CS 422: Data Mining

Department of Computer Science
Illinois Institute of Technology
Vijay K. Gurbani, Ph.D.

Spring 2018: Homework 2 (10 points)

Due date: Sunday, Mar 04, 2018 11:59:59 PM Chicago Time

Please read all of the parts of the homework carefully before attempting any question. If you detect any ambiguities in the instructions, please let me know right away instead of waiting until after the homework has been graded.

This is the final homework writeup.

1 Exercises (3 points divided evenly among the questions). Please submit a PDF file containing answers to these questions. Any other file format will lead to a loss of 1 point. Non-PDF files that cannot be opened by the TAs will lead to a loss of 3 points.

1.1 Tan, Chapter 3

Exercises 8, 9, 10

1.2 Tan, Chapter 4

Exercise 2, 3, 5.

1.3 Tan, Chapter 5.

Exercise 20.

1.4 Zaki, Chapter 8.

Exercises 1, 4, 6(a), 6(b).

2 Practicum problems (7 points distributed as indicated) Please label your answers clearly, see Homework 0 R notebook for an example (Homework 0 R notebook is available in "Blackboard→Assignment and Projects → Homework 0".) Each answer must be preceded by the R markdown as shown in the Homework 0 R notebook (### Part 2.1-A-ii, for example). Failure to clearly label the answers in the submitted R notebook will lead to a loss of 2 points.

2.1 Decision tree classification (2.5 points divided evenly by all sub-questions in 2.1)

You are provided two datasets from the 1994 US Census database: a training dataset (adult-train.csv) and a testing dataset (adult-test.csv). Each observation of the datasets has 15 attributes as described below. The class variable (response) is stored in the last attribute and indicates whether a person makes more than $50K per year.

The attributes are as follows:

age: Age of the person (numeric)
workclass: Factor, one of Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: Final sampling weight (used by Census Bureau to handle over and under-sampling of particular groups).
education: Factor, one of Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: Number of years of education (numeric).
marital-status: Factor, one of Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Factor, one of Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Factor, one of Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: Factor, one of White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Factor, one of Female, Male
capital-gain: Continuous
capital-loss: Continuous
hours-per-week: Continuous
native-country: Factor, one of United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
income: class variable (response), factor, one of “>50K”, “<=50K”.

You are to build a decision tree using rpart to predict whether a person makes more than $50K per year.

Both the training and testing dataset are not clean; some fields have ‘?’ in them. You will remove those observations that contain ‘?’.

Do the questions below in the order that they appear since there will be dependency between some questions.

First, set the seed as follows before reading in the datasets.

```
> set.seed(1122)
```

(a) Remove all the observations that have ‘?’ in them. Hints: To find out which attributes have a ‘?’ in them, use sum(df$occupation == “?”). If this method returns a non-zero value, the value returned represents the number of times a ‘?’ is seen in that column. Then, use which(df$occupation == “?”) to determine the index of the rows containing the attribute that has the ‘?’. Recall that which() accepts as a parameter a logical vector (or array), and returns the indices where a TRUE occurs in the vector (or array). Consequently, the return value of which() will be a vector of indices. (See R-intro-1.r in Lecture 1 for an example that involves the use of which().) Collect all the indices of the columns where a ‘?’ occurs into a vector, and use that vector to weed out the rows containing the columns with ‘?’

As a sanity check, when you are done with weeding out the ‘?’, you should be left with 30,161 observations in the training set.

Do the same thing for the test dataset. Again, as a sanity check, you should be left with 15,060 observations in the test dataset after you have removed the rows containing ‘?’ in a column.

The rest of the questions below assume that the training and testing datasets are clean.

(b) Build a decision tree model using rpart() to predict whether a person makes <=50K or >50K using all of the predictors. Answer the following questions through model introspection:
(i) Name the top three important predictors in the model?

(ii) The first split is done on which predictor? What is the predicted class of the first node? What is the distribution of observations between the "<=50K" and ">50K" classes at this node?

(c) Use the trained model from (b) to predict the test dataset. Answer the following questions based on the outcome of the prediction and examination of the confusion matrix: (for floating point answers, assume 3 decimal place accuracy):
(i) What is the balanced accuracy of the model? (Note that in our test dataset, we have more observations of class "<=50" than we do of class ">50". Thus, we are more interested in the *balanced accuracy*, instead of just accuracy. Balanced accuracy is calculated as the average of sensitivity and specificity.)
(ii) What is the balanced error rate of the model? (Again, because our test data is imbalanced, a balanced error rate makes more sense. Balanced error rate = 1.0 – balanced accuracy.)
(iii) What is the sensitivity? Specificity?
(iv) What is the AUC of the ROC curve. Plot the ROC curve.

(d) Print the complexity table of the model you trained. Examine the complexity table and state whether the tree would benefit from a pruning. If the tree would benefit from a pruning, at what complexity level would you prune it? If the tree would not benefit from a pruning, provide reason why you think this is the case.
(e) Besides the class imbalance problem we see in the test dataset, we also have a class imbalance problem in the training dataset. To solve this class imbalance problem in the training dataset, we will use undersampling, i.e., we will undersample the majority class such that both classes have the same number of observations in the training dataset.
(i) In the *training* dataset, how many observations are in the class "<=50K"? How many are in the class ">50K"?
(ii) Create a new training dataset that has equal representation of both classes; i.e., number of observations of class "<=50K" must be the same as number of observations of class ">50K". Call this new training dataset. (Use the sample() method on the majority class to sample as many observations as there are in the minority class. **Do not use any other method for undersampling as your results will not match expectation if you do so.**)
(iii) Train a new model on the new training dataset, and then fit this model to the testing dataset. Answer the following questions based on the outcome of the prediction and examination of the confusion matrix: (for floating point answers, assume 3 decimal place accuracy):
i) What is the balanced accuracy of this model?
(ii) What is the balanced error rate of this model?
(iii) What is the sensitivity? Specificity?
(iv) What is the AUC of the ROC curve. Plot the ROC curve.

(f) Comment on the differences in the balanced accuracy, sensitivity, specificity and AUC of the models used in (c) and (e).

2.2 **Random Forest** (2.5 points divided evenly by all sub-questions in 2.2)

Use the same (cleaned) dataset from Question 2.1 for Random Forest (RF). Ensure that the seed is set to 1122 before attempting to do the problems below.

```
> set.seed(1122)
```

(a) Create a RF model using the entire training dataset. When you create the model, use the importance=T parameter to randomForest(). This parameter allows you to examine which variables were important to the model. Once you have created the model, use the testing dataset to predict the response class. Execute the confusionMatrix() method on the predictions versus the true class response and answer the following questions:
(i) What is the balanced accuracy of the model?
(ii) What is the accuracy of the model?
(iii) What is the sensitivity and specificity of the model?
(iv) What is the response class distribution in the test dataset? (That is, how many observations are labeled ">50K" and how many are labeled "<=50K")?
(v) Given the response class distribution, does the sensitivity and specificity make sense?
(vi) RF allow you to see what the variable importance is by running varImpPlot(model). Run this method on the model you created and answer the following questions: For MeanDecreaseAccuracy, which is the most important variable and which is the least important one? For MeanDecreaseGini, which is the most important variable and which is the least important one?

(vii) Examine the model you created by invoking the print() method on it. What is the number of variables tried at each split?

(b) You will now tune the RF model by finding out what is the best value to use for number of predictors to select at each split. To do so, you will use a method called tuneRF(). Make sure you read the manual page for tuneRF(). Invoke tuneRF() as follows:

```
> mtry <- tuneRF(X, Y, ntreeTry=500,
  stepFactor=1.5,improve=0.01, trace=TRUE, plot=TRUE)
> print(mtry)
```

Where X is the training data frame of predictor variables and Y is the response variable. You can look up the remaining options in the manual page, but do not change the value of the remaining options. tuneRF() starts with the default value of mtry and search for the optimal value (with respect to Out-of-Bag error estimate) of mtry for randomForest.
(i) What is the default value of mtry given the number of predictors in your model?
(ii) Once the tuneRF() method completes, examine the *mtry* object by printing it. It will print a table that contains relevant information on OOB (Out of Bag) errors. You want to choose the row that exhibits the lowest OOB error and use the *mtry* value associated with that row. For the tuneRF() method you ran above, what is the optimal value of *mtry* suggested by the method?
(iii) Using the optimal value of *mtry* from (ii), create a RF model by specifying the optimal value from (ii) in the mtry parameter to RF. When you create the model, use the importance=T parameter to randomForest(). Once you have created the model, use the testing dataset to predict the response class. Execute the confusionMatrix() method on the predictions versus the true class response and answer the following questions:
(1) What is the balanced accuracy of the model?
(2) What is the accuracy of the model?
(3) What is the sensitivity and specificity of the model?
(5) Run the varImpPlot(model)and answer the following questions: For MeanDecreaseAccuracy, which is the most important variable and which is the least important one? For MeanDecreaseGini, which is the most important variable and which is the least important one?
(iv) Comment on how the accuracy, balanced accuracy, sensitivity, specificity and variable importance from this model compareto the model you created in 2.2(a).

2.3 **Association Rules** (2.0 points divided evenly by all sub-questions in 2.3)

You are provided a file groceries.csv in which you will find 9,835 transactions for market-basket analysis. Each transaction contains between 1 – 32 items. This file is organized as a transaction database (i.e., each line contains a comma-separated list of items). To read this file into R, use the read.transactions() method in the arules library. (Make sure you install the library if you do not have it already in your R environment, and also make sure you load the library into your R environment for access to the methods.)

(i) Run apriori() on the transaction set. By default, apriori() uses a minimum support value of 0.1. How many rules do you get at this support value?
(ii) Manipulate the support value so you get at least 400 rules. At what support value do you get at least 400 rules?

For the support value used in (ii), answer the following questions:

(iii) Which item is the most frequently bought and what is its frequency?
(iv) Which item is the least frequently bought and what is its frequency?
(v) What are the top 5 rules, sorted by *support*?
(vi) What are the top 5 rules, sorted by *confidence*?

(vii) What are the bottom 5 rules, sorted by *support*?
(viii) What are the bottom 5 rules, sorted by *confidence*?