The Analytics Edge Summer 2024

Test your knowledge of CART and Random Forests

1. The United States government periodically collects demographic information by conducting a census. In this problem, you are going to use census information about an individual to predict how much a person earns - in particular, whether the person earns more than $50,000 per year. This data comes from the UCI Machine Learning Repository. The file **census.csv** contains 1994 census data for 31,978 individuals in the United States. The dataset includes the following 13 variables:

   - **age**: the age of the individual in years
   - **workclass**: the classification of the individual's working status (does the person work for the federal government, work for the local government, work without pay, and so on)
   - **education**: the level of education of the individual (e.g., 5th-6th grade, high school graduate, PhD, so on)
   - **maritalstatus**: the marital status of the individual
   - **occupation**: the type of work the individual does (e.g., administrative/clerical work, farming/fishing, sales and so on)
   - **relationship**: relationship of individual to his/her household
   - **race**: the individual's race
   - **sex**: the individual's sex
   - **capitalgain**: the capital gains of the individual in 1994 (from selling an asset such as a stock or bond for more than the original purchase price)
   - **capitalloss**: the capital losses of the individual in 1994 (from selling an asset such as a stock or bond for less than the original purchase price)
   - **hoursperweek**: the number of hours the individual works per week
   - **nativecountry**: the native country of the individual
   - **over50k**: whether or not the individual earned more than $50,000 in 1994

   (a) Let's begin by building a logistic regression model to predict whether an individual's earnings are above $50,000 (the variable "over50k") using all of the other variables as independent variables. You need to first transform the variable "over50k" to a value 1 when the individual earned more than $50,000 and 0 otherwise. Split the data randomly into a training set and a testing set, setting the seed to 2000 before creating the split. Split the data so that the training set contains 60% of the observations, while the testing set contains 40% of the observations. Use the caTools package to do so such that the "over50k" variable is balanced in the training and testing set. Next, build a logistic regression model to predict the dependent variable "over50k", using all of the other variables in the dataset as independent variables. Use the training set to build the model. What is the log-likelihood of the model? (Use 0.1 as your significance threshold. You might see a warning message here - you can ignore it and proceed. This message is a warning that we might be overfitting our model to the training set.)

(b) What is the accuracy of the model on the testing set? Use a threshold of 0.5. (You might see a warning message when you make predictions on the test set - you can safely ignore it.)

(c) What is the baseline accuracy on the testing set?

(d) What is the area-under-the-curve (AUC) for the logistic regression model on the test set?

(e) We have just seen how the logistic regression model for this data achieves a high accuracy. Moreover, the significances of the variables give us a way to gauge which variables are relevant for this prediction task. However, it is not immediately clear which variables are more important than the others, especially due to the large number of factor variables in this problem. Let us now build a classification tree to predict "over50k". Use the training set to build the model, and all of the other variables as independent variables. Use the default parameters. After you are done building the model, plot the resulting tree.

(f) How many splits does the tree have in total? How many leaf nodes does the tree have in total?

(g) Which variable does the tree split on at the first level (the very first split of the tree)?

(h) Which variables does the tree split on at the second level (immediately after the first split of the tree)?

(i) What is the accuracy of the model on the testing set? Use a threshold of 0.5.

(j) Let us now consider the ROC curve and AUC for the CART model on the test set. You will need to get predicted probabilities for the observations in the test set to build the ROC curve and compute the AUC. Plot the ROC curve for the CART model you have estimated. Also plot the ROC curve for logistic regression. Observe that compared to the logistic regression ROC curve, the CART ROC curve is less smooth. Which of the following explanations for this behavior is most correct?

- The number of variables that the logistic regression model is based on is larger than the number of variables used by the CART model, so the ROC curve for the logistic regression model will be smoother.
- CART models require a higher number of observations in the testing set to produce a smoother/more continuous ROC curve; there is simply not enough data.
- The probabilities from the CART model take only a handful of values (five, one for each end bucket/leaf of the tree); the changes in the ROC curve correspond to setting the threshold to one of those values.

- The CART model uses fewer continuous variables than the logistic regression model (capitalgain for CART versus age, capitalgain, capitallosses, hoursperweek), which is why the CART ROC curve is less smooth than the logistic regression one.

(k) What is the AUC of the CART model on the test set?

(l) Before building a random forest model, we'll down-sample our training set. While some modern personal computers can build a random forest model on the entire training set, others might run out of memory when trying to train the model since random forests is much more computationally intensive than CART or Logistic Regression. For this reason, before continuing we will define a new training set to be used when building our random forest model, that contains 2000 randomly selected observations from the original training set. Do this by running the following commands in your R console (assuming your training set is called "train"):

```
> set.seed(1)
> trainSmall <− train[sample(nrow(train), 2000), ]
```

Let us now build a random forest model to predict "over50k", using the dataset "trainSmall" to build the model. Set the seed to 1 again right before building the model, and use all of the other variables in the dataset as independent variables. (If you get an error that random forest "can not handle categorical predictors with more than 32 categories", rebuild the model without the nativecountry variable as one of the independent variables.) Then, make predictions using this model on the entire test set. What is the accuracy of the model on the test set, using a threshold of 0.5? (Remember that you don't need a "type" argument when making predictions with a random forest model if you want to use a threshold of 0.5. Also, note that your accuracy might be different from since the random forest models can still differ depending on your operating system, even when the random seed is set.)

(m) As we discussed in class, random forest models work by building a large collection of trees. As a result, we lose some of the interpretability that comes with CART in terms of seeing how predictions are made and which variables are important. However, we can still compute metrics that give us insight into which variables are important. One metric that we can look at is the number of times, aggregated over all of the trees in the random forest model, that a certain variable is selected for a split. To view this metric, run the following lines of R code (replace "MODEL" with the name of your random forest model):

```
> vu <− varUsed(MODEL, count=TRUE)
> vusorted <− sort(vu, decreasing = FALSE, index.return = TRUE)
> dotchart(vusorted$x, names(MODEL$forest$xlevels[vusorted$ix]))
```

This code produces a chart that for each variable measures the number of times that variable was selected for splitting (the value on the x-axis). Which of the variables is the most important in terms of the number of splits?

(n) A different metric we can look at is related to "impurity", which measures how homogenous each bucket or leaf of the tree is. In each tree in the forest, whenever we select a variable and perform a split, the impurity is decreased. Therefore, one way to measure the importance of a variable is to average the reduction in impurity, taken over all the times that variable is selected for splitting in all of the trees in the forest. To compute this metric, run the following command in R (replace "MODEL" with the name of your random forest model):

```
> varImpPlot(MODEL)
```

Which of the following variables is the most important in terms of mean reduction in impurity?

(o) We now conclude our study of this dataset by looking at how CART behaves with different choices of its parameters. Let us select the cost complexity parameter for our CART model using k-fold cross validation, with k = 10 folds. Modify the minimum complexity parameter cp = 0.0001. Suggest a reasonable value of the cost complexity parameter from the plot and plot the corresponding tree.

(p) What is the prediction accuracy on the test set? Comment on how the model compares with the accuracy of the original CART model.

2. In this problem, you will fit regression trees to the **Boston** dataset. This data was part of an important paper in 1978 by Harrison and Rubinfeld titled - "Hedonic housing prices and the demand for clean air" published in the Journal of Environmental Economics and Management 5(1): 81-102. The dataset has the following fields:

   - **crim**: per capita crime rate by town
   - **zn**: proportion of residential land zoned for lots over 25,000 sq.ft
   - **indus**: proportion of non-retail business acres per town
   - **chas**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
   - **nox**: nitrogen oxides concentration (parts per 10 million)
   - **rm**: average number of rooms per dwelling
   - **age**: proportion of owner-occupied units built prior to 1940
   - **dis**: weighted mean of distances to five Boston employment centres
   - **rad**: index of accessibility to radial highways
   - **tax**: full-value property-tax rate per $10,000
   - **ptratio**: pupil-teacher ratio by town
   - **black**: $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
   - **lstat**: lower status of the population (percent)
   - **medv**: median value of owner-occupied homes in $1000s

   We will try to predict the median house value using thirteen predictors

(a) Use a seed of 1. Split the dataset into a training set and a test set using the **sample** function where half the observations lie in each set. Fit a regression tree to the training set. Plot the tree. Which predictor variables were used in the regression tree?

(b) What is the test set mean squared error? Draw a scatter plot of the fitted and true values.

(c) Use the default settings and cross-validation in order to determine the optimal level of tree complexity. Would you prune the tree based on the result?

(d) If you decide to prune the tree in the previous step, plot the new tree. What is the test set mean squared error? Compare with the result in part (b).

(e) Use random forests to analyze this data. Set the seed to 1 before running the method. What test set mean squared error do you obtain? How does this compare to the CART model? How many variables does the randomForest function try at each split?

(f) Use the **importance()** function to determine the two variables which are most important. Plot the importance measures using the **varImpPlot()**.

(g) Describe the effect of the number of variables considered at each split controlled by the **mtry** argument in **randomForest()**, on the error obtained.

3. In this question, we will look at the data on the US Supreme Court decisions from 1994 to 2001 and build a predictive model to forecast Supreme Court decisions. The data is provided in the file **supremeexercise.csv** and contains the following variables:

- **docket**: Case number
- **term**: Case year
- **party_1**: First party in the case
- **party_2**: Second party in the case
- **rehndir**: Direction of Judge Rehnquist ruling (1 = conservative, 0 = liberal)
- **stevdir**: Direction of Judge Stevens ruling (1 = conservative, 0 = liberal)
- **ocondir**: Direction of Judge O'Connors ruling (1 = conservative, 0 = liberal)
- **scaldir**: Direction of Judge Scalia ruling (1 = conservative, 0 = liberal)
- **kendir**: Direction of Judge Kennedy ruling (1 = conservative, 0 = liberal)
- **soutdir**: Direction of Judge Souter ruling (1 = conservative, 0 = liberal)
- **thomdir**: Direction of Judge Thomas ruling (1 = conservative, 0 = liberal)
- **gindir**: Direction of Judge Ginsburg ruling (1 = conservative, 0 = liberal)
- **brydir**: Direction of Judge Breyer ruling (1 = conservative, 0 = liberal)
- **petit**: Petitioner type (BUSINESS, CITY, DEF (defendant), EE (employee), ER (employer), INDIAN, IP (injured person), OF (official), OTHER, POL (politician), STATE, US)
- **respon**: Respondent type (Same as petitioner types)

- **circuit**: Circuit of origin of case (1st-11th, DC, and FED)
- **unconst**: Case argued to be as unconstitutional by law by petitioner (1 = yes, 0 = no)
- **lctdir**: Lower Court direction of ruling (liberal, conser)
- **issue**: Issue of the case (AT = attorneys, CP = criminal procedure, CR = civil rights, DP = due process, ECN = economic activity, FA = first amendment, FED = federalism, IR = interstate relations, JUD = judicial power, PRIV = privacy, TAX = federal taxation, UN = unions)
- **result**: Result of the case (1 = conservative, 0 = liberal)

(a) Read the data into the dataframe **supreme**. What is the fraction of cases in which the Supreme Court reversed the decision of the Lower Court?

(b) Define a new variable **unCons** that takes a value of 1 if the decision made by the judges was an unanimous conservative decision and 0 otherwise. What is the total number of cases that had an unanimous conservative decision?

(c) Define a new variable **unLib** that takes a value of 1 if the decision made by the judges was an unanimous liberal decision and 0 otherwise. What is the total number of cases that had an unanimous liberal decision?

(d) You will now develop a two step CART model for this data. In the first step, you will build two classification trees to predict the unanimous conservative and liberal decisions respectively and in the second step, you will build nine judge-specific trees to predict the outcome for cases for which the predictions from the first step are ambiguous. Start by building a CART model to predict **unCons** using the six predictor variables **petit**, **respon**, **circuit**, **unconst**, **lctdir** and **issue**. Use the rpart package in R to build the model. Use the default parameter settings to build the CART model. Remember that you want to build a classification tree rather than a regression tree. Use all the observations to build the model. How many node splits are there in the resulting tree?

(e) List all the variables that this tree splits on.

(f) What is the area under the curve for the receiver operating characteristic (ROC) curve for this model?

(g) Similarly build a CART model to predict **unLib** using the six variables **petit**, **respon**, **circuit**, **unconst**, **lctdir** and **issue** as the predictor variables. Use the default parameter settings to build the CART model. Use all the observations to build the model. Which variable does the tree split on at the first level?

(h) Using the CART tree plot for the model in question (g), identify the leaf node with the fewest number of observations in it. What is the fraction of cases that has an unanimous liberal decision at this node?

(i) We will now combine the results from the two trees. What is the total number of cases where the two trees predict an unanimous outcome for the conservative and liberal judgement simultaneously, thus contradicting each other?

(j) What is the total number of cases where neither tree predicts an unanimous outcome?

(k) We now build the second part of our model which is nine judge-specific classification trees to provide predictions for the cases when either both trees predict an unanimous outcome or neither does (the harder cases). Build a CART model to predict each of the variables **rehndir** up to **brydir** using the six predictor variables **petit**, **respon**, **circuit**, **unconst**, **lctdir** and **issue**. Build you model using only those cases identified in questions (i) and (j). Use the majority of the judge predictions to make a prediction for each of these cases. What is the accuracy of the model on these cases?

(l) What is the overall accuracy of your two step CART model?

(m) We consider the Moseley versus Victoria Secret Catalogue case in 2002, the details of which are as follows: The owner of the "Victoria's Secret" **business** brought a trademark dilution action against Victor Moseley in the Lower Court. They claimed that the "Victoria's Secret" trademark was diluted and tarnished by Moseley's adult specialty **business** named "Victor's Secret". The U.S. Lower Court for the **Sixth Circuit** ruled the judgment in a **conservative** direction by ruling in favor of Victoria's Secret. Moseley petitioned against this decision to the Supreme Court. Use your two step model to predict the outcome of this case which deals with **economic activities**. You can look at the tree plots to make your conclusion.

(n) We will now build a random forest model directly to predict the outcome **result** using the six predictor variables **petit**, **respon**, **circuit**, **unconst**, **lctdir** and **issue**. Use all the observations to build you model. Use the default settings to build the random forest model. What is the accuracy of the model?

(o) The CART model and the random forest models have their respective advantages. Briefly provide one reason each as to why the CART model might be preferred to the random forest model and one reason why the random forest model might be preferred to the CART model.