STAT303-3 Assignment 3

May 1, 2022

Instructions:

a. You may talk to a friend, discuss the questions and potential directions for solving them. However, you need to write your own solutions and code separately, and not as a group activity.

b. Do not write your name on the assignment. (1 point)

c. Export your Jupyter notebook as a PDF file. If you get an error, make sure you have downloaded the MikTex software (for windows) or MacTex (for mac). Note that after installing MikTex/MacTex, you will need to check for updates, install the updates if needed, and re-start your system. Submit the PDF file. (1 point)

d. Please include each question (or question number) followed by code and your answer (if applicable). Write your code in the 'Code' cells and your answer in the 'Markdown' cells of the Jupyter notebook. Ensure that the solution is well-organized, clear, and concise (3 points)

1. It's easy enough to identify different sections of the homework assignment (e.g., if there are different sections of an assignment, they're clearly distinguishable by section headers or the like)

2. It's clear which code/markdown blocks correspond to which questions.
3. There aren't excessively long outputs of extraneous information (e.g., no printouts of entire data frames without good reason)

This assignment is **due at 11:59pm on Wednesday, May 11th**. Good luck!

Submissions will be graded with a maximum of 40 points – 35 points for code & answers, 5 points for anonymity and proper formatting. However, your final grade in the assignment will be scaled to be out of 100 points. For example, if you scored 30/40 in the assigment, your score will be scaled to 75/100

Q1a

Develop a bagged decision tree model on the data *house_feature_train.csv* to predict *house_price*. Use all the predictors except *house_id*. Follow the steps below:

(i) Make a plot of out-of-bag RMSE (Root mean squared error) vs number of trees. *(3 points for code, 1 point for plot)*

(ii) Based on the plot in (i), develop a bagged tree model with an appropriate number of trees. *(2 points for code)*

(iii) Use the model developed in (ii) to predict house prices in *house_feature_test.csv*. Report the RMSE (should be less than 350). *(1 point for code, 1 point for answer)*

Q1b

Develop a random forest model on the data *house_feature_train.csv* to predict *house_price*. Use all the predictors except *house_id*. For tuning the model, compute the out-of-bag scores (OOB R-squared) for the following set of parameter values:

(i) Number of trees = [75,100,125]

(ii) Number of predictors considered at each split: [1,2,3,4,5],

(iii) Minimum number of observations required in a non-terminal node to split it: [2,3,4,5,6],

(iv) Minimum number of observations required in a leaf: [1,2,3]

Use the parameter values corresponding to the highest OOB score to develop the model. Use the developed model to predict house prices in *house_feature_test.csv*. Report the RMSE (should be less than the RMSE in *Q1a*).

(7 points for code, 1 point for answer)

Q2

Refer to *Q2a of assignment 2*. Make an attempt to tune a random forest model (instead of a single decision tree model as in *Q2a of assignment 2*) that has both **precision and recall higher than 40%** on *train.csv*, *test1.csv* and *test2.csv*. Attempt to tune the number of trees (`n_estimators`), and the number of predictors to consider at each split (`max_features`) to achieve the objective, but do not tune any parameters that prune trees (i.e., do not tune `max_depth`, `max_leaf_nodes`, `min_samples_split`, and `min_samples_leaf`, etc.).

Tune the parameters as follows. Compute the out-of-bag (OOB) precision for different parameter values of `n_estimators`, `max_features` and/or any other parameters that do not prune the tree, and choose the parameter values corresponing to the maximum OOB precision. Show that tuned model fails to meet the objective. You can show that by making a precision-recall plot for the tuned model, where both precision and recall are never simultaneously greater than 40%.

Purpose of this exercise: This exercise shows that pruning a single decision tree model may sometimes (though not usually) lead to a higher reduction in prediction variance (or a more accurate model), as compared to a random forest with unpruned trees. Thus,

(i) A random forest model can sometimes be further improved by pruning individual trees.

(ii) A pruned tree may lead to enough reduction in prediction variance, so that developing a random forest with such pruned trees leads to negligible value addition (this can be shown by a similar exercise).

(7 points for code, 1 point for the precision-recall plot of the optimized random forest model)

Q3

We aim to classify stars as Galaxies or Quasars *(Quasars are extremely luminous active galactic nucleus, powered by a supermassive black hole)* based on their spectral characteristics. The data (reference) consists of observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 9 feature columns and 1 class column which identifies it to be either a galaxy or quasar:

`alpha` = Right Ascension angle (at J2000 epoch)
`delta` = Declination angle (at J2000 epoch)
`u` = Ultraviolet filter in the photometric system
`g` = Green filter in the photometric system
`r` = Red filter in the photometric system
`i` = Near Infrared filter in the photometric system
`z` = Infrared filter in the photometric system
`cam_col` = Camera column to identify the scanline within the run
`class` = object class (galaxy or quasar object)
`redshift` = redshift value based on the increase in wavelength

Develop and tune a random forest model using *Stellar_Classification_train.csv* to classify an observations as a Galaxy (*y=0*) or Quasar (*y=1*). The model must have a ROC-AUC of at least 99% on both *Stellar_Classification_train.csv* and *Stellar_Classification_test.csv*. Print the ROC-AUC for both the datasets.

Hint: Consider 500 trees, and find the value of `max_features` that maximizes the out-of-bag (OOB) ROC-AUC.

(7 points for code, 1 point for answer)

Q4

Can a non-linear monotonic transformation of predictors (such as *log(), sqrt()* etc.) be useful in improving the accuracy of decision tree models?

(3 points for answer)