26 BE 7082 + 26 PH 7028 + 20 BME 7082
Introduction to Data Science
Autumn 2020

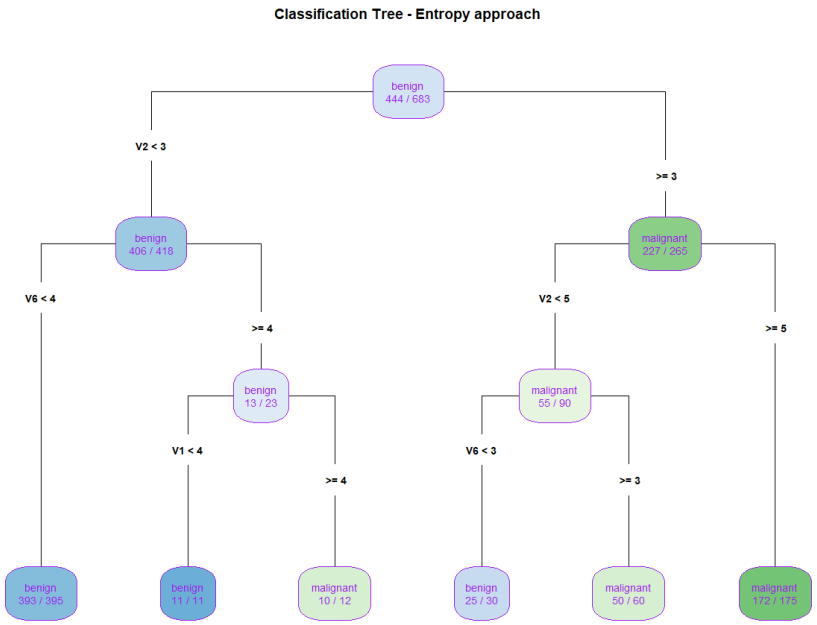
MB Rao

Homework No. 6 Due date: October 08, 2020 Maximum points: 30

Theme: Decision Trees – p value approach

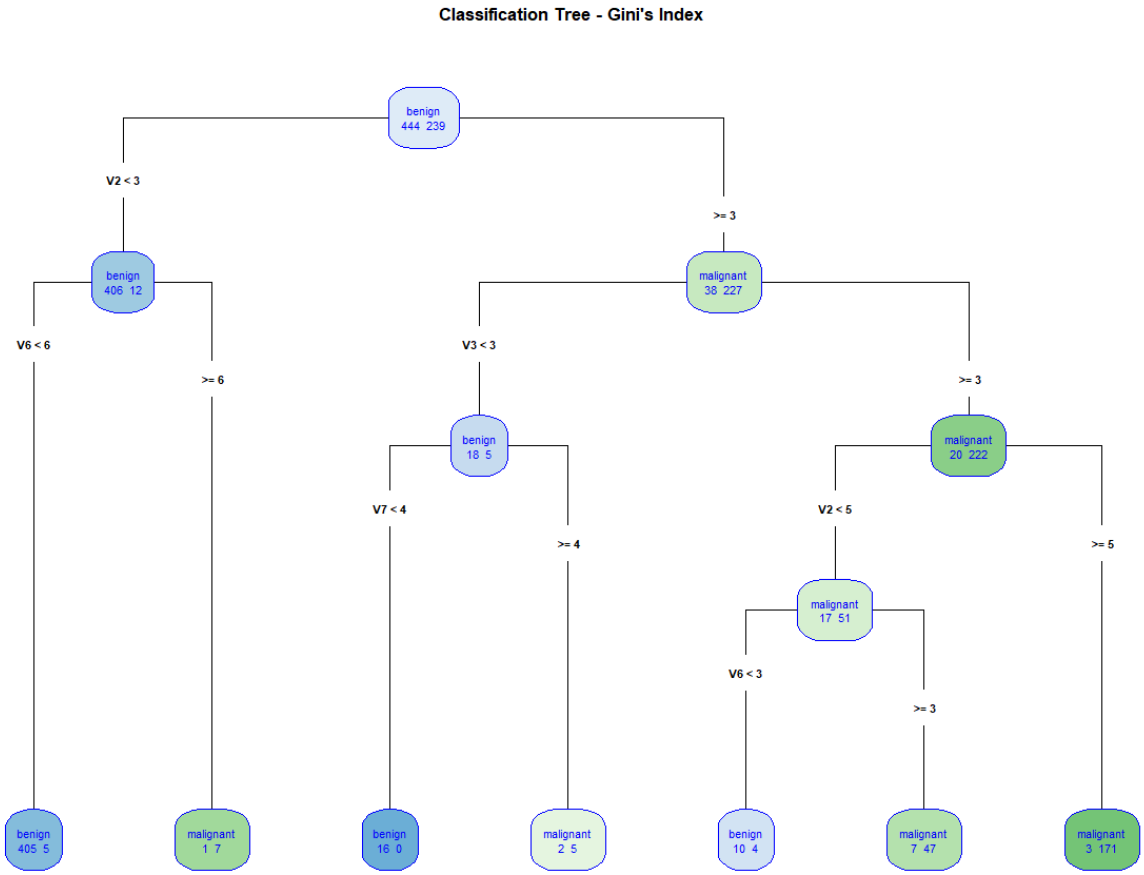
This homework will be light. I have touched upon how to build a decision tree using the p value approach in my last lecture. The response variable was numeric.

1. Go back to the Wisconsin Breast Cancer Data. Build classification trees using the following optimality criteria.
I got rid of NA's and ID column using following lines of code:
data(biopsy)
head(biopsy)
df=na.omit(biopsy)
df1=subset(df, select=-c(ID))
a. Entropy
SF<-rpart(class~., data=df1, parms=list(split="information"))
rpart.plot(SF, type=4, col="purple", digits=3, extra=2, main="Classification Tree - Entropy approach")



b. Gini

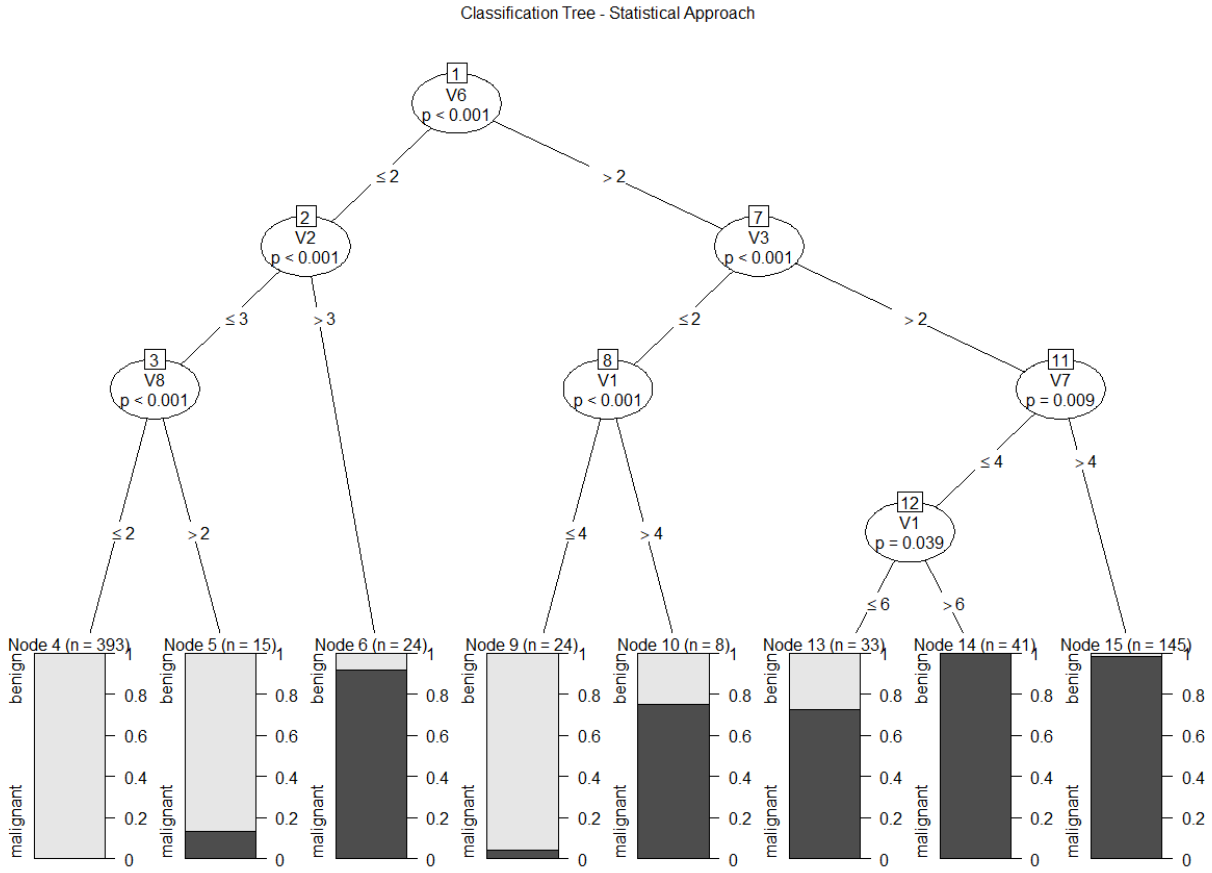
df3<-rpart(class~., data=df1)
rpart.plot(df3, col="blue", type=4, extra=1, main="Classification Tree - Gini's Index")



c. p value

df2<-ctree(class~., data=df1)

plot(df2, main="Classification Tree - Statistical Approach")



In the case of entropy and Gini criteria, the default pruning principle is stop splitting the tree if the size of the node is 20 or less. What is the default in the case of the p value criterion?

I used ctree_control to extract default control values for the ctree function. It indicated that a default split criterion, i.e. **mincriterion is defined as (1-alpha) and alpha is taken as 0.05.** i.e., the test statistics criterion is maximized to 0.95. This further means that a node is split once the **p-value is less than 0.05**. I have highlighted this below.

```
> ctree_control
function (teststat = c("quadratic", "maximum"), splitstat = c("quadratic",
    "maximum"), splittest = FALSE, testtype = c("Bonferroni",
    "MonteCarlo", "Univariate", "Teststatistic"),
    pargs = GenzBretz(), nmax = c(yx = Inf, z = Inf), alpha = 0.05,
    mincriterion = 1 - alpha, logmincriterion = log(mincriterion),
    minsplit = 20L, minbucket = 7L, minprob = 0.01, stump = FALSE,
    lookahead = FALSE, MIA = FALSE, nresample = 9999L, tol = sqrt(.Machine$double.eps),
    maxsurrogate = 0L, numsurrogate = FALSE, mtry = Inf, maxdepth = Inf
```

Contrast the trees. **8 + 8 + 8 + 6 points**

Classification tree optimized using Entropy.

Using information gain the classification tree was created to split a node when it reaches minimum entropy or in other words the node is split once highest information gain (reduction in entropy) is returned. The classification tree thus resulting is symmetrical and relies on node homogeneity to stop the split. The implementation is carried out by defining parms (optional parameters) as split='information' indicating splitting of nodes is dependent on information gain.
Description of tree is given below:

a) The cancer is classified as benign when V<3 and V6<4.
b) The cancer is classified as benign when V2<3, V6>=4 and V1<4.
c) The cancer is classified as malignant when V2<3, V6>=4 and V1>=4.
d) The cancer is classified as benign when V2<5 and V6<3.
e) The cancer is classified as malignant when V2<5 and V6>=3.
f) The cancer is classified as malignant when V2>=5.

Misclassification Rate=2+0+2+5+10+8/683=**3.95%**

Classification tree optimized using GINI's Index

In rpart, the default criterion for optimizing classification/regression tree is GINI's index thus the only control parameter is control= rpart.control(minsplit=n) where n is 20 by default. This hyperparameter can be altered to prune and fine tune the tree. However, I have chosen to use the default node split size of 20. This resulted in a relatively better (deeper) classification tree. The tree is described below:

a) The cancer is classified as benign if V2<3 and V6<6.
b) The cancer is classified as malignant if V2<3 and V6>=6.
c) The cancer is classified as benign if V2>=3, V3<3 and V7<4.
d) The cancer is classified as malignant if V2>=3, V3<3 and V7>=4.
e) The cancer is classified as benign if V2<6, V3>=3 and V6<3.
f) The cancer is classified as malignant if V2<6, V3>=3 and V6>=4.
g) The cancer is classified as malignant if V2>=5 and V3>=3.

As obvious an additional terminal daughter node is added and moreover an additional variable V3 was utilized for splitting the node compared to classification tree drawn using Entropy or information gain as node splitting criterion.

Misclassification rate= (5+1+0+2+4+7+3)/683=**3.22%**

Classification tree using p-value criterion.

A much more graphically informative graph with additional statistical information as well. The tree contains 15 nodes compared to a total of 11 nodes using entropy and 13 using GINI based optimization. The tree has a total of 8 terminal nodes compared to 6 and 7 terminal nodes of entropy and GINI based optimized trees. The terminal nodes use bar diagrams to depict classification of cancer being malignant or benign. Each node contains the p value below which the node was split. The highest p value is denoted as 0.039 for node no. 12 opposed to 0.05 as set default and described in previous part of the question. This results in very high accuracy compared to both entropy or GINI index criterion.

The description of tree can be obtained using simple r statement calling the name of folder that was used for storing ctree output.

```
> df2

Model formula:
class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9

Fitted party:
[1] root
|   [2] V6 <= 2
|   |   [3] V2 <= 3
|   |   |   [4] V8 <= 2: benign (n = 393, err = 0.0%)
|   |   |   [5] V8 > 2: benign (n = 15, err = 13.3%)
|   |   [6] V2 > 3: malignant (n = 24, err = 8.3%)
|   [7] V6 > 2
|   |   [8] V3 <= 2
|   |   |   [9] V1 <= 4: benign (n = 24, err = 4.2%)
|   |   |   [10] V1 > 4: malignant (n = 8, err = 25.0%)
|   |   [11] V3 > 2
|   |   |   [12] V7 <= 4
|   |   |   |   [13] V1 <= 6: malignant (n = 33, err = 27.3%)
|   |   |   |   [14] V1 > 6: malignant (n = 41, err = 0.0%)
|   |   |   [15] V7 > 4: malignant (n = 145, err = 1.4%)

Number of inner nodes:    7
Number of terminal nodes: 8
```

Here,

A. Node 4 is describing cancer as benign if V6<=2, V2<=3 and V8<=2 with 0% error.
B. Node 5 is describing cancer as benign with an error of 13.3% if V6<=2, V2<=3 and V8>2.
C. Node 6 is describing cancer as malignant with an error of 8.3% if V6<2 but V2>3.
D. Node 9 is describing cancer as benign with an error of 4.2% if V6>2, V3<=2 and V1<=4.
E. Node 10 describes cancer as malignant with an error of 25% if V6>2, V3<=2 and V1>4.
F. Node 13 describes cancer as malignant with an error of 27.3% if V6>2, V3>2, V7<=4 and V1<=6.
G. Node 14 describes cancer as malignant with an error of 0% if V6>2, V3>2, V7<=4 and V1>6.
H. Node 15 describes cancer as malignant with an error of 1.4% if V6>2, V3>2, V7>4.

Misclassification Rate:
(15*0.133+24*0.083+24*0.042+8*0.25+13*0.273+145*0.014)/(683)=**0.3%**

A significant improvement over Classification trees built using Entropy or GINI Index as optimality criterion.