CS 422: Data Mining

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

Fall 2024: Homework 4 (10 points)

Due date: Thu, Oct 10, 2024 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.

1. Exercises (2 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 2, 3, 5

1.2 Tan, Chapter 4

Exercise 18 (show your work, don't just provide the answer without showing how you derived it).

1.3 Multiclass classification

Using the confusion matrix from multiclass.Rmd notebook (from Lecture 7), create a binary-class confusion matrix using the "one-vs-many" strategy for each class. Then, for each class, compute the sensitivity, specificity and precision to **two decimal places**. Show all work, including the binary class confusion matrices.

2. Practicum problem (8 points)

Decision Trees (Advanced)

You will use the dataset in the file HR-Employee-Attrition.csv for this programming assignment. This dataset consists of 1,470 observations across 35 dimensions. The data is collected from an HR department with the aim of predicting whether the employee will leave the job or not (the variable 'Attrition').

2.1 (a) **[0.25 points]** Read the observations into a data frame. Is the dataset class balanced? Or is it class-imbalanced? Support your answer empirically by noting how many observations you have of each response class. Your function should return a string in the form shown below:

```
yes=XX,no=YY
```

(b) Set the seed to 1121. Create a test and train split where 80% of the data is used for training and the remaining 20% for testing.

(i) **[1.15 point]** Create a decision tree to predict the response variable 'Attrition' using all of the predictors from the training set. Once the decision tree has been created, fit the held-out test dataset to the model and create a confusion matrix. (Use the confusion matrix API from package caret, do not use any other package to create the confusion matrix.) In the confusion matrix, **please make sure that the positive class is 'Yes'.**

Your function will return the confusion matrix object.

(ii) **[0.10 points]** Plot the decision tree using rpart.plot(). Use the following parameters to rpart.plot():

```
extra=104, fallen.leaves = T, type=4, main="hr employee attrition tree"
```

Return the name of the file that contains your plot.

(iii) **[0.25 points]** Between sensitivity and specificity, focus on the value that is lower between the two, and provide some justification for why that is the case. (1 sentence only.) Put the answer to this question in the PDF you submit. Label the answer with "Q2.1(b)(iii) ..."

(iv) **[0.25 points]** Plot a ROC curve and print the area under the curve. Use the same plot parameters shown in sample code. Return an un-named list containing two elements: first element is name of the file that contains your plot, and the second element is the AUC (as a floating point numeric type) rounded up to two decimal digits.

(v) **[0.75 points]** Do you think this is a good model? Justify your answer. (1-2 sentences only.) Put the answer to this question in the PDF you submit. Label the answer with "Q2.1(b)(v) …"

2.2 We will now balance the dataset, i.e., ensure that the number of observations where Attrition = "Yes" are the same as the number of observations where Attrition = "No".

(a) **[2 points]** Set the seed to 1121. We will undersample the majority class to balance the dataset. Create a data frame (let's call it balanced.df) that contains all of the observations where Attrition = "Yes" (237 observations). Then, sample 250 observations from the original dataset where Attrition = "No". Add these 250 observations to balanced.df so you have 487 observations in all.

Note: It could be that after you have created balanced.df, your data frame is such that the first 237 observations all have Attrition = "Yes", and the last 250 observations all have Attrition = "No". When this happens, the sampling process may choose all observations of the same class. To avoid this, shuffle the balanced.df before you use it for (b) below. The following code will shuffle the data frame:

```
balanced.df <- balanced.df[sample(1:nrow(balanced.df)), ]
```

Your function should return the balanced.df data frame.

(b) [**0.25 points]** From balanced.df, create a test and train split where 80% of the data is used for training and the rest for testing.

Using balanced.df, create a decision tree to predict the response variable 'Attrition' using all of the predictors from the training set (balanced_train.df). Once the decision tree has been created, fit the held-out test dataset (balanced_test.df) to the model and print out a confusion matrix. In the confusion matrix, **please make sure that the positive class is 'Yes'.** Once the decision tree has been created, fit the held-out test dataset to the model and create a confusion matrix. (Use the confusion matrix API from package caret, do not use any other package to create the confusion matrix.) In the confusion matrix, please make sure that the positive class is 'Yes'.

Your function will return the confusion matrix object.

(i) **[0.10 points]** Plot a ROC curve and print the area under the curve. Use the same plot parameters shown in sample code. Return an un-named list containing two elements: first element is name of the file that contains your plot, and the second element is the AUC (as a floating point numeric type) rounded up to two decimal digits.

(ii) **[0.35 points]** Do you think this is a good model? Justify your answer. (1-2 sentences only.) Label the answer with "Q2.2(b)(ii) …"

2.3 We will now determine if a pruned tree performs better than the tree in Question 2.2.

(a) **[0.55 points]** For the model created in 2.2 (b), look at the complexity table and determine where to prune the tree. At what value of CP will you prune the tree? Prune the tree at that level of complexity level. Your function should return the complexity level as a floating point numeric type rounded to two decimal digits.

(b) **[1.0 point]** Now, use the pruned tree to fit the held-out test dataset (balanced_test.df) to the model and create a confusion matrix. In the confusion matrix, please make sure that the positive class is 'Yes'. Your function should return the confusion matrix object.

(c) **[1.0 point]** Did the pruning help improve the tree? Please explain your answer in no more than 2-3 sentences. Put the answer to this question in the PDF you submit. Label the answer with "Q2.3(c) …"