## $ residence_since        : int  4 2 3 4 4 4 4 2 4 2 ...
## $ property               : Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4
4 2 3 1 3 ...
## $ age                    : int  67 22 49 45 53 35 53 35 61 28 ...
## $ other_installment_plans: Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3
3 3 3 3 3 ...
## $ housing                : Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3
3 2 1 2 2 ...
## $ existing_credits       : int  2 1 1 1 2 1 1 1 1 2 ...
## $ job                    : Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3
2 3 4 2 4 ...
## $ num_dependents         : int  1 1 2 2 2 2 1 1 1 1 ...
## $ telephone              : Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2
1 2 1 1 ...
## $ foreign_worker         : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1
1 1 1 1 ...
## $ credit_risk            : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1
2 ...
```

splitting data for logistic regression models

```
set.seed(97)
trainrows <- sample(nrow(data), nrow(data) * 0.70)
data.train <- data[trainrows, ]
data.test <- data[-trainrows,]
```

testing accuracy od the first logistic regression model which contains all predictor variables in the data set(accuracy was 0.24)

```
#modeling the data
risk.glm0 <- glm(credit_risk ~ ., family = binomial, data.train) #model with
all explanatory variables
summary(risk.glm0)
```

```
##
## Call:
## glm(formula = credit_risk ~ ., family = binomial, data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1490  -0.6984  -0.3400   0.6366   2.6650
##
## Coefficients:
##                                 Estimate   Std. Error  z value
Pr(>|z|)
## (Intercept)                   0.90993041   1.37164433   0.663
0.507083
## checking_account_statusA12   -0.40811807   0.27225845  -1.499
0.133871
## checking_account_statusA13   -0.78218907   0.46457105  -1.684
0.092243 .
## checking_account_statusA14   -1.65656177   0.28409013  -5.831
0.00000000551 ***
## duration                      0.03629934   0.01127744   3.219
0.001287 **
## credit_historyA31             0.37772830   0.72730468   0.519
0.603514
## credit_historyA32            -0.52474838   0.54981018  -0.954
0.339872
## credit_historyA33            -0.53064479   0.58828273  -0.902
0.367044
## credit_historyA34            -1.61599187   0.55448456  -2.914
0.003564 **
## purposeA41                   -1.77870511   0.45014561  -3.951
0.00007769553 ***
## purposeA410                  -1.48265419   0.94541067  -1.568
0.116819
## purposeA42                   -0.90835749   0.32652159  -2.782
0.005404 **
## purposeA43                   -1.12967811   0.30736694  -3.675
0.000238 ***
## purposeA44                   -0.43161898   0.89382054  -0.483
0.629172
## purposeA45                   -0.52601881   0.65794360  -0.799
0.424007
## purposeA46                    0.50091563   0.49347028   1.015
```

```
0.310064
## purposeA48                  -15.36754818 471.75675832  -0.033
0.974013
## purposeA49                   -0.82715763   0.41429422  -1.997
0.045874 *
## credit_amount                 0.00010859   0.00005617   1.933
0.053228 .
## savings_account_statusA62    -0.41044777   0.34480139  -1.190
0.233894
## savings_account_statusA63    -0.43705807   0.48060640  -0.909
0.363145
## savings_account_statusA64    -1.27459829   0.60420699  -2.110
0.034898 *
## savings_account_statusA65    -1.30332469   0.33532125  -3.887
0.000102 ***
## employment_statusA72         -0.15137091   0.52117210  -0.290
0.771477
## employment_statusA73         -0.34878866   0.49631419  -0.703
0.482207
## employment_statusA74         -0.95585852   0.54498233  -1.754
0.079443 .
## employment_statusA75         -0.38051429   0.50411309  -0.755
0.450357
## installment_rate              0.22849102   0.10928613   2.091
0.036550 *
## personal_statusA92           -0.28593093   0.50418216  -0.567
0.570634
## personal_statusA93           -0.65838656   0.49716983  -1.324
0.185414
## personal_statusA94           -0.37156841   0.57787825  -0.643
0.520232
## other_debtorsA102            -0.09876632   0.48949769  -0.202
0.840096
## other_debtorsA103            -0.54561172   0.48590098  -1.123
0.261486
## residence_since               0.02919682   0.10656876   0.274
0.784106
## propertyA122                  0.13381253   0.31426657   0.426
0.670259
## propertyA123                  0.27351748   0.28907290   0.946
0.344052
## propertyA124                  1.10223402   0.57028545   1.933
0.053264 .
## age                          -0.01700166   0.01137227  -1.495
0.134912
## other_installment_plansA142  -0.11889630   0.52767460  -0.225
0.821729
## other_installment_plansA143  -0.62269142   0.28212626  -2.207
0.027304 *
## housingA152                  -0.49695776   0.28487504  -1.744
```

```
0.081076 .
## housingA153                -1.03308457   0.61614164  -1.677
0.093601 .
## existing_credits            0.35552292   0.23006894   1.545
0.122276
## jobA172                     0.56004084   0.91540835   0.612
0.540674
## jobA173                     0.77353045   0.87653385   0.882
0.377513
## jobA174                     0.44738047   0.87652313   0.510
0.609769
## num_dependents             -0.05011305   0.31084346  -0.161
0.871923
## telephoneA192              -0.29625868   0.24597145  -1.204
0.228418
## foreign_workerA202         -1.47976692   0.83087655  -1.781
0.074917 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 871.48  on 699  degrees of freedom
## Residual deviance: 610.22  on 651  degrees of freedom
## AIC: 708.22
##
## Number of Fisher Scoring iterations: 14
```

```
anova(risk.glm0, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: credit_risk
##
## Terms added sequentially (first to last)
##
##
##                         Df Deviance Resid. Df Resid. Dev
Pr(>Chi)
## NULL                                      699     871.48
## checking_account_status  3  100.900       696     770.58 <
0.00000000000000022
## duration                 1   34.726       695     735.86
0.000000003796
## credit_history           4   25.163       691     710.69
0.000046660516
## purpose                  9   31.331       682     679.36
0.0002597
```

```
## credit_amount            1    0.602       681     678.76
0.4378726
## savings_account_status   4   21.354       677     657.41
0.0002694
## employment_status        4    9.444       673     647.96
0.0509166
## installment_rate         1    3.779       672     644.18
0.0518860
## personal_status          3    4.556       669     639.63
0.2073247
## other_debtors            2    2.196       667     637.43
0.3334973
## residence_since          1    0.156       666     637.28
0.6932177
## property                 3    3.298       663     633.98
0.3478778
## age                      1    4.166       662     629.81
0.0412429
## other_installment_plans  2    5.617       660     624.19
0.0602819
## housing                  2    3.726       658     620.47
0.1551723
## existing_credits         1    2.388       657     618.08
0.1222582
## job                      3    2.471       654     615.61
0.4804837
## num_dependents           1    0.034       653     615.57
0.8542341
## telephone                1    1.217       652     614.36
0.2700171
## foreign_worker           1    4.137       651     610.22
0.0419647
##
## NULL
## checking_account_status ***
## duration                ***
## credit_history          ***
## purpose                 ***
## credit_amount
## savings_account_status  ***
## employment_status       .
## installment_rate        .
## personal_status
## other_debtors
## residence_since
## property
## age                     *
## other_installment_plans .
## housing
## existing_credits
```

```
## job
## num_dependents
## telephone
## foreign_worker          *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 4.2.3
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
pR2(risk.glm0)#how good does the model fit the data
```

```
## fitting null model for pseudo-r2
```

```
##          llh      llhNull           G2     McFadden         r2ML
r2CU
## -305.1107146 -435.7408445  261.2602597    0.2997886    0.3114924
0.4374577
```

```
fitted.results <-
predict(risk.glm0,newdata=subset(data.test,select=c("checking_account_status"
, "duration", "credit_history", "purpose", "credit_amount",
"savings_account_status", "employment_status", "installment_rate",
"personal_status", "other_debtors", "residence_since", "property", "age",
"other_installment_plans", "housing", "existing_credits", "job",
"num_dependents", "telephone", "foreign_worker")),type='response')
fitted.results <- ifelse(fitted.results >0.5, 1,2)
misClasificError <- mean(fitted.results != data.test$credit_risk)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.24"
```

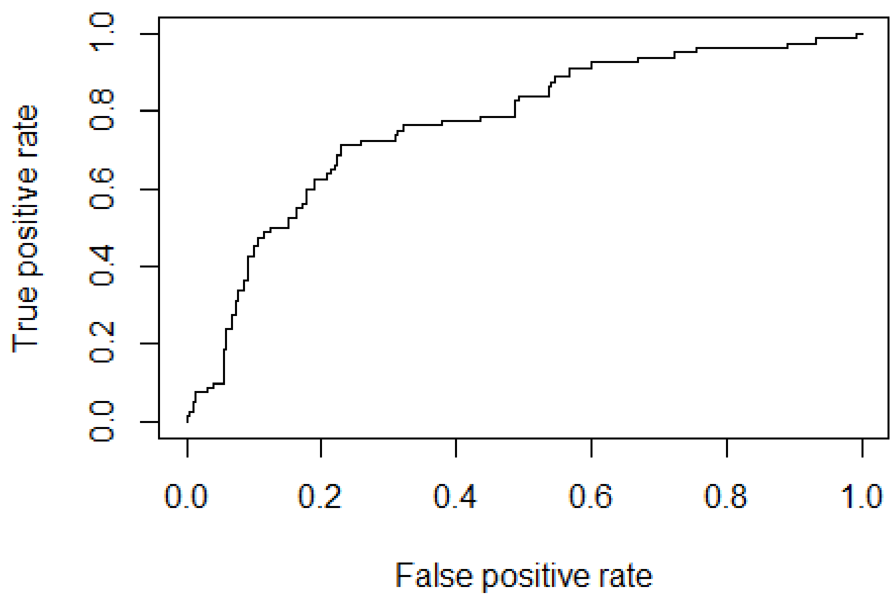
measuring the area under the curve of the first logistic regressiion model the result was 0.766

```
#measuring the auc
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.2.3
```

```
p <- predict(risk.glm0,
newdata=subset(data.test,select=c("checking_account_status", "duration",
"credit_history", "purpose", "credit_amount", "savings_account_status",
"employment_status", "installment_rate", "personal_status", "other_debtors",
```

```
"residence_since", "property", "age", "other_installment_plans", "housing",
"existing_credits", "job", "num_dependents", "telephone", "foreign_worker")),
type="response")
pr <- prediction(p, data.test$credit_risk)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```

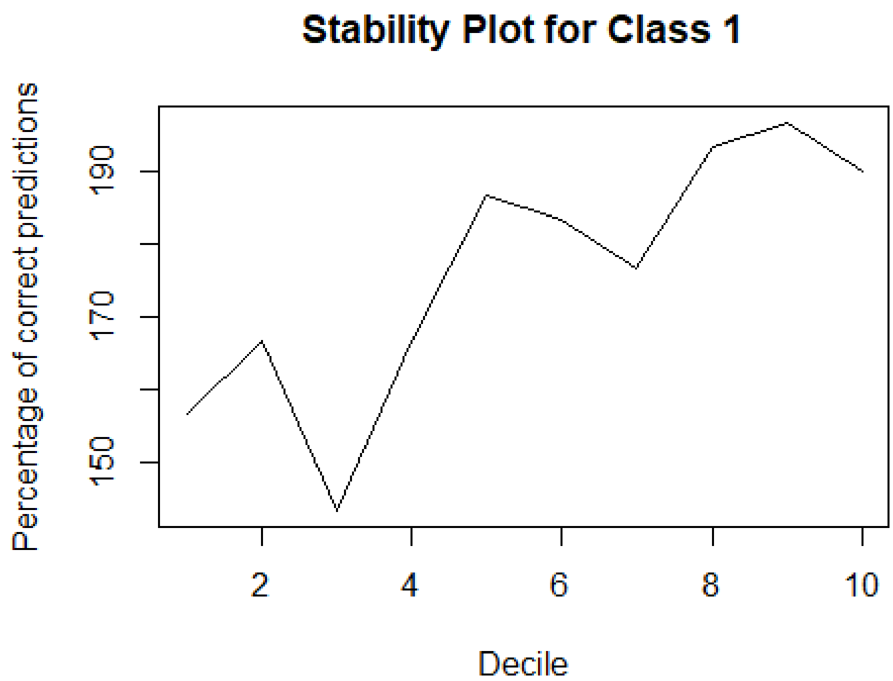



```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc

## [1] 0.7666477

auc

## [1] 0.7666477
```

testing the stability of the model. the results stated that the model was unstable because of the how low the the model performs at the first and third decile while, the performance increases at lower deciles

```
#stability test
probA<-data.frame(data.test$credit_risk,fitted.results,p)#dataframe showing
results and probabilities

probA<-probA[order(probA$p,decreasing=TRUE),]
```

```r
#-----Create empty df-------
decileDF<- data.frame(matrix(ncol=4,nrow = 0))
colnames(decileDF)<-
c("Decile","per_correct_preds","No_correct_Preds","cum_preds")
#-----Initialize varables
num_of_deciles=10
Obs_per_decile<-nrow(probA)/num_of_deciles
decile_count=1
start=1
stop=(start-1) + Obs_per_decile
prev_cum_pred<-0
x=0
#-----Loop through DF and create deciles
while (x < nrow(probA)) {
  subset<-probA[c(start:stop),]
  correct_count<-
ifelse(subset$data.test.credit_risk==subset$fitted.results,1,2)
  no_correct_Preds<-sum(correct_count,na.rm = TRUE)
  per_correct_Preds<-(no_correct_Preds/Obs_per_decile)*100
  cum_preds<-no_correct_Preds+prev_cum_pred
  addRow<-
data.frame("Decile"=decile_count,"per_correct_preds"=per_correct_Preds,"No_co
rrect_Preds"=no_correct_Preds,"cum_preds"=cum_preds)
  decileDF<-rbind(decileDF,addRow)
  prev_cum_pred<-prev_cum_pred+no_correct_Preds
  start<-stop+1
  stop=(start-1) + Obs_per_decile
  x<-x+Obs_per_decile
  decile_count<-decile_count+1
}
#------Stability plot (correct preds per decile)
plot(decileDF$Decile,decileDF$per_correct_preds,type = "l",xlab =
"Decile",ylab = "Percentage of correct predictions",main="Stability Plot for
Class 1")
```



Logistic regression model 2 was created using step wise variable selection, this model was using the 10 most statistically signigicant predictor variables as x values to predict credit risk

```
#logistic regression model 2

library(caret)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3

## Loading required package: lattice

library(leaps)

## Warning: package 'leaps' was built under R version 4.2.3

library(MASS)
nullModel = glm(credit_risk ~ 1, family =  binomial,data = data.train) #
model with the intercept only
model.step<-stepAIC(nullModel, # start with a model containing no variables
                direction = 'forward', # run forward selection
                scope = list(upper = risk.glm0, # the maximum to consider is
a model with all variables
                             lower = nullModel), # the minimum to consider is
a model with no variables
                trace = 0) # do not show the step-by-step process of model
```

```
selection
summary(model.step)

##
## Call:
## glm(formula = credit_risk ~ checking_account_status + duration +
##     credit_history + purpose + savings_account_status + foreign_worker +
##     housing + other_installment_plans + age + property, family = binomial,
##     data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3203  -0.6939  -0.3640   0.6868   2.5657
##
## Coefficients:
##                               Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                   2.214979   0.734967   3.014      0.002581 **
## checking_account_statusA12   -0.402037   0.255576  -1.573      0.115704
## checking_account_statusA13   -0.915819   0.447599  -2.046      0.040749 *
## checking_account_statusA14   -1.636563   0.273964  -5.974 0.00000000232
***
## duration                      0.043402   0.009007   4.819 0.00000144285
***
## credit_historyA31            -0.070361   0.678618  -0.104      0.917421
## credit_historyA32            -0.901969   0.504209  -1.789      0.073634 .
## credit_historyA33            -0.675107   0.567028  -1.191      0.233808
## credit_historyA34            -1.710629   0.533584  -3.206      0.001346 **
## purposeA41                   -1.606941   0.423814  -3.792      0.000150
***
## purposeA410                  -1.627981   0.858350  -1.897      0.057876 .
## purposeA42                   -0.827832   0.311240  -2.660      0.007819 **
## purposeA43                   -1.081911   0.292630  -3.697      0.000218
***
## purposeA44                   -0.149137   0.851320  -0.175      0.860935
## purposeA45                   -0.419931   0.653457  -0.643      0.520465
## purposeA46                    0.587317   0.482437   1.217      0.223453
## purposeA48                  -15.198752 490.497312  -0.031      0.975280
## purposeA49                   -0.865477   0.400148  -2.163      0.030550 *
## savings_account_statusA62    -0.410167   0.330737  -1.240      0.214915
## savings_account_statusA63    -0.538546   0.463004  -1.163      0.244766
## savings_account_statusA64    -1.054739   0.575404  -1.833      0.066797 .
## savings_account_statusA65    -1.245181   0.318942  -3.904 0.00009457749
***
## foreign_workerA202           -1.639500   0.810628  -2.023      0.043124 *
## housingA152                  -0.580741   0.262713  -2.211      0.027067 *
## housingA153                  -1.132565   0.578856  -1.957      0.050399 .
## other_installment_plansA142  -0.015500   0.504159  -0.031      0.975473
## other_installment_plansA143  -0.597846   0.273609  -2.185      0.028886 *
## age                          -0.017808   0.009981  -1.784      0.074408 .
## propertyA122                  0.216132   0.298783   0.723      0.469451
```

```
## propertyA123                    0.349223   0.272457   1.282      0.199929
## propertyA124                    1.259026   0.517691   2.432      0.015016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 871.48  on 699  degrees of freedom
## Residual deviance: 631.61  on 669  degrees of freedom
## AIC: 693.61
##
## Number of Fisher Scoring iterations: 14
```

```r
# model contains 10 predictor variables as opposed the original 19

risk.step.glm<-glm(formula = credit_risk ~ checking_account_status + duration
+ credit_history + purpose + savings_account_status + foreign_worker +
housing + other_installment_plans + age + property, family = binomial, data =
data.train)
summary(risk.step.glm)
```

```
##
## Call:
## glm(formula = credit_risk ~ checking_account_status + duration +
##     credit_history + purpose + savings_account_status + foreign_worker +
##     housing + other_installment_plans + age + property, family = binomial,
##     data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3203  -0.6939  -0.3640   0.6868   2.5657
##
## Coefficients:
##                                 Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                     2.214979   0.734967   3.014      0.002581 **
## checking_account_statusA12     -0.402037   0.255576  -1.573      0.115704
## checking_account_statusA13     -0.915819   0.447599  -2.046      0.040749 *
## checking_account_statusA14     -1.636563   0.273964  -5.974 0.00000000232
***
## duration                        0.043402   0.009007   4.819 0.00000144285
***
## credit_historyA31              -0.070361   0.678618  -0.104      0.917421
## credit_historyA32              -0.901969   0.504209  -1.789      0.073634 .
## credit_historyA33              -0.675107   0.567028  -1.191      0.233808
## credit_historyA34              -1.710629   0.533584  -3.206      0.001346 **
## purposeA41                     -1.606941   0.423814  -3.792      0.000150
***
## purposeA410                    -1.627981   0.858350  -1.897      0.057876 .
## purposeA42                     -0.827832   0.311240  -2.660      0.007819 **
## purposeA43                     -1.081911   0.292630  -3.697      0.000218
```

```
***
## purposeA44                    -0.149137   0.851320  -0.175      0.860935
## purposeA45                    -0.419931   0.653457  -0.643      0.520465
## purposeA46                     0.587317   0.482437   1.217      0.223453
## purposeA48                   -15.198752 490.497312  -0.031      0.975280
## purposeA49                    -0.865477   0.400148  -2.163      0.030550 *
## savings_account_statusA62     -0.410167   0.330737  -1.240      0.214915
## savings_account_statusA63     -0.538546   0.463004  -1.163      0.244766
## savings_account_statusA64     -1.054739   0.575404  -1.833      0.066797 .
## savings_account_statusA65     -1.245181   0.318942  -3.904 0.00009457749
***
## foreign_workerA202            -1.639500   0.810628  -2.023      0.043124 *
## housingA152                   -0.580741   0.262713  -2.211      0.027067 *
## housingA153                   -1.132565   0.578856  -1.957      0.050399 .
## other_installment_plansA142   -0.015500   0.504159  -0.031      0.975473
## other_installment_plansA143   -0.597846   0.273609  -2.185      0.028886 *
## age                           -0.017808   0.009981  -1.784      0.074408 .
## propertyA122                   0.216132   0.298783   0.723      0.469451
## propertyA123                   0.349223   0.272457   1.282      0.199929
## propertyA124                   1.259026   0.517691   2.432      0.015016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 871.48  on 699  degrees of freedom
## Residual deviance: 631.61  on 669  degrees of freedom
## AIC: 693.61
##
## Number of Fisher Scoring iterations: 14
```

```
anova(risk.step.glm, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: credit_risk
##
## Terms added sequentially (first to last)
##
##
##                         Df Deviance Resid. Df Resid. Dev
Pr(>Chi)
## NULL                                      699     871.48
## checking_account_status  3  100.900       696     770.58 <
0.00000000000000022
## duration                 1   34.726       695     735.86
0.000000003796
## credit_history           4   25.163       691     710.69
```

```
0.000046660516
## purpose                   9   31.331       682     679.36
0.0002597
## savings_account_status   4   20.866       678     658.50
0.0003367
## foreign_worker            1    5.385       677     653.11
0.0203136
## housing                   2    5.845       675     647.27
0.0538030
## other_installment_plans  2    6.185       673     641.08
0.0453896
## age                       1    3.110       672     637.97
0.0778358
## property                  3    6.367       669     631.61
0.0950739
##
## NULL
## checking_account_status ***
## duration                ***
## credit_history          ***
## purpose                 ***
## savings_account_status  ***
## foreign_worker          *
## housing                 .
## other_installment_plans *
## age                     .
## property                .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

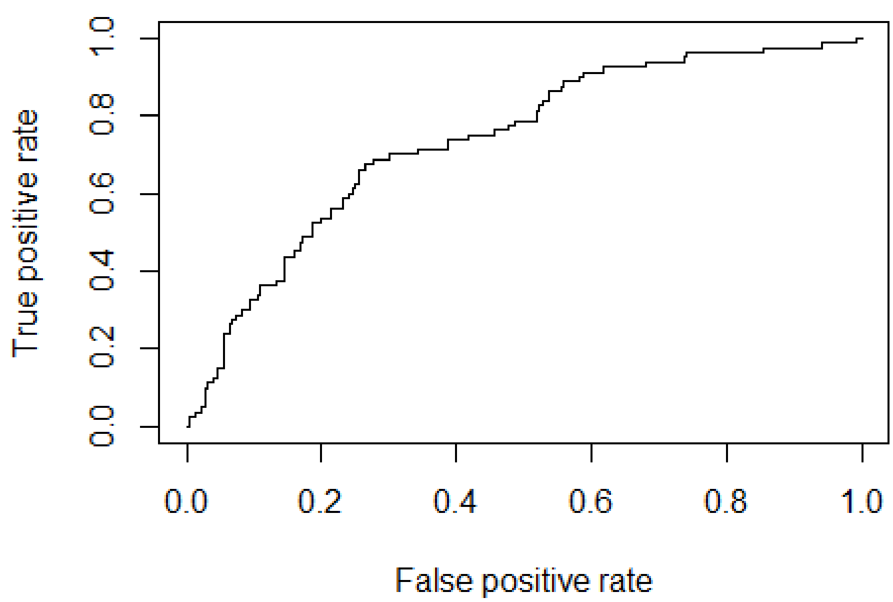
The accuracy of the second logistic regression model was 0.24 as well, which is giving the impression that logist regression may not be the best model for classifying this data

```
#MEASURING ACCURACY
pR2(risk.step.glm)#how good does the model fit the data

## fitting null model for pseudo-r2

##          llh      llhNull           G2     McFadden         r2ML
r2CU
## -315.8032158 -435.7408445  239.8752573    0.2752499    0.2901339
0.4074619

fitted.results2 <-
predict(risk.step.glm,newdata=subset(data.test,select=c('checking_account_sta
tus' , 'duration', 'credit_history', 'purpose', 'savings_account_status',
'foreign_worker','housing','other_installment_plans', 'age',
'property')),type='response')
fitted.results2 <- ifelse(fitted.results2 >0.5, 1,2)
misClasificError2 <- mean(fitted.results2 != data.test$credit_risk)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.24"
```

the area under the curve for this model was 0.737

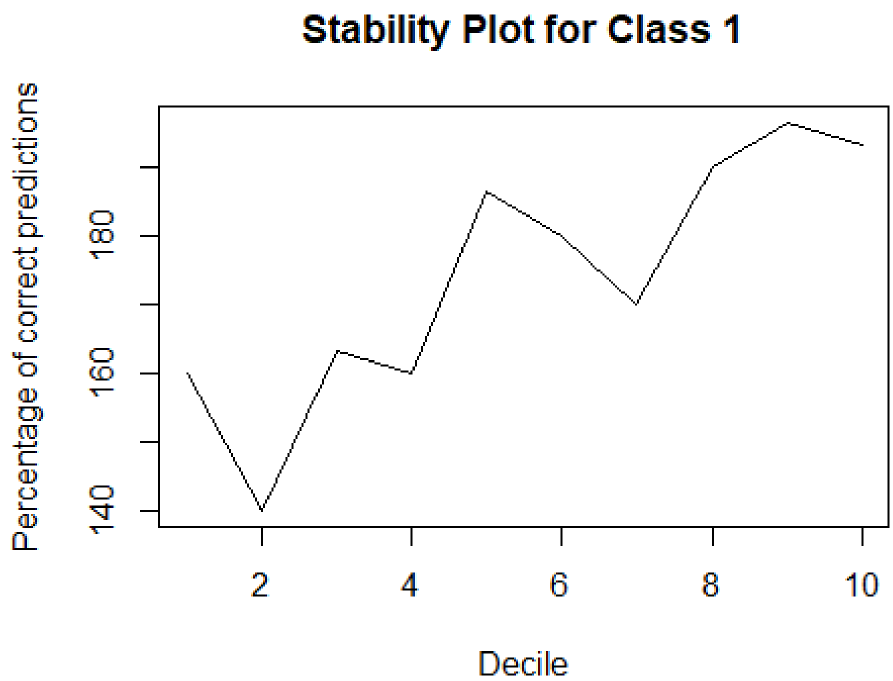
```
#meausring the auc

p2 <- predict(risk.step.glm,
newdata=subset(data.test,select=c('checking_account_status' , 'duration',
'credit_history', 'purpose', 'savings_account_status',
'foreign_worker','housing','other_installment_plans', 'age', 'property')),
type="response")
pr_2 <- prediction(p2, data.test$credit_risk)
prf2 <- performance(pr_2, measure = "tpr", x.measure = "fpr")
plot(prf2)
```




```
auc2 <- performance(pr_2, measure = "auc")
auc2 <- auc2@y.values[[1]]
auc2

## [1] 0.7370455

auc2

## [1] 0.7370455
```

this model was also unstable because it performed very poorly at the higher deciles compared to the lower ones

```
#stability test
probB<-data.frame(data.test$credit_risk,fitted.results2,p2)#dataframe showing
results and probabilities

probB<-probB[order(probB$p2,decreasing=TRUE),]

#-----Create empty df-------
decileDF2<- data.frame(matrix(ncol=4,nrow = 0))
colnames(decileDF2)<-
c("Decile","per_correct_preds","No_correct_Preds","cum_preds")
#-----Initialize varables
num_of_decilesB=10
Obs_per_decile2<-nrow(probB)/num_of_decilesB
decile_countB=1
startB=1
stopB=(startB-1) + Obs_per_decile2
prev_cum_predB<-0
B=0
#-----Loop through DF and create deciles
while (B < nrow(probB)) {
  subsetB<-probB[c(startB:stopB),]
  correct_countB<-
ifelse(subsetB$data.test.credit_risk==subsetB$fitted.results2,1,2)
  no_correct_PredsB<-sum(correct_countB,na.rm = TRUE)
  per_correct_PredsB<-(no_correct_PredsB/Obs_per_decile2)*100
  cum_predsB<-no_correct_PredsB+prev_cum_predB
  addRowB<-
data.frame("Decile"=decile_countB,"per_correct_preds"=per_correct_PredsB,"No_
correct_Preds"=no_correct_PredsB,"cum_preds"=cum_predsB)
  decileDF2<-rbind(decileDF2,addRowB)
  prev_cum_predB<-prev_cum_predB+no_correct_PredsB
  startB<-stopB+1
  stopB=(startB-1) + Obs_per_decile2
  B<-B+Obs_per_decile2
  decile_countB<-decile_countB+1
}
#------Stability plot (correct preds per decile)
plot(decileDF2$Decile,decileDF2$per_correct_preds,type = "l",xlab =
"Decile",ylab = "Percentage of correct predictions",main="Stability Plot for
Class 1")
```



Decsion tree models two decision tree models were made one with complexity of 0.01 and another with 0.001, this was done to see whether a high complexity or simpler model would better fit and predict the data.

```
rm(list=ls())

library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.2.3

library(caret)
library(pROC)

## Warning: package 'pROC' was built under R version 4.2.3

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

# Read in the dataset
data <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-
databases/statlog/german/german.data",header=FALSE,sep=" ")
```

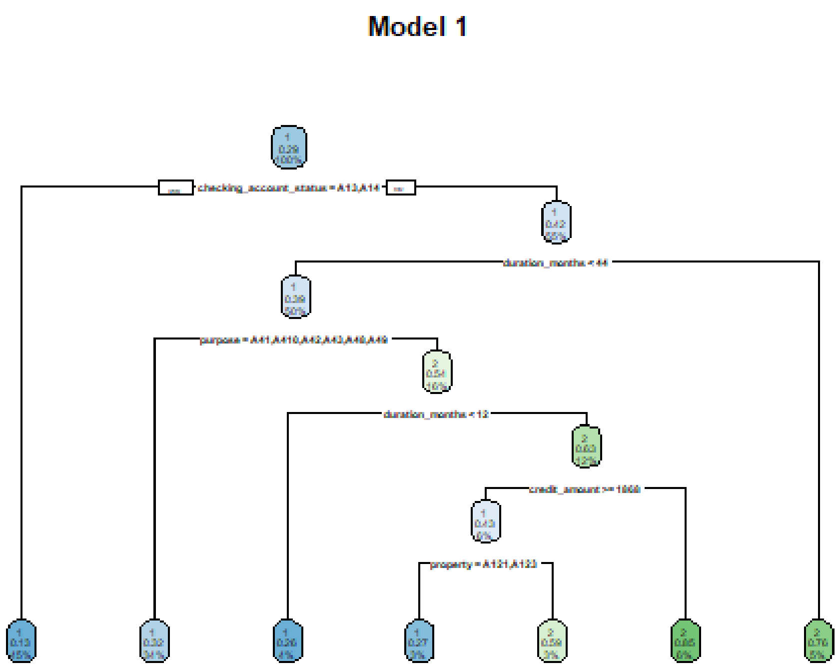
```
# Rename the columns
colnames(data) <- c("checking_account_status", "duration_months",
"credit_history",
                    "purpose", "credit_amount", "savings_account_status",
"employment_status",
                    "installment_rate", "personal_status_sex",
"other_debtors_guarantors",
                    "present_residence_since", "property", "age_years",
"other_installment_plans",
                    "housing", "number_existing_credits", "job",
"number_people_liable",
                    "telephone", "foreign_worker", "credit_risk")
View(data)
str(data)

## 'data.frame':    1000 obs. of  21 variables:
##  $ checking_account_status : chr  "A11" "A12" "A14" "A11" ...
##  $ duration_months         : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ credit_history          : chr  "A34" "A32" "A34" "A32" ...
##  $ purpose                 : chr  "A43" "A43" "A46" "A42" ...
##  $ credit_amount           : int  1169 5951 2096 7882 4870 9055 2835 6948
3059 5234 ...
##  $ savings_account_status  : chr  "A65" "A61" "A61" "A61" ...
##  $ employment_status       : chr  "A75" "A73" "A74" "A74" ...
##  $ installment_rate        : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ personal_status_sex     : chr  "A93" "A92" "A93" "A93" ...
##  $ other_debtors_guarantors: chr  "A101" "A101" "A101" "A103" ...
##  $ present_residence_since : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ property                : chr  "A121" "A121" "A121" "A122" ...
##  $ age_years               : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ other_installment_plans : chr  "A143" "A143" "A143" "A143" ...
##  $ housing                 : chr  "A152" "A152" "A152" "A153" ...
##  $ number_existing_credits : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ job                     : chr  "A173" "A173" "A172" "A173" ...
##  $ number_people_liable    : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ telephone               : chr  "A192" "A191" "A191" "A191" ...
##  $ foreign_worker          : chr  "A201" "A201" "A201" "A201" ...
##  $ credit_risk             : int  1 2 1 1 2 1 1 1 1 2 ...

summary(data)

##  checking_account_status duration_months credit_history       purpose
##  Length:1000             Min.   : 4.0    Length:1000        Length:1000
##  Class :character        1st Qu.:12.0    Class :character   Class
:character
##  Mode  :character        Median :18.0    Mode  :character   Mode
:character
##                          Mean   :20.9
##                          3rd Qu.:24.0
```

```
##                           Max.   :72.0  
##  credit_amount   savings_account_status employment_status 
installment_rate
##  Min.   :  250   Length:1000            Length:1000        Min.   :1.000  
##  1st Qu.: 1366   Class :character       Class :character   1st Qu.:2.000  
##  Median : 2320   Mode  :character       Mode  :character   Median :3.000  
##  Mean   : 3271                                             Mean   :2.973  
##  3rd Qu.: 3972                                             3rd Qu.:4.000  
##  Max.   :18424                                             Max.   :4.000  
##  personal_status_sex other_debtors_guarantors present_residence_since
##  Length:1000         Length:1000              Min.   :1.000          
##  Class :character    Class :character         1st Qu.:2.000          
##  Mode  :character    Mode  :character         Median :3.000          
##                                               Mean   :2.845          
##                                               3rd Qu.:4.000          
##                                               Max.   :4.000          
##    property           age_years     other_installment_plans   housing         
##  Length:1000        Min.   :19.00   Length:1000              Length:1000       
##  Class :character   1st Qu.:27.00   Class :character         Class 
:character  
##  Mode  :character   Median :33.00   Mode  :character         Mode  
:character  
##                     Mean   :35.55                                              
##                     3rd Qu.:42.00                                              
##                     Max.   :75.00                                              
##  number_existing_credits     job             number_people_liable
##  Min.   :1.000           Length:1000        Min.   :1.000       
##  1st Qu.:1.000           Class :character   1st Qu.:1.000       
##  Median :1.000           Mode  :character   Median :1.000       
##  Mean   :1.407                              Mean   :1.155       
##  3rd Qu.:2.000                              3rd Qu.:1.000       
##  Max.   :4.000                              Max.   :2.000       
##   telephone         foreign_worker      credit_risk 
##  Length:1000        Length:1000        Min.   :1.0  
##  Class :character   Class :character   1st Qu.:1.0  
##  Mode  :character   Mode  :character   Median :1.0  
##                                        Mean   :1.3  
##                                        3rd Qu.:2.0  
##                                        Max.   :2.0

# Convert categorical variables to factors
cat_cols <- c(1,3,4,6,7,9,10,12,14,15,17,19,20)
data[,cat_cols] <- lapply(data[,cat_cols], as.factor)

# Split the dataset into training and testing sets
set.seed(123)
train_index <- createDataPartition(data$credit_risk, p=0.7, list=FALSE)
train_data <- data[train_index,]
test_data <- data[-train_index,]
```

```r
# Define the two different decision tree models
model1 <- rpart(credit_risk ~ ., data=train_data, method="class",
minbucket=10, cp=0.01)
model2 <- rpart(credit_risk ~ ., data=train_data, method="class",
minbucket=5, cp=0.001)

# Make predictions on the test set
predictions1 <- predict(model1, newdata=test_data, type="class")
predictions2 <- predict(model2, newdata=test_data, type="class")
```

plotting decision tress model 1

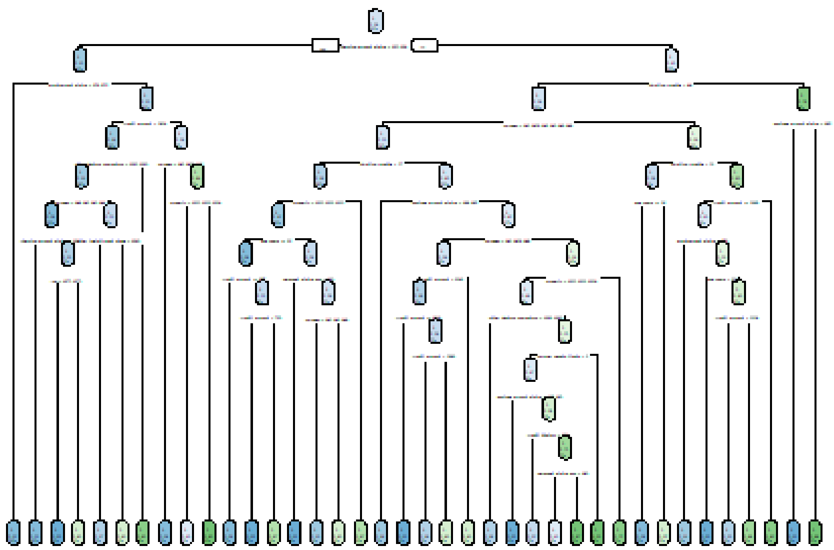
```r
# Plot the two models
rpart.plot(model1, main="Model 1")
```



model 2

```r
# Plot the two models

rpart.plot(model2, main="Model 2")

## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

Model 2



in evaluating the accuracy of the models it was noted that both provided an accuracy of 0.707

```
# Evaluate the accuracy of the two models
accuracy1 <- sum(predictions1 == test_data$credit_risk) / nrow(test_data)
accuracy2 <- sum(predictions2 == test_data$credit_risk) / nrow(test_data)

# Print the accuracy of the two models
cat("Accuracy of Model 1:", round(accuracy1, 3), "\n")

## Accuracy of Model 1: 0.707

cat("Accuracy of Model 2:", round(accuracy2, 3), "\n")

## Accuracy of Model 2: 0.707
```

the results for area under the curve of decision tree 1 was 0.72 and decision tree 2 was 0.66

```
# Calculate the AUC for the two models
prob1 <- predict(model1, newdata=test_data, type="prob")[,2]
roc1 <- roc(test_data$credit_risk, prob1)

## Setting levels: control = 1, case = 2

## Setting direction: controls < cases

auc1 <- auc(roc1)
```

```
prob2 <- predict(model2, newdata=test_data, type="prob")[,2]
roc2 <- roc(test_data$credit_risk, prob2)

## Setting levels: control = 1, case = 2
## Setting direction: controls < cases

auc2 <- auc(roc2)

# Print the AUC for the two models
cat("AUC of Model 1:", round(auc1, 3), "\n")

## AUC of Model 1: 0.72

cat("AUC of Model 2:", round(auc2, 3), "\n")

## AUC of Model 2: 0.66
```

in evaluating the stability of each model both were given a result of 1 meaning that they were both stable models

```
# Evaluate the stability of the two models
set.seed(456)
predictions1b <- predict(model1, newdata=test_data, type="class")
set.seed(789)
predictions1c <- predict(model1, newdata=test_data, type="class")
set.seed(456)
predictions2b <- predict(model2, newdata=test_data, type="class")
set.seed(789)
predictions2c <- predict(model2, newdata=test_data, type="class")

stability1 <- sum(predictions1 == predictions1b) / nrow(test_data)
stability2 <- sum(predictions2 == predictions2b) / nrow(test_data)

# Print the stability of the two models
cat("Stability of Model 1:", round(stability1, 3), "\n")

## Stability of Model 1: 1

cat("Stability of Model 2:", round(stability2, 3), "\n")

## Stability of Model 2: 1
```

it is to be noted that based on the evaluation criteria logistic regression model 1 and decision tree model 2 were deemed as not simple and logistic resgression model 2 and decision tree model 1 were deemed as the perfec level of simplicity giving them 1 on the simplicity scale and the other 2 models a 0.

Final evaluation

Evaluating models

Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions.

AUC ROC stands for "Area Under the Curve" of the "Receiver Operating Characteristic" curve. The AUC ROC curve is basically a way of measuring the performance of an ML model. AUC measures the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve.

Simplicity refers to how many nodes or variables the model uses to accurately make predictions.

Stability, also known as algorithmic stability, is a notion in computational learning theory of how a machine learning algorithm output is changed with small perturbations to its inputs. A stable learning algorithm is one for which the prediction does not change much when the training data is modified slightly

Making stability and simplicity into numbers between 1 and 0

stability

For evaluating these models stability will be measured as a binary value with 1 meaning the model is stable and 0 meaning not stable

simplicity

In the case of simplicity the ideal number of nodes/leaves/variables is between 8 and 10, hence 10 nodes will equal 1 on a scale for simplicity

The maximum number of nodes is 13, while the minimum is 5 hence values greater than 13 or less than 5 will equal 0

If the number of nodes is less than 8 but greater than 5 then (Nonodes – 5)/8-5

If number of nodes is greater than 10 but less than 12 then (13-Nonodes)/13-10

Weights

Accuracy = 0.45

Simplicity = 0.10

Auc = 0.30

Stability = 0.15

Threshold

Accuracy = <=0.60

AUC <= 0.65

The best model is decision tree model 1 based on the evaluation

Model Description	Performance Measures			Overall Score
	Accuracy Score	Simplicity		
		No of Leaves/Layers/Nodes	Score	
Logistic regression 1	0.24	19	0	0.3378
Logistic regression 2	0.24	10	1	0.4291
Decision tree 1	0.707	13	1	0.78415
Decision tree 2	0.707	Above 20	0	0.6665

Deployment

Deployment: Integrate the model into the lending institution's decision-making process. This may involve developing an application that allows loan officers to enter loan application data and receive a prediction from the model, telling them whether someone is a high or low credit risk which would allow them to asses whether or not to provide them with a loan.