The German Credit data set

The German Credit data set contains observations on 30 variables for 1000 past applicants for credit. Each applicant was rated as "good credit" or "bad credit". New applicants for credit can also be evaluated on these 30 "predictor" variables. We want to develop a credit scoring rule that can be used to determine if a new applicant is a good credit risk or a bad credit risk, based on values for one or more of the predictor variables. All the variables are explained in the document "GermanCreditVariablesDefinitions.pdf". You can find this data set in spreadsheet German Credit.xls.

Please use R to answer the following questions.

(a) The first step before constructing any predictive models is exploring data and learning about the variables. So try to explore the data. For example, what is the proportion of "Good" to "Bad" cases? Obtain descriptions of the predictor (independent) variables – mean, standard deviations, etc. for real-values attributes, frequencies of different category values. Look at the relationship of the input variables with the target variable. Draw any plots that you think help with your analysis. Which variables do you think is imporant to predict "good" and "bad" cases? Anything noteworthy in the data? Only include interesting observations in your reports.

(b) Check the effect of the size of training and testing data on the quality of your predictions. To do so, consider different size of Training and Test sets. For example, consider a partition of the data into 50% for Training and 50% for Test. What model performance do you obtain? Is the model reliable (why or why not)?

Consider partitions of the data into 70% for Training and 30% for Test, and 80% for Training and 20% for Test and report on model and performance comparisons. Feel free to experiment with other size partitions on the data. Is there any specific model you would prefer to implement? Please explain your reasoning.

In developing the models above, change some of the decision tree options and see if and how they affect performance (for example, the minimum number of cases at a leaf node, the split criteria). Also, does pruning give a better model – please explain why or why not?.

Also, consider different type of decision tree operators – for example, C5.0 and C&R tree – play around with the parameters till you get a 'good' model. Do you see any performance differences across different types of decision tree learners?

(c) The consequences of misclassification have been assessed as follows: the costs of a false positive (incorrectly saying an applicant is good creit risk) outwiegh the cost of a false negative (incorrectly saying an applicant is a bad credit risk) by a factor of five. This can be summerized in the following "Opportunity Cost Table":

		Preicted	
Actual		Good (Accept)	Bad (Reject)
	Good	0	100DM
	Bad	500DM	0

Use the misclassification costs in obtaining a model (To do this, you can use the "loss" argument in the rpart() function. Please look at the link at the end of this document for an example). Do you observe any changes in the model and /or performance? Are there any benefits from specifying misclassification costs?

(d) What are the best decision rules for classifying "Good" applicants? Please justify your answer.

(e) Summarize your findings.

An example of including different costs of mis-classification in rpart:
Predicting Fraud (http://www.togaware.com/datamining/survivor/Predicting_Fraud.html)