CS 422: Data Mining

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

Fall 2021: Homework 6 (10 points)

Due date: Thu, October 28 2021, 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.

Remember: This is an individual assignment. Sharing of code is strictly prohibited, and affected students will loose points if code is determined to have been shared.

1. Questions

1.1 (2 points) Tan, Chapter 4 Exercise 14, 15.

2. Problems

2.1 (8 points) Grid search and Random Forests

For this problem you will use the same model that you used in HW 5, Q 2.3 (the resulting model from pruning).

A grid search is a hyperparameter optimization method; it chooses a set of optimal hyperparameters for a learning algorithm. For example, if you wanted to tune two parameters, **a** and **b**, and **a** can take the values between 1-3, and **b** can take the values between 4-6, you will do a grid search like so:

```
for a = 1 to 3 {
  for b = 4 to 6 {
     result = train_ML_model(a, b)
     save result
  }
}
```

At the end of the loops, you examine result and see which value of **a** and **b** leads to a model you will consider to be best.

You will now implement a RandomForest model to predict whether a booking will be canceled or not. Use the same training and test sets split (90%/10%) used earlier, and the same seed (1122).

Use the same number of predictors you used in Q 2.1 to build the RandomForest model.

You will use two hyper-parameters: ntree (number of trees in the forest) and mtry (randomly chosen attributes for each split).

Use the following values for the ntree: 250, 500, 750

Use the following values for the mtry parameter: $\lfloor\sqrt{n}\rfloor$, $\lfloor\sqrt{n}\rfloor + 1$, $\lfloor\sqrt{n}\rfloor + 2$ (where n is the number of predictors in your base model of Q 2.1).

For each model, save the OOB error estimate you obtained during training, and the confusion matrix that results from fitting the model on the held-out test dataset. (Run the grid search programmatically, i.e., using loops, instead of manually building nine models. At the end of each iteration of the grid search, save the confusion matrix and the OOB estimate in a collection data structure. Hint: examine the err.rate field of the model to determine the mean OOB estimate. This field will contain as many rows as the value of ntree under consideration. The first column in the err.rate field is the OOB error, and the i-th element is the OOB error for all trees up to the i-th tree.)

(i) Determine which model is the **<u>best</u>** by examining balanced accuracy, sensitivity, and specificity as shown in the confusion matrix from the held-out test dataset, and picking the model that shows the **maximum** balanced accuracy, sensitivity and specificity.

(ii) Determine which model is **<u>best</u>** by examining the **lowest (minimum)** OOB error.

(iii) Is the best model as determined by (i) the same model as determined by (ii). If yes, why do you think this may be the case? If no, why do you think this is the case?