Basics of Predictive modeling (Binary Classification)

CHINDU

In this report we will look into the basics of predictive modeling. It is aways important to carry out data cleaning, data preparatioan and feature engineering before modeling. These steps will not be covered in this report (Refer to to the Data-Preparation rep if you are intrested in learning about data-preparation)

Here lets assume the data is cleaned and prepared for modeling.

We will be using the dataset 'German Credit'. This well-known data set is used to classify customers as having good or bad credit based on customer attributes (e.g. information on bank accounts or property). The data can be found at the UC Irvine Machine Learning Repository and in the caret R package.

In this report we will build a few models to predict if a customer has a good or bad credit based on customer attributes.

```
#Read the data
german_credit = read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/germ

#Understand the structure
str(german_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ V1 : Factor w/ 4 levels "A11","A12","A13",..: 1 2 4 1 1 4 4 2 4 2 ...
##  $ V2 : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ V3 : Factor w/ 5 levels "A30","A31","A32",..: 5 3 5 3 4 3 3 3 3 5 ...
##  $ V4 : Factor w/ 10 levels "A40","A41","A410",..: 5 5 8 4 1 8 4 2 5 1 ...
##  $ V5 : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ V6 : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ V7 : Factor w/ 5 levels "A71","A72","A73",..: 5 3 4 4 3 3 5 3 4 1 ...
##  $ V8 : int  4 2 2 2 3 2 3 2 2 4 ...
##  $ V9 : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...
##  $ V10: Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ V11: int  4 2 3 4 4 4 4 2 4 2 ...
##  $ V12: Factor w/ 4 levels "A121","A122",..: 1 1 1 2 4 4 2 3 1 3 ...
##  $ V13: int  67 22 49 45 53 35 53 35 61 28 ...
##  $ V14: Factor w/ 3 levels "A141","A142",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ V15: Factor w/ 3 levels "A151","A152",..: 2 2 2 3 3 3 2 1 2 2 ...
##  $ V16: int  2 1 1 1 2 1 1 1 1 2 ...
##  $ V17: Factor w/ 4 levels "A171","A172",..: 3 3 2 3 3 2 3 4 2 4 ...
##  $ V18: int  1 1 2 2 2 2 1 1 1 1 ...
##  $ V19: Factor w/ 2 levels "A191","A192": 2 1 1 1 1 2 1 2 1 1 ...
##  $ V20: Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ V21: int  1 2 1 1 2 1 1 1 1 2 ...
```

The data is not labeled. Lets first label our data.

```r
colnames(german_credit) = c("chk_acct", "duration", "credit_his", "purpose",
                            "amount", "saving_acct", "present_emp", "installment_rate", "sex", "other_d
                            "present_resid", "property", "age", "other_install", "housing", "n_credits"
                            "job", "n_people", "telephone", "foreign", "CustomerCredit")
str(german_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ chk_acct        : 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_his      : 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 ...
##  $ amount          : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ saving_acct     : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ present_emp     : 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 ...
##  $ sex             : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...
##  $ other_debtor    : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ present_resid   : 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_install   : 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 ...
##  $ n_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 ...
##  $ n_people        : 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         : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CustomerCredit  : int  1 2 1 1 2 1 1 1 1 2 ...
```

For binary classification we need to first ensure that our target/repose variable is a factor.

```r
#Changing CustomerCredit to a factor
german_credit$CustomerCredit <- as.factor(german_credit$CustomerCredit)
str(german_credit)
```

```
## 'data.frame':    1000 obs. of  21 variables:
##  $ chk_acct        : 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_his      : 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 ...
##  $ amount          : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ saving_acct     : Factor w/ 5 levels "A61","A62","A63",..: 5 1 1 1 1 5 3 1 4 1 ...
##  $ present_emp     : 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 ...
##  $ sex             : Factor w/ 4 levels "A91","A92","A93",..: 3 2 3 3 3 3 3 3 1 4 ...
##  $ other_debtor    : Factor w/ 3 levels "A101","A102",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ present_resid   : 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_install   : 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 ...
##  $ n_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 ...
##  $ n_people        : 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         : Factor w/ 2 levels "A201","A202": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CustomerCredit  : Factor w/ 2 levels "1","2": 1 2 1 1 2 1 1 1 1 2 ...
```

In this data, 1 refers to "good" and 2 refers to "bad".

Next lets explore the distribution of our target variable. This is a key step to understand if there is a need to apply sampling methods (RUS,ROS,SMOTE,Ensemble different resampled datasets, etc ..) to solve data imbalance problems.

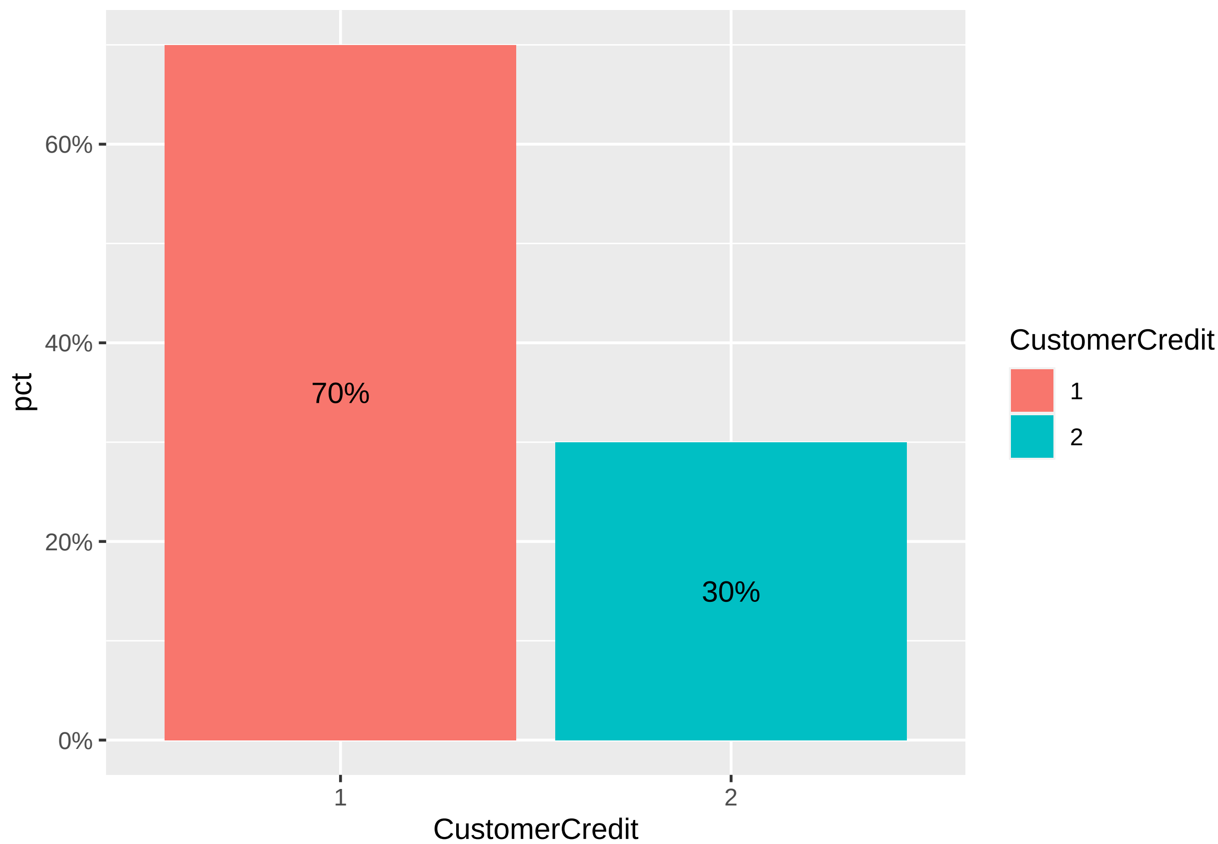
```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages ------------------ tidyverse 1.3.0 --
```

```
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts --------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
germanNew <- german_credit %>% group_by(CustomerCredit) %>%
 summarize(count = n()) %>%  # count records by species
 mutate(pct = count/sum(count))
ggplot(germanNew, aes(CustomerCredit, pct, fill = CustomerCredit)) +
  geom_bar(stat='identity') +
  geom_text(aes(label=scales::percent(pct)), position = position_stack(vjust = .5))+
  scale_y_continuous(labels = scales::percent)
```



The data contains 70% good credit and 30% bad credit. The data has a minor data imbalance. The ideal distribution is 50% good and 50% bad. Some algorithms are able to handle such data imbalance but in a situation where the algorithm performs poorly, sampling methods should be used. In this analysis, we will use this imbalanced data and determine if our algorithms are able to handle the data imbalance.

Data Splitting

One of the most important step in data modeling is to decide how to utilize the available data. A common technique is to split the data into testing and training sets. Training Set - used to develop the model (example estimaitng parameters and comparing models) Testing Set - used is used to estimate an unbiased assessment of the model's performance.

So what is a good split or data? The proportion of data can be driven by many factors, including the size of the original pool of samples and the total number of predictors.

There are a number of ways to split the data into training and testing sets. The most common approach is to use some version of random sampling. Completely random sampling is a straightforward strategy to implement and usually protects the process from being biased towards any characteristic of the data. However this approach can be problematic when the response is not evenly distributed across the outcome such as in our case. A less risky splitting strategy would be to use a stratified random sample based on the outcome. For classification models, this is accomplished by selecting samples at random within each class. This approach ensures that the frequency distribution of the outcome is approximately equal within the training and test sets

Split data into training and testing with a 70/30 split

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

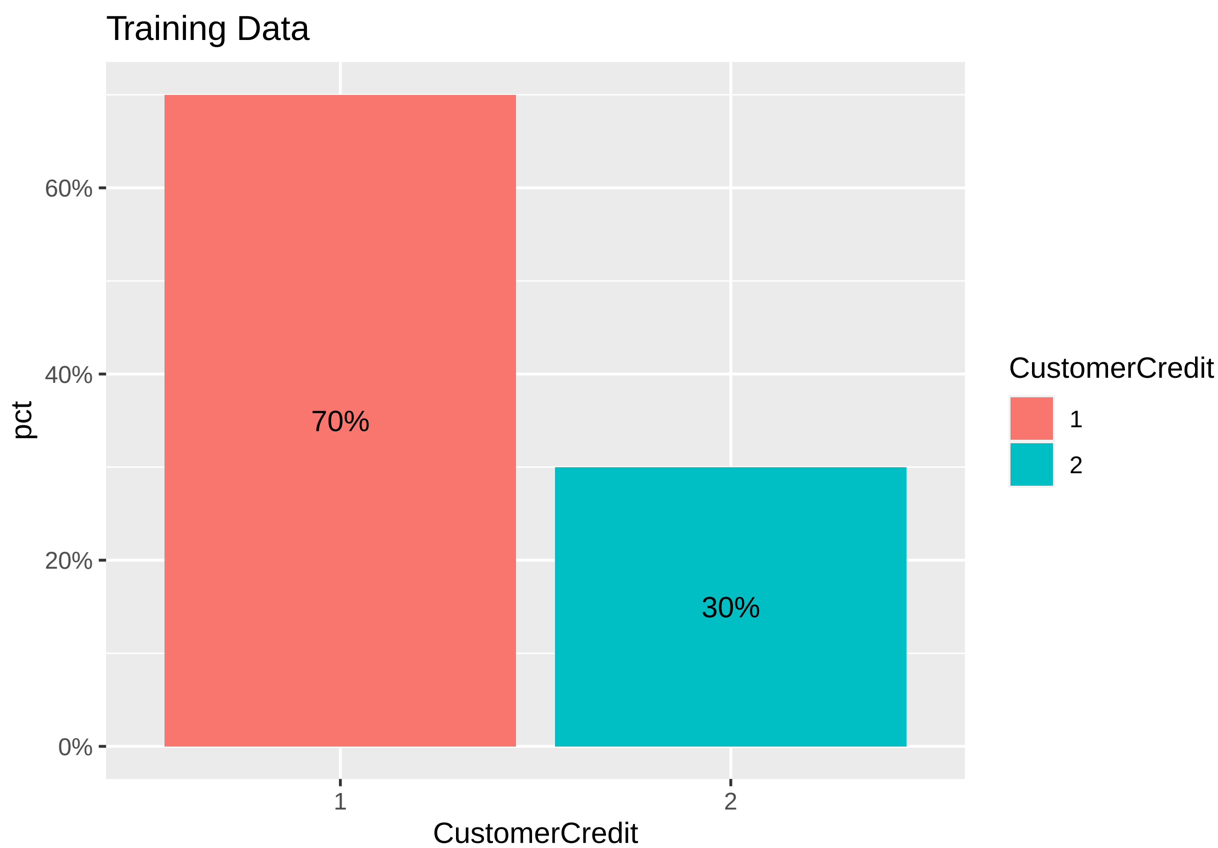
```r
set.seed(123)
in.train <- createDataPartition(as.factor(german_credit$CustomerCredit), p=0.7, list=FALSE)
train <- german_credit[in.train,]
test <- german_credit[-in.train,]
# Training Data
table(train$CustomerCredit)
```

```
##
##   1   2
## 490 210
```

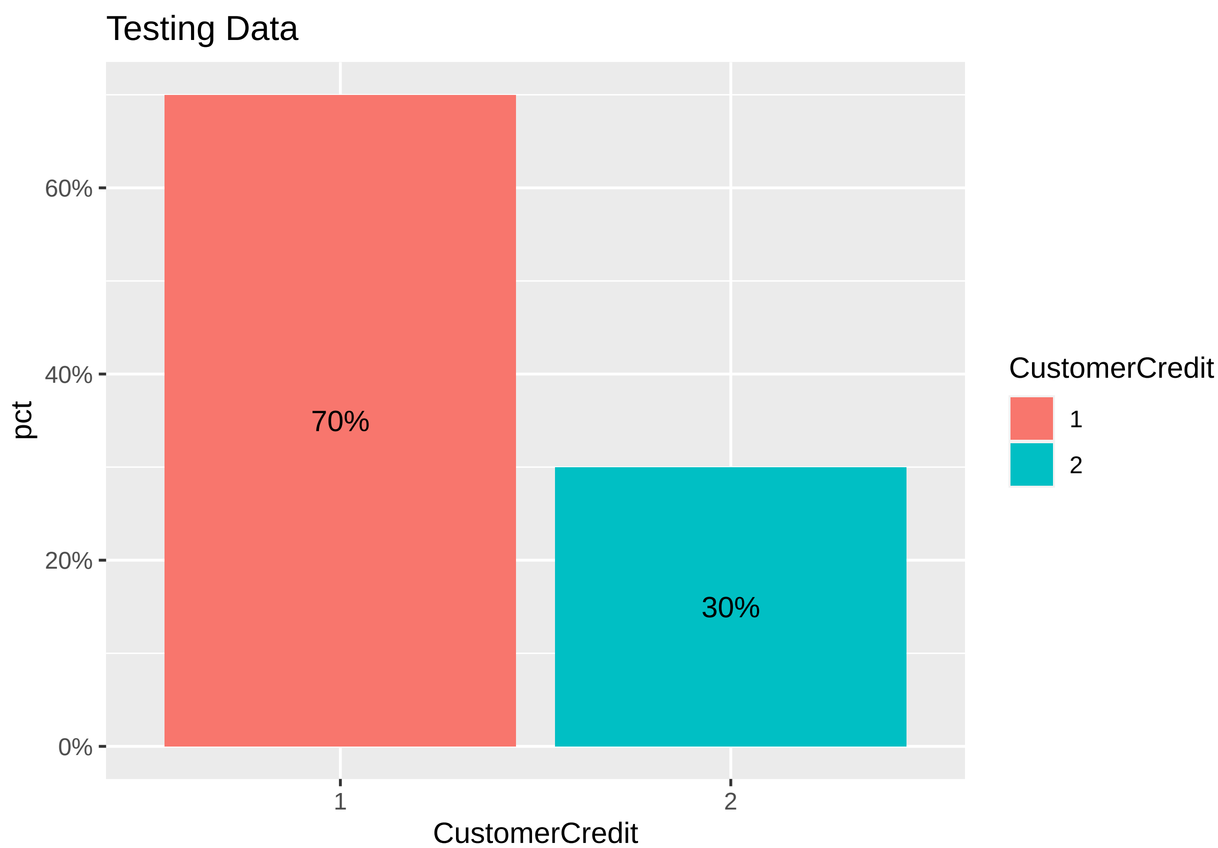
```r
# Testing Data
table(test$CustomerCredit)
```

```
##
##   1   2
## 210  90
```

```r
germanNew2 <- train %>% group_by(CustomerCredit) %>%
 summarize(count = n()) %>%  # count records by species
 mutate(pct = count/sum(count))
ggplot(germanNew2, aes(CustomerCredit, pct, fill = CustomerCredit)) +
  geom_bar(stat='identity') +
  geom_text(aes(label=scales::percent(pct)), position = position_stack(vjust = .5))+
  scale_y_continuous(labels = scales::percent) + labs(title= "Training Data")
```




```r
germanNew3 <- test %>% group_by(CustomerCredit) %>%
 summarize(count = n()) %>%  # count records by species
 mutate(pct = count/sum(count))
ggplot(germanNew3, aes(CustomerCredit, pct, fill = CustomerCredit)) +
  geom_bar(stat='identity') +
  geom_text(aes(label=scales::percent(pct)), position = position_stack(vjust = .5))+
  scale_y_continuous(labels = scales::percent) + labs(title = "Testing Data")
```



Here we used the stratified random sampling based on the reponse/target variable The histograms shows the same distribution of good and bad for the test and train dataset

Logistic Regression

Building basic models for binary classification/prediction

```
# Basic logistic regression model
model1 <- glm(CustomerCredit ~ ., family = binomial, train)
model1
```

```
##
## Call:  glm(formula = CustomerCredit ~ ., family = binomial, data = train)
##
## Coefficients:
##      (Intercept)       chk_acctA12       chk_acctA13       chk_acctA14
##        0.8350089        -0.3428658        -1.1524184        -1.6836104
##         duration      credit_hisA31     credit_hisA32     credit_hisA33
##        0.0354387        -0.1676064        -0.7207542        -0.9380245
##    credit_hisA34        purposeA41       purposeA410        purposeA42
##       -1.4946956        -1.4758022        -2.5975762        -0.7272795
##       purposeA43        purposeA44        purposeA45        purposeA46
##       -0.8615727        -2.1178130         0.6800610         0.1234344
##       purposeA48        purposeA49            amount     saving_acctA62
```

```
##        -1.7900136         -0.4886075          0.0001057         -0.3092232
##    saving_acctA63     saving_acctA64     saving_acctA65     present_empA72
##        -0.2710700         -1.6881466         -0.9253318          0.2728161
##    present_empA73     present_empA74     present_empA75   installment_rate
##         0.1043439         -0.6454993          0.0635785          0.1484235
##            sexA92             sexA93             sexA94   other_debtorA102
##         0.0181399         -0.3809067          0.0270642          0.8825550
##  other_debtorA103      present_resid       propertyA122       propertyA123
##        -1.0285213          0.0264030          0.1873464          0.2845013
##      propertyA124                age  other_installA142  other_installA143
##         0.4253772         -0.0022446         -0.4271456         -0.9200867
##       housingA152        housingA153          n_credits            jobA172
##        -0.4916757         -0.6548619          0.0904883          0.1388349
##           jobA173            jobA174           n_people      telephoneA192
##        -0.0169445          0.0818012          0.2234447         -0.1979044
##       foreignA202
##        -1.9280987
##
## Degrees of Freedom: 699 Total (i.e. Null);  651 Residual
## Null Deviance:       855.2
## Residual Deviance: 627.2     AIC: 725.2
```

AIC is a goodness of fit measure that favours smaller residual error in the model, but penalises for including further predictors and helps avoiding overfitting.

```r
#variable selection using AIC (both,backward,forward). Here we are using both

StepModel1 <- step(model1, direction = "both")
```

```
## Start:  AIC=725.19
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + sex +
##     other_debtor + present_resid + property + age + other_install +
##     housing + n_credits + job + n_people + telephone + foreign
##
##                    Df Deviance    AIC
## - job               3   627.57 719.57
## - property          3   628.40 720.40
## - sex               3   630.09 722.09
## - age               1   627.23 723.23
## - present_resid     1   627.25 723.25
## - n_credits         1   627.37 723.37
## - n_people          1   627.73 723.73
## - telephone         1   627.86 723.86
## - present_emp       4   634.33 724.33
## - housing           2   630.58 724.58
## - installment_rate  1   629.15 725.15
## <none>                  627.19 725.19
## - amount            1   631.03 727.03
## - other_debtor      2   634.87 728.87
## - credit_his        4   640.42 730.42
## - foreign           1   635.01 731.01
## - saving_acct       4   641.90 731.90
```

```
## - other_install     2   638.23 732.23
## - duration          1   636.73 732.73
## - purpose           9   657.42 737.42
## - chk_acct          3   673.04 765.04
##
## Step:  AIC=719.57
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + sex +
##     other_debtor + present_resid + property + age + other_install +
##     housing + n_credits + n_people + telephone + foreign
##
##                    Df Deviance    AIC
## - property          3   628.72 714.72
## - sex               3   630.47 716.47
## - age               1   627.59 717.59
## - present_resid     1   627.64 717.64
## - n_credits         1   627.72 717.72
## - n_people          1   628.19 718.19
## - telephone         1   628.41 718.41
## - housing           2   631.00 719.00
## - present_emp       4   635.01 719.01
## <none>                  627.57 719.57
## - installment_rate  1   629.63 719.63
## - amount            1   631.91 721.91
## - other_debtor      2   635.24 723.24
## - credit_his        4   640.72 724.72
## - foreign           1   635.17 725.17
## + job               3   627.19 725.19
## - other_install     2   638.79 726.79
## - duration          1   636.89 726.89
## - saving_acct       4   643.00 727.00
## - purpose           9   657.97 731.97
## - chk_acct          3   673.81 759.81
##
## Step:  AIC=714.72
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + sex +
##     other_debtor + present_resid + age + other_install + housing +
##     n_credits + n_people + telephone + foreign
##
##                    Df Deviance    AIC
## - sex               3   631.61 711.61
## - age               1   628.75 712.75
## - present_resid     1   628.78 712.78
## - n_credits         1   628.85 712.85
## - telephone         1   629.31 713.31
## - n_people          1   629.33 713.33
## - housing           2   631.88 713.88
## - present_emp       4   636.23 714.23
## <none>                  628.72 714.72
## - installment_rate  1   630.99 714.99
## - amount            1   633.67 717.67
## - other_debtor      2   637.22 719.22
## + property          3   627.57 719.57
```

```
## - credit_his        4   642.25 720.25
## + job               3   628.40 720.40
## - foreign           1   636.47 720.47
## - saving_acct       4   644.36 722.36
## - duration          1   639.01 723.01
## - other_install     2   641.01 723.01
## - purpose           9   659.68 727.68
## - chk_acct          3   675.70 755.70
##
## Step:  AIC=711.61
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     present_resid + age + other_install + housing + n_credits +
##     n_people + telephone + foreign
##
##                    Df Deviance    AIC
## - age               1   631.67 709.67
## - present_resid     1   631.68 709.68
## - n_credits         1   631.70 709.70
## - n_people          1   631.72 709.72
## - telephone         1   632.03 710.03
## - installment_rate  1   633.14 711.14
## <none>                  631.61 711.61
## - housing           2   635.95 711.95
## - present_emp       4   640.62 712.62
## - amount            1   635.66 713.66
## + sex               3   628.72 714.72
## - other_debtor      2   639.99 715.99
## + property          3   630.47 716.47
## + job               3   631.27 717.27
## - credit_his        4   645.54 717.54
## - foreign           1   639.91 717.91
## - saving_acct       4   647.00 719.00
## - other_install     2   643.39 719.39
## - duration          1   641.60 719.60
## - purpose           9   662.32 724.32
## - chk_acct          3   678.66 752.66
##
## Step:  AIC=709.67
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     present_resid + other_install + housing + n_credits + n_people +
##     telephone + foreign
##
##                    Df Deviance    AIC
## - present_resid     1   631.72 707.72
## - n_credits         1   631.75 707.75
## - n_people          1   631.77 707.77
## - telephone         1   632.10 708.10
## - installment_rate  1   633.19 709.19
## <none>                  631.67 709.67
## - housing           2   636.36 710.36
## - present_emp       4   640.69 710.69
## + age               1   631.61 711.61
```

```
## - amount           1   635.69 711.69
## + sex              3   628.75 712.75
## - other_debtor     2   640.05 714.05
## + property         3   630.51 714.51
## + job              3   631.36 715.36
## - credit_his       4   645.66 715.66
## - foreign          1   639.97 715.97
## - saving_acct      4   647.11 717.11
## - other_install    2   643.45 717.45
## - duration         1   641.97 717.97
## - purpose          9   662.36 722.36
## - chk_acct         3   679.34 751.34
##
## Step:  AIC=707.72
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     other_install + housing + n_credits + n_people + telephone +
##     foreign
##
##                   Df Deviance    AIC
## - n_credits        1   631.82 705.82
## - n_people         1   631.83 705.83
## - telephone        1   632.15 706.15
## - installment_rate 1   633.28 707.28
## <none>                 631.72 707.72
## - present_emp      4   640.70 708.70
## - housing          2   636.80 708.80
## + present_resid    1   631.67 709.67
## + age              1   631.68 709.68
## - amount           1   635.71 709.71
## + sex              3   628.80 710.80
## - other_debtor     2   640.13 712.13
## + property         3   630.58 712.58
## + job              3   631.42 713.42
## - credit_his       4   645.66 713.66
## - foreign          1   640.07 714.07
## - saving_acct      4   647.13 715.13
## - other_install    2   643.47 715.47
## - duration         1   642.07 716.07
## - purpose          9   662.43 720.43
## - chk_acct         3   679.38 749.38
##
## Step:  AIC=705.82
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     other_install + housing + n_people + telephone + foreign
##
##                   Df Deviance    AIC
## - n_people         1   631.95 703.95
## - telephone        1   632.21 704.21
## - installment_rate 1   633.35 705.35
## <none>                 631.82 705.82
## - present_emp      4   640.71 706.71
## - housing          2   636.98 706.98
```

```
## + n_credits         1   631.72 707.72
## + present_resid     1   631.75 707.75
## + age               1   631.78 707.78
## - amount            1   635.89 707.89
## + sex               3   628.93 708.93
## - other_debtor      2   640.26 710.26
## + property          3   630.68 710.68
## + job               3   631.53 711.53
## - foreign           1   640.28 712.28
## - credit_his        4   647.01 713.01
## - saving_acct       4   647.40 713.40
## - other_install     2   643.84 713.84
## - duration          1   642.08 714.08
## - purpose           9   662.89 718.89
## - chk_acct          3   679.39 747.39
##
## Step:  AIC=703.95
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     other_install + housing + telephone + foreign
##
##                    Df Deviance    AIC
## - telephone         1   632.35 702.35
## - installment_rate  1   633.43 703.43
## <none>                  631.95 703.95
## - present_emp       4   640.73 704.73
## - housing           2   637.01 705.01
## + n_people          1   631.82 705.82
## + n_credits         1   631.83 705.83
## + present_resid     1   631.87 705.87
## + age               1   631.92 705.92
## - amount            1   636.00 706.00
## + sex               3   629.60 707.60
## - other_debtor      2   640.40 708.40
## + property          3   630.82 708.82
## + job               3   631.64 709.64
## - foreign           1   640.34 710.34
## - credit_his        4   647.32 711.32
## - saving_acct       4   647.46 711.46
## - duration          1   642.18 712.18
## - other_install     2   644.27 712.27
## - purpose           9   663.18 717.18
## - chk_acct          3   679.47 745.47
##
## Step:  AIC=702.35
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + other_debtor +
##     other_install + housing + foreign
##
##                    Df Deviance    AIC
## - installment_rate  1   633.78 701.78
## <none>                  632.35 702.35
## - housing           2   637.43 703.43
## - present_emp       4   641.79 703.79
```

```
## + telephone         1   631.95 703.95
## - amount            1   636.08 704.08
## + n_people          1   632.21 704.21
## + n_credits         1   632.27 704.27
## + present_resid     1   632.29 704.29
## + age               1   632.31 704.31
## + sex               3   630.16 706.16
## - other_debtor      2   640.78 706.78
## + property          3   631.42 707.42
## + job               3   632.03 708.03
## - foreign           1   640.55 708.55
## - credit_his        4   648.03 710.03
## - saving_acct       4   648.03 710.03
## - duration          1   642.57 710.57
## - other_install     2   644.60 710.60
## - purpose           9   663.43 715.43
## - chk_acct          3   680.75 744.75
##
## Step:  AIC=701.78
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + other_debtor + other_install +
##     housing + foreign
##
##                    Df Deviance    AIC
## <none>                 633.78 701.78
## - amount            1   636.30 702.30
## + installment_rate  1   632.35 702.35
## - housing           2   638.57 702.57
## - present_emp       4   643.29 703.29
## + telephone         1   633.43 703.43
## + present_resid     1   633.67 703.67
## + n_people          1   633.69 703.69
## + n_credits         1   633.72 703.72
## + age               1   633.75 703.75
## + sex               3   632.10 706.10
## - other_debtor      2   642.56 706.56
## + property          3   632.67 706.67
## + job               3   633.40 707.40
## - foreign           1   642.02 708.02
## - credit_his        4   649.02 709.02
## - saving_acct       4   649.27 709.27
## - other_install     2   645.88 709.88
## - duration          1   647.55 713.55
## - purpose           9   664.77 714.77
## - chk_acct          3   681.81 743.81
```

```r
summary(StepModel1)
```

```
##
## Call:
## glm(formula = CustomerCredit ~ chk_acct + duration + credit_his +
##     purpose + amount + saving_acct + present_emp + other_debtor +
##     other_install + housing + foreign, family = binomial, data = train)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2479  -0.7196  -0.3885   0.7435   2.6404
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.734e+00  7.668e-01   2.261 0.023771 *
## chk_acctA12       -3.639e-01  2.543e-01  -1.431 0.152321
## chk_acctA13       -1.224e+00  4.302e-01  -2.844 0.004456 **
## chk_acctA14       -1.689e+00  2.748e-01  -6.144 8.03e-10 ***
## duration           3.935e-02  1.072e-02   3.672 0.000241 ***
## credit_hisA31     -1.995e-01  6.725e-01  -0.297 0.766774
## credit_hisA32     -7.481e-01  5.222e-01  -1.433 0.151995
## credit_hisA33     -9.995e-01  5.791e-01  -1.726 0.084329 .
## credit_hisA34     -1.488e+00  5.452e-01  -2.730 0.006340 **
## purposeA41        -1.425e+00  4.147e-01  -3.437 0.000588 ***
## purposeA410       -2.598e+00  1.006e+00  -2.582 0.009830 **
## purposeA42        -7.003e-01  3.108e-01  -2.253 0.024245 *
## purposeA43        -7.958e-01  2.851e-01  -2.791 0.005253 **
## purposeA44        -2.050e+00  1.291e+00  -1.588 0.112270
## purposeA45         7.802e-01  6.857e-01   1.138 0.255172
## purposeA46         2.396e-01  4.602e-01   0.521 0.602644
## purposeA48        -1.350e+00  1.278e+00  -1.057 0.290719
## purposeA49        -4.518e-01  3.907e-01  -1.156 0.247514
## amount             7.316e-05  4.637e-05   1.578 0.114599
## saving_acctA62    -2.573e-01  3.394e-01  -0.758 0.448466
## saving_acctA63    -3.633e-01  4.219e-01  -0.861 0.389212
## saving_acctA64    -1.634e+00  7.097e-01  -2.303 0.021289 *
## saving_acctA65    -9.476e-01  3.112e-01  -3.044 0.002331 **
## present_empA72     3.346e-01  4.540e-01   0.737 0.461179
## present_empA73     7.716e-02  4.333e-01   0.178 0.858660
## present_empA74    -7.161e-01  4.886e-01  -1.466 0.142744
## present_empA75     5.118e-02  4.438e-01   0.115 0.908189
## other_debtorA102   9.112e-01  5.137e-01   1.774 0.076077 .
## other_debtorA103  -1.073e+00  5.028e-01  -2.135 0.032774 *
## other_installA142 -4.947e-01  4.704e-01  -1.052 0.292950
## other_installA143 -9.468e-01  2.758e-01  -3.433 0.000598 ***
## housingA152       -5.639e-01  2.604e-01  -2.166 0.030318 *
## housingA153       -5.469e-01  3.871e-01  -1.413 0.157707
## foreignA202       -1.936e+00  8.025e-01  -2.412 0.015860 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 855.21  on 699  degrees of freedom
## Residual deviance: 633.78  on 666  degrees of freedom
## AIC: 701.78
##
## Number of Fisher Scoring iterations: 5
```

```
#chi-square test for significance of variables
```

```r
# Chi square test on our initial model
drop1(model1, test ="Chi")
```

```
## Single term deletions
##
## Model:
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + installment_rate + sex +
##     other_debtor + present_resid + property + age + other_install +
##     housing + n_credits + job + n_people + telephone + foreign
##                  Df Deviance    AIC    LRT Pr(>Chi)
## <none>                627.19 725.19
## chk_acct          3   673.04 765.04 45.848 6.11e-10 ***
## duration          1   636.73 732.73  9.546 0.002004 **
## credit_his        4   640.42 730.42 13.235 0.010181 *
## purpose           9   657.42 737.42 30.229 0.000401 ***
## amount            1   631.03 727.03  3.844 0.049913 *
## saving_acct       4   641.90 731.90 14.708 0.005347 **
## present_emp       4   634.33 724.33  7.138 0.128777
## installment_rate  1   629.15 725.15  1.962 0.161252
## sex               3   630.09 722.09  2.898 0.407638
## other_debtor      2   634.87 728.87  7.680 0.021494 *
## present_resid     1   627.25 723.25  0.065 0.798558
## property          3   628.40 720.40  1.214 0.749765
## age               1   627.23 723.23  0.044 0.833402
## other_install     2   638.23 732.23 11.045 0.003996 **
## housing           2   630.58 724.58  3.389 0.183713
## n_credits         1   627.37 723.37  0.182 0.669912
## job               3   627.57 719.57  0.381 0.944187
## n_people          1   627.73 723.73  0.544 0.460586
## telephone         1   627.86 723.86  0.675 0.411241
## foreign           1   635.01 731.01  7.826 0.005151 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Chi Square test on model after AIC selection
drop1(StepModel1,test="Chi")
```

```
## Single term deletions
##
## Model:
## CustomerCredit ~ chk_acct + duration + credit_his + purpose +
##     amount + saving_acct + present_emp + other_debtor + other_install +
##     housing + foreign
##               Df Deviance    AIC    LRT  Pr(>Chi)
## <none>             633.78 701.78
## chk_acct       3   681.81 743.81 48.034 2.094e-10 ***
## duration       1   647.55 713.55 13.772 0.0002064 ***
## credit_his     4   649.02 709.02 15.244 0.0042205 **
## purpose        9   664.77 714.77 30.992 0.0002970 ***
## amount         1   636.30 702.30  2.519 0.1124987
## saving_acct    4   649.27 709.27 15.491 0.0037843 **
```

```
## present_emp    4   643.29 703.29  9.515 0.0494311 *
## other_debtor   2   642.56 706.56  8.785 0.0123676 *
## other_install  2   645.88 709.88 12.107 0.0023500 **
## housing        2   638.57 702.57  4.789 0.0912029 .
## foreign        1   642.02 708.02  8.245 0.0040860 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the AIC selection has provided us with the variable that has the most impact on our target variable.

using significant variables based on AIC selection

```r
# logistic regression
Model2<- glm(formula = CustomerCredit ~ chk_acct + duration + credit_his + purpose +
               amount + saving_acct + present_emp + other_debtor + other_install +
               housing + foreign, family = binomial, data = train)
```

Next we are going to use our testing data on our model to see how well the model performs on unseen data

```r
pred <- predict(Model2, type = "response",newdata = test)
y_act <- test$CustomerCredit
```

#We can customise the probability rate at which the model determines if the outcome is good or bad. #In this case we are setting the probability to greater than 0.5. This means that if probabily is >0.5 then bad else good.

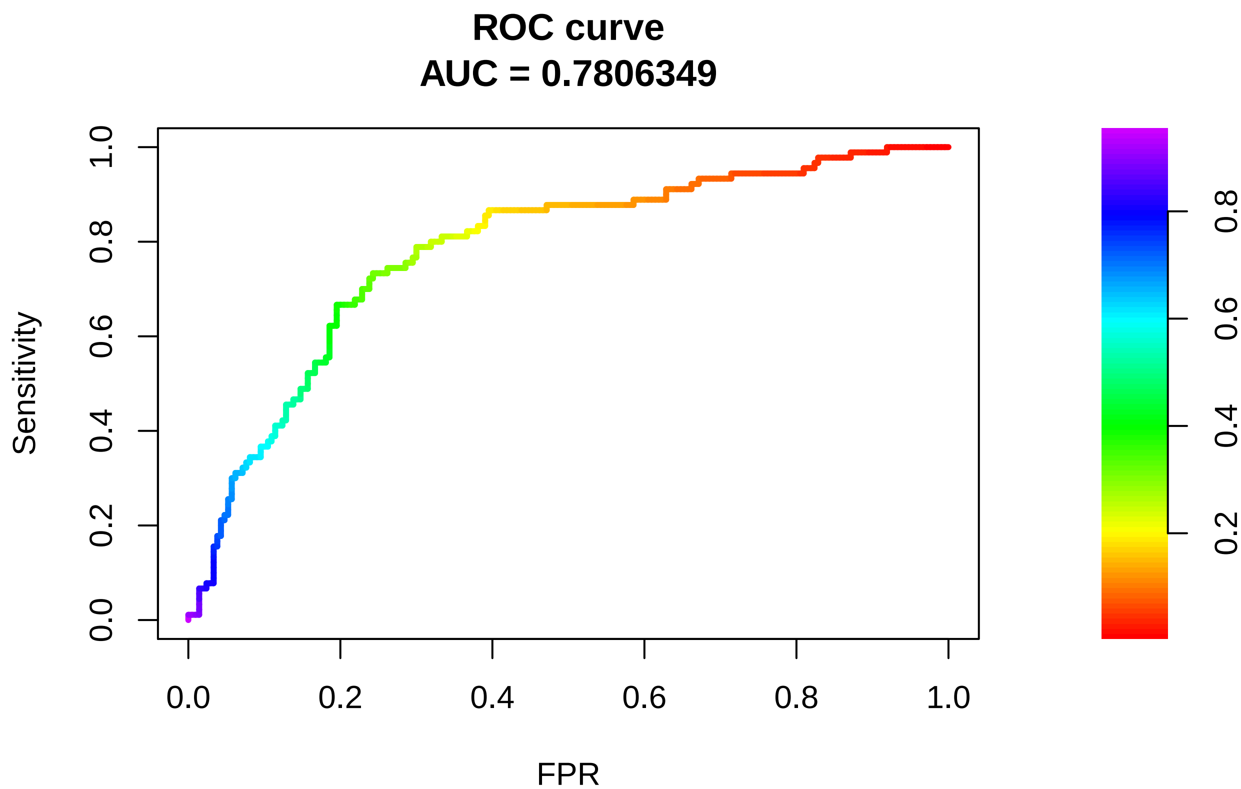
```r
pred1<- ifelse(pred > 0.5,2,1)
```

Confusion matrix

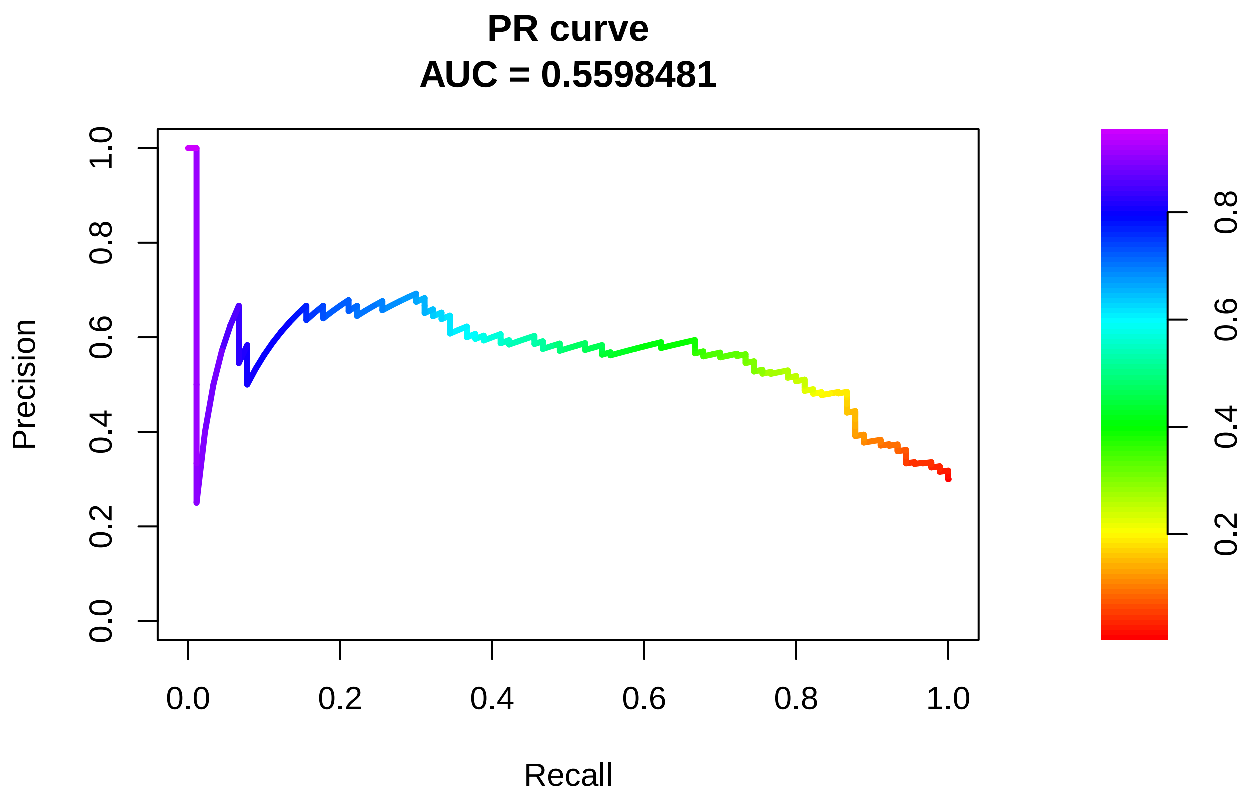
confusionMatrix(table(pred1,y_act), positive='2') confusionMatrix(table(pred1,y_act), positive='2',mode = "prec_recall")

```
In situations were data is imbalanced, comparing accuracy is not ideal. Imagine the model predicts all c

So how do you determine if our model is good in such cases?
We can use metrics such as specificity,sensitivity,precision,recall and F-measure.

Sensitivity: metric that evaluates a model's ability to predict true positives of each available catego
Specificity: metric that evaluates a model's ability to predict true negatives of each available catego
Recall: measure of completeness; the proportion of positive class examples that are classified correctl
Precision: measure of exactness, the proportion of positive class examples that are classified correctl
F-measure: incorporates both recall and precision to express the trade-off between them.

## ROC & PR curve (Logistic Regression)
```

```r
library(PRROC)
fg1 <- pred[test$CustomerCredit == 2]
bg1 <- pred[test$CustomerCredit == 1]

# ROC Curve
roc1 <- PRROC::roc.curve(scores.class0 = fg1, scores.class1 = bg1, curve = T)
plot(roc1)
```




```r
# PR CUrve
pr <- pr.curve(scores.class0 = fg1, scores.class1 = bg1, curve = T)
plot(pr)
```



Decision Tree

```
library(rpart)
m1 <- rpart(CustomerCredit~.,
            method="class", data=train)

pdata <- as.data.frame(predict(m1, newdata = test, type = "p"))
pdata$my_custom_predicted_class <- ifelse(pdata$`2`> .5, 2,1)
pdata$my_custom_predicted_class<-factor(pdata$my_custom_predicted_class)
test$CustomerCredit<-factor(test$CustomerCredit)
# confusion matrix
caret::confusionMatrix(data = pdata$my_custom_predicted_class,
                       reference = test$CustomerCredit, positive = "2")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 169  42
##          2  41  48
##
##                Accuracy : 0.7233
##                  95% CI : (0.669, 0.7732)
##     No Information Rate : 0.7
```

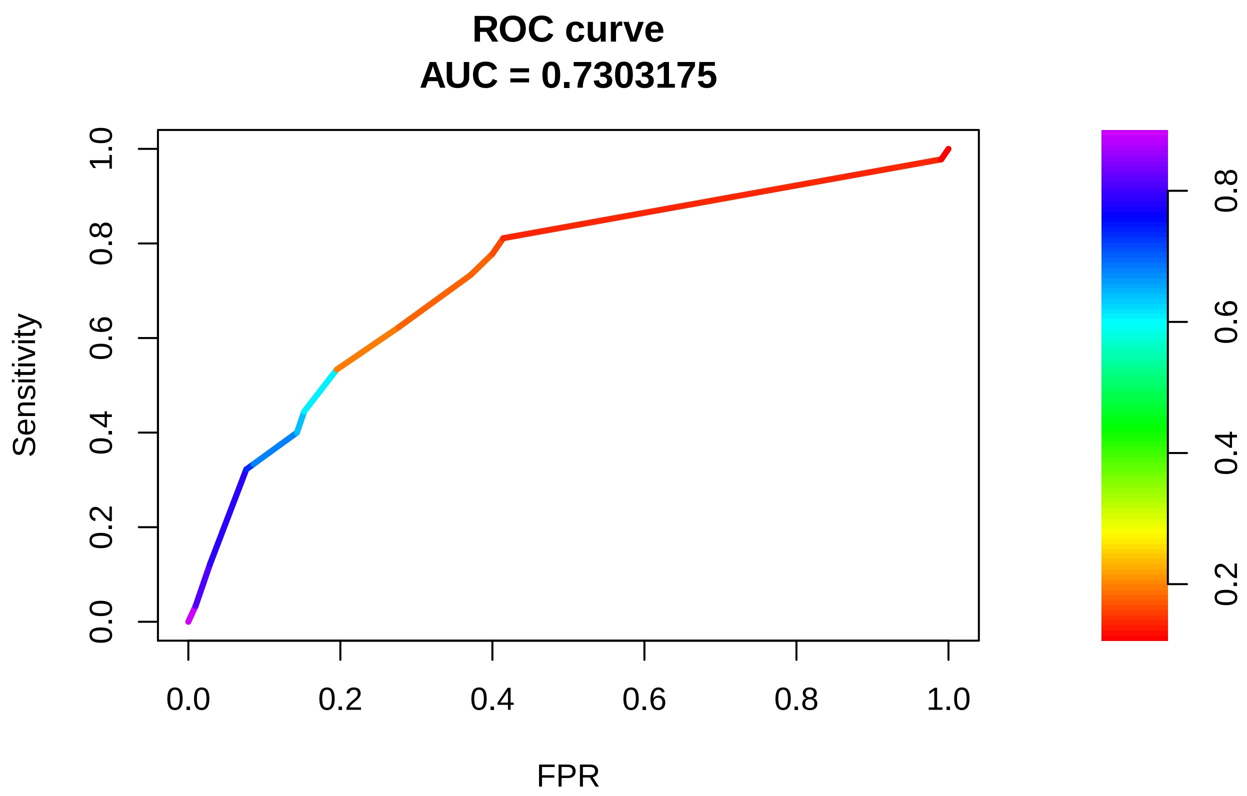
```
##     P-Value [Acc > NIR] : 0.2072
##
##                   Kappa : 0.3392
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.5333
##             Specificity : 0.8048
##          Pos Pred Value : 0.5393
##          Neg Pred Value : 0.8009
##              Prevalence : 0.3000
##          Detection Rate : 0.1600
##    Detection Prevalence : 0.2967
##       Balanced Accuracy : 0.6690
##
##        'Positive' Class : 2
##
```

ROC & PR curve (Decision Tree)

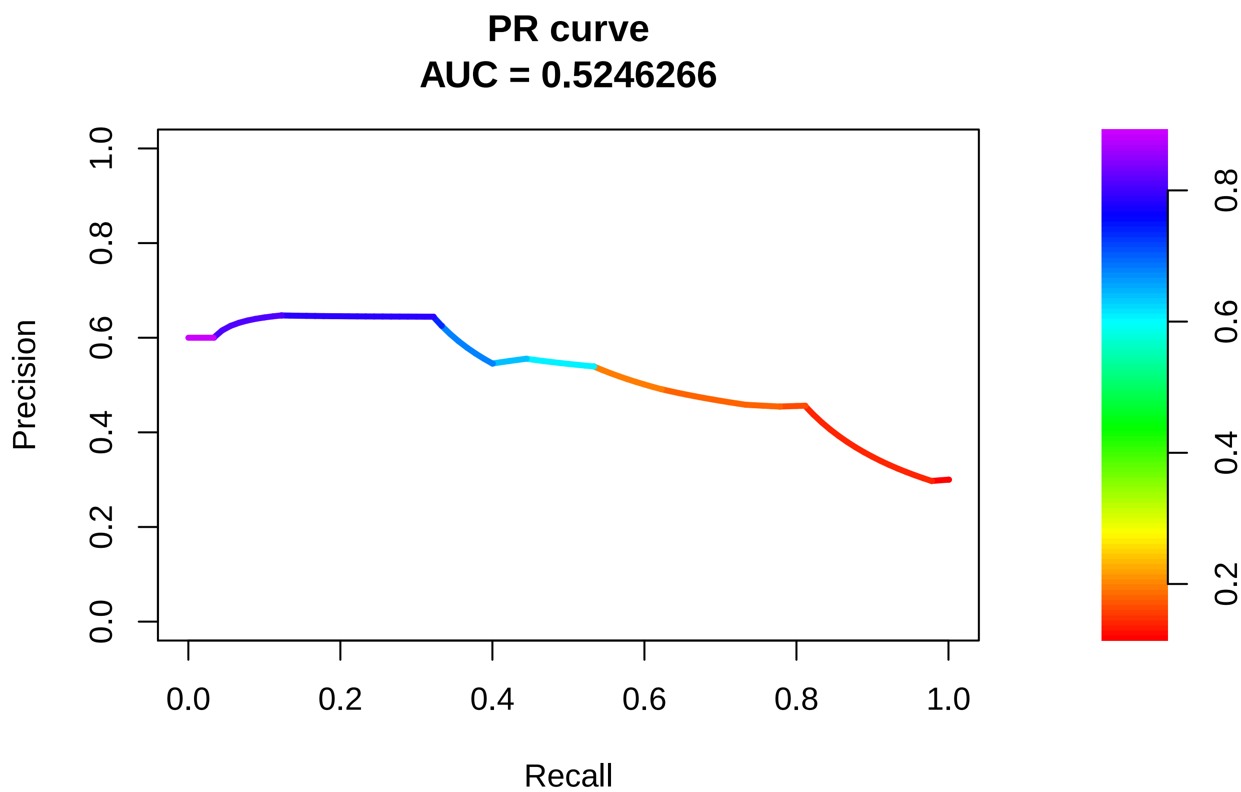
```r
caret::confusionMatrix(data = pdata$my_custom_predicted_class,
                       reference = test$CustomerCredit, positive = "2",mode="prec_recall")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2
##          1 169  42
##          2  41  48
##
##                Accuracy : 0.7233
##                  95% CI : (0.669, 0.7732)
##     No Information Rate : 0.7
##     P-Value [Acc > NIR] : 0.2072
##
##                   Kappa : 0.3392
##
##  Mcnemar's Test P-Value : 1.0000
##
##               Precision : 0.5393
##                  Recall : 0.5333
##                      F1 : 0.5363
##              Prevalence : 0.3000
##          Detection Rate : 0.1600
##    Detection Prevalence : 0.2967
##       Balanced Accuracy : 0.6690
##
##        'Positive' Class : 2
##
```

```r
fg2 <- pdata$`2`[test$CustomerCredit == 2]
bg2 <- pdata$`2`[test$CustomerCredit == 1]

roc2 <- PRROC::roc.curve(scores.class0 = fg2,
                         scores.class1 = bg2, curve = T)
plot(roc2)
```

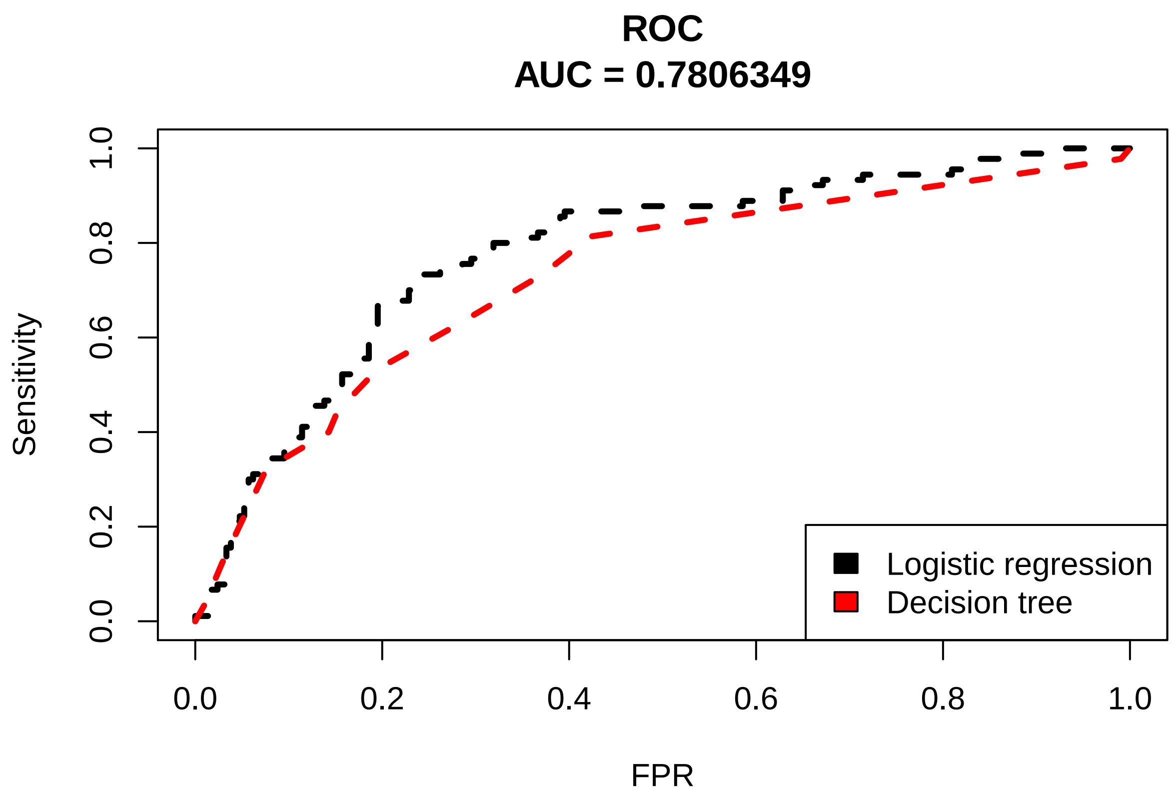



```r
pr2 <- pr.curve(scores.class0 = fg2, scores.class1 = bg2, curve = T)
plot(pr2)
```

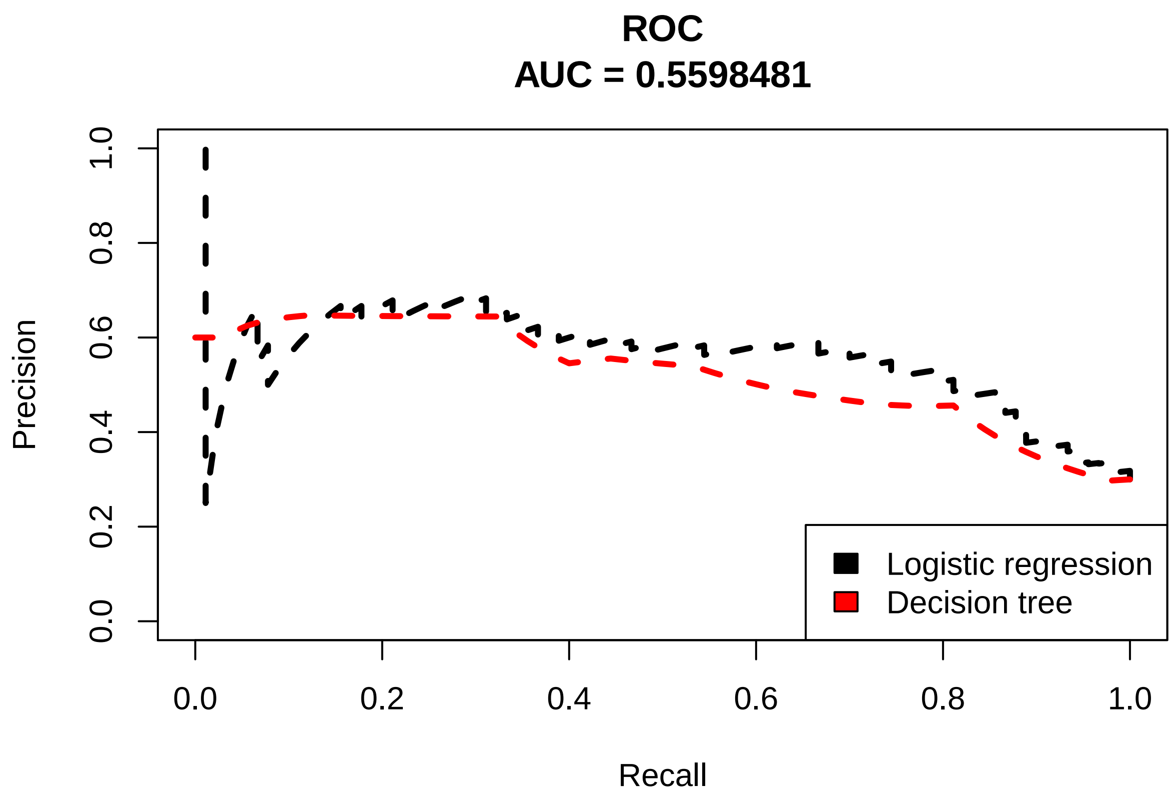


Model comparison

```
plot(roc1,col=1,lty=2,main="ROC")
plot(roc2,col=2,lty=2,add=TRUE)


legend(x="bottomright",
       legend= c("Logistic regression",
                 "Decision tree"),
       fill = 1:5)
```




```
plot(pr,col=1,lty=2,main="ROC")
plot(pr2,col=2,lty=2,add=TRUE)


legend(x="bottomright",
       legend= c("Logistic regression",
                 "Decision tree"),
       fill = 1:5)
```



To understand how to analyse ROC/PR curve, please refer to the PDF on “Machine learning Ensemble”