# EPLORATORY DATA ANALYSIS ON THE DATA

Good is encoded as 1 and bad as zero, therefore G/(G+B) = Proportion of good to bad =sum(Dataset$RESPONSE) = 700. There are 1000 cases, therefore70% of cases are good.

Yes, there are missing values in two instances. First, variables 5 through 10, i.e. the ‘purpose of credit’ variables were intended to be encoded as 1,0. However the ‘0’ values are not present in the dataset. Therefore, we will impute them.

Example:

|  |
| --- |
| dataset$USED\_CAR[is.na(dataset$USED\_CAR)] <- 0  dataset$`RADIO/TV`[is.na(dataset$`RADIO/TV`)] <- 0 |

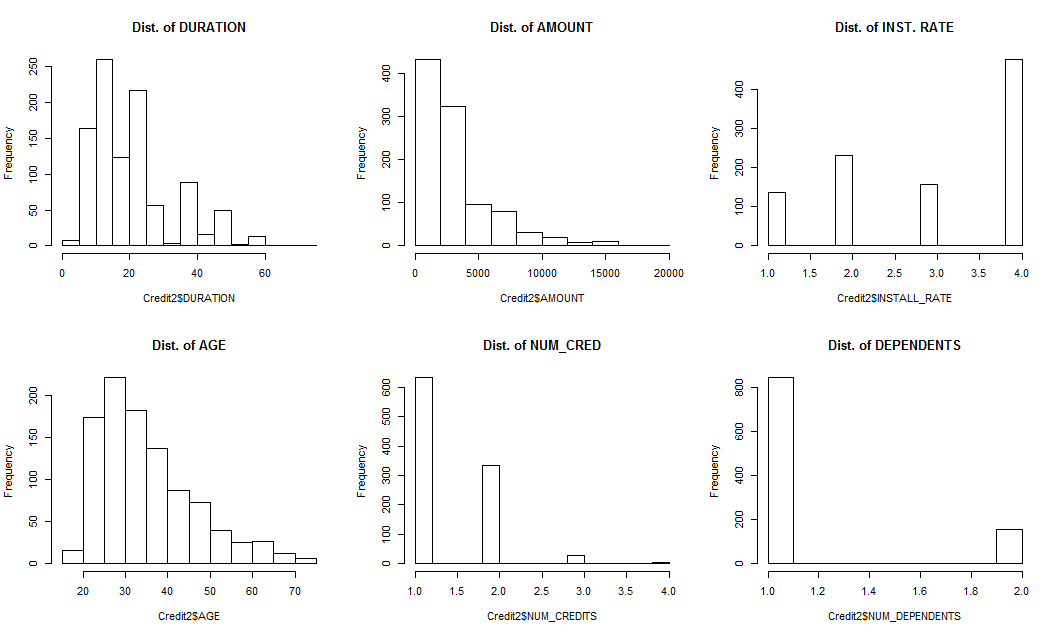
Secondly, the variable ‘age’ is missing values in 9 cases.

|  |
| --- |
| > sum(!complete.cases(dataset$AGE)) |

Therefore, we will take the median and impute in the missing cases.

|  |
| --- |
| > var <- median(dataset$AGE, na.rm= TRUE)  > Credit2$AGE[is.na(data$AGE)] <- 33) |

Six numeric continuous independent ‘X’ variables are described below.



Analysis:Duration is skewed right, amount is skewed right, age is skewed right, num\_credits are skewed right. The independent variables are not normally distributed and would need to be transformed for use in logistic or linear models.

Summary of independent numeric variables:

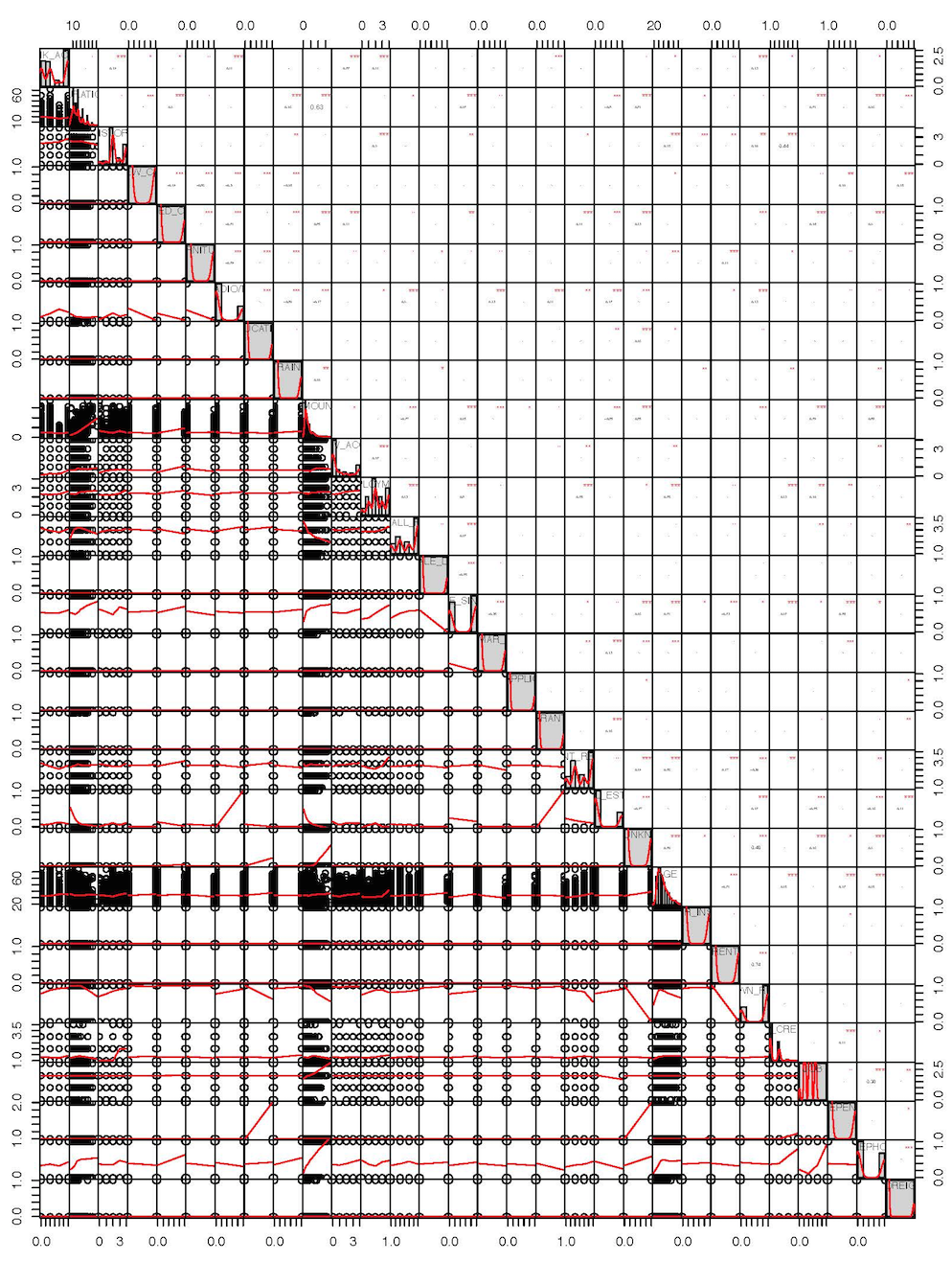
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Duration** | **Amount** | **Install rate** | **Age** | **Num Credits** | **Dependents** |
| **median** | 18 | 2320 | 3 | 33 | 1 | 1 |
| **mean** | 21 | 3271 | 3 | 35 | 1 | 1 |
| **SE.mean** | 0 | 89 | 0 | 0 | 0 | 0 |
| **CI.mean.0.95** | 1 | 175 | 0 | 1 | 0 | 0 |
| **var** | 145 | 7967212 | 1 | 128 | 0 | 0 |
| **std.dev** | 12 | 2823 | 1 | 11 | 1 | 0 |
| **coef.var** | 1 | 1 | 0 | 0 | 0 | 0 |

Counts and Frequencies of Categorical and Binary Variables

|  |  |  |  |
| --- | --- | --- | --- |
| CHK\_ACCT freq  1 0 274  2 1 269  3 2 63  4 3 394 | HISTORY freq  1 0 40  2 1 49  3 2 530  4 3 88  5 4 293 | SAV\_ACCT freq  1 0 603  2 1 103  3 2 63  4 3 48  5 4 183 | EMPLOYMENT freq  1 0 62  2 1 172  3 2 339  4 3 174  5 4 253 |
| PRESENT\_RESIDENT freq  1 1 130  2 2 308  3 3 149  4 4 413 | JOB freq  1 0 22  2 1 200  3 2 630  4 3 148 | NEW\_CAR freq freq.1  0 766  1 234 | USED\_CAR freq freq.1  0 897  1 103 |
| FURNITURE  0 819  1 181 | EDUCATION  0 950  1 50 | RETRAINING  0 903  1 97 |  |

Analysis:The largest segment of applicants are new applicants without a checking account.

The below figure displays a correlation matrix of the independent variables. We can see that there are several highly correlated independent variables.



Although the assignment does not explicitly call for a model, in this case a logistic can be so direct and illustrative of the variable importance. that we have called that model to investigate this question.

|  |
| --- |
| glm(formula = Credit2$RESPONSE ~ ., family = binomial(link = "logit"),  data = Credit2) |
| Deviance Residuals:  Min 1Q Median 3Q Max  -2.6527 -0.7189 0.3874 0.7063 2.3592  Coefficients:  *The following independent variables are significant predictors of the outcome variable.*  Estimate Std. Error z value Pr(>|z|)  CHK\_ACCT 5.637e-01 7.251e-02 7.774 7.59e-15 \*\*\*  DURATION -2.686e-02 9.010e-03 -2.981 0.00287 \*\*  HISTORY 4.012e-01 8.975e-02 4.470 7.82e-06 \*\*\*  NEW\_CAR -7.917e-01 3.846e-01 -2.059 0.03952 \*  USED\_CAR 8.343e-01 4.823e-01 1.730 0.08366 .  EDUCATION -8.660e-01 5.009e-01 -1.729 0.08383 .  AMOUNT -1.178e-04 4.264e-05 -2.763 0.00573 \*\*  SAV\_ACCT 2.496e-01 6.061e-02 4.118 3.82e-05 \*\*\*  INSTALL\_RATE -3.210e-01 8.629e-02 -3.720 0.00020 \*\*\*  MALE\_SINGLE 5.385e-01 2.049e-01 2.628 0.00859 \*\*  GUARANTOR 9.446e-01 4.195e-01 2.252 0.02433 \*  OTHER\_INSTALL -6.203e-01 2.040e-01 -3.040 0.00236 \*\*  TELEPHONE 3.541e-01 1.951e-01 1.815 0.06960 .  FOREIGN 1.452e+00 6.221e-01 2.334 0.01959 \*  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 1221.73 on 999 degrees of freedom  Residual deviance: 909.06 on 969 degrees of freedom  AIC: 971.06  Number of Fisher Scoring iterations: 5 |

Analysis:Our results are intuitive. It is reasonable that ***Checking Account Balance*** and ***Savings Account Balance*** would be the most significant predictors of the outcome variable. Some of the ‘x’ variables are correlated and thus these results are not optimal.

# MODELLING ON THE ENTIRE DATA SET

Four decision trees were built on the full data. They produced similar results in terms of accuracy on the whole set. The second model shown below produced the highest accuracy.

|  |  |
| --- | --- |
|  | Parameters |
| DT1 | DT1 <- rpart(RESPONSE ~., data = Credit2 , method="class") |
| DT2 | DT2<- rpart(RESPONSE ~., data = Credit2 , method="class", maxdepth = 15,  minsplit = 15, xval = 10, cp=.01, parms = list(split = 'information')) |
| DT3 | DT3<- rpart(RESPONSE ~., data = Credit2 , method="class", maxdepth = 15,  minsplit = 40, xval = 10, parms = list(split = 'gini')) |
| Z | DT4<- rpart(RESPONSE ~., data = Credit2 , method="class", maxdepth = 5,  minsplit = 50, xval = 15, cp= .02, parms = list(split = 'information')) |

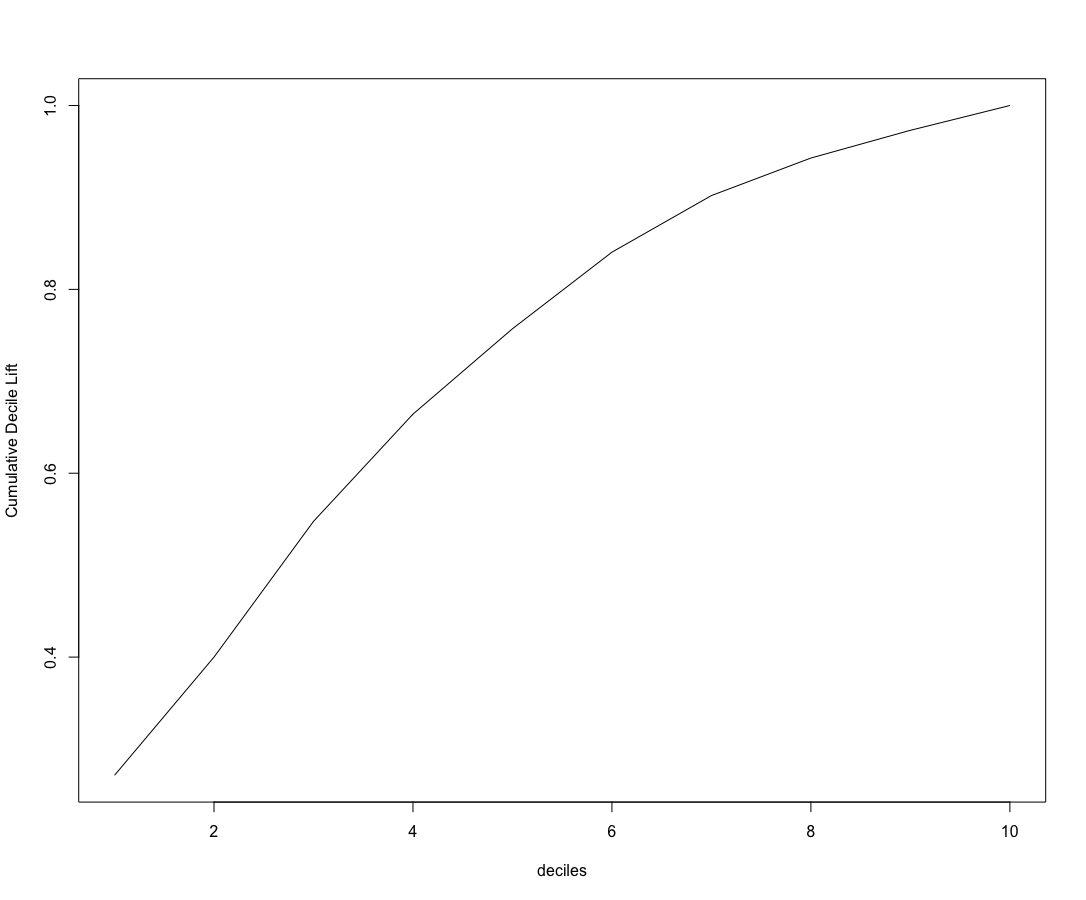
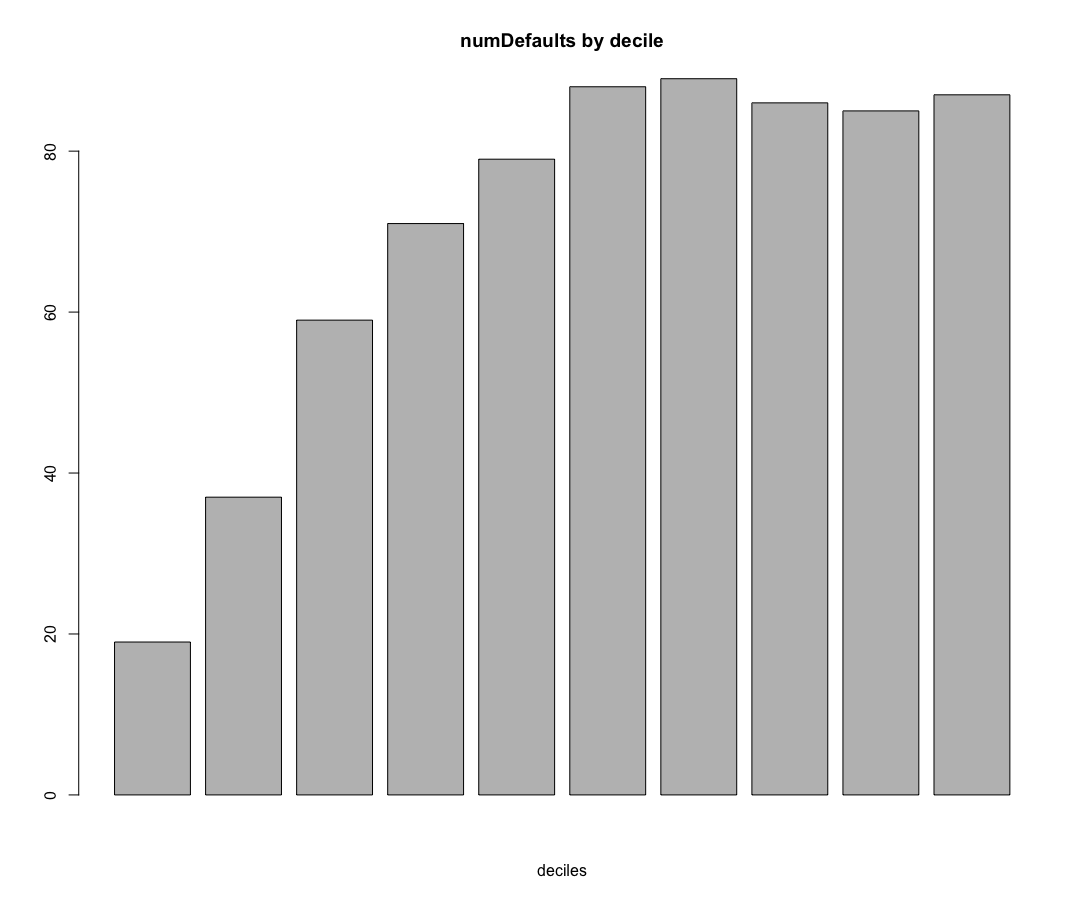
The main parameters that we found to have an effect on the levels of the tree and the accuracy against the whole set were: Minsplit (the minimum number of observations that must exist in a node in order for a split to be attempted) which we did not want to be too high, cp (a rule that will not split if the result does not produce given improvement in fit) which we did not want to be too high, and maxdepth (the max levels of the tree) which we did not want to be too low as it would not classify accurately enough.

|  |  |
| --- | --- |
| MODELS | Accuracy |
| DT 1 | 0.794 |
| DT 2 | **0.796** |
| DT 3 | 0.786 |
| DT 4 | 0.762 |

|  |  |
| --- | --- |
| DT1 | DT2 |
| CP nsplit rel error xerror xstd  1 0.05166667 0 1.0000000 1.000000 0.04830459  2 0.04666667 3 0.8400000 1.006667 0.04839605  3 0.01833333 4 0.7933333 0.840000 0.04576462  4 0.01666667 6 0.7566667 0.880000 0.04646432  5 0.01111111 7 0.7400000 0.890000 0.04663225  6 0.01000000 11 0.6866667 0.900000 0.04679744 | CP nsplit rel error xerror xstd  1 0.05166667 0 1.0000000 1.0000000 0.04830459  2 0.04666667 3 0.8400000 0.9900000 0.04816534  3 0.01833333 4 0.7933333 0.8900000 0.04663225  4 0.01111111 6 0.7566667 0.8733333 0.04635084  5 0.01000000 12 0.6800000 0.8433333 0.04582467 |
| DT3 | DT4 |
| CP nsplit rel error xerror xstd  1 0.05166667 0 1.0000000 1.0000000 0.04830459  2 0.04666667 3 0.8400000 0.9700000 0.04787936  3 0.01833333 4 0.7933333 0.8533333 0.04600290  4 0.01444444 6 0.7566667 0.8500000 0.04594381  5 0.01000000 9 0.7133333 0.8866667 0.04657658 | CP nsplit rel error xerror xstd  1 0.05166667 0 1.0000000 1.0000000 0.04830459  2 0.04666667 3 0.8400000 1.0066667 0.04839605  3 0.02000000 4 0.7933333 0.8266667 0.04552118 |

We found that increasing minsplit > 15 pruned the tree too much and lowered our accuracy. The levels in the tree with minsplit = 50 was three while the levels with minsplit = 15 was 5 (a more complex model). It is impossible to know how complex the fit should be as we are testing on the training set, but in these conditions model two performs best.

Unfortunately, it appears our table is upside-down, the ones and zeros must be switched? We expected a lift curve and decile plot that are the opposite of what we have here. I was unable to figure out how to switch the zeros to ones and vice-versa in the RESPONSE column.



# SPLIT THE ENTIRE DATA INTO 70-30

No, the above models are not the reliable ones, as we will need to divide our data into training and testing to train the model on the training data and test the model on the test data set. Hence, we split the data in the following ratio

TRAINING DATA = 70%

TESTING DATA = 30%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DT | CP | Maxdepth | Minsplit | Train accuracy | Test accuracy |
| Model1 | 0.01 | 15 | 15 | .844 | .706 |
| Model2 | 0.02 | 10 | 20 | .782 | .7 |
| Model3 | 0.0001 | 20 | 50 | .776 | .696 |
| Model4 | 0.01 | 30 | 150 | .736 | .704 |

Results of four different trees, splitting on ***‘gini’***, data split = 50/50

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DT | CP | Maxdepth | Minsplit | Train accuracy | Test accuracy |
| Model1 | 0.01 | 15 | 15 | .85 | **.706** |
| Model2 | 0.02 | 10 | 20 | .782 | .696 |
| Model3 | 0.0001 | 20 | 50 | .776 | .696 |
| Model4 | 0.01 | 30 | 150 | .736 | .704 |

No difference in accuracy, therefore we will use info.

Performance Measures:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision** | | **Sensitivity** | | **Recall** | | **AUC** |
| Class | 0 | 1 | 0 | 1 | 0 | 1 |
| Model1 | .5 | . 7614213 | 0.3605442 | 0.8498584 | 0.4189723 | 0.8032129 | 0.6739223 |
| Model2 | 0.4835165 | 0.7481663 | 0.2993197 | 0.8668555 | 0.3697479 | 0.803149 | 0.7150566 |
| Model3 | 0.480315 | 0.769437 | 0.4149660 | 0.8130312 | 0.4452555 | 0.7906336 | **0.7150951** |
| Model4 | 0.4963504 | 0.7823691 | 0.4625850 | 0.8045326 | 0.4788732 | 0.7932961 | 0.7085429 |

Based on AUC we will choose Model3 as our model as it slightly outperforms Model2.

Therefore, the decision tree parameters that provide the best results are those that prune the tree somewhat. Model2 and Model3 have only five levels, while the default setting produces a tree of six levels. Having a larger min split prunes, the tree, prevents over-fitting and improves our performance on unseen data. The best parameter was achieved with the following parameters, although we believe that minsplit is the important parameter.

|  |  |  |
| --- | --- | --- |
| CP | Maxdepth | Minsplit |
| 0.0001 | 20 | 50 |

We select the best-pruned tree based on AUC. The cp parameter can be helpful in pruning the tree, as can minsplit.

# C5.0 Parameters

We experimented with many combinations of parameters and measured accuracy on the test set. Only the rules parameter affected our accuracy.

|  |  |
| --- | --- |
| 1 | tree1 <- C5.0(RESPONSE~., data=mdTrn, method="class") |
| 2 | tree2 <- C5.0(RESPONSE~., data=mdTrn, method="class", subset = TRUE, winnow = TRUE, rules = TRUE) |
| 3 | tree3 <- C5.0(RESPONSE~., data=mdTrn, method="class", winnow = FALSE, rules = TRUE, earlyStopping = TRUE, CF = .00001, miniCases = 10, seed = sample.int(4096, size = 1) -  1L) |

Results:

|  |  |  |
| --- | --- | --- |
|  | Confusion matrix | Accuracy on Test Set |
| Tree1 | True  pred 0 1  0 55 52  1 96 297 | 0.704 |
| Tree2 | True  Pred 0 1  0 52 46  1 99 303 | .71 |
| Tree3 | True  pred 0 1  0 52 46  1 99 303 | .71 |

Analysis: There were no effects on accuracy on the test set given many varied parameters. A variety of settings were chosen for each of the parameters shown in model1.

Performance measures for C5.0 Decision Trees

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision** | | **Recall** | | **F1** | | **AUC** | **Optimal Threshold** |
| Class | 0 | 1 | 0 | 1 | 0 | 1 |  | |
| 1 | 0.514 | 0.755 | 0.364 | 0.851 | 0.426 | 0.801 | 0.650 | 0.855 |
| 2 | 0.531 | 0.754 | 0.344 | 0.868 | 0.417 | 0.806 | 0.710 | 0.235 |
| 3 | 0.531 | 0.754 | 0.344 | 0.868 | 0.417 | 0.806 | 0.710 | 0.235 |

Tree2 is our best model. A set of rules derived from this model is displayed below. Each rule is in rank order in terms of its ability to correctly predict the outcome. In the case of these rules, class ‘0’ can be predicted quite well.

|  |  |
| --- | --- |
| Rule 1: (8, lift 3.0)  HISTORY = 0  OWN\_RES = 0  -> class 0 [0.900] | Rule 2: (8, lift 3.0)  CHK\_ACCT in {0, 1}  HISTORY = 0  AGE <= 29  -> class 0 [0.900] |
| Rule 3: (5, lift 2.9)  CHK\_ACCT in {0, 1}  HISTORY = 2  SAV\_ACCT = 0  INSTALL\_RATE <= 3  REAL\_ESTATE = 0  PROP\_UNKN\_NONE = 0  AGE > 37  -> class 0 [0.857] | Rule 4: (12/1, lift 2.9)  CHK\_ACCT in {0, 1}  HISTORY = 2  SAV\_ACCT = 0  INSTALL\_RATE > 3  GUARANTOR = 0  PRESENT\_RESIDENT = 2  -> class 0 [0.857] |

Comparing different learns was done based on AUC in evaluating the performance of different learners on the test data. The best models of each type of learner produced roughly the same AUC.

RPart Measures

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision** | | **Sensitivity** | | **Recall** | | **AUC** |
| Class | 0 | 1 | 0 | 1 | 0 | 1 |
| Model1 | 0.500 | 0.761 | 0.360 | 0.849 | 0.418 | 0.803 | 0.673 |
| Model2 | 0.484 | 0.748 | 0.299 | 0.867 | 0.370 | 0.803 | 0.715 |
| Model3 | 0.480 | 0.769 | 0.415 | 0.813 | 0.445 | 0.791 | ***0.715*** |
| Model4 | 0.496 | 0.782 | 0.463 | 0.805 | 0.479 | 0.793 | 0.709 |

C5.0 Measures

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision** | | **Recall** | | **F1** | | **AUC** | **Optimal Threshold** |
|  | 0 | 1 | 0 | 1 | 0 | 1 |  | |
| tree1 | 0.514 | 0.755 | 0.364 | 0.851 | 0.426 | 0.801 | 0.65 | 0.855 |
| tree2 | 0.531 | 0.754 | 0.344 | 0.868 | 0.417 | 0.806 | ***0.71*** | 0.235 |
| tree3 | 0.531 | 0.754 | 0.344 | 0.868 | 0.417 | 0.806 | 0.71 | 0.235 |

Best model from the DT’s is tested in this case. The accuracy of the best model DT2is 0.795 whereas the accuracy of that model with cost matrix included is 0.73. This is a negligible difference, however there are noticeable differences in precision, recall and AUC between the models as shown above.

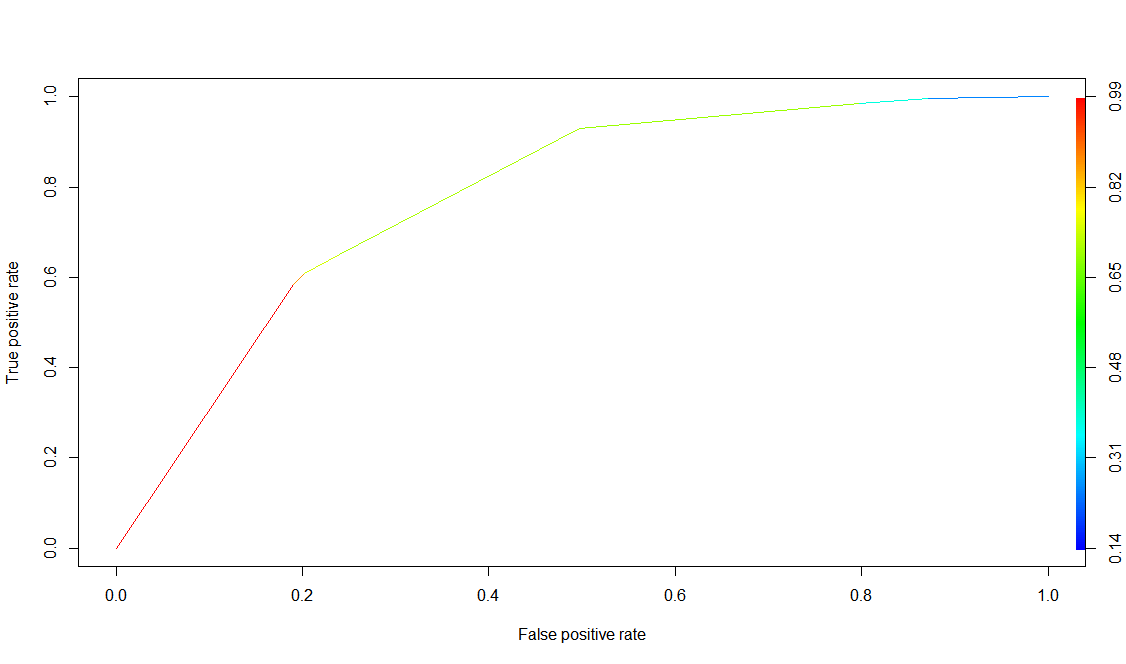
The difference between the models with and without cost matrix for different cutoff values for classification threshold are as follows

.

|  |  |  |
| --- | --- | --- |
| Probability Threshold | Accuracy of Model DT2 | Accuracy of Model from Q2 with cost matrix |
| .3 | 0.74 | 0.737 |
| .4 | 0.773 | 0.737 |
| .6 | 0.777 | 0.762 |
| .7 | 0.665 | 0.652 |
| .8 | 0.637 | 0.627 |

As we see, the accuracy of the entire model with cost matrix reduces because the prediction of 1s increases and predictions which cause the loss of 500DM reduces, this is clearly seen in the confusion matrix.

Optimal threshold from the ROC curve for model with cost matrix is 0.6942446 and accuracy for the same is 0.773. This can also be seen in the table above.



Theoretical threshold is 0.833 and the accuracy of the model with cost matrix with this threshold is 0.617. The accuracy with theoretical threshold is less than the accuracy with the optimal threshold. This once again holds true with the table above.

Let’s examine your ‘best’ decision tree model obtained. What is the tree depth? And how many nodes does it have? What are the important variables for classifying “Good’ vs. ‘Bad’ credit? Identify two relatively pure leaf nodes. What are the ‘probabilities for ‘Good’ and ‘Bad’ in these nodes?

In this Decision tree we used Cost Matrix and split as Information as the parameters to build the tree.

DT2<- rpart(RESPONSE ~., data = mdTrn , method="class", maxdepth = 15, minsplit = 15, xval = 10, cp=.01, parms = list(split = 'information'))

Minimal Leaf size: 5

Maximal Depth of tree: 10

Decision Criterion: information

Number of nodes: 27

|  |
| --- |
|  |

|  |
| --- |
|  |

Pure Leaf Nodes:

One of the pure leaf nodes follows this path:

*DURATION (>=29) ⇒ HISTORY (0,1) ⇒ USED\_CAR (=0).*

It has a size of 17 and there are 0 bad cases and 17 good cases, it perfectly classifies our observations and therefore P(1) = 1. Another of the pure leaf nodes follows this path:

*DURATION (>=29) ⇒ CHK\_ACCT (0,1) ⇒ JOB (1,2)*

It has a size of 18 and they’re 17 Bad cases and 1 good cases, therefore the P(1) = .056, P(0) =.944

The predicted probabilities can be used to determine how the model may be implemented. We can sort the data from high to low on predicted probability of “good” credit risk. Then, going down the cases from high to low probabilities, one may be able to determine an appropriate cutoff probability – values above this can be considered acceptable credit risk.

The use of cost figures given above can help in this analysis. For this, first sort the validation data on predicted probability. Then, for each validation case, calculate the actual cost/benefit of extending credit. Add a separate column for the cumulative net cost/benefit. How far into the validation data would you go to get maximum net benefit? In using this model to score future credit applicants, what cutoff value for predicted probability would you recommend? Provide appropriate performance values to back up your recommendation.

We have plotted a graph for our model based on the above guidelines. Below is the scatter plot for the same.

|  |
| --- |
|  |

As we can see from the graph, we are getting maximum profit of 5000DM with probability of 0.8461.

Use the best model that you identified after splitting the data in order to find the behavior of the model after inclusion of the Cost Matrix

# Cost Matrix Model:

In DT2 we measured accuracy of the model against the entire dataset, therefore we will use that measure to compare with the performance of our new (with cost matrix incorporated), against the test set.

Accuracy of Model of DT2 = 0.796 (built and measured against same set).

Accuracy of Model with Cost Matrix incorporated = 0.722 (against test data)

Therefore, it is perfectly reasonable to have lower accuracy as this model was applied to unseen data.

ADD precision, recall, f1, etc… and Compare.

Explain why we are ‘testing’ or ‘measuring’ results of Model from Q2 using entire data (no un-seen data)

|  |  |  |
| --- | --- | --- |
| Probability Threshold | Accuracy of Model from Q2 | Accuracy of w/ Model (CostMatrix) |
| .6 | .796 | **.648** |
| .7 | .796 | .31 |
| .8 | .705 | .31 |
| .9 | .316 | .31 |

Highest value of Youden Index = Best Cutoff Threshold

