BUSINESS DATA MINING (IDS 572)

HOMEWORK 5
DUE DATE: WEDNESDAY, OCTOBER 05 AT 3:00 PM

- Please provide succinct answers to the questions below.
- Please include all the R codes you use.
- You should submit an electronic pdf or word file in blackboard.
- Please include the names of all team-members in your write up and in the name of the file.

Problem 1. Do not use R to answer this question. The following table contains data from an employee database. The database includes the status, department, age range and salary of each employee.

Status	Department	Age	Salary
Senior	Sales	31-35	46K-50K
Junior	Sales	26-30	26K-30K
Junior	Sales	31-35	31K-35K
Junior	Systems	21-25	46K-50K
Senior	Systems	36-40	66K-70K
Junior	Systems	26-30	66K-70K
Senior	Systems	41-45	66K-70K
Senior	Marketing	36-40	46K-50K
Junior	Marketing	31-35	41K-45K
Senior	Secretary	46-50	36K-40K
Junior	Secretary	26-30	26K-30K

This problem asks you to learn a Naïve Bayes classifier for predicting the employee status.

(a) What is the prior probability of status $p(status)$?

(b) What is the conditional probability (given status) for department, age and salary respectively? Please write down your answers in three tables for the three conditional probabilities, respectively.

(c) Use your naïve Bayes classifier, predict the status for two instances A={Marketing, 31-35, 46K-50K} and B={Sales, 31-35, 66K-70K}.

(d) Suppose we add another feature called "SalaryDuplicate", which takes on the same value as "Salary" for all training examples. What are the prediction results for the above two instances, if we train a naïve Bayes classifier on the same dataset with this extra feature?

(e) Why do you observe the differences in parts (c) and (d)? What property of the naïve Bayes classifier was affected?

Problem 2. 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) Explore the data: 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. Anything noteworthy in the data?

(b) We will first focus on a descriptive model i.e. assume we are not interested in prediction. Develop a decision tree on the full data. What levels of accuracy/error are obtained? What is the accuracy for the "good" and "bad" cases? Do you think this is a reliable (robust?) description?

(c) Next consider developing a model for prediction. For this, we should divide the data into Training and Test sets. 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?

(d) In implementation, the model will be used to identify "good" vs. "bad" credit applicants. It is conceivable that certain applications will not carry reliable data on the different attributes (data on some attributes may be missing for the case you want to predict) used in the model. Will the decision tree model still be usable? How do you propose to use the model in such cases?

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

The opportunity cost table was derived from the average net profit per loan as shown bleow:

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

(f) Change some of the decision tree options and try the above. Also, see what happens when Pruning is set on.

(g) What are the best nodes for classifying "Good" applicants? Output rules corresponding to these. Please explain why you chose these nodes.

(h) 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)