Assignment 4, Part I Work

Matthew Dunne

August 1, 2018

Data

First we load the data, convert the necessary variables into factors and then split into train and test data.

```
setwd("C:/Users/mjdun/Desktop/Data Mining/Assignments")
#using csv which has it in factor variables
MyData <- read.csv(file="German.Credit.csv", header=TRUE, sep=",")
columns<-c(1,2,4,5,7,8,9,10,11,12,13,15,16,17,18,19,20,21)
MyData[columns] <- lapply(MyData[columns], factor)
```

Split into train and test.

```
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
#split data into train and test
set.seed(1234)
s1<-sample(1:nrow(MyData), nrow(MyData)*.7, replace=FALSE)
```

Choosing Variables to Include in the Logistic Regression Model

To decide which variables we will include in the logistic regression let us first include all variables in a logistic regression and see what is statistically significant. Below are the p-values for the coefficient of each variable for a logistic regression model including all variables.

```
model0<-glm(Creditability~., data=MyData[s1, ], family=binomial(link = logit))
summary(model0)$coefficients
```

```
##                                        Estimate   Std. Error     z value
## (Intercept)                       -0.4923381203 1.236659e+00 -0.39811955
## Account.Balance2                   0.3475991474 2.718258e-01  1.27875710
## Account.Balance3                   0.7324247325 4.526348e-01  1.61813621
## Account.Balance4                   1.7100419036 2.919057e-01  5.85819955
## Duration.of.Credit..month.        -0.0286658497 1.168159e-02 -2.45393416
## Payment.Status.of.Previous.Credit1 -0.4484453208 7.244021e-01 -0.61905576
## Payment.Status.of.Previous.Credit2  0.6813212925 5.796628e-01  1.17537524
## Payment.Status.of.Previous.Credit3  1.3724201102 6.287002e-01  2.18294825
## Payment.Status.of.Previous.Credit4  1.7575804469 5.895897e-01  2.98102308
## Purpose1                           1.8958184635 4.722633e-01  4.01432508
## Purpose2                           0.7840464570 3.297886e-01  2.37742127
## Purpose3                           1.1211469592 3.148298e-01  3.56112040
## Purpose4                           0.5294219266 8.618837e-01  0.61426145
## Purpose5                           0.7956333517 7.143395e-01  1.11380287
## Purpose6                           0.0385157130 4.675250e-01  0.08238215
## Purpose8                          16.3862378929 7.505877e+02  0.02183121
## Purpose9                           0.7683923575 4.126132e-01  1.86225848
```

```
## Purpose10                          2.3370903511 1.099594e+00  2.12541159
## Credit.Amount                     -0.0001976897 5.735769e-05 -3.44661205
## Value.Savings.Stocks2              0.2539912163 3.466295e-01  0.73274546
## Value.Savings.Stocks3             -0.0817713893 4.598831e-01 -0.17780907
## Value.Savings.Stocks4              1.3366355610 5.968950e-01  2.23931439
## Value.Savings.Stocks5              1.1431162998 3.371531e-01  3.39049570
## Length.of.current.employment2      0.2847203234 5.306883e-01  0.53651138
## Length.of.current.employment3      0.4374456647 5.139366e-01  0.85116666
## Length.of.current.employment4      1.2459776985 5.692823e-01  2.18868165
## Length.of.current.employment5      0.4348321171 5.228494e-01  0.83165851
## Instalment.per.cent2              -0.3011484007 3.917663e-01 -0.76869399
## Instalment.per.cent3              -1.0257724280 4.182136e-01 -2.45274757
## Instalment.per.cent4              -1.2852826419 3.806364e-01 -3.37666744
## Sex...Marital.Status2              0.1834175423 4.798710e-01  0.38222260
## Sex...Marital.Status3              0.9111270237 4.682451e-01  1.94583368
## Sex...Marital.Status4              0.4203989211 5.699040e-01  0.73766625
## Guarantors2                       -0.3808092275 4.945468e-01 -0.77001663
## Guarantors3                        1.0966546134 5.518888e-01  1.98709350
## Duration.in.Current.address2      -0.9806385674 3.740466e-01 -2.62170147
## Duration.in.Current.address3      -1.0033398989 4.146237e-01 -2.41988073
## Duration.in.Current.address4      -0.7230324681 3.881329e-01 -1.86284781
## Most.valuable.available.asset2    -0.0478166257 3.199513e-01 -0.14944971
## Most.valuable.available.asset3    -0.1984139553 2.940753e-01 -0.67470460
## Most.valuable.available.asset4    -0.5873723085 5.234154e-01 -1.12219157
## Age..years.                        0.0149723280 1.144347e-02  1.30837252
## Concurrent.Credits2               -0.0534930441 5.577382e-01 -0.09591068
## Concurrent.Credits3                0.4531192900 3.075113e-01  1.47350467
## Type.of.apartment2                 0.2666034365 2.940825e-01  0.90656006
## Type.of.apartment3                 0.5628036924 6.022666e-01  0.93447595
## No.of.Credits.at.this.Bank2       -0.5990225898 3.116426e-01 -1.92214593
## No.of.Credits.at.this.Bank3        0.2501294386 7.822432e-01  0.31975919
## No.of.Credits.at.this.Bank4        0.1091059524 1.460465e+00  0.07470630
## Occupation2                       -0.5142689030 8.235319e-01 -0.62446753
## Occupation3                       -0.2502486024 7.916950e-01 -0.31609219
## Occupation4                       -0.0186017369 8.104251e-01 -0.02295306
## No.of.dependents2                 -0.3441576083 3.106146e-01 -1.10798929
## Telephone2                         0.2520320524 2.582512e-01  0.97591826
## Foreign.Worker2                    0.9063905192 7.778933e-01  1.16518624
##                                       Pr(>|z|)
## (Intercept)                       6.905421e-01
## Account.Balance2                  2.009826e-01
## Account.Balance3                  1.056332e-01
## Account.Balance4                  4.679122e-09
## Duration.of.Credit..month.        1.413028e-02
## Payment.Status.of.Previous.Credit1 5.358796e-01
## Payment.Status.of.Previous.Credit2 2.398446e-01
## Payment.Status.of.Previous.Credit3 2.903962e-02
## Payment.Status.of.Previous.Credit4 2.872871e-03
## Purpose1                          5.961614e-05
## Purpose2                          1.743416e-02
## Purpose3                          3.692757e-04
## Purpose4                          5.390426e-01
## Purpose5                          2.653638e-01
## Purpose6                          9.343428e-01
```

```
## Purpose8                          9.825826e-01
## Purpose9                          6.256666e-02
## Purpose10                         3.355228e-02
## Credit.Amount                     5.676631e-04
## Value.Savings.Stocks2             4.637137e-01
## Value.Savings.Stocks3             8.588729e-01
## Value.Savings.Stocks4             2.513547e-02
## Value.Savings.Stocks5             6.976635e-04
## Length.of.current.employment2     5.916052e-01
## Length.of.current.employment3     3.946768e-01
## Length.of.current.employment4     2.861999e-02
## Length.of.current.employment5     4.056017e-01
## Instalment.per.cent2              4.420750e-01
## Instalment.per.cent3              1.417698e-02
## Instalment.per.cent4              7.336970e-04
## Sex...Marital.Status2             7.022963e-01
## Sex...Marital.Status3             5.167472e-02
## Sex...Marital.Status4             4.607173e-01
## Guarantors2                       4.412900e-01
## Guarantors3                       4.691204e-02
## Duration.in.Current.address2      8.749203e-03
## Duration.in.Current.address3      1.552560e-02
## Duration.in.Current.address4      6.248368e-02
## Most.valuable.available.asset2    8.811988e-01
## Most.valuable.available.asset3    4.998635e-01
## Most.valuable.available.asset4    2.617810e-01
## Age..years.                       1.907470e-01
## Concurrent.Credits2               9.235915e-01
## Concurrent.Credits3               1.406150e-01
## Type.of.apartment2                3.646395e-01
## Type.of.apartment3                3.500584e-01
## No.of.Credits.at.this.Bank2       5.458740e-02
## No.of.Credits.at.this.Bank3       7.491509e-01
## No.of.Credits.at.this.Bank4       9.404484e-01
## Occupation2                       5.323206e-01
## Occupation3                       7.519325e-01
## Occupation4                       9.816877e-01
## No.of.dependents2                 2.678664e-01
## Telephone2                        3.291050e-01
## Foreign.Worker2                   2.439436e-01
```

Some of these variables are significant and some are not.

Now we will use the step() function to determine if we should use a subset of these variables. We use the parameter trace=0 so as only to return the formula with the lowest AIC.

```
step(model1<-glm(Creditability~., data=MyData[s1, ], family=binomial(link = logit)), direction = "both"
```

```
##
## Call:  glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
##     Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
##     Value.Savings.Stocks + Instalment.per.cent + Sex...Marital.Status +
##     Guarantors + Duration.in.Current.address + Telephone, family = binomial(link = logit),
##     data = MyData[s1, ])
##
## Coefficients:
```

```
##                         (Intercept)                    Account.Balance2
##                           0.3783202                           0.3306060
##                    Account.Balance3                    Account.Balance4
##                           0.8418371                           1.6802727
##             Duration.of.Credit..month.  Payment.Status.of.Previous.Credit1
##                          -0.0278439                          -0.4366447
## Payment.Status.of.Previous.Credit2  Payment.Status.of.Previous.Credit3
##                           0.9069549                           1.2007277
## Payment.Status.of.Previous.Credit4                            Purpose1
##                           1.6843430                           1.7131507
##                            Purpose2                            Purpose3
##                           0.6788654                           1.0564755
##                            Purpose4                            Purpose5
##                           0.4784419                           0.6922853
##                            Purpose6                            Purpose8
##                          -0.0217938                          15.1382158
##                            Purpose9                           Purpose10
##                           0.6083848                           1.7158640
##                       Credit.Amount               Value.Savings.Stocks2
##                          -0.0001835                           0.2139996
##               Value.Savings.Stocks3               Value.Savings.Stocks4
##                           0.0378286                           1.2086323
##               Value.Savings.Stocks5                Instalment.per.cent2
##                           1.1716854                          -0.1779563
##                Instalment.per.cent3                Instalment.per.cent4
##                          -0.8351003                          -1.1948762
##               Sex...Marital.Status2               Sex...Marital.Status3
##                           0.0038724                           0.7636152
##               Sex...Marital.Status4                         Guarantors2
##                           0.2834001                          -0.5323598
##                         Guarantors3        Duration.in.Current.address2
##                           1.2335305                          -1.0339815
##        Duration.in.Current.address3        Duration.in.Current.address4
##                          -0.8974537                          -0.7484014
##                          Telephone2
##                           0.4569613
##
## Degrees of Freedom: 699 Total (i.e. Null);  665 Residual
## Null Deviance:       855.2
## Residual Deviance: 622.4     AIC: 692.4
```

The **model with the lowest AIC (at 692.4) has as its variables: Account Balance, Duration of Credit, Payment Status of Previous Credit, Purpose, Credit Amount, Value of Savings Stocks, Installment Percent, Sex/Marital Status, Guarantors, Duration in Current Address, Telephone.**

Let us recreate this model on the training data.

```
chosen_model<-glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
    Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
    Value.Savings.Stocks + Instalment.per.cent + Sex...Marital.Status +
    Guarantors + Duration.in.Current.address + Telephone, family = binomial(link = logit),
    data = MyData[s1, ])
chosen_model
```

```
##
```

```
## Call:  glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
##     Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
##     Value.Savings.Stocks + Instalment.per.cent + Sex...Marital.Status +
##     Guarantors + Duration.in.Current.address + Telephone, family = binomial(link = logit),
##     data = MyData[s1, ])
##
## Coefficients:
##                       (Intercept)                    Account.Balance2
##                         0.3783202                           0.3306060
##                  Account.Balance3                    Account.Balance4
##                         0.8418371                           1.6802727
##        Duration.of.Credit..month.  Payment.Status.of.Previous.Credit1
##                        -0.0278439                          -0.4366447
## Payment.Status.of.Previous.Credit2  Payment.Status.of.Previous.Credit3
##                         0.9069549                           1.2007277
## Payment.Status.of.Previous.Credit4                            Purpose1
##                         1.6843430                           1.7131507
##                          Purpose2                            Purpose3
##                         0.6788654                           1.0564755
##                          Purpose4                            Purpose5
##                         0.4784419                           0.6922853
##                          Purpose6                            Purpose8
##                        -0.0217938                          15.1382158
##                          Purpose9                           Purpose10
##                         0.6083848                           1.7158640
##                     Credit.Amount               Value.Savings.Stocks2
##                        -0.0001835                           0.2139996
##             Value.Savings.Stocks3               Value.Savings.Stocks4
##                         0.0378286                           1.2086323
##             Value.Savings.Stocks5                Instalment.per.cent2
##                         1.1716854                          -0.1779563
##              Instalment.per.cent3                Instalment.per.cent4
##                        -0.8351003                          -1.1948762
##             Sex...Marital.Status2               Sex...Marital.Status3
##                         0.0038724                           0.7636152
##             Sex...Marital.Status4                         Guarantors2
##                         0.2834001                          -0.5323598
##                       Guarantors3        Duration.in.Current.address2
##                         1.2335305                          -1.0339815
##      Duration.in.Current.address3        Duration.in.Current.address4
##                        -0.8974537                          -0.7484014
##                        Telephone2
##                         0.4569613
##
## Degrees of Freedom: 699 Total (i.e. Null);  665 Residual
## Null Deviance:       855.2
## Residual Deviance: 622.4     AIC: 692.4
```

Performance of the Model

Confusion Matrix for Training Observations

First we create a confusion matrix which shows how good our chosen model is at predicting whether a given individual's Creditability is 0=Bad and 1=Good. Overall, We know that 300 individuals have a Bad rating and 700 individuals have a Good rating, however this confusion matrix will just use the training data (700 of the original 1000 observations). Initially we will use a classification bound of 0.5 (Good if >= 0.5, Bad if < 0.5).

```
chosen_model_p=chosen_model$fitted.values
chosen_model_p[chosen_model_p>=0.5]=1
chosen_model_p[chosen_model_p<0.5]=0
table(MyData[s1, 1],chosen_model_p)
```

```
##    chosen_model_p
##       0   1
##   0 110 100
##   1  46 444
```

```
round(prop.table(table(MyData[s1, 1],chosen_model_p),1),2)
```

```
##    chosen_model_p
##        0    1
##   0 0.52 0.48
##   1 0.09 0.91
```

Given that it is worse to classify a customer as good when they are bad than it is to class a customer as bad when they are good, we may change the classification bound.

```
chosen_model_p=chosen_model$fitted.values
chosen_model_p[chosen_model_p>=0.65]=1
chosen_model_p[chosen_model_p<0.65]=0
table(MyData[s1, 1],chosen_model_p)
```

```
##    chosen_model_p
##       0   1
##   0 153  57
##   1  99 391
```

```
round(prop.table(table(MyData[s1, 1],chosen_model_p),1),2)
```

```
##    chosen_model_p
##        0    1
##   0 0.73 0.27
##   1 0.20 0.80
```

A classification bound of 0.65 gives us a nice balance. A high percentage of Good and Bad are correctly detected: about 20% more Bad at the cost of about 10% less Good

Confusion Matrix for the Test Data

Now we use the model (and the classification bound we decided on) to predict values for the test data.

```
test_p<-predict(chosen_model, newdata=MyData[-s1, -1], type="response")
test_p[test_p>=0.65]=1
test_p[test_p<0.65]=0
table(MyData[-s1, 1], test_p)
```

```
##    test_p
##       0   1
##   0  55  35
##   1  52 158
```

```
round(prop.table(table(MyData[-s1, 1], test_p),1),2)
```

```
##    test_p
##        0    1
##   0 0.61 0.39
##   1 0.25 0.75
```

We see the model performs slightly worse in terms of correctly prediction Good (80% vs 75%) but significantly worse in terms of correctly predicting Bad (73% vs 61%).

Lift Modeling

When we do lift modeling on the data we find the following:

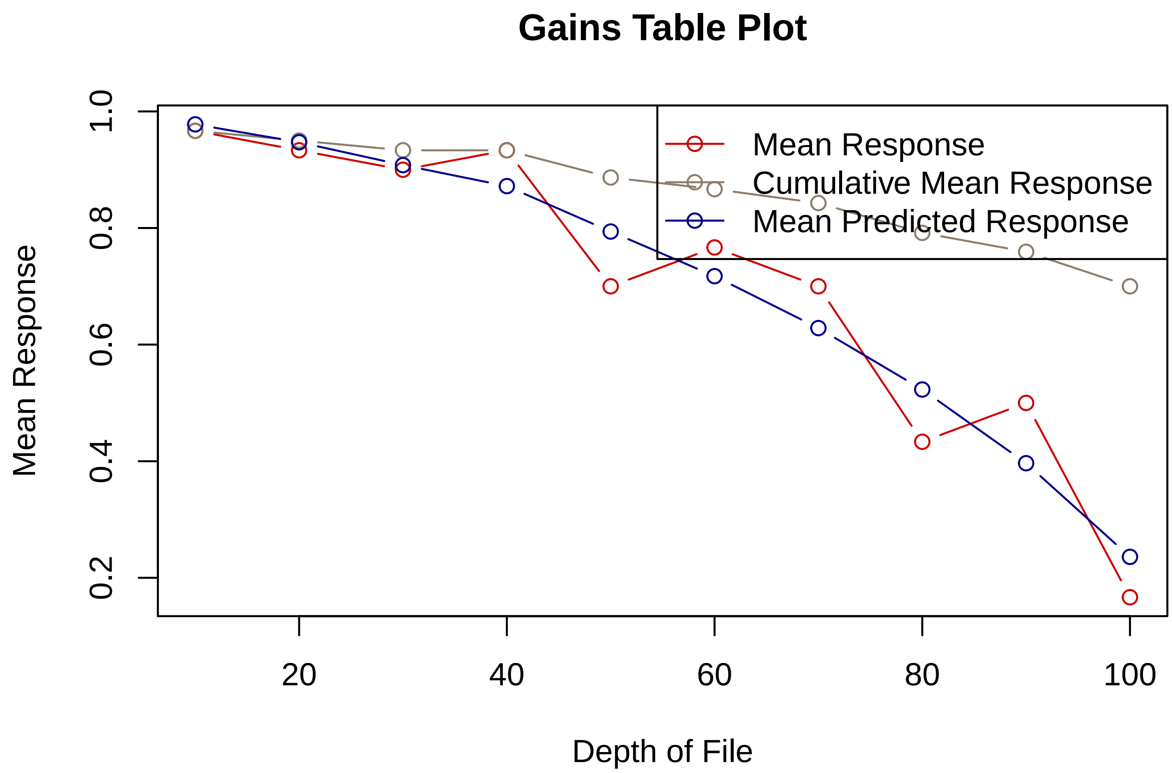
```r
require(gains)
```

```
## Loading required package: gains
```

```r
#had to redefine model in this function for whatever reason
chosen_model<-glm(formula = Creditability ~ Account.Balance + Duration.of.Credit..month. +
    Payment.Status.of.Previous.Credit + Purpose + Credit.Amount +
    Value.Savings.Stocks + Instalment.per.cent + Sex...Marital.Status +
    Guarantors + Duration.in.Current.address + Telephone, family = binomial(link = logit),
    data = MyData[-s1, ])
#get the PROBABILITIES not the predictions (0,1)
testp<-chosen_model$fitted.values
gains(as.numeric(MyData[-s1, 1])-1, testp, 10)
```

```
## Depth                            Cume   Cume Pct                     Mean
##  of           Cume     Mean      Mean   of Total    Lift   Cume     Model
## File     N      N      Resp      Resp      Resp     Index  Lift     Score
## -------------------------------------------------------------------------
##   10    30     30      0.97      0.97      13.8%     138    138      0.98
##   20    30     60      0.93      0.95      27.1%     133    136      0.95
##   30    30     90      0.90      0.93      40.0%     129    133      0.91
##   40    30    120      0.93      0.93      53.3%     133    133      0.87
##   50    30    150      0.70      0.89      63.3%     100    127      0.79
##   60    30    180      0.77      0.87      74.3%     110    124      0.72
##   70    30    210      0.70      0.84      84.3%     100    120      0.63
##   80    30    240      0.43      0.79      90.5%      62    113      0.52
##   90    30    270      0.50      0.76      97.6%      71    108      0.40
##  100    30    300      0.17      0.70     100.0%      24    100      0.24
```

```r
plot(gains(as.numeric(MyData[-s1, 1])-1, testp, 10))
```



Remember that we are predicting Good ratings, of which there are 700 of the 1000. So we would expect there to be mostly 1's in the top deciles. This is what we see, with at least 90% 1's in the top four deciles of probability. Then we see it decrease as we get to the lower deciles, which is also what we would expect to see if the lower probabilities are associated with scores of 0.

The ROC Curve

Now plot the ROC Curve:

```r
library(AUC)
```

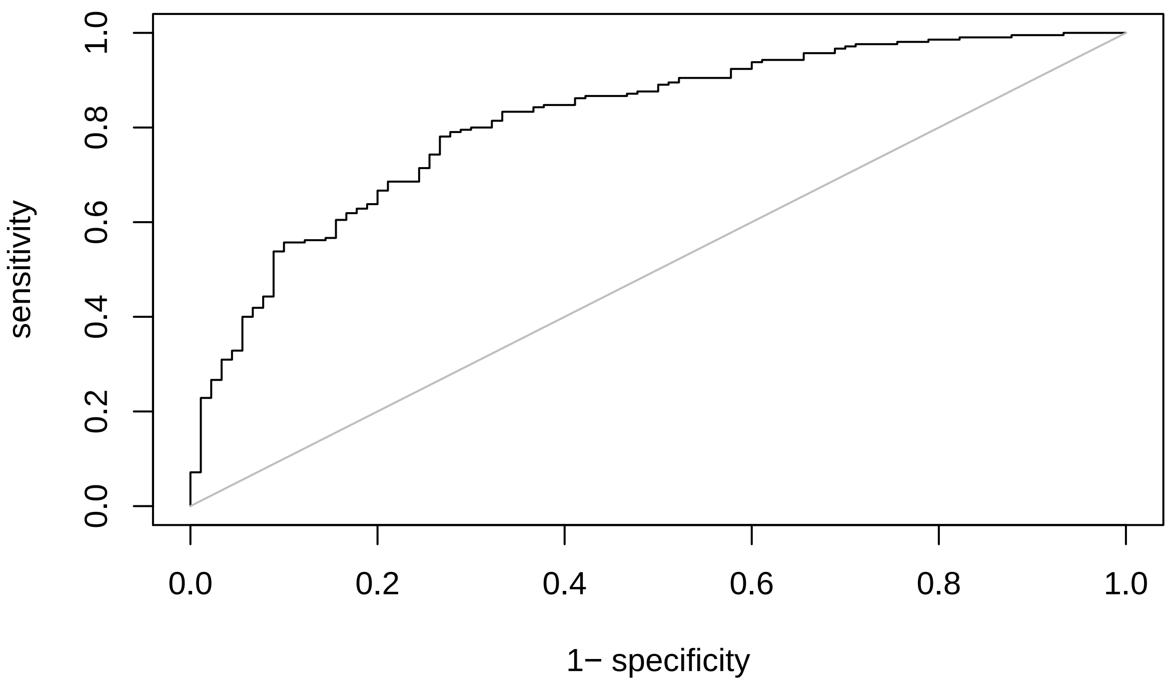
```
## AUC 0.3.0
```

```
## Type AUCNews() to see the change log and ?AUC to get an overview.
```

```
##
## Attaching package: 'AUC'
```

```
## The following objects are masked from 'package:caret':
##
##     sensitivity, specificity
```

```r
#don't convert to 0,1 numbers, input has to be a factor
plot(roc(testp, MyData[-s1, 1]))
```



The main diagonal line represents a random model (the equivalent of flipping a coin). The further the curve is from this line, the better our model. Our True Positive rate (accuracy) is on the y-axis. Our False Positive rate (1-specificity) is on the x-axis. To get better accuracy we end up having to increase our false positive rate as a tradeoff.