FUNDAMENTALS OF DATA SCIENCE – DSC 441

HOMEWORK 3

Turn your assignment in as a PDF file to the D2L submission folder by the due date associated with that folder (unless otherwise specified explicitly by the Instructor). Late assignments will incur a penalty. Please ask if you need an extension and come to office hours if you need assistance.

Academic integrity is key to maintaining the impact of your degree. You may discuss the problems and methods with others, but everything you submit must be your own work. For violating, you may incur significant sanctions including failing the course or penalties from the University. If in doubt, ask first.

To get credit, you must clearly label your responses with the questions they answer. Include results and visualizations that you generate, and clearly written answers to any questions included in the problem. Your code file must be submitted alongside your document in D2L.

This homework assignment covers material from Modules 5 and 6. You will begin using of our second type of data mining algorithm, decision trees. There are multiple problems where you will build and evaluate decision trees to give you the chance to develop comfort not only with this particular type of algorithm, but with the general classification and evaluation process.

Problem 1 (15 points):

For this problem, you will perform a straightforward training and evaluation of a decision tree, as well as generate rules by hand. Load the *breast_cancer_updated.csv* data. These data are visual features computed from samples of breast tissue being evaluated for cancer[1]. As a preprocessing step, remove the *IDNumber* column and exclude rows with *NA* from the dataset.

a. Apply decision tree learning (use *rpart*) to the data to predict breast cancer malignancy (*Class)* and report the accuracy using 10-fold cross validation.
b. Generate a visualization of the decision tree.
c. Generate the full set of rules using IF-THEN statements.

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Problem 2 (15 points):

In this problem you will generate decision trees with a set of parameters. You will be using the *storms* data, a subset of the NOAA Atlantic hurricane database[2] , which includes the positions and attributes of 198 tropical storms (potential hurricanes), measured every six hours during the lifetime of a storm. It is part of the *dplyr* library, so load the library and you will be able to access it. As a preprocessing step, view the data and make sure the target variable (*category*) is converted to a factor (as opposed to character string).

a. Build a decision tree using the following hyperparameters, *maxdepth=2*, *minsplit=5* and *minbucket=3*. Be careful to use the right method of training so that you are not automatically tuning the *cp* parameter, but you are controlling the aforementioned parameters specifically. Use cross validation to report your accuracy score. These parameters will result in a relatively small tree.
b. To see how this performed with respect to the individual classes, we could use a confusion matrix. We also want to see if that aspect of performance is different on the train versus the test set. Create a train/test partition. Train on the training set. By making predictions with that model on the train set and on the test set separately, use the outputs to create two separate confusion matrices, one for each partition. Remember, we are testing if the model built with the training data performs differently on data used to train it (train set) as opposed to new data (test set). Compare the confusion matrices and report which classes it has problem classifying. Do you think that both are performing similarly and what does that suggest about overfitting for the model?

Problem 3 (15 points):

This is will be an extension of Problem 2, using the same data and class. Here you will build many decision trees, manually tuning the parameters to gain intuition about the tradeoffs and how these tree parameters affect the complexity and quality of the model. The goal is to find the best tree model, which means it should be accurate but not too complex that the model overfits the training data. We will achieve this by using multiple sets of parameters and creating a graph of accuracy versus complexity for the training and the test sets (refer to the tutorial). This problem may require a significant amount of effort because you will need to train a substantial number of trees (at least 10).

a. Partition your data into 80% for training and 20% for the test data set
b. Train at least 10 trees using different sets of parameters, through you made need more. Create the graph described above such that you can identify the inflection point where the tree is overfitting and pick a high-quality decision tree. Your strategy should be to make at least one very simple model and at least one very complex model and work towards the center by changing different parameters. Generate a table that contains all of the parameters (*maxdepth, minsplit*, *minbucket, etc*) used along with the number of nodes created, and the training and testing set accuracy values. The number of rows will be equal to the number of sets of parameters used. You will use the data in the table to generate the graph. The final results to be reported for this problem are the table and graph.

[2] https://www.nhc.noaa.gov/data/#hurdat

c. Identify the final choice of model, list it parameters and evaluate with a the confusion matrix to make sure that it gets balanced performance over classes. Also get a better accuracy estimate for this tree using cross validation.

Problem 4 (25 points)

In this problem you will identify the most important independent variables used in a classification model. Use the *Bank_Modified.csv* data. As a preprocessing step, remove the *ID* column and make sure to convert the target variable, *approval*, from a string to a factor.

a. Build your initial decision tree model with *minsplit=10 and maxdepth=20*
b. Run variable importance analysis on the model and print the result.
c. Generate a plot to visualize the variables by importance.
d. Rebuild your model with the top six variables only, based on the variable relevance analysis. Did this change have an effect on the accuracy?
e. Visualize the trees from (a) and (d) and report if reducing the number of variables had an effect on the size of the tree?