The Analytics Edge

Test your knowledge of CART and Random Forests in R

Note to all. I have compiled the answers in the following format – for each question, the qualitative or "written" solutions will be provided together with their sub-questions. The R scripts (as well as the console outputs) will be provided *after* each whole question, followed by all the relevant plots. If I have missed anything in the solutions, or if you have any questions, you may email me at benjamin_tanwj@mymail.sutd.edu.sg. Thank you!

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. 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. 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. Identify all the variables that are significant, or have factors that are significant? (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.)

Solution. The significant variables are:

- *age*
- *workclass (factor variable):*
  - *Federal-gov, Local-gov, Private, Self-emp-inc, State-gov*
- *education (factor variable):*
  - *12th, Assoc-acdm, Assoc-voc, Bachelors, Doctorate, HS-grad, Masters, Prof-school, Some-college*
- *maritalstatus (factor variable):*
  - *Married-AF-spouse, Married-civ-spouse, Never-married*
- *occupation (factor variable):*
  - *Exec-managerial, Farming-fishing, Other-service, Prof-speciality, Protective-serv, Sales, Tech-support*
- *relationship (factor variable):*
  - *Not-in-family, Own-child, Unmarried, Wife*
- *sex (factor variable):*
  - *Male*
- *capitalgain*
- *capitalloss*
- *hoursperweek*

(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.)

Solution. The confusion matrix is as follows:

	$FALSE$	$TRUE$
$\leq 50K$	9051	662
$> 50K$	1190	1888

$$Accuracy = sum(diag(table1))/sum(table1) = 0.8552$$

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

Solution. Baseline accuracy is 0.7593.

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

Solution. AUC for the model on the test set is given as 0.9061.

(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.

Solution. Plot found here.

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

Solution. Total number of splits is 4.

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

Solution. At the top level, the tree splits on the *relationship* variable.

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

Solution. At the second level, the tree splits on the *capitalgain* and *education* variables.

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

Solution. The confusion matrix is as follows:

	$FALSE$	$TRUE$
$\leq 50K$	9243	470
$> 50K$	1480	1596

$$Accuracy = sum(diag(table2))/sum(table2) = 0.8473$$

(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. 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.

Solution. Plots found here. The plots indicate that the ROC for logistic regression is more smooth. By tabulating the probability results (as opposed to majority votes) from the CART model, we can see that there are only 5 possible predicted probability values. Since they can only take on these handful of values, the breakpoints of the curve correspond to the TPR and FPR when the threshold is set to these values. Option 3 is correct.

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

Solution. The AUC is given as 0.8470.

(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.)

Solution. The confusion matrix is as follows:

	$FALSE$	$TRUE$
$\leq 50K$	9586	127
$> 50K$	1985	1093

$$Accuracy = sum(diag(table3))/sum(table3) = 0.834$$

(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?

Solution. The first command *varUsed* is part of the random forest package and helps identify which variables are actually used in the random forest (by providing frequencies of variables used).

The second command gives sorted frequencies while *dotchart* draws a Cleveland dot plot – like a bar chart, but with dots – and we can see that *age* is the most important variable in random forests in terms of splits. Plot found here.

(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?

Solution. From the dot chart here, we see that in terms of average reduction in impurity, the *occupation* variable is the most important. While *age* and *occupation* are important in both models, the order of importance changes.

(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.

Solution. From the figure here, I am proposing to prune the tree at $cp = 0.002$ roughly.

(p) What is the prediction accuracy on the test set? Comment on how the model compares with the model in part (e).

Solution. Confusion matrix is as follows (predicted as columns, actuals as rows)

	$\leq 50K$	$> 50K$
$\leq 50K$	9178	535
$> 50K$	1240	1838

$$Accuracy = sum(diag(table4))/sum(table4) = 0.8612$$

The accuracy on the test set is greater but the model is much more complicated and less interpretable.

R Scripts.

```
> #1)
> #a)
> census <- read.csv("census.csv")
> str(census)
'data.frame': 31978 obs. of  13 variables:
 $ age          : int  39 50 38 53 28 37 49 52 31 42 ...
```

```
 $ workclass    : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
 $ education    : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
 $ maritalstatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ occupation   : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
 $ relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
 $ race         : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
 $ sex          : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ capitalgain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ capitalloss  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hoursperweek : int  40 13 40 40 40 40 16 45 50 40 ...
 $ nativecountry: Factor w/ 41 levels " Cambodia"," Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
 $ over50k      : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
> library(caTools)
> set.seed(2000)
> spl <- sample.split(census$over50k, SplitRatio = 0.6)
> train <- subset(census, spl == TRUE)
> test <- subset(census, spl == FALSE)
> census.glm <- glm(over50k ~ ., family = "binomial", data = train)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(census.glm)

Call:
glm(formula = over50k ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.1065  -0.5037  -0.1804  -0.0008   3.3383

Coefficients: (1 not defined because of singularities)
                                        Estimate Std. Error z value Pr(>|z|)
(Intercept)                           -8.658e+00  1.379e+00  -6.279 3.41e-10 ***
age                                    2.548e-02  2.139e-03  11.916  < 2e-16 ***
workclass Federal-gov                  1.105e+00  2.014e-01   5.489 4.03e-08 ***
workclass Local-gov                    3.675e-01  1.821e-01   2.018 0.043641 *
workclass Never-worked                -1.283e+01  8.453e+02  -0.015 0.987885
workclass Private                      6.012e-01  1.626e-01   3.698 0.000218 ***
workclass Self-emp-inc                 7.575e-01  1.950e-01   3.884 0.000103 ***
workclass Self-emp-not-inc             1.855e-01  1.774e-01   1.046 0.295646
workclass State-gov                    4.012e-01  1.961e-01   2.046 0.040728 *
workclass Without-pay                 -1.395e+01  6.597e+02  -0.021 0.983134
education 11th                         2.225e-01  2.867e-01   0.776 0.437738
education 12th                         6.380e-01  3.597e-01   1.774 0.076064 .
education 1st-4th                     -7.075e-01  7.760e-01  -0.912 0.361897
```

education 5th-6th	-3.170e-01	4.880e-01	-0.650	0.516008	
education 7th-8th	-3.498e-01	3.126e-01	-1.119	0.263152	
education 9th	-1.258e-01	3.539e-01	-0.355	0.722228	
education Assoc-acdm	1.602e+00	2.427e-01	6.601	4.10e-11	***
education Assoc-voc	1.541e+00	2.368e-01	6.506	7.74e-11	***
education Bachelors	2.177e+00	2.218e-01	9.817	< 2e-16	***
education Doctorate	2.761e+00	2.893e-01	9.544	< 2e-16	***
education HS-grad	1.006e+00	2.169e-01	4.638	3.52e-06	***
education Masters	2.421e+00	2.353e-01	10.289	< 2e-16	***
education Preschool	-2.237e+01	6.864e+02	-0.033	0.973996	
education Prof-school	2.938e+00	2.753e-01	10.672	< 2e-16	***
education Some-college	1.365e+00	2.195e-01	6.219	5.00e-10	***
maritalstatus Married-AF-spouse	2.540e+00	7.145e-01	3.555	0.000378	***
maritalstatus Married-civ-spouse	2.458e+00	3.573e-01	6.880	6.00e-12	***
maritalstatus Married-spouse-absent	-9.486e-02	3.204e-01	-0.296	0.767155	
maritalstatus Never-married	-4.515e-01	1.139e-01	-3.962	7.42e-05	***
maritalstatus Separated	3.609e-02	1.984e-01	0.182	0.855672	
maritalstatus Widowed	1.858e-01	1.962e-01	0.947	0.343449	
occupation Adm-clerical	9.470e-02	1.288e-01	0.735	0.462064	
occupation Armed-Forces	-1.008e+00	1.487e+00	-0.677	0.498170	
occupation Craft-repair	2.174e-01	1.109e-01	1.960	0.049972	*
occupation Exec-managerial	9.400e-01	1.138e-01	8.257	< 2e-16	***
occupation Farming-fishing	-1.068e+00	1.908e-01	-5.599	2.15e-08	***
occupation Handlers-cleaners	-6.237e-01	1.946e-01	-3.204	0.001353	**
occupation Machine-op-inspct	-1.862e-01	1.376e-01	-1.353	0.176061	
occupation Other-service	-8.183e-01	1.641e-01	-4.987	6.14e-07	***
occupation Priv-house-serv	-1.297e+01	2.267e+02	-0.057	0.954385	
occupation Prof-specialty	6.331e-01	1.222e-01	5.180	2.22e-07	***
occupation Protective-serv	6.267e-01	1.710e-01	3.664	0.000248	***
occupation Sales	3.276e-01	1.175e-01	2.789	0.005282	**
occupation Tech-support	6.173e-01	1.533e-01	4.028	5.63e-05	***
occupation Transport-moving	NA	NA	NA	NA	
relationship Not-in-family	7.881e-01	3.530e-01	2.233	0.025562	*
relationship Other-relative	-2.194e-01	3.137e-01	-0.699	0.484263	
relationship Own-child	-7.489e-01	3.507e-01	-2.136	0.032716	*
relationship Unmarried	7.041e-01	3.720e-01	1.893	0.058392	.
relationship Wife	1.324e+00	1.331e-01	9.942	< 2e-16	***
race Asian-Pac-Islander	4.830e-01	3.548e-01	1.361	0.173504	
race Black	3.644e-01	2.882e-01	1.265	0.206001	
race Other	2.204e-01	4.513e-01	0.488	0.625263	
race White	4.108e-01	2.737e-01	1.501	0.133356	
sex Male	7.729e-01	1.024e-01	7.545	4.52e-14	***
capitalgain	3.280e-04	1.372e-05	23.904	< 2e-16	***
capitalloss	6.445e-04	4.854e-05	13.277	< 2e-16	***

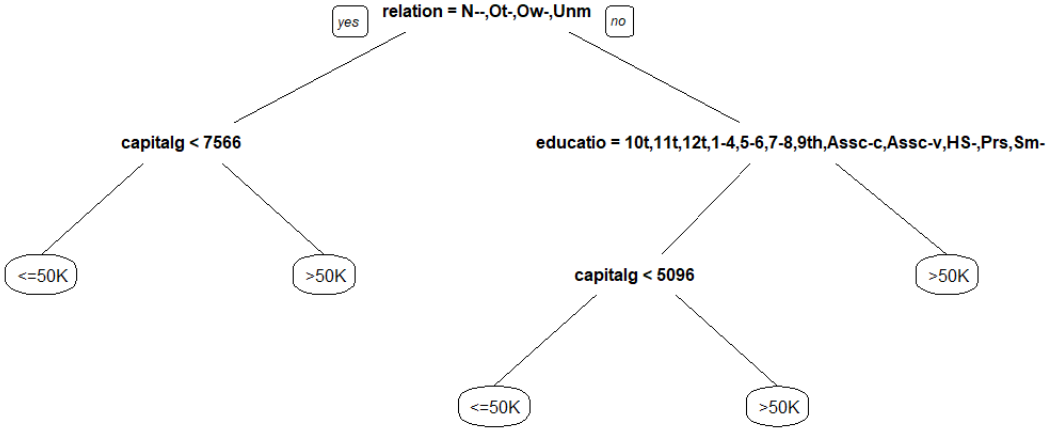
```
hoursperweek                                2.897e-02  2.101e-03  13.791  < 2e-16 ***
nativecountry Canada                        2.593e-01  1.308e+00   0.198 0.842879
nativecountry China                        -9.695e-01  1.327e+00  -0.730 0.465157
nativecountry Columbia                     -1.954e+00  1.526e+00  -1.280 0.200470
nativecountry Cuba                          5.735e-02  1.323e+00   0.043 0.965432
nativecountry Dominican-Republic           -1.435e+01  3.092e+02  -0.046 0.962972
nativecountry Ecuador                      -3.550e-02  1.477e+00  -0.024 0.980829
nativecountry El-Salvador                  -6.095e-01  1.395e+00  -0.437 0.662181
nativecountry England                      -6.707e-02  1.327e+00  -0.051 0.959686
nativecountry France                        5.301e-01  1.419e+00   0.374 0.708642
nativecountry Germany                       5.474e-02  1.306e+00   0.042 0.966572
nativecountry Greece                       -2.646e+00  1.714e+00  -1.544 0.122527
nativecountry Guatemala                    -1.293e+01  3.345e+02  -0.039 0.969180
nativecountry Haiti                        -9.221e-01  1.615e+00  -0.571 0.568105
nativecountry Holand-Netherlands           -1.282e+01  2.400e+03  -0.005 0.995736
nativecountry Honduras                     -9.584e-01  3.412e+00  -0.281 0.778775
nativecountry Hong                         -2.362e-01  1.492e+00  -0.158 0.874155
nativecountry Hungary                       1.412e-01  1.555e+00   0.091 0.927653
nativecountry India                        -8.218e-01  1.314e+00  -0.625 0.531661
nativecountry Iran                         -3.299e-02  1.366e+00  -0.024 0.980736
nativecountry Ireland                       1.579e-01  1.473e+00   0.107 0.914628
nativecountry Italy                         6.100e-01  1.333e+00   0.458 0.647194
nativecountry Jamaica                      -2.279e-01  1.387e+00  -0.164 0.869467
nativecountry Japan                         5.072e-01  1.375e+00   0.369 0.712179
nativecountry Laos                         -6.831e-01  1.661e+00  -0.411 0.680866
nativecountry Mexico                       -9.182e-01  1.303e+00  -0.705 0.481103
nativecountry Nicaragua                    -1.987e-01  1.507e+00  -0.132 0.895132
nativecountry Outlying-US(Guam-USVI-etc)   -1.373e+01  8.502e+02  -0.016 0.987115
nativecountry Peru                         -9.660e-01  1.678e+00  -0.576 0.564797
nativecountry Philippines                   4.393e-02  1.281e+00   0.034 0.972640
nativecountry Poland                        2.410e-01  1.383e+00   0.174 0.861624
nativecountry Portugal                      7.276e-01  1.477e+00   0.493 0.622327
nativecountry Puerto-Rico                  -5.769e-01  1.357e+00  -0.425 0.670837
nativecountry Scotland                     -1.188e+00  1.719e+00  -0.691 0.489616
nativecountry South                        -8.183e-01  1.341e+00  -0.610 0.541809
nativecountry Taiwan                       -2.590e-01  1.350e+00  -0.192 0.847878
nativecountry Thailand                     -1.693e+00  1.737e+00  -0.975 0.329678
nativecountry Trinadad&Tobago              -1.346e+00  1.721e+00  -0.782 0.434105
nativecountry United-States                -8.594e-02  1.269e+00  -0.068 0.946020
nativecountry Vietnam                      -1.008e+00  1.523e+00  -0.662 0.507799
nativecountry Yugoslavia                    1.402e+00  1.648e+00   0.851 0.394874
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 21175  on 19186  degrees of freedom
Residual deviance: 12104  on 19090  degrees of freedom
AIC: 12298

Number of Fisher Scoring iterations: 15


> #b)
> pred.glm <- predict(census.glm, newdata = test, type = "response")
Warning message:
In predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type ==  :
  prediction from a rank-deficient fit may be misleading
> table1 <- table(test$over50k, pred.glm >= 0.5)
> sum(diag(table1))/sum(table1)
[1] 0.8552107

> #c)
> table(train$over50k)

 <=50K   >50K
 14570   4617
> table(test$over50k)

 <=50K   >50K
  9713   3078
> 9713/(9713 + 3078)
[1] 0.7593621

> #d)
> library(ROCR)
> ROCRpred <- prediction(pred.glm, test$over50k)
> performance(ROCRpred, "auc")@y.values
[[1]]
[1] 0.9061598


> #e)
> library(rpart)
> library(rpart.plot)
> tree1 <- rpart(over50k ~., data = train)
> prp(tree1)
```

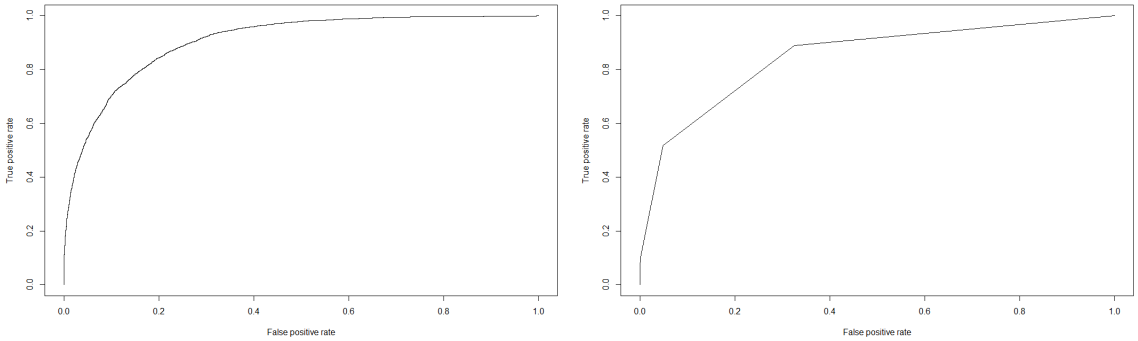
```
> #i)
> pred.tree <- predict(tree1, newdata = test, type = "class")
> table2 <- table(test$over50k, pred.tree)
> sum(diag(table2))/sum(table2)
[1] 0.8473927

> #j)
> perf1 <- performance(ROCRpred, measure = "tpr", x.measure = "fpr")
> plot(perf1)
> pred.tree1 <- predict(tree1, newdata = test, type = "prob")
> ROCRpred2 <- prediction(pred.tree1[,2], test$over50k)
> perf2 <- performance(ROCRpred2, measure = "tpr", x.measure = "fpr")
> plot(perf2)

> #k)
> performance(ROCRpred2, measure = "auc")@y.values
[[1]]
[1] 0.8470256


> #l)
> set.seed(1)
> trainSmall <- train[sample(nrow(train), 2000),]
> library(randomForest)
> set.seed(1)
> censusrf <- randomForest(over50k ~ ., data = trainSmall)
> pred.rf <- predict(censusrf, newdata = test)
> table3 <- table(test$over50k, pred.rf)
> sum(diag(table3))/sum(table3)
[1] 0.8348839

> #m)
> vu <- varUsed(censusrf, count = TRUE)
> vusorted <- sort(vu, decreasing = FALSE, index.return = TRUE)
> dotchart(vusorted$x, names(censusrf$forest$xlevel[vusorted$ix]))

> #n)
> varImpPlot(censusrf)

> #o)
> tree2 <- rpart(over50k ~., data = train, cp=0.0001)
> printcp(tree2)

Classification tree:
```

```
rpart(formula = over50k ~ ., data = train, cp = 1e-04)

Variables actually used in tree construction:
 [1] age           capitalgain   capitalloss   education     hoursperweek
 [6] maritalstatus nativecountry occupation    race          relationship
[11] sex           workclass

Root node error: 4617/19187 = 0.24063

n= 19187

           CP nsplit rel error  xerror     xstd
1  0.12183236      0   1.00000 1.00000 0.012825
2  0.06562703      2   0.75634 0.77020 0.011658
3  0.03747022      3   0.69071 0.71107 0.011299
4  0.00758068      4   0.65324 0.65952 0.010962
5  0.00595625      8   0.62292 0.64306 0.010850
6  0.00433182     10   0.61100 0.63201 0.010774
7  0.00422352     11   0.60667 0.61317 0.010640
8  0.00346545     13   0.59822 0.60819 0.010604
9  0.00324886     16   0.58696 0.60602 0.010589
10 0.00216591     17   0.58371 0.59866 0.010535
11 0.00194932     18   0.58155 0.60191 0.010559
12 0.00151614     19   0.57960 0.59974 0.010543
13 0.00144394     24   0.57050 0.59974 0.010543
14 0.00129955     28   0.56444 0.60104 0.010552
15 0.00108295     29   0.56314 0.59736 0.010525
16 0.00103964     31   0.56097 0.60191 0.010559
17 0.00086636     39   0.55187 0.59757 0.010527
18 0.00075807     62   0.52697 0.59887 0.010537
19 0.00064977     66   0.52393 0.60017 0.010546
20 0.00054148     70   0.52090 0.60039 0.010548
21 0.00051982     80   0.51527 0.59996 0.010544
22 0.00043318     96   0.50162 0.59931 0.010540
23 0.00039708    109   0.49556 0.60039 0.010548
24 0.00032489    116   0.49274 0.60667 0.010593
25 0.00030323    135   0.48625 0.61057 0.010621
26 0.00028879    140   0.48473 0.61035 0.010620
27 0.00025991    154   0.48018 0.61057 0.010621
28 0.00021659    164   0.47693 0.62313 0.010711
29 0.00017327    191   0.47109 0.62421 0.010719
30 0.00016244    205   0.46827 0.62833 0.010748
31 0.00015471    215   0.46524 0.62920 0.010754
32 0.00014439    242   0.45896 0.62876 0.010751
```

```
33 0.00010830    255   0.45701 0.63288 0.010780
34 0.00010000    289   0.45332 0.64306 0.010850
> plotcp(tree2)
> tree3 <- prune(tree2, cp = 0.002)
> prp(tree3)

> #p)
> pred.tree3 <- predict(tree3, newdata = test, type = "class")
> table4 <- table(test$over50k, pred.tree3)
> table4
        pred.tree3
          <=50K  >50K
   <=50K   9178   535
   >50K    1240  1838
```



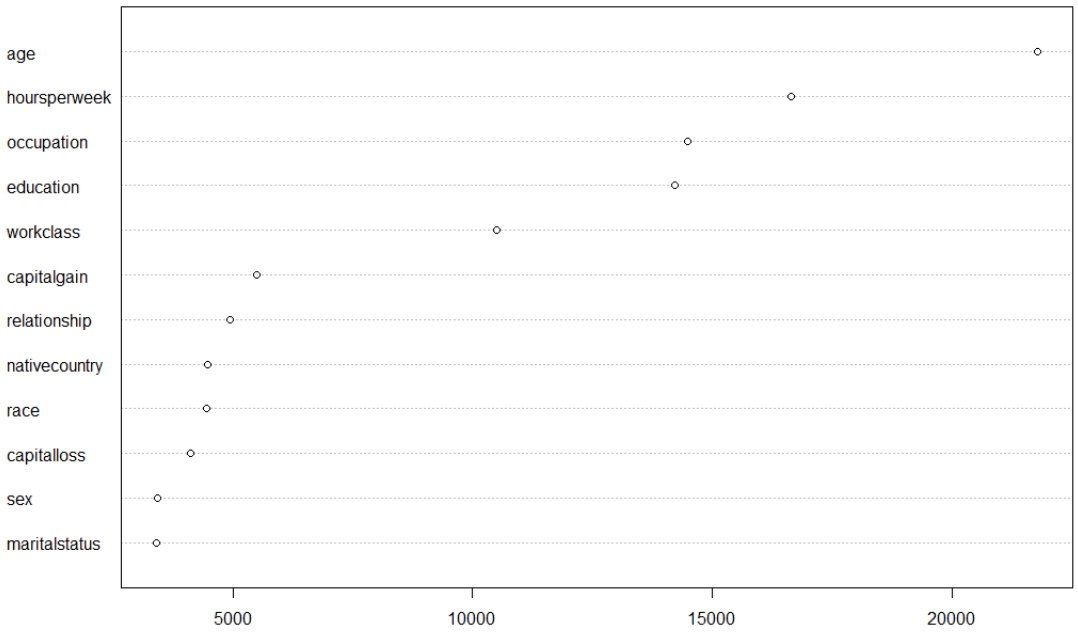
Plot for Q1e. Click here to go back to the question.



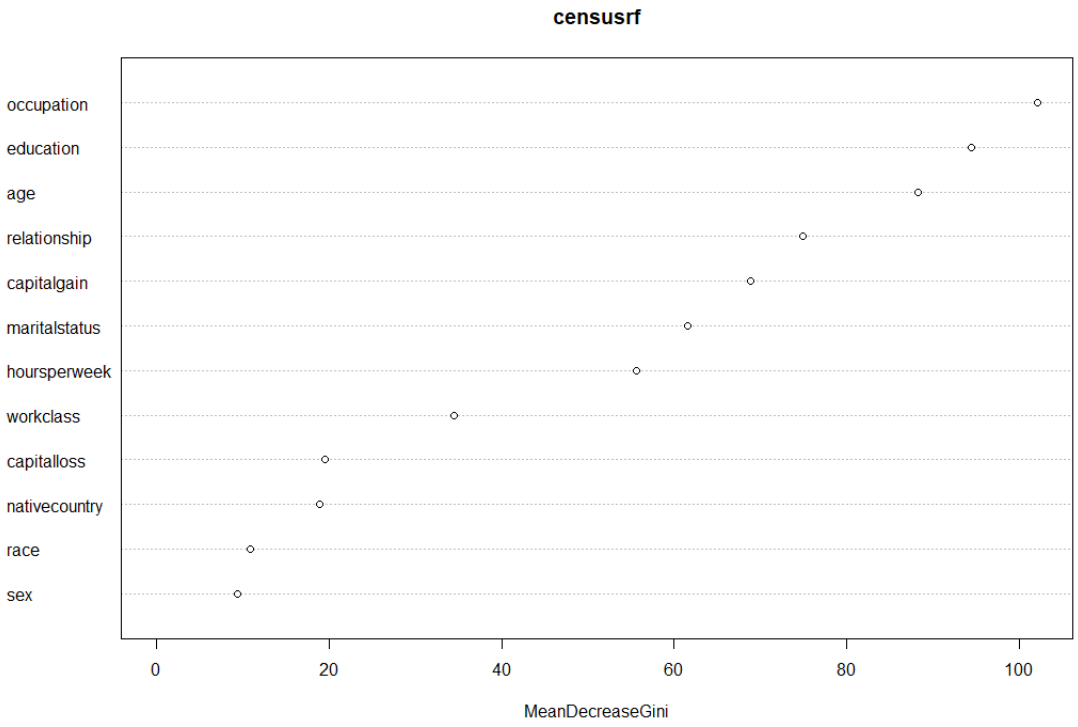
(a) ROC for the logistic regression model

(b) ROC for the CART model

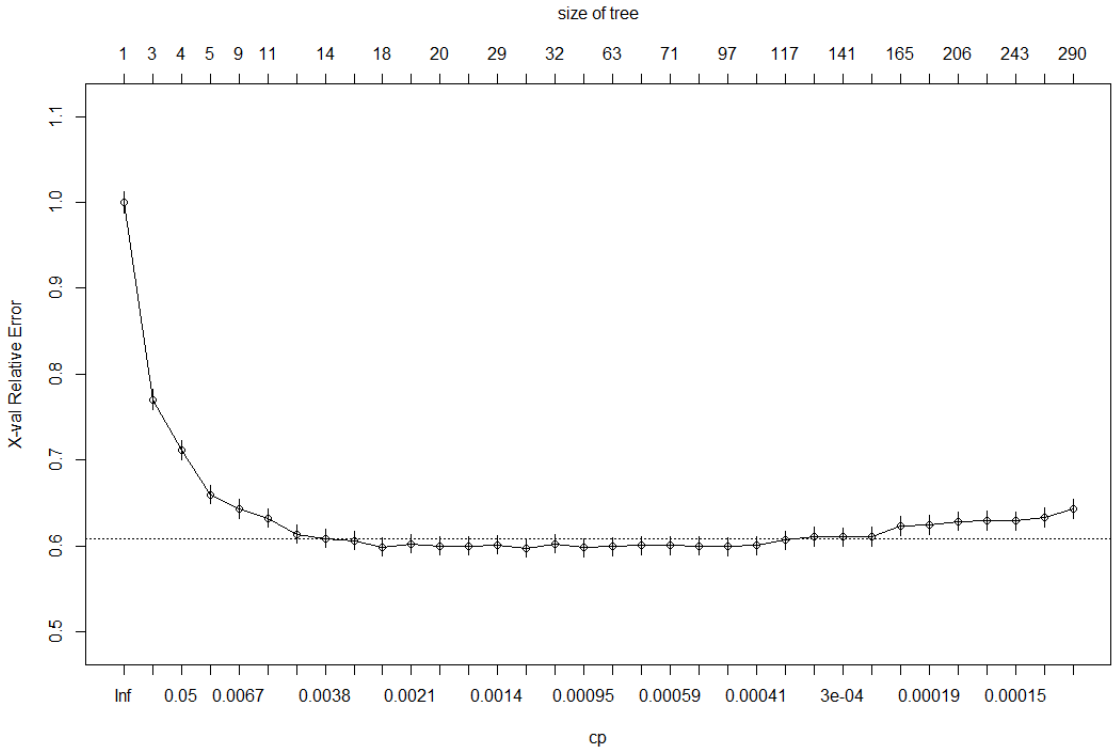
Plot for Q1j. Click here to go back to the question.

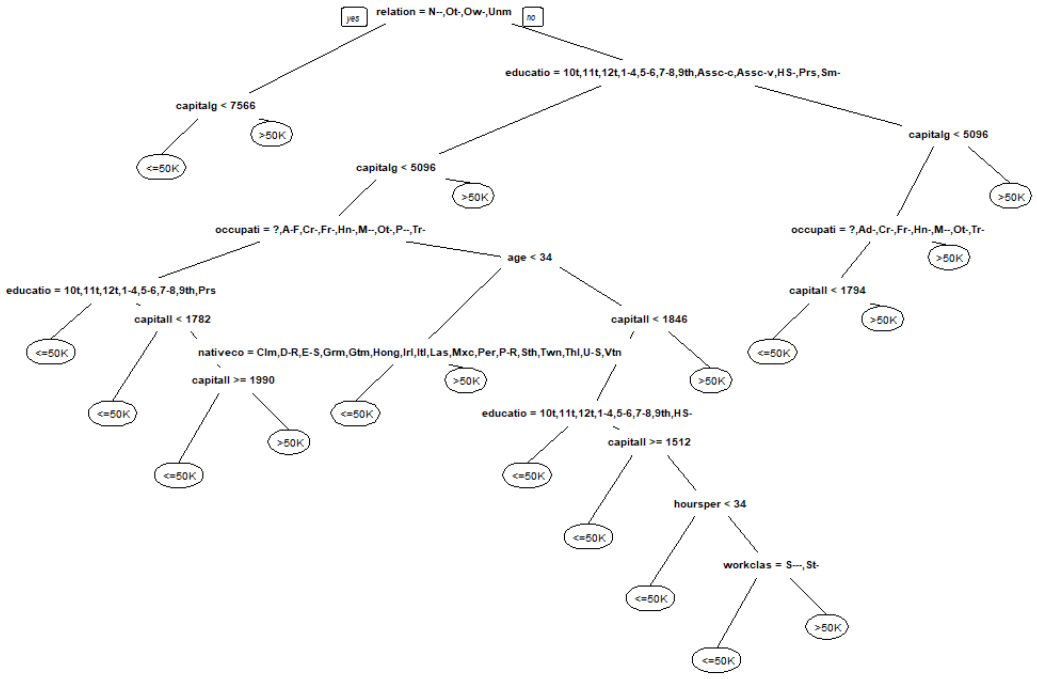


Plot for Q1m. Click here to go back to the question.



Plot for Q1n. Click here to go back to the question.





Plots for Q1o. Click here to go back to the question.

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. How many predictor variables were used in the regression tree?

   Solution. Plot found here. The variables used are *lstat*, *rm*, and *dis*.

   (b) What is the test set mean squared error? Draw a scatter plot of the fitted and true values. On average, the test predictions are within what range of the true median home value for the suburb?

   Solution. Plot found here. Test MSE from the regression tree is 25.35. On average, the predictions are within 5.035 of the true value.

   (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?

   Solution. By using the default settings, we see that te smallest value of *xerror* (or *xerror* +

xstd) is given at $cp = 0.01$ (7 splits). Hence, we would not prune the tree developed earlier.

(d) Suppose you prune the tree to 5 nodes. Plot the new tree. What is the test set mean squared error? Compare with the result in part (b).

Solution. Plot found here. The new MSE is 28.25, and is higher than 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?

Solution. The test MSE is 11.93 (Your number could be different). The result from the random forest, in terms of test error, is significantly smaller than CART. It tries 4 variables at each split by default.

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

Solution. Plot found here. *importance()* prints out the importance of the variables in terms of increase in node purity. The highest values are obtained for *lstat* and *rm* (most important).

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

Solution. By default, random forests use $p/3$ variables in regression and $\sqrt{p}$ variables in classification, where p is the number of variables to randomly sample as candidates at each split. Here, $13/3 \approx 4$, as in part (e).
We can try out different values of *mtry* arguments in random forests to control this. Using all variables gives highly correlated trees, and as such, the prediction power of such a model would not be as good.

R Scripts.

```
> #2)
> #a)
> boston <- read.csv("Boston.csv")
> set.seed(1)
> trainID <- sample(1:nrow(boston), nrow(boston)/2)
> train <- boston[trainID,]
> test <- boston[-trainID,]
```

```
> library(rpart)
> library(rpart.plot)
> model.tree <- rpart(medv ~ ., data = train)
> summary(model.tree)
Call:
rpart(formula = medv ~ ., data = train)
  n= 253

          CP nsplit rel error    xerror       xstd
1 0.46257558      0 1.0000000 1.0127304 0.11772795
2 0.20467339      1 0.5374244 0.5633934 0.05969933
3 0.07461842      2 0.3327510 0.3494856 0.03994467
4 0.03919129      3 0.2581326 0.2960757 0.03997597
5 0.03208187      4 0.2189413 0.3107221 0.04807582
6 0.02162884      5 0.1868595 0.2950782 0.04805304
7 0.01114973      6 0.1652306 0.2614557 0.04260435
8 0.01000000      7 0.1540809 0.2503241 0.03701355

Variable importance
  lstat      rm   indus     nox    crim      zn     dis     age ptratio     rad
     28      23      12      12      11       9       3       2       1       1

Node number 1: 253 observations,    complexity param=0.4625756
  mean=22.67312, MSE=82.58758
  left son=2 (150 obs) right son=3 (103 obs)
  Primary splits:
      lstat   < 9.715    to the right, improve=0.4625756, (0 missing)
      rm      < 6.803    to the left,  improve=0.4235645, (0 missing)
      indus   < 3.985    to the right, improve=0.2666899, (0 missing)
      ptratio < 18.55    to the right, improve=0.2571745, (0 missing)
      nox     < 0.6695   to the right, improve=0.2405937, (0 missing)
  Surrogate splits:
      indus < 6.66     to the right, agree=0.822, adj=0.563, (0 split)
      nox   < 0.5085   to the right, agree=0.810, adj=0.534, (0 split)
      rm    < 6.4895   to the left,  agree=0.810, adj=0.534, (0 split)
      crim  < 0.092825 to the right, agree=0.771, adj=0.437, (0 split)
      zn    < 16.25    to the left,  agree=0.763, adj=0.417, (0 split)

Node number 2: 150 observations,    complexity param=0.07461842
  mean=17.55133, MSE=23.0981
  left son=4 (30 obs) right son=5 (120 obs)
  Primary splits:
      lstat < 21.49    to the right, improve=0.4500014, (0 missing)
      dis   < 2.0643   to the left,  improve=0.3505294, (0 missing)
```

```
      crim  < 5.84803  to the right, improve=0.3450068, (0 missing)
      nox   < 0.6635   to the right, improve=0.3341687, (0 missing)
      age   < 82.85    to the right, improve=0.2587750, (0 missing)
  Surrogate splits:
      crim < 14.80775 to the right, agree=0.873, adj=0.367, (0 split)
      rm   < 5.548    to the left,  agree=0.867, adj=0.333, (0 split)
      dis  < 1.61495  to the left,  agree=0.867, adj=0.333, (0 split)
      age  < 99.65    to the right, agree=0.820, adj=0.100, (0 split)

Node number 3: 103 observations,    complexity param=0.2046734
  mean=30.13204, MSE=75.38431
  left son=6 (89 obs) right son=7 (14 obs)
  Primary splits:
      rm      < 7.437    to the left,  improve=0.5507803, (0 missing)
      lstat   < 4.475    to the right, improve=0.3880045, (0 missing)
      ptratio < 14.75    to the right, improve=0.1921520, (0 missing)
      nox     < 0.574    to the left,  improve=0.1880380, (0 missing)
      dis     < 3.231    to the right, improve=0.1547792, (0 missing)
  Surrogate splits:
      lstat < 4.15     to the right, agree=0.913, adj=0.357, (0 split)

Node number 4: 30 observations
  mean=11.10333, MSE=10.39632

Node number 5: 120 observations,    complexity param=0.02162884
  mean=19.16333, MSE=13.28082
  left son=10 (58 obs) right son=11 (62 obs)
  Primary splits:
      lstat < 14.48    to the right, improve=0.2835713, (0 missing)
      age   < 82.85    to the right, improve=0.2001015, (0 missing)
      crim  < 5.84803  to the right, improve=0.1857460, (0 missing)
      nox   < 0.665    to the right, improve=0.1723284, (0 missing)
      black < 114.685  to the left,  improve=0.1691283, (0 missing)
  Surrogate splits:
      age  < 82.05    to the right, agree=0.767, adj=0.517, (0 split)
      dis  < 2.6368   to the left,  agree=0.725, adj=0.431, (0 split)
      crim < 4.036045 to the right, agree=0.675, adj=0.328, (0 split)
      nox  < 0.5835   to the right, agree=0.675, adj=0.328, (0 split)
      tax  < 417.5    to the right, agree=0.642, adj=0.259, (0 split)

Node number 6: 89 observations,    complexity param=0.03919129
  mean=27.5764, MSE=37.19281
  left son=12 (61 obs) right son=13 (28 obs)
  Primary splits:
```

```
      rm    < 6.7815   to the left,  improve=0.2473863, (0 missing)
      nox   < 0.589    to the left,  improve=0.2144312, (0 missing)
      dis   < 2.1398   to the right, improve=0.2144312, (0 missing)
      age   < 88.75    to the left,  improve=0.1861958, (0 missing)
      lstat < 4.91     to the right, improve=0.1403299, (0 missing)
  Surrogate splits:
      ptratio < 15.25    to the right, agree=0.798, adj=0.357, (0 split)
      lstat   < 6.2      to the right, agree=0.742, adj=0.179, (0 split)
      zn      < 65       to the left,  agree=0.730, adj=0.143, (0 split)
      indus   < 4.01     to the right, agree=0.730, adj=0.143, (0 split)
      crim    < 0.01375  to the right, agree=0.697, adj=0.036, (0 split)

Node number 7: 14 observations
  mean=46.37857, MSE=12.70311

Node number 10: 58 observations
  mean=17.1569, MSE=12.81521

Node number 11: 62 observations
  mean=21.04032, MSE=6.427245

Node number 12: 61 observations,    complexity param=0.03208187
  mean=25.52131, MSE=32.69873
  left son=24 (54 obs) right son=25 (7 obs)
  Primary splits:
      dis   < 2.85155  to the right, improve=0.3360735, (0 missing)
      age   < 84.6     to the left,  improve=0.2642544, (0 missing)
      indus < 13.36    to the left,  improve=0.2264315, (0 missing)
      crim  < 0.40549  to the left,  improve=0.1554861, (0 missing)
      black < 368.405  to the right, improve=0.1457973, (0 missing)
  Surrogate splits:
      age   < 87.6     to the left,  agree=0.967, adj=0.714, (0 split)
      crim  < 1.02739  to the left,  agree=0.951, adj=0.571, (0 split)
      indus < 16.57    to the left,  agree=0.951, adj=0.571, (0 split)
      nox   < 0.5905   to the left,  agree=0.951, adj=0.571, (0 split)
      rad   < 16       to the left,  agree=0.934, adj=0.429, (0 split)

Node number 13: 28 observations
  mean=32.05357, MSE=17.73749

Node number 24: 54 observations,    complexity param=0.01114973
  mean=24.32778, MSE=10.07423
  left son=48 (22 obs) right son=49 (32 obs)
  Primary splits:
```

```
      rm    < 6.36     to the left,  improve=0.42824660, (0 missing)
      lstat < 6.69     to the right, improve=0.25624760, (0 missing)
      dis   < 4.6371   to the right, improve=0.10778880, (0 missing)
      indus < 3.085    to the right, improve=0.09410025, (0 missing)
      tax   < 278      to the right, improve=0.09291026, (0 missing)
  Surrogate splits:
      ptratio < 19.05    to the right, agree=0.648, adj=0.136, (0 split)
      lstat   < 7.465    to the right, agree=0.648, adj=0.136, (0 split)
      tax     < 420.5    to the right, agree=0.630, adj=0.091, (0 split)
      black   < 369.675  to the left,  agree=0.630, adj=0.091, (0 split)
      zn      < 65       to the right, agree=0.611, adj=0.045, (0 split)

Node number 25: 7 observations
  mean=34.72857, MSE=111.4678

Node number 48: 22 observations
  mean=21.82273, MSE=2.144483

Node number 49: 32 observations
  mean=26.05, MSE=8.245625

> model.tree
n= 253

node), split, n, deviance, yval
      * denotes terminal node

 1) root 253 20894.66000 22.67312
   2) lstat>=9.715 150  3464.71500 17.55133
     4) lstat>=21.49 30   311.88970 11.10333 *
     5) lstat< 21.49 120  1593.69900 19.16333
      10) lstat>=14.48 58   743.28220 17.15690 *
      11) lstat< 14.48 62   398.48920 21.04032 *
   3) lstat< 9.715 103  7764.58400 30.13204
     6) rm< 7.437 89  3310.16000 27.57640
      12) rm< 6.7815 61  1994.62200 25.52131
        24) dis>=2.85155 54   544.00830 24.32778
          48) rm< 6.36 22    47.17864 21.82273 *
          49) rm>=6.36 32   263.86000 26.05000 *
        25) dis< 2.85155 7   780.27430 34.72857 *
      13) rm>=6.7815 28   496.64960 32.05357 *
     7) rm>=7.437 14   177.84360 46.37857 *
> prp(model.tree)
```

```
> #b)
> pred.tree <- predict(model.tree, newdata = test)
> mse <- mean((pred.tree - test$medv)^2)
> mse
[1] 25.35825
> plot(pred.tree, test$medv)
> abline(1:50,1:50)
> sqrt(mse)
[1] 5.035698

> #c)
> model.tree
n= 253

node), split, n, deviance, yval
      * denotes terminal node

 1) root 253 20894.66000 22.67312
   2) lstat>=9.715 150  3464.71500 17.55133
     4) lstat>=21.49 30   311.88970 11.10333 *
     5) lstat< 21.49 120  1593.69900 19.16333
      10) lstat>=14.48 58   743.28220 17.15690 *
      11) lstat< 14.48 62   398.48920 21.04032 *
   3) lstat< 9.715 103  7764.58400 30.13204
     6) rm< 7.437 89  3310.16000 27.57640
      12) rm< 6.7815 61  1994.62200 25.52131
        24) dis>=2.85155 54   544.00830 24.32778
          48) rm< 6.36 22    47.17864 21.82273 *
          49) rm>=6.36 32   263.86000 26.05000 *
        25) dis< 2.85155 7   780.27430 34.72857 *
      13) rm>=6.7815 28   496.64960 32.05357 *
     7) rm>=7.437 14   177.84360 46.37857 *
> summary(model.tree)
Call:
rpart(formula = medv ~ ., data = train)
  n= 253

          CP nsplit rel error    xerror       xstd
1 0.46257558      0 1.0000000 1.0127304 0.11772795
2 0.20467339      1 0.5374244 0.5633934 0.05969933
3 0.07461842      2 0.3327510 0.3494856 0.03994467
4 0.03919129      3 0.2581326 0.2960757 0.03997597
5 0.03208187      4 0.2189413 0.3107221 0.04807582
6 0.02162884      5 0.1868595 0.2950782 0.04805304
```

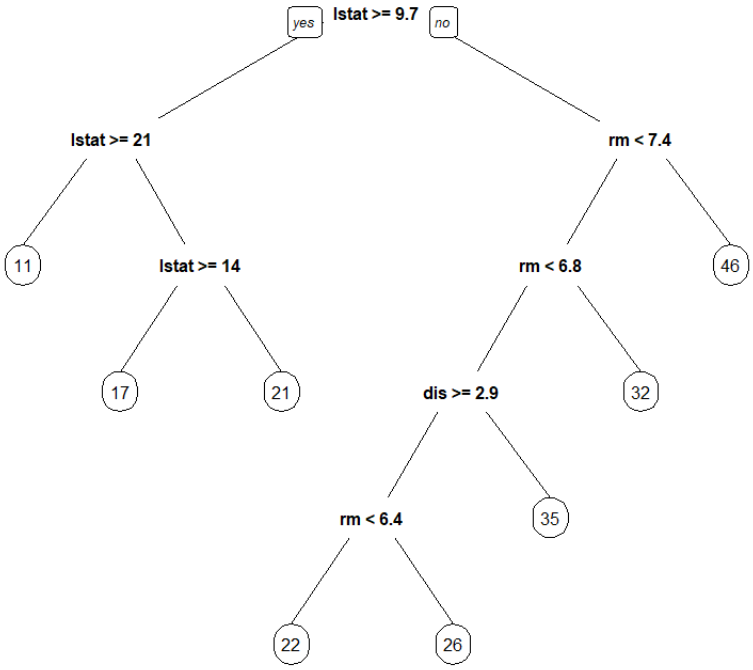
```
7 0.01114973      6 0.1652306 0.2614557 0.04260435
8 0.01000000      7 0.1540809 0.2503241 0.03701355

Variable importance
  lstat      rm   indus     nox    crim      zn     dis     age ptratio     rad
     28      23      12      12      11       9       3       2       1       1

Node number 1: 253 observations,    complexity param=0.4625756
  mean=22.67312, MSE=82.58758
  left son=2 (150 obs) right son=3 (103 obs)
  Primary splits:
      lstat   < 9.715    to the right, improve=0.4625756, (0 missing)
      rm      < 6.803    to the left,  improve=0.4235645, (0 missing)
      indus   < 3.985    to the right, improve=0.2666899, (0 missing)
      ptratio < 18.55    to the right, improve=0.2571745, (0 missing)
      nox     < 0.6695   to the right, improve=0.2405937, (0 missing)
  Surrogate splits:
      indus < 6.66     to the right, agree=0.822, adj=0.563, (0 split)
      nox   < 0.5085   to the right, agree=0.810, adj=0.534, (0 split)
      rm    < 6.4895   to the left,  agree=0.810, adj=0.534, (0 split)
      crim  < 0.092825 to the right, agree=0.771, adj=0.437, (0 split)
      zn    < 16.25    to the left,  agree=0.763, adj=0.417, (0 split)

Node number 2: 150 observations,    complexity param=0.07461842
  mean=17.55133, MSE=23.0981
  left son=4 (30 obs) right son=5 (120 obs)
  Primary splits:
      lstat < 21.49    to the right, improve=0.4500014, (0 missing)
      dis   < 2.0643   to the left,  improve=0.3505294, (0 missing)
      crim  < 5.84803  to the right, improve=0.3450068, (0 missing)
      nox   < 0.6635   to the right, improve=0.3341687, (0 missing)
      age   < 82.85    to the right, improve=0.2587750, (0 missing)
  Surrogate splits:
      crim < 14.80775 to the right, agree=0.873, adj=0.367, (0 split)
      rm   < 5.548    to the left,  agree=0.867, adj=0.333, (0 split)
      dis  < 1.61495  to the left,  agree=0.867, adj=0.333, (0 split)
      age  < 99.65    to the right, agree=0.820, adj=0.100, (0 split)

Node number 3: 103 observations,    complexity param=0.2046734
  mean=30.13204, MSE=75.38431
  left son=6 (89 obs) right son=7 (14 obs)
  Primary splits:
      rm      < 7.437    to the left,  improve=0.5507803, (0 missing)
      lstat   < 4.475    to the right, improve=0.3880045, (0 missing)
```

```
      ptratio < 14.75    to the right, improve=0.1921520, (0 missing)
      nox     < 0.574    to the left,  improve=0.1880380, (0 missing)
      dis     < 3.231    to the right, improve=0.1547792, (0 missing)
  Surrogate splits:
      lstat < 4.15     to the right, agree=0.913, adj=0.357, (0 split)

Node number 4: 30 observations
  mean=11.10333, MSE=10.39632

Node number 5: 120 observations,    complexity param=0.02162884
  mean=19.16333, MSE=13.28082
  left son=10 (58 obs) right son=11 (62 obs)
  Primary splits:
      lstat < 14.48    to the right, improve=0.2835713, (0 missing)
      age   < 82.85    to the right, improve=0.2001015, (0 missing)
      crim  < 5.84803  to the right, improve=0.1857460, (0 missing)
      nox   < 0.665    to the right, improve=0.1723284, (0 missing)
      black < 114.685  to the left,  improve=0.1691283, (0 missing)
  Surrogate splits:
      age  < 82.05    to the right, agree=0.767, adj=0.517, (0 split)
      dis  < 2.6368   to the left,  agree=0.725, adj=0.431, (0 split)
      crim < 4.036045 to the right, agree=0.675, adj=0.328, (0 split)
      nox  < 0.5835   to the right, agree=0.675, adj=0.328, (0 split)
      tax  < 417.5    to the right, agree=0.642, adj=0.259, (0 split)

Node number 6: 89 observations,    complexity param=0.03919129
  mean=27.5764, MSE=37.19281
  left son=12 (61 obs) right son=13 (28 obs)
  Primary splits:
      rm    < 6.7815   to the left,  improve=0.2473863, (0 missing)
      nox   < 0.589    to the left,  improve=0.2144312, (0 missing)
      dis   < 2.1398   to the right, improve=0.2144312, (0 missing)
      age   < 88.75    to the left,  improve=0.1861958, (0 missing)
      lstat < 4.91     to the right, improve=0.1403299, (0 missing)
  Surrogate splits:
      ptratio < 15.25    to the right, agree=0.798, adj=0.357, (0 split)
      lstat   < 6.2      to the right, agree=0.742, adj=0.179, (0 split)
      zn      < 65       to the left,  agree=0.730, adj=0.143, (0 split)
      indus   < 4.01     to the right, agree=0.730, adj=0.143, (0 split)
      crim    < 0.01375  to the right, agree=0.697, adj=0.036, (0 split)

Node number 7: 14 observations
  mean=46.37857, MSE=12.70311
```

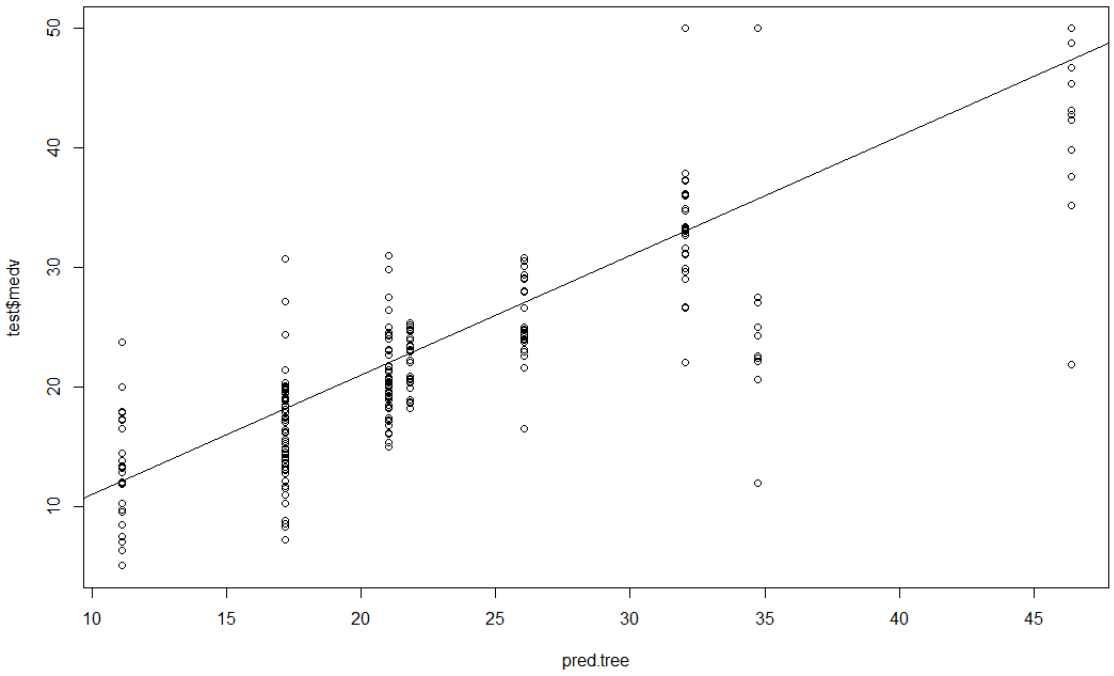
```
Node number 10: 58 observations
  mean=17.1569, MSE=12.81521

Node number 11: 62 observations
  mean=21.04032, MSE=6.427245

Node number 12: 61 observations,    complexity param=0.03208187
  mean=25.52131, MSE=32.69873
  left son=24 (54 obs) right son=25 (7 obs)
  Primary splits:
      dis   < 2.85155  to the right, improve=0.3360735, (0 missing)
      age   < 84.6     to the left,  improve=0.2642544, (0 missing)
      indus < 13.36    to the left,  improve=0.2264315, (0 missing)
      crim  < 0.40549  to the left,  improve=0.1554861, (0 missing)
      black < 368.405  to the right, improve=0.1457973, (0 missing)
  Surrogate splits:
      age   < 87.6     to the left,  agree=0.967, adj=0.714, (0 split)
      crim  < 1.02739  to the left,  agree=0.951, adj=0.571, (0 split)
      indus < 16.57    to the left,  agree=0.951, adj=0.571, (0 split)
      nox   < 0.5905   to the left,  agree=0.951, adj=0.571, (0 split)
      rad   < 16       to the left,  agree=0.934, adj=0.429, (0 split)

Node number 13: 28 observations
  mean=32.05357, MSE=17.73749

Node number 24: 54 observations,    complexity param=0.01114973
  mean=24.32778, MSE=10.07423
  left son=48 (22 obs) right son=49 (32 obs)
  Primary splits:
      rm    < 6.36     to the left,  improve=0.42824660, (0 missing)
      lstat < 6.69     to the right, improve=0.25624760, (0 missing)
      dis   < 4.6371   to the right, improve=0.10778880, (0 missing)
      indus < 3.085    to the right, improve=0.09410025, (0 missing)
      tax   < 278      to the right, improve=0.09291026, (0 missing)
  Surrogate splits:
      ptratio < 19.05    to the right, agree=0.648, adj=0.136, (0 split)
      lstat   < 7.465    to the right, agree=0.648, adj=0.136, (0 split)
      tax     < 420.5    to the right, agree=0.630, adj=0.091, (0 split)
      black   < 369.675  to the left,  agree=0.630, adj=0.091, (0 split)
      zn      < 65       to the right, agree=0.611, adj=0.045, (0 split)

Node number 25: 7 observations
  mean=34.72857, MSE=111.4678
```

```
Node number 48: 22 observations
  mean=21.82273, MSE=2.144483

Node number 49: 32 observations
  mean=26.05, MSE=8.245625


> #d)
> model.tree1 <- prune(model.tree, cp = 0.021629)
> prp(model.tree1)
> pred.tree1 <- predict(model.tree1, newdata = test)
> mse1 <- mean((pred.tree1 - test$medv)^2)
> mse1
[1] 28.25719

> #e)
> library(randomForest)
> set.seed(1)
> model.forest <- randomForest(medv ~ ., data = train)
> model.forest

Call:
 randomForest(formula = medv ~ ., data = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 12.81989
                    % Var explained: 84.48
> pred.forest <- predict(model.forest, newdata = test)
> mse2 <- mean((pred.forest - test$medv)^2)
> mse2
[1] 11.93146

> #f)
> importance(model.forest)
        IncNodePurity
crim        1303.9189
zn           114.0047
indus       1309.5213
chas         108.5740
nox         1310.5029
rm          5481.3672
age          690.2379
```

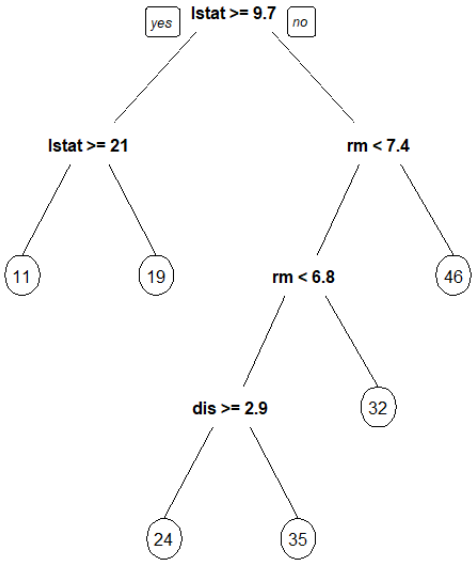
```
dis         1463.2625
rad          183.6832
tax          714.5192
ptratio     1416.2867
black        445.7410
lstat       5844.3476
> varImpPlot(model.forest)

> #g)
> set.seed(1)
> model.forest1 <- randomForest(medv ~ ., data = train, mtry = 6)
> model.forest1

Call:
 randomForest(formula = medv ~ ., data = train, mtry = 6)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 11.72238
                    % Var explained: 85.81
> pred.forest1 <- predict(model.forest1, newdata = test)
> mse3 <- mean((pred.forest1 - test$medv)^2)
> mse3
[1] 11.50837
> set.seed(1)
> model.forest2 <- randomForest(medv ~ ., data = train, mtry = 13)
> model.forest2

Call:
 randomForest(formula = medv ~ ., data = train, mtry = 13)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 13

          Mean of squared residuals: 10.76175
                    % Var explained: 86.97
> pred.forest2 <- predict(model.forest2, newdata = test)
> mse4 <- mean((pred.forest2 - test$medv)^2)
> mse4
[1] 13.39501
```



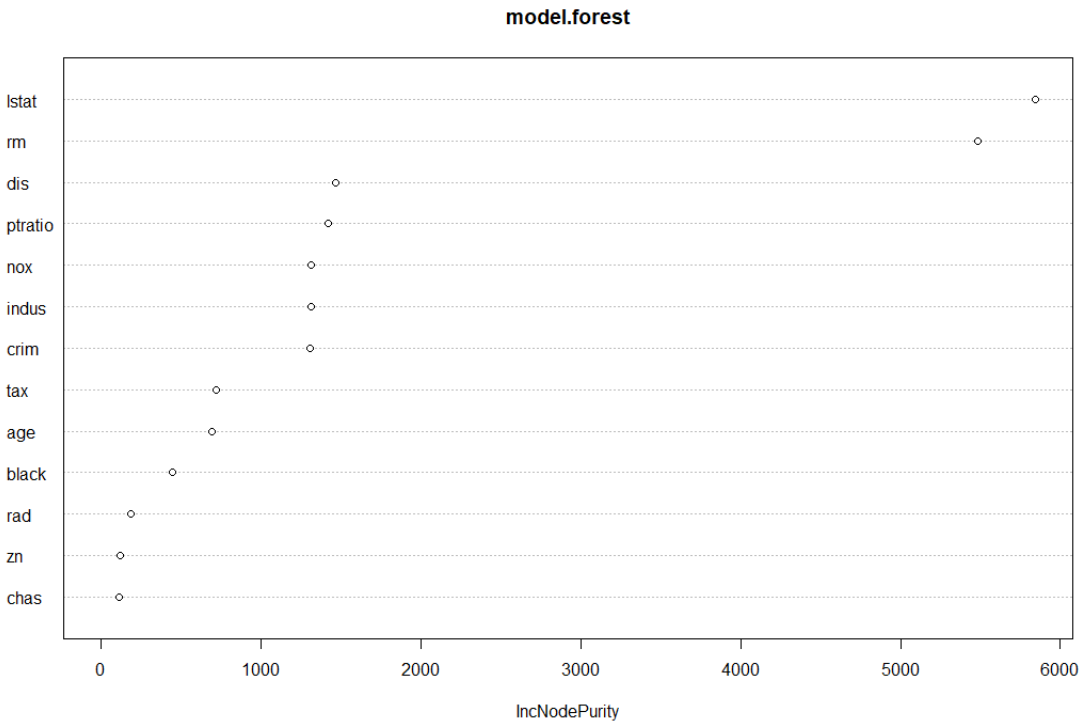
Plot for Q2a. Click here to go back to the question.



Plot for Q2b. Click here to go back to the question.



Plot for Q2d. Click here to go back to the question.



Plot for Q2f. Click here to go back to the question.

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?

   Solution. Table (*lctdir* as rows, *result* as columns) is as shown:

	Lib	*Conser*
Conser	170	136
Lib	94	198

The fraction of cases where the Lower Court decision was reversed by the Supreme Court is

$$\frac{170 + 198}{170 + 198 + 136 + 94} = 0.6153$$

(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. Write down the R command(s) that you used to define this variable. What is the total number of cases that had an unanimous conservative decision?

Solution. 143 cases 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?

Solution. 124 cases 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?

Solution. Plot found here. There are 7 node splits in the tree.

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

Solution. The tree splits on the *circuit, issue, petit, respon* variables.

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

Solution. The AUC is given as 0.6519.

(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?

Solution. Plot found here. The tree splits on the *respon* variable at the top node.

(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?

Solution. Plot found here. The leaf node has "8/3" observations and the output is 0 (not liberal). The fraction of cases with unanimus liberal decision at this node is 3/11.

(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?

Solution. The table (first model as rows, second as columns) is as follows:

	0	1
0	502	36
1	58	2

There are 2 cases with unanimous conservative and liberal decisions predicted by the trees.

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

Solution. From the same table, we have 502 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?

Solution. The table (majority prediction as rows, actual result as columns) is as shown:

	Lib	*Cons*
Lib	149	71
Cons	73	211

$$Accuracy = \frac{149 + 211}{149 + 211 + 71 + 73} = 0.714.$$

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

Solution. Total accuracy of the 2 step model is

$$\frac{360 + 76}{598} = 0.729.$$

(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.

Solution. The variables to be subbed in are *petit="business", respon="business", circuit=6, lctdir="conser", issue="ECN".* From model 1, at the top node, we get *unCons=0.* From model 2, we get *unLib=1*, suggesting a unanimous liberal vote from the Supreme Court, overturning the Lower Court vote.

(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?

Solution. Remember to do a classification and not regression. The table (prediction as rows, actuals as columns) is as shown:

	0	1
0	225	21
1	39	313

$$Accuracy = \frac{225 + 313}{598} = 0.899.$$

(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.

Solution. CART is more interpretable than the random forest, but the latter has a

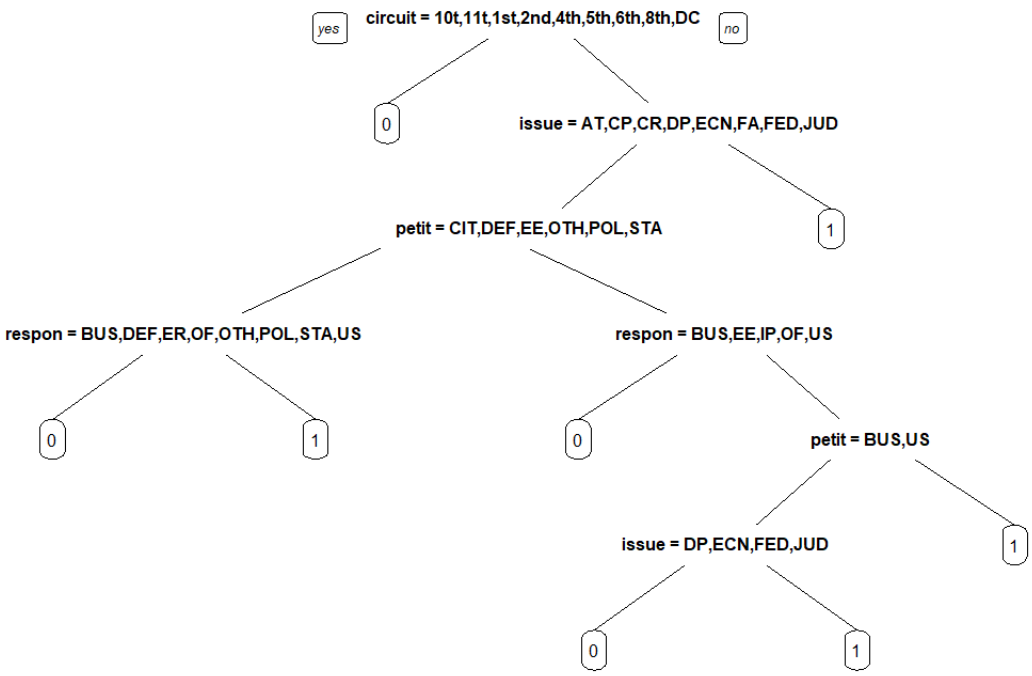
greater accuracy.

R Scripts.

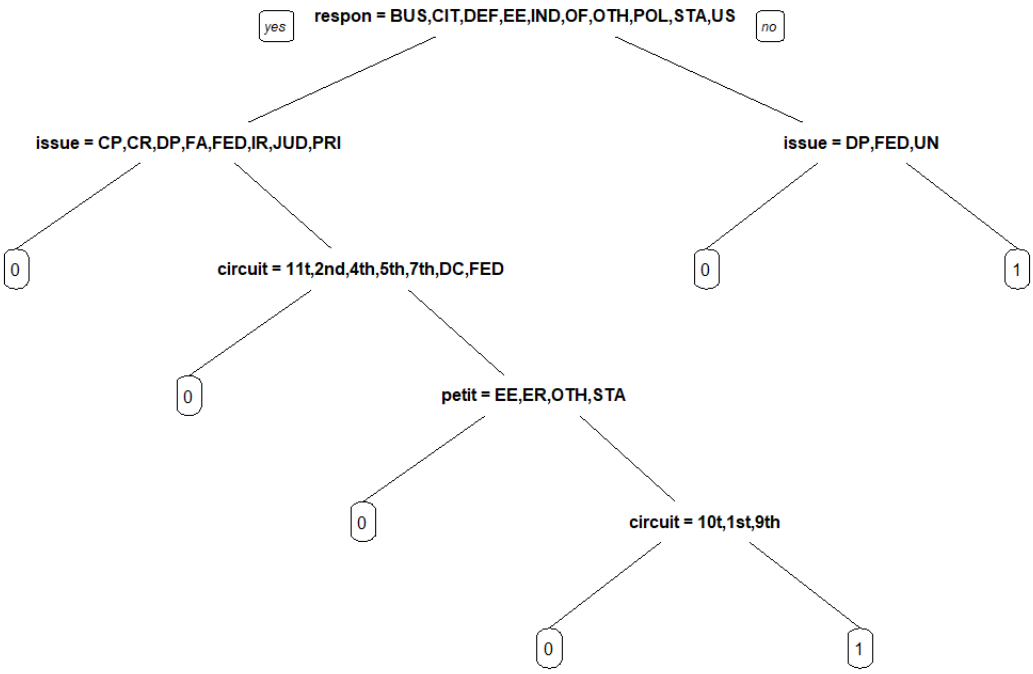
```
> #3)
> #a)
> supreme <- read.csv("supremeexercise.csv")
> table(supreme$lctdir,supreme$result)

           0   1
  conser  170 136
  liberal  94 198

> #b)
> supreme$unCons <- as.integer(rowSums(supreme[,5:13])==9)
> table(supreme$unCons)

  0   1
455 143

> #c)
> supreme$unLib <- as.integer(rowSums(supreme[,5:13])==0)
> table(supreme$unLib)

  0   1
474 124

> #d)
> library(rpart)
> library(rpart.plot)
> model1 <- rpart(as.factor(unCons)~petit+respon+circuit+unconst+lctdir+issue,data=supreme)
> prp(model1)

> #f)
> library(ROCR)
> predict1 <- predict(model1,newdata=supreme)
> ROCRpred1 <- prediction(predict1[,2],supreme$unCons)
> performance(ROCRpred1,"auc")@y.values
[[1]]
[1] 0.6519788


> #g)
> model2 <- rpart(as.factor(unLib)~petit+respon+circuit+unconst+lctdir+issue,data=supreme)
> prp(model2)
```

```
> #h)
> prp(model2, extra=1)

> #i)
> predict1a <- predict(model1,newdata=supreme,type="class")
> predict2a <- predict(model2,newdata=supreme,type="class")
> v1 <- subset(supreme$result,predict1a==1 & predict2a == 0)
> v2 <- subset(supreme$result,predict1a==0 & predict2a == 1)
> table(v1)
v1
 0  1
12 46
> table(v2)
v2
 0  1
30  6

> #k)
> supreme1 <- subset(supreme,predict1a==predict2a)
> str(supreme1)
'data.frame': 504 obs. of  22 variables:
 $ docket : Factor w/ 598 levels "00-1011","00-1021",..: 93 94 95 96 97 98 99 100 103 104 ...
 $ term   : int  1994 1994 1994 1994 1994 1994 1994 1994 1994 1994 ...
 $ party_1: Factor w/ 209 levels "0","AC","AG",..: 75 78 201 194 88 12 84 194 82 164 ...
 $ party_2: Factor w/ 202 levels "AC","AIRLINE",..: 54 133 195 188 83 91 81 177 92 196 ...
 $ rehndir: int  1 0 0 1 1 1 1 1 1 1 ...
 $ stevdir: int  0 0 0 0 0 1 1 0 1 0 ...
 $ ocondir: int  1 0 0 0 1 0 0 1 1 1 ...
 $ scaldir: int  1 1 0 1 1 1 0 1 1 1 ...
 $ kendir : int  1 0 0 0 0 1 1 1 1 0 ...
 $ soutdir: int  1 0 0 0 0 0 0 0 1 0 ...
 $ thomdir: int  1 1 0 1 1 1 0 1 1 1 ...
 $ gindir : int  1 0 0 0 0 1 1 0 1 0 ...
 $ brydir : int  1 0 0 0 0 0 1 0 1 0 ...
 $ petit  : Factor w/ 12 levels "BUSINESS","CITY",..: 1 9 9 12 4 9 9 12 1 1 ...
 $ respon : Factor w/ 12 levels "BUSINESS","CITY",..: 1 9 9 9 9 9 1 9 1 10 ...
 $ circuit: Factor w/ 13 levels "10th","11th",..: 13 2 13 12 5 8 8 7 4 10 ...
 $ unconst: int  0 0 0 0 0 0 0 0 0 0 ...
 $ lctdir : Factor w/ 2 levels "conser","liberal": 2 1 2 2 1 1 2 1 2 2 ...
 $ issue  : Factor w/ 12 levels "AT","CP","CR",..: 5 5 3 6 5 3 9 7 5 3 ...
 $ result : int  1 0 0 0 0 1 1 1 1 0 ...
 $ unCons : int  0 0 0 0 0 0 0 0 1 0 ...
 $ unLib  : int  0 0 1 0 0 0 0 0 0 0 ...
> model3 <- rpart(as.factor(rehndir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
```

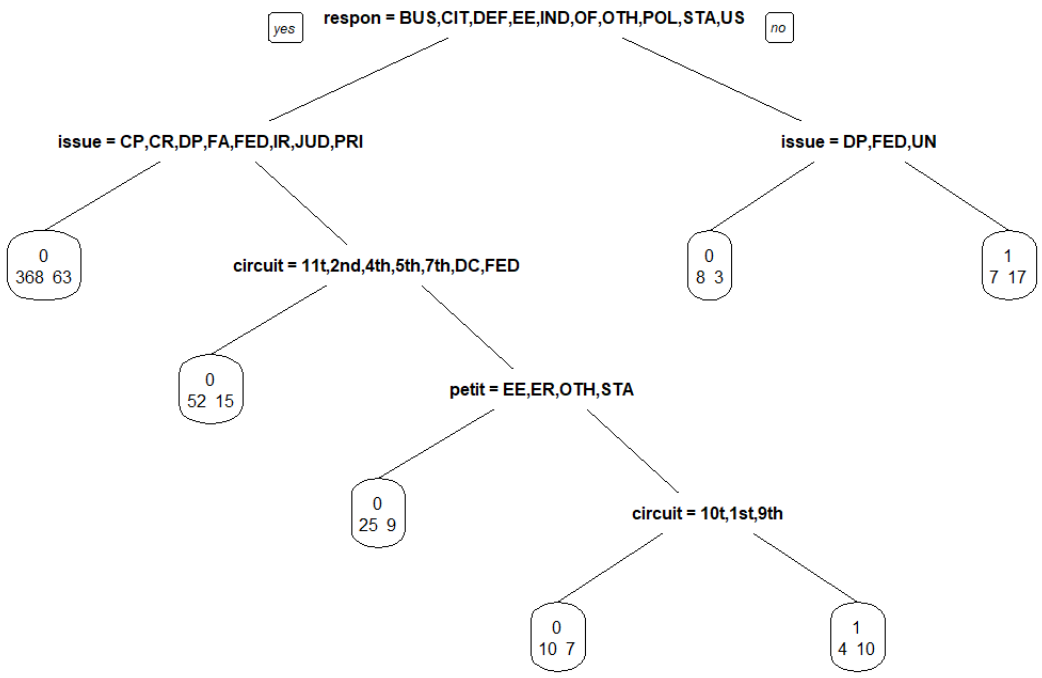
```
> model4 <- rpart(as.factor(stevdir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model5 <- rpart(as.factor(ocondir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model6 <- rpart(as.factor(scaldir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model7 <- rpart(as.factor(kendir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model8 <- rpart(as.factor(soutdir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model9 <- rpart(as.factor(thomdir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model10 <- rpart(as.factor(gindir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> model11 <- rpart(as.factor(brydir)~petit+respon+circuit+unconst+lctdir+issue,data=supreme1)
> predict3 <- predict(model3,newdata=supreme1,type="class")
> predict4 <- predict(model4,newdata=supreme1,type="class")
> predict5 <- predict(model5,newdata=supreme1,type="class")
> predict6 <- predict(model6,newdata=supreme1,type="class")
> predict7 <- predict(model7,newdata=supreme1,type="class")
> predict8 <- predict(model8,newdata=supreme1,type="class")
> predict9 <- predict(model9,newdata=supreme1,type="class")
> predict10 <- predict(model10,newdata=supreme1,type="class")
> predict11 <- predict(model11,newdata=supreme1,type="class")
> totalcons <-  as.numeric(as.character(predict3))+ as.numeric(as.character(predict4))+ as.numeric(as.cha
> table(totalcons>=5,supreme1$result)

         0   1
  FALSE 149  71
  TRUE   73 211

> #m)
> prp(model1)
> prp(model2)

> #n)
> library(randomForest)
> set.seed(1)
> forest <- randomForest(as.factor(result)~issue+circuit+lctdir+unconst+petit+respon,data=supreme)
> predictforest <- predict(forest,newdata=supreme)
> table(predictforest,supreme$result)

predictforest   0   1
            0 225  21
            1  39 313
```



Plot for Q3d. Click here to go back to the question.



Plot for Q3g. Click here to go back to the question.



Plot for Q3h. Click here to go back to the question.