James Taylor

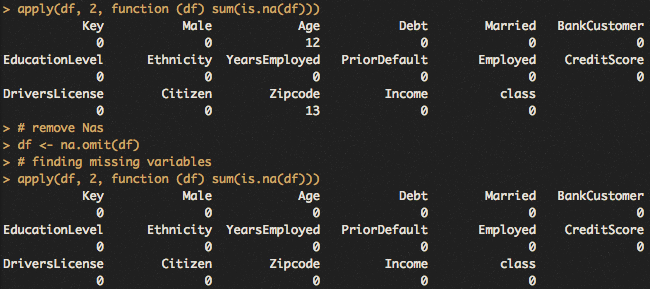
Exercise 5

1) Introduction

The dependent variable is named ‘class’. This indicates whether a particular applicant was approved for credit or not. The independent variables are Male, Age, Debt, Married, BankCustomer, EducationLevel, Ethnicity, YearsEmployed, PriorDefault, Employed, CreditScore, DriversLicense, Citizen, Zipcode and Income.

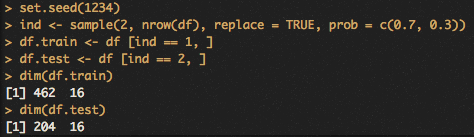
I expect the ‘ctree’ method to discover rules that allow the model to accurately predict who will be approved for credit.

2) Preprocessing



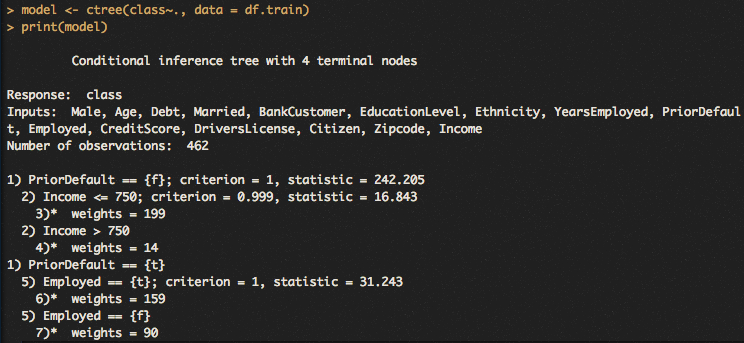
Above shows the removal of observations that had at least on missing value in the its variables. The variables ‘Key’ was also removed because it was a unique identifier for each observation. It is useless for our purposes.

3) Training and Test Data



Above is splitting the data into training and test data sets. Training data made up 70% of the original data, and the remaining is test data.

4) Build model



The above is a simple view of the model. There were no specified parameters besides from dependent and independent variables, and the source of the data being the training set.

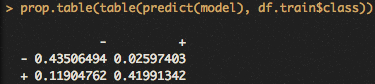
It lists the nodes and how they split the data. It parses the observations by prior default first. Then, for those who had no prior default, it splits them by income by above or below 750. For those who did have prior default, it breaks them into employed or unemployed. These variables, income and employment, are crucial to paying back borrowed money obviously.

5) Visualize



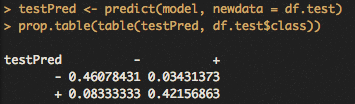
Above is a visualization of the model. It shows four terminal nodes. The root node is prior default. The goal is to have nodes which create ‘pure’ terminal sets. That means a terminal node where they are all one type of class of the dependent variable, for us it would be all approved or declined credit approvals. Above, node 3, 4 and 6 do a good job at purity but node 7 is not.

6) Confusion Matrix



The model had an accuracy of 85% on the training data. A confusion matrix also shows the Type I and II errors the model makes.

7) Evaluate on test data



The model had an accuracy of 88% on the test data. The accuracy was higher than the training data, which is a surprise but not unheard of.

8) New Instance

The model can be used easily to extract a prediction. The branches from the nodes are rules that determine how the model classifies observations. In our model it would require looking if an applicant had prior default, are unemployed or earn more/less than 750 in income. It will lead you to the prediction.

9) Summary

The main difference between the Apriori and ctree approach is the ctree approach has a lineage to it. All the predictions from the ctree model has to start from the root node. Apriori has no real starting point, only just predictions based on variable values.

Although I removed missing values, it does effect the way the model makes predictions. It creates other nodes that are used when the primary node has a missing value, so it can handle missing data but it’s not preferred.

I didn’t find the any particular part too laborious. Getting used to the outputs of the ctree and their meanings will come with practice and curiosity.