edX (https://www.edx.org)

MITx: 15.071x The Analytics Edge

sandipan_dey (/dashboard) ▾

Courseware (/courses/MITx/15.071x/1T2014/courseware) Course Info (/courses/MITx/15.071x/1T2014/info)

Discussion (/courses/MITx/15.071x/1T2014/discussion/forum) Progress (/courses/MITx/15.071x/1T2014/progress)

Help

yllabus (/courses/MITx/15.071x/1T2014/4264e68418f34d839cf0b33a5da644b2/)

chedule (/courses/MITx/15.071x/1T2014/2891f8bf120945b9aa12e6601739c3e6/)

PREDICTING EARNINGS FROM CENSUS DATA

The United States government periodically collects demographic information by conducting a census.

In this problem, we 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 (http://archive.ics.uci.edu/ml/datasets/Adult).

The file census.csv (/c4x/MITx/15.071x/asset/census.csv) contains 1994 census data for almost 32,000 individuals in the United States.

The available variables include:

- *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 pe ek
- *nativecountry* = the native country of the individual
- *over50k* = whether or not the individual earned more than $50,000 in 1994

PROBLEM 1.1 - A LOGISTIC REGRESSION MODEL (1/1 point)

As we did in lecture, let's begin by building a logistic regression model to predict whether an individual's earnings are above $50,000 using the other variables. First, read the dataset census.csv into R.

Then, 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 using all of the independent variables to predict the dependent variable "over50k", and use the training set to build the model.

Which variables are significant, or have factors that are significant? (Use 0.1 as your significance threshold, so variables with a period or dot in the stars column should be counted too. You might see a warning message here - ignore it.)

☑ age ✓

☑ workclass ✓

☑ education ✓

- ☑ maritalstatus ✓
- ☑ occupation ✓
- ☑ relationship ✓
- ☐ race
- ☑ sex ✓
- ☑ capitalgain ✓
- ☑ capitalloss ✓
- ☑ hoursperweek ✓
- ☐ nativecountry

EXPLANATION

To read census.csv, set your working directory to the same directory that census.csv is in, and run the following command:

```
census = read.csv("census.csv")
```

We now need to split the data. Load the caTools package, and set the seed to 2000:

```
library(caTools)

set.seed(2000)
```

Split the data set according to the over50k variable:

```
spl = sample.split(census$over50k, SplitRatio = 0.6)

train = subset(census, spl==TRUE)

test = subset(census, spl==FALSE)
```

We are now ready to run logistic regression. Build the logistic regression model:

```
censusglm = glm( over50k ~ . , family="binomial", data = train)
```

Finally, look at the model summary to identify the significant factors:

```
summary(censusglm)
```

Hide Answer *You have used 3 of 3 submissions*

PROBLEM 1.2 - A LOGISTIC REGRESSION MODEL (1/1 point)

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.)

0.8552107

0.8552107

Answer: 0.8552

EXPLANATION

Generate the predictions for the test set:

```
predictTest = predict(censusglm, newdata = test, type = "response")
```

Then we can generate the confusion matrix:

```
table(test$over50k, predictTest >= 0.5)
```

If we divide the sum of the main diagonal by the sum of all of the entries in the matrix, we obtain the accuracy:

(9051+1888)/(9051+662+1190+1888) = 0.8552107

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 1.3 - A LOGISTIC REGRESSION MODEL (1/1 point)

What is the baseline accuracy for the testing set?

0.7593621

0.7593621

Answer: 0.7594

EXPLANATION

We need to first determine the most frequent outcome in the training set. To do that, we table the dependent variable in the training set:

```
table(train$over50k)
```

"<=50K" is the more frequent outcome (14570 observations), so this is what the baseline predicts. To generate the accuracy of the baseline on the test set, we can table the dependent variable in the test set:

```
table(test$over50k)
```

The baseline accuracy is

9713/(9713+3078) = 0.7593621.

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 1.4 - A LOGISTIC REGRESSION MODEL (1/1 point)

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

0.9061598

0.9061598

Answer: 0.9062

EXPLANATION

First, load the ROCR package:

```
library(ROCR)
```

Then you can use the following commands to compute the AUC (assuming your test set predictions are called "predictTest"):

```
ROCRpred = prediction(predictTest, test$over50k)

as.numeric(performance(ROCpred, "auc")@y.values)
```

Hide Answer *You have used 3 of 3 submissions*

PROBLEM 2.1 - A CART MODEL (1/1 point)

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 for this model. Using the same training set, fit a CART model, and plot the tree. Use the default parameters, so don't set a value for minbucket or cp. Remember to specify method="class" as an argument to rpart, since this is a classification problem.

How many splits does the tree have in total?

4

4

Answer: 4

EXPLANATION

To get started, load the rpart package:

```
library(rpart)
```

Estimate the CART tree:

```
censustree = rpart( over50k ~ . , method="class", data = train)
```

Plot the tree:

```
prp(censustree)
```

There are four splits in total.

Hide Answer

You have used 1 of 5 submissions

PROBLEM 2.2 - A CART MODEL (1/1 point)

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

- ○ age
- ○ workclass
- ○ education
- ○ maritalstatus
- ○ occupation
- ● relationship ✓
- ○ race
- ○ sex
- ○ capitalgain
- ○ capitalloss
- ○ hoursperweek
- ○ nativecountry

EXPLANATION

Plot the tree and examine the first split:

```
prp(censustree)
```

The first split uses the variable relationship.

Hide Answer *You have used 1 of 2 submissions*

PROBLEM 2.3 - A CART MODEL (1/1 point)

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

□ age
□ workclass
☑ education ✓
□ maritalstatus
□ occupation
□ relationship
□ race
□ sex
☑ capitalgain ✓
□ capitalloss
□ hoursperweek
□ nativecountry

EXPLANATION

Plot the tree and examine the second splits:

prp(censustree)

The second splits are on capitalgains and education.

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 2.4 - A CART MODEL (1/1 point)

What is the accuracy of the model on the testing set? (Use a threshold of 0.5, so add the argument type="class".)

0.8473927

0.8473927

Answer: 0.8474

This highlights a very regular phenomenon when comparing CART and logistic regression. CART often performs a little worse than logistic regression in out-of-sample accuracy. However, as is the case here, the CART model is often much simpler to describe and understand.

EXPLANATION

First, generate predictions on the test set using the CART tree:

predictTest = predict(censustree, newdata = test, type = "class")

Then create the confusion matrix:

table(test$over50k, predictTest)

To compute the accuracy, sum the diagonal entries and divide by the sum of all of the terms:

(9243+1596)/(9243+470+1482+1596) = 0.8473927

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 2.5 - A CART MODEL (1/1 point)

Let us now consider the ROC curve and AUC for the CART model. 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 than the logistic regression ROC curve. Which of the following explanations for this behavior is most correct? (HINT: Think about what the ROC curve is plotting and what changing the threshold does.)

○ 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.

EXPLANATION

Choice 1 is on the right track, but is incorrect, because the number of variables that you use in a model does not determine how the ROC curve looks. In particular, try fitting logistic regression with hourperweek as the only variable; you will see that the ROC curve is very smooth.

Choice 2 is not correct. The smoothness of the ROC curve will generally depend on the number of data points, but in the case of the particular CART model we have estimated, varying the amount of testing set data will not change the qualitative behavior of the ROC curve.

Choice 3 is the correct answer. The breakpoints of the curve correspond to the false and true positive rates when the threshold is set to the five possible probability values.

Choice 4 is also not correct. In logistic regression, the continuity of an independent variable means that you will have a large range of predicted class probabilities in your test set data; this, in turn, means that you will see a large range of true and false positive rates as you change the threshold for generating predictions. In CART, the continuity of the variables does not at all affect the continuity of the predicted class probabilities; for our CART tree, there are only five possible probability values.

Hide Answer *You have used 1 of 2 submissions*

PROBLEM 2.6 - A CART MODEL (1/1 point)

What is the AUC of the CART model on the test set?

0.8470256

0.8470256

Answer: 0.8470

EXPLANATION

First, if you haven't already, load the ROCR package:

library(ROCR)

Generate the predictions for the tree. Note that unlike the previous question, when we call the predict function, we leave out the argument type = "class" from the function call. Without this extra part, we will get the raw probabilities of the dependent variable values for each observation, which we need in order to generate the AUC. We need to take the second column of the output:

```
predictTest = predict(censustree, newdata = test)

predictTest = predictTest[,2]
```

Compute the AUC:

```
ROCRpred = prediction(predictTest, test$over50k)

as.numeric(performance(ROCRpred, "auc")@y.values)
```

Hide Answer *You have used 2 of 3 submissions*

PROBLEM 3.1 - A RANDOM FOREST MODEL (1/1 point)

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 obervations 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 try to build a random forest model using the dataset "trainSmall" as the data used to build the model. Go ahead and attempt to build a random forest model. You should get an error that random forest "can not handle categorical predictors with more than 32 categories". This means that we have a factor variable with more than 32 different possible values. Which one of your variables is causing this error?

○ age
○ workclass
○ education
○ maritalstatus
○ occupation
○ relationship
○ race
○ sex
○ capitalgain
○ capitalloss
○ hoursperweek
● nativecountry ✓

EXPLANATION

To generate the random forest model with all of the variables, just run:

```
censusrf = randomForest( over50k ~ . , data = trainSmall)
```

The error you get is:

```
Error in randomForest.default(m, y, ...) :

Can not handle categorical predictors with more than 32 categories.
```

To figure out which variable is problematic, look at the data frame - for example:

```
str(census)
```

Of all the factor variables, nativecountry fits the error message -- from the str output:

```
$ nativecountry: Factor w/ 41 levels " Cambodia"," Canada",..: 39 39 39 39 39 39 39 39 39 39 ...
```

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 3.2 - A RANDOM FOREST MODEL (1/1 point)

Now, build your random forest model without the problematic variable identified in the previous problem. Set the seed to 1 before building the model. Remember to use the dataset "trainSmall" to build the model.

Then, make predictions using this model on the entire test set. What is the accuracy of the model on the test set? (Remember that you don't need a "type" argument when making predictions with a random forest model.)

0.8535689

0.8535689

Answer: 0.8520835

EXPLANATION

You can build your random forest model with the following commands:

```
set.seed(1)

censusrf = randomForest(over50k ~ . - nativecountry, data=trainSmall)
```

And then you can make predictions on the test set by using the following command:

```
predictTest = predict(censusrf, newdata=test)
```

And to compute the accuracy, you can create the confusion matrix:

```
table(test$over50k, predictTest)
```

The accuracy of the model is

(8861+2038)/nrow(test) = 0.8520835

Hide Answer *You have used 1 of 3 submissions*

PROBLEM 3.3 - A RANDOM FOREST MODEL (1/1 point)

As we discussed in lecture, 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 following variables is the most important in terms of the number of splits?

● age ✓

○ maritalstatus

○ capitalgain
○ education

EXPLANATION

If you run the three lines of R code given in this problem, you can see that age is used significantly more than the other variables.

Hide Answer *You have used 1 of 2 submissions*

PROBLEM 3.4 - A RANDOM FOREST MODEL (1/1 point)

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

○ workclass
● occupation ✓
○ sex
○ capitalloss

EXPLANATION

If you generate the plot with the command varImpPlot(MODEL), you can see that occupation gives a much larger reduction in impurity than the other variables.

Notice that the importance as measured by the average reduction in impurity is in general different from the importance as measured by the number of times the variable is selected for splitting. Although age and occupation are important variables in both metrics, the order of the variables is not the same in the two plots.

Hide Answer *You have used 1 of 2 submissions*

PROBLEM 4.1 - SELECTING CP BY CROSS-VALIDATION (1 point possible)

We now conclude our study of this data set by looking at how CART behaves with different choices of its parameters.

Let us select the cp parameter for our CART model using k-fold cross validation, with k = 10 folds. Do this by using the train function. Set the seed beforehand to 2. Test cp values from 0.002 to 0.1 in 0.002 increments, by using the following command:

cartGrid = expand.grid(.cp = seq(0.002,0.1,0.002))

Also, remember to use the entire training set "train" when building this model. The train function might take some time to run.

Which value of cp does the train function recommend?

Answer: 0.002

EXPLANATION

Before doing anything, load the caret package:

```
library(caret)
```

Set the seed to 2:

```
set.seed(2)
```

Specify that we are going to use k-fold cross validation with 10 folds:

```
fitControl = trainControl( method = "cv", number = 10 )
```

Specify the grid of cp values that we wish to evaluate:

```
cartGrid = expand.grid( .cp = seq(0.002,0.1,0.002))
```

Finally, run the train function and view the result:

```
train( over50k ~ . , data = train, method = "rpart", trControl = fitControl, tuneGrid = cartGrid )
```

The final output should read

```
Accuracy was used to select the optimal model using the largest value.

The final value used for the model was cp = 0.002.
```

In other words, the best value was cp = 0.002, corresponding to the lowest cp value. If we look more closely at the accuracy at different cp values, we can see that it seems to be decreasing steadily as the cp value increases. Often, the cp value needs to become quite low before the accuracy begins to deteriorate.

Hide Answer *You have used 0 of 4 submissions*

PROBLEM 4.2 - SELECTING CP BY CROSS-VALIDATION (1/1 point)

Fit a CART model to the training data using this value of cp. What is the prediction accuracy on the test set?

0.8612306

0.8612306

Answer: 0.8612

EXPLANATION

You can create a CART model with the following command:

```
model = rpart(over50k~., data=train, cp=0.002)
```

You can make predictions on the test set using the following command (where "model" is the name of your CART model):

```
predictTest = predict(model, newdata=test, type="class")
```

Then you can generate the confusion matrix with the command

```
table(test$over50k, predictTest)
```

The accuracy is (9178+1838)/(9178+535+1240+1838) = 0.8612306.

Hide Answer *You have used 1 of 4 submissions*

PROBLEM 4.3 - SELECTING CP BY CROSS-VALIDATION (1/1 point)

Compared to the original accuracy using the default value of cp, this new CART model is an improvement, and so we should clearly favor this new model over the old one -- or should we? Plot the CART tree for this model. How many splits are there?

18

18

Answer: 18

This highlights one important tradeoff in building predictive models. By tuning cp, we improved our accuracy by over 1%, but our tree became significantly more complicated. In some applications, such an improvement in accuracy would be worth the loss in interpretability. In others, we may prefer a less accurate model that is simpler to understand and describe over a more accurate -- but more complicated -- model.

EXPLANATION

If you plot the tree with prp(model), where "model" is the name of your CART tree, you should see that there are 18 splits!

Hide Answer *You have used 1 of 3 submissions*

Please remember not to ask for or post complete answers to homework questions in this discussion forum.

Show Discussion New Post