Lab 11 Decision tree

Lusine Zilfimian

April 29 (Wednesday), 2020

Contents

Needed packages

```
library(rpart)       # Recursive Partitioning and Regression Trees
library(rpart.plot)  # Plotting an rpart Model
library(CHAID) # for CHAID DT:
# https://r-forge.r-project.org/R/?group_id=343
library(rattle) # for fancyRpartPlot
library(ROCR) # for ROC curve
library(caret) # for createDataPartition()
library(dplyr) # again, you know it
library(ggplot2) # you know it, too
```

Data Preparation / Undestanding the data

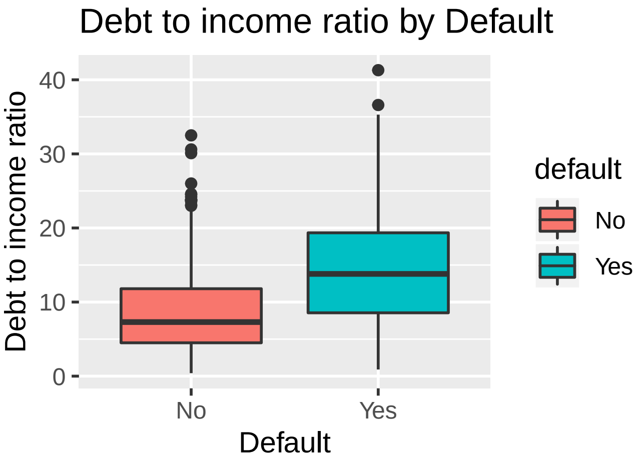
- Read the data Credit.csv

```
credit <- read.csv("credit.csv")
str(credit)
```

```
## 'data.frame':    700 obs. of  9 variables:
##  $ age     : int  41 27 40 41 24 41 39 43 24 36 ...
##  $ ed      : Factor w/ 5 levels "college degree",..: 1 3 3 3 2 2 3 3 3 3
##  $ employ  : int  17 10 15 15 2 5 20 12 3 0 ...
##  $ address : int  12 6 14 14 0 5 9 11 4 13 ...
##  $ income  : int  176 31 55 120 28 25 67 38 19 25 ...
##  $ debtinc : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
##  $ creddebt: num  11.359 1.362 0.856 2.659 1.787 ...
##  $ othdebt : num  5.009 4.001 2.169 0.821 3.057 ...
##  $ default : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 1 1 2 1 ...
```

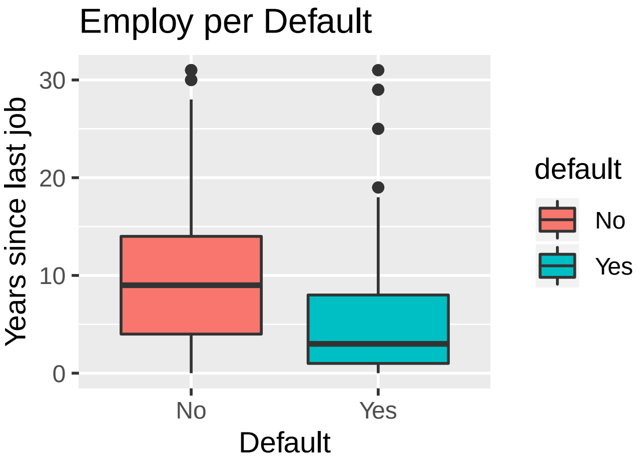
Undestanding the data

```
ggplot(data = credit, aes(x = default, y = debtinc, fill = default))+
  geom_boxplot() + labs(x = "Default", y = "Debt to income ratio",
    title = "Debt to income ratio by Default")
```



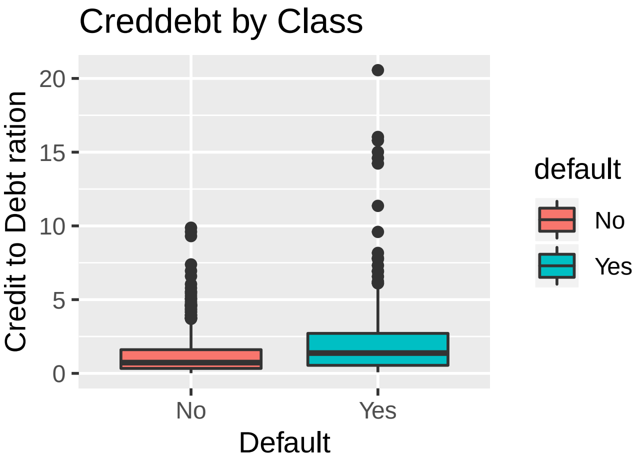
Undestanding the data

```
ggplot(data = credit, aes(x = default, y = employ, fill = default))+
  geom_boxplot() + labs(x = "Default", y = "Years since last job",
    title = "Employ per Default")
```



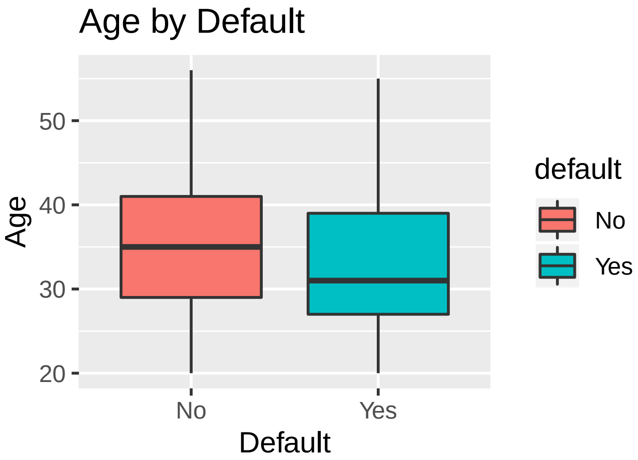
Undestanding the data

```
ggplot(data = credit, aes(x = default, y = creddebt, fill = default))+
  geom_boxplot() + labs(x = "Default", y = "Credit to Debt ration",
    title = "Creddebt by Class")
```



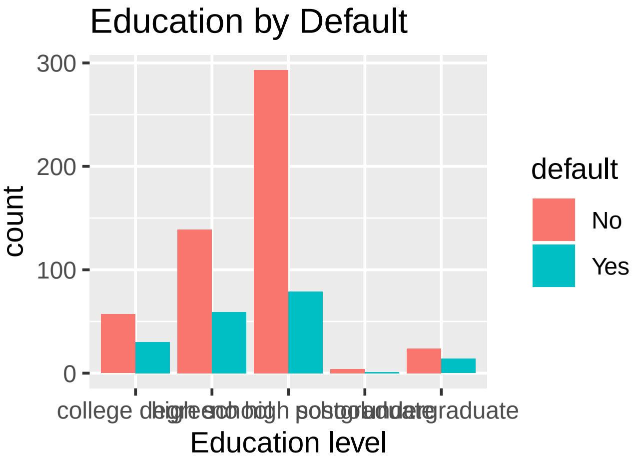
Undestanding the data

```
ggplot(data = credit, aes(x = default, y = age, fill = default))+
  geom_boxplot() + labs(x = "Default", y = "Age",
    title = "Age by Default")
```



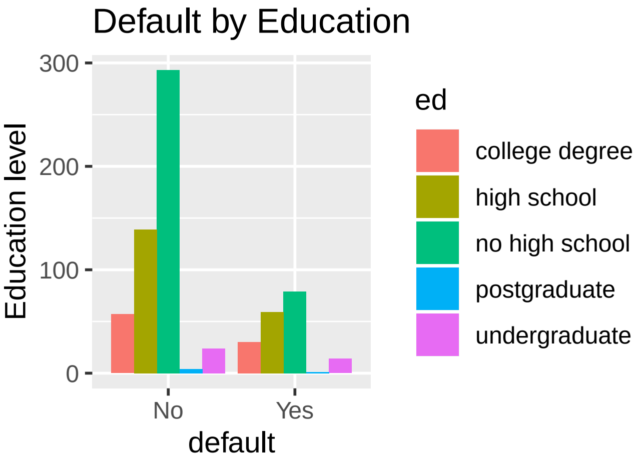
Undestanding the data

```
ggplot(data = credit, aes( x = ed, fill = default))+
  geom_bar(position = "dodge") + labs( x = "Education level",
    title = "Education by Default")
```



Undestanding the data

```
ggplot(data = credit, aes( fill = ed, x = default))+
  geom_bar(position = "dodge") + labs(y = "Education level",
    title = "Default by Education")
```



Undestanding the data

```
table(credit$default)/dim(credit)[1]
```

```
##
##        No       Yes
## 0.7385714 0.2614286
```

Data Preparation

- Divide the dataset into training and testing sets:

```
set.seed(2708)
split <- credit$default %>% createDataPartition(p = 0.8, list = FALSE)

train.data  <- credit[split, ]
test.data <- credit[-split, ]

table(train.data$default)/dim(train.data)[1]
```

```
##
##        No       Yes
## 0.7379679 0.2620321
```

DT: Classification

- Specify the formula, with independent and dependent variables

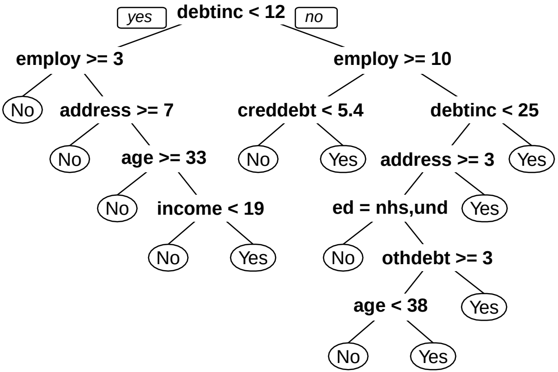
```
model_c <- rpart(formula = default ~ .,
  data = train.data, method = "class")
model_c
```

```
## n= 561
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 561 147 No (0.73796791 0.26203209)
##     2) debtinc< 12.35 369  55 No (0.85094851 0.14905149)
##       4) employ>=2.5 301  30 No (0.90033223 0.09966777) *
##       5) employ< 2.5 68  25 No (0.63235294 0.36764706)
##        10) address>=6.5 23   3 No (0.86956522 0.13043478) *
##        11) address< 6.5 45  22 No (0.51111111 0.48888889)
##          22) age>=33 9   1 No (0.88888889 0.11111111) *
##          23) age< 33 36  15 Yes (0.41666667 0.58333333)
##            46) income< 19 14   5 No (0.64285714 0.35714286) *
##            47) income>=19 22   6 Yes (0.27272727 0.72727273) *
```

DT: Classification

- Plot using rpart.plot library

```
prp(model_c)
```



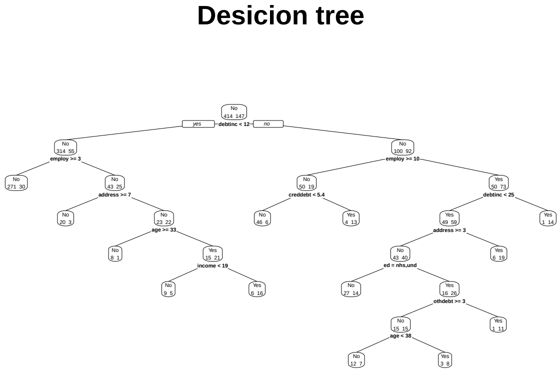
DT: Classification

- `type` argument is used to get different layouts for the tree.
- `extra` argument is used to add extra information
- Look for help for more info: `?rpart.plot::prp`

DT: Classification

- Number of observations that fall in the node per class

```r
prp(model_c, type = 2, extra = 1, main  = "Desicion tree")
```

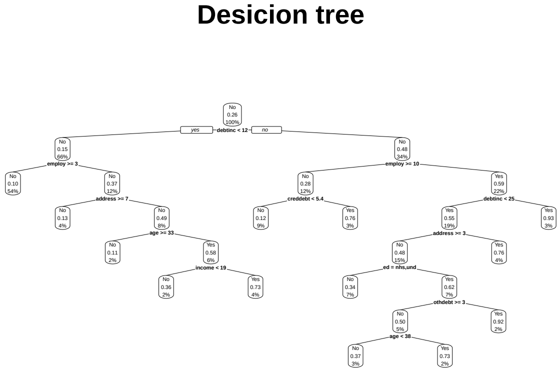



```r
# add proportion
```

DT: Classification

- Proportion of largest class or classification rate at the node (extra = 2)
- Misclassification rate at the node (extra = 3)
- Probabilities per class (extra = 4)

```r
prp(model_c, type = 2, extra = 106, main  = "Desicion tree")
```



DT: Classification

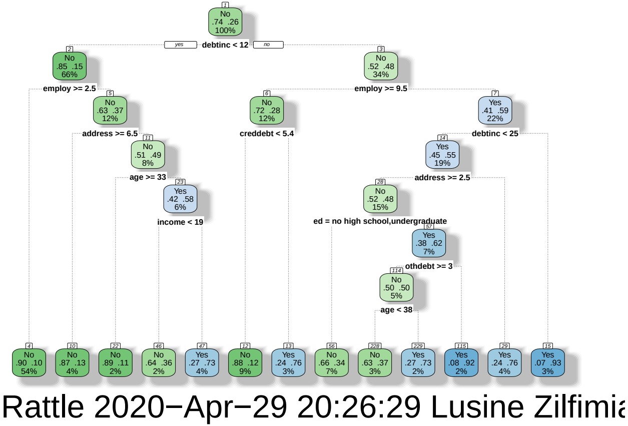
- Look at the decision rules using rattle package
- The predicted class is Yes, it covers 3% of the data, overall 15 cases in the terminal node, probability of Yes is 0.93

```
asRules(model_c) # from rattle
```

```
##
##  Rule number: 15 [default=Yes cover=15 (3%) prob=0.93]
##    debtinc>=12.35
##    employ< 9.5
##    debtinc>=24.85
##
##  Rule number: 115 [default=Yes cover=12 (2%) prob=0.92]
##    debtinc>=12.35
##    employ< 9.5
##    debtinc< 24.85
##    address>=2.5
##    ed=college degree,high school
##    othdebt< 3.028
##
##  Rule number: 13 [default=Yes cover=17 (3%) prob=0.76]
##    debtinc>=12.35
```

DT: Classification

```
fancyRpartPlot(model_c)
```



DT: Classification

- What about Rule number 4
- The terminal node predicts No,
- covers 54% of the data (301 cases)
- prob = 0.10, probability of Yes is 0.10, for No is 0.9

Controlling the tree

- by default Gini coefficient is the impurity measure
- Tree is pruned using Complexity parameter
- Other parameters to control tree growth
- **minsplit**: the minimum number of observations that must exist in a node in order for a split to be attempted
- **minbucket**: the minmum number of observations in any terminal node

Make predictions

```r
pred_prob <- predict(model_c, test.data, type = "prob")
pred_prob[1:10,]
```

```
##           No        Yes
## 1  0.9003322 0.09966777
## 3  0.9003322 0.09966777
## 5  0.2400000 0.76000000
## 18 0.9003322 0.09966777
## 20 0.8695652 0.13043478
## 22 0.9003322 0.09966777
## 30 0.2400000 0.76000000
## 46 0.6428571 0.35714286
## 49 0.2727273 0.72727273
## 53 0.8846154 0.11538462
```

Make predictions

```r
pred_class <- predict(model_c, test.data, type = "class")
pred_class[1:20]
```

```
##   1   3   5  18  20  22  30  46  49  53  56  63  77  79  85  90  92  96
##  No  No Yes  No  No  No Yes  No Yes  No  No Yes Yes  No  No Yes  No  No
##  98 101
##  No  No
## Levels: No Yes
```

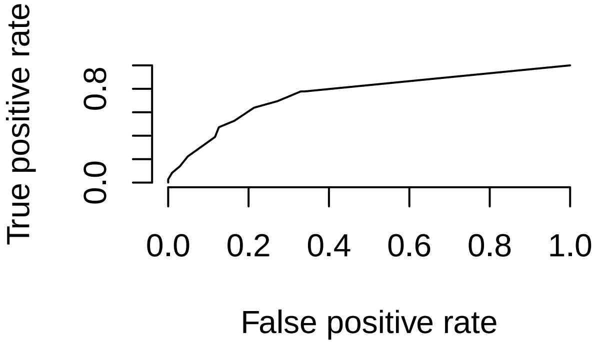
Accuracy

```r
confusionMatrix(pred_class, test.data$default, positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction No Yes
##        No  91  22
##        Yes 12  14
##
##                Accuracy : 0.7554
##                  95% CI : (0.6753, 0.8243)
##     No Information Rate : 0.741
##     P-Value [Acc > NIR] : 0.3913
##
##                   Kappa : 0.2994
##
##  Mcnemar's Test P-Value : 0.1227
##
##             Sensitivity : 0.3889
##             Specificity : 0.8835
##          Pos Pred Value : 0.5385
##          Neg Pred Value : 0.8053
```

Accuracy

```
P_Test <- prediction(pred_prob[,2], test.data$default)
perf <- performance(P_Test, "tpr", "fpr")
plot(perf)
```



AUC

```
performance(P_Test,"auc")@y.values
```

```
## [[1]]
## [1] 0.7549892
```

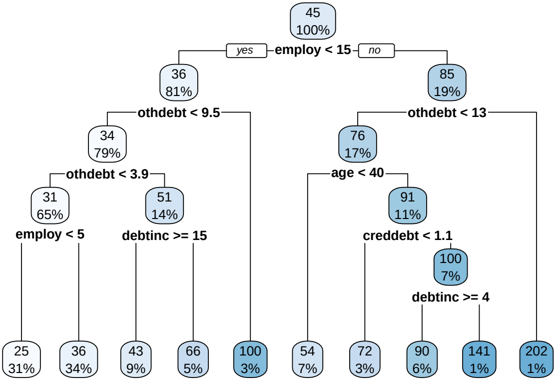
DT: Regression

```
model_r <- rpart(formula = income ~ ., data = train.data)
model_r
```

```
## n= 561
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 561 731402.000  45.34046
##    2) employ< 14.5 457 218524.400  36.33917
##      4) othdebt< 9.507397 442 117774.800  34.18552
##        8) othdebt< 3.866795 365  65950.300  30.61644
##         16) employ< 4.5 174  13968.810  24.84483 *
##         17) employ>=4.5 191  40904.980  35.87435 *
##        9) othdebt>=3.866795 77  25135.170  51.10390
##         18) debtinc>=14.65 50   7847.380  43.18000 *
##         19) debtinc< 14.65 27   8334.667  65.77778 *
##      5) othdebt>=9.507397 15  38290.400  99.80000 *
##    3) employ>=14.5 104 313141.800  84.89423
##      6) othdebt< 12.69019 97 118617.900  76.44330
##       12) age< 39.5 38   6802.868  53.76316 *
```

DT: Regression

```r
rpart.plot(model_r)
```



DT: Regression

- The percentage of data that fall to that node and the average income for that branch.
- We have 8 internal nodes and 10 terminal node
- This tree is partitioning on 6 variables to produce its model. However, there are 9-1 variables in `train.data`.

CHAID

```r
summary(credit$debtinc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.40    5.00    8.60   10.26   14.12   41.30
```

```r
credit$diratio <- as.factor(ifelse(credit$debtinc > 10, "High", "Low"))

addmargins(table(credit$diratio,credit$default))
```

```
##
##         No Yes Sum
##   High 173 124 297
##   Low  344  59 403
##   Sum  517 183 700
```

```r
addmargins(table(credit$ed,credit$default))
```

```
##
##                     No Yes Sum
##   college degree    57  30  87
##   high school      139  59 198
```

CHAID

```r
model_c2<-chaid(default ~ ed + diratio, data = credit)
print(model_c2)
```

```
##
## Model formula:
## default ~ ed + diratio
##
## Fitted party:
## [1] root
## |   [2] diratio in High: No (n = 297, err = 41.8%)
## |   [3] diratio in Low: No (n = 403, err = 14.6%)
##
## Number of inner nodes:    1
## Number of terminal nodes: 2
```

CHAID

```r
plot(model_c2,  gp = gpar(fontsize = 8))
```