ACTSC 632 – Project for Module 5 – Solutions

In this project you will work with the same credit data set as for the project for Module 4, and with the same team. Whether you want to use the same team leader or not as for the previous project is left up to you.

The goal of this project is to compare different classification methods based on trees for this problem and make a recommendation on what is the best method to use.

1. First randomly split your data in 70% of the observations for training and 30% for testing.

   Solution: see code CodeProjMod5July15.text for this first question and all others.

2. Simply using recursive binary partitioning, obtain a tree for this classification problem.

   (a) How many leaves does your tree have?

   (b) How many factors were used to build this tree?

   (c) What is the deviance for this tree? (If you used something else than the default definition of deviance in R, please specify how is deviance determined).

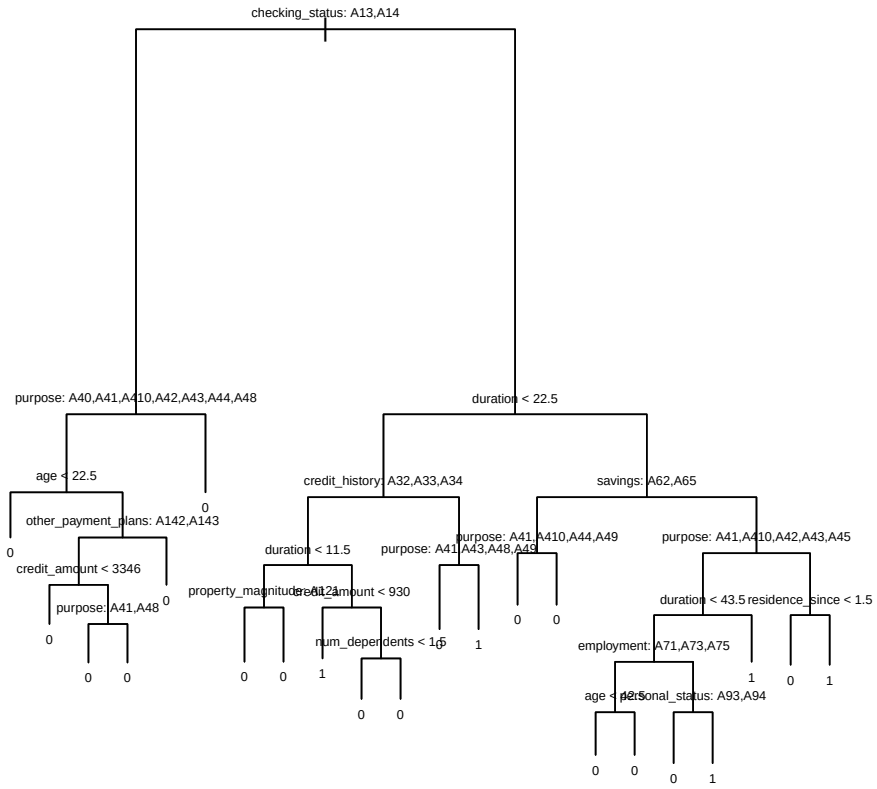
   Solution: the summary of the tree model is given by

```
  Classification tree:
tree(formula = class ~ ., data = credit, subset = train)
Variables actually used in tree construction:
 [1] "checking_status"     "purpose"       "age"       "other_payment_plans"
 [5] "credit_amount"   "duration"  "credit_history"   "property_magnitude"
 [9] "num_dependents"      "savings"   "employment"    "personal_status"
[13] "residence_since"
Number of terminal nodes:  22
Residual mean deviance:  0.7722 = 523.6 / 678
Misclassification error rate: 0.1871 = 131 / 700
```

   So there are 22 leaves and 13 factors were used. The deviance is 0.7722. Note that since the tree will depend on the training set, which is randomly chosen, you may have obtained a tree that has a quite different structure, with a different subset of factors used, and a different number of terminal nodes. As mentioned in class, this method has a high variance, which is why the results can be so different.

   (d) Plot the tree you obtained using R. There should be enough information that given an observation, one could determine in which leaf it ends up.

   Solution: we obtain the following tree



We recall that the categories listed on a node are those used to determine which observations go in the left child.

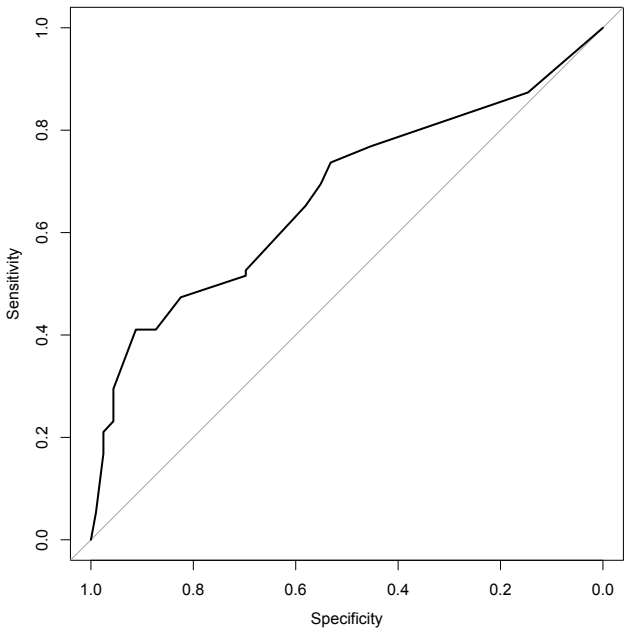
(e) Use your tree to make predictions for the test data set. Produce the confusion matrix corresponding to your tree and plot the ROC curve.

Solution: we get the confusion table

	0	1
0	196	73
1	9	22

and corresponding overall error, type-1 and type 2 of 0.2733, 0.0439, and 0.7684211.

The ROC curve is given below and has AUC of 0.6729:



3. Now try to use pruning to see if you can improve your results.

   (a) Using the function `cv.tree` in R, determine the optimal level of complexity for the tree, i.e., the number of terminal nodes in the tree that minimizes the test prediction error (estimated by cross-validation). Does pruning the tree improve the deviance? The prediction error? Use the test set to answer the latter two questions.
   Solution: first we find the optimal size of tree

```
$size
[1] 22 13 11 10  8  5  4  1

$dev
[1] 209 209 197 194 204 202 208 211
```

   and see that the optimal size (based on misclassification rate) is 10. We then build a pruned tree of size 10 and get the following

```
Classification tree:
snip.tree(tree = tree.credit, nodes = c(2L, 14L, 24L, 51L, 120L,
31L, 121L, 13L))
Variables actually used in tree construction:
[1] "checking_status" "duration"        "credit_history"  "credit_amount"   "savings"
[6] "purpose"         "employment"
Number of terminal nodes:  10
Residual mean deviance:  0.9536 = 658 / 690
Misclassification error rate: 0.1943 = 136 / 700
```

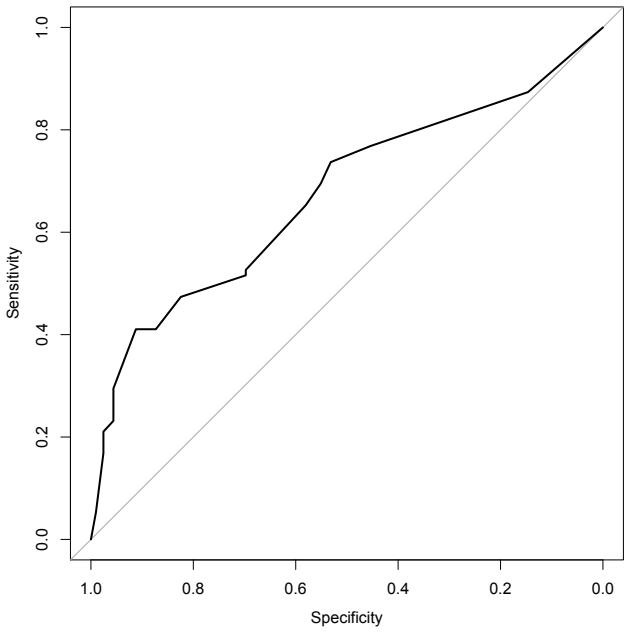
   On the test set, the overall error of 0.26 is a bit smaller than for the unpruned tree.
   If we look at the deviance on the test set, we see that it goes down from 495.1042 to 327.5961 when using the pruned tree, so there is a clearly an improvement

(b) If pruning helps, then produce the confusion matrix for the pruned tree based on the optimal level of complexity and plot the corresponding ROC curve.

Solution: I realize that some students might have determined based on their results that "pruning doesn't help" and did not answer this question. With my numbers, the pruned tree has a slightly lower misclassification rate so I interpret this as "pruning helps", and thus computed the confusion matrix

```
    #               credit.test
#pred.prune.cred   0   1
#               0 197  68
#               1   8  27
```

As mentioned before, the prediction error is 0.26, with type-1 error of and type-2 error of with ROC curve (of corresponding AUC of 0.7408)



4. Now try bagging and random forests to see if you can improve your results.

(a) Provide the confusion table obtained using bagging.

Solution: with bagging we obtain

	0	1
0	183	58
1	22	37

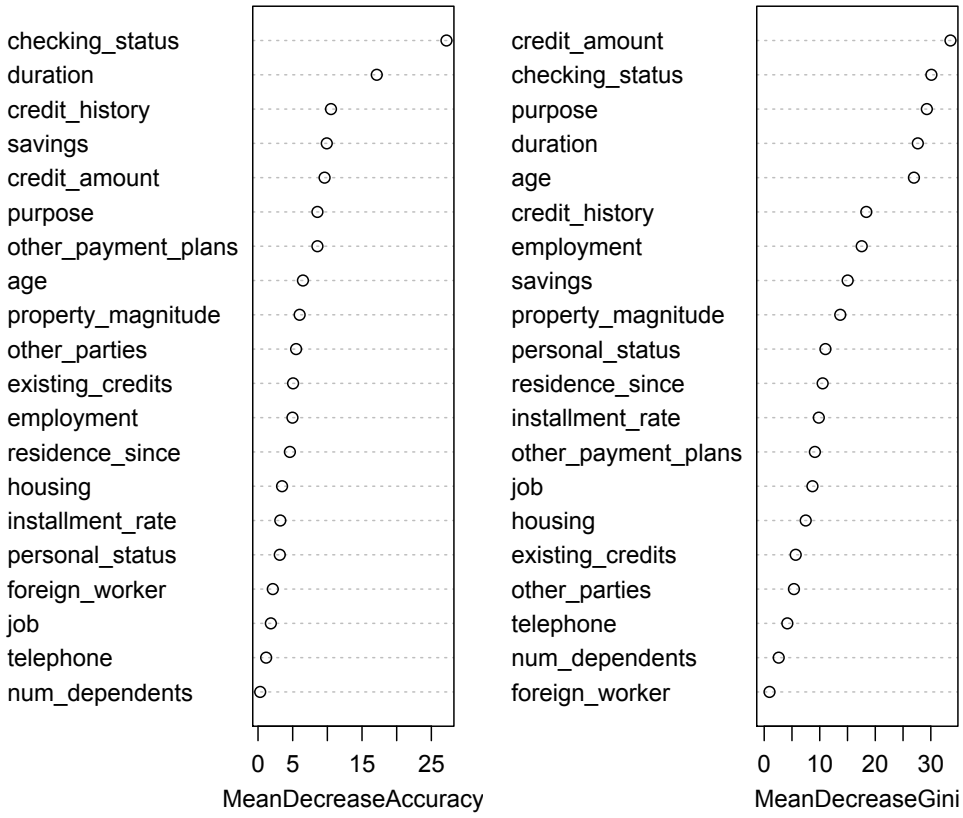
(b) Provide the confusion table obtained using random forests. How many variables were each split chosen from?

Solution: with random forests based on a subset of 5 randomly chosen predictors at each split, we obtain

	0	1
0	189	63
1	16	32

(c) According to the results obtained based on random forests, which predictors seem the most important? Provide data and/or plots to answer this question.

Solution: using the importance function, we get the plots



We see that whether we use the accuracy (prediction error) or Gini index, the predictor checking_status seems very important, as well as duration. Other important predictors are credit_history and savings, purpose, age and credit_amount. These are pretty consistent with the (limited set of) predictors that were used to construct the pruned tree.

5. Conclude by making a suggestion as to which of the above methods is the best for this problem.

Solution: we compare the overall, type-1 and type-2 errors on the test set for the 4 methods:

single tree	0.2733	0.0439	0.7684
pruned tree	0.26	0.0585	0.6947
bagging	0.2667	0.1073	0.6105
random forest	0.2633	0.0780	0.6632

Based on our results, the single tree is definitely not a good choice. Among the three other methods, the pruned tree has the best overall prediction error and type-1 error, but bagging has the lowest type-2 error. Given that bagging and random forest have much less variance, our

assessment of their performance is much more reliable than for the single tree and pruned tree and as such, we would recommend using one of those two methods.