Introduction to Data Science

DSA1101

Semester 1, 2018/2019
Week 8

Classification methods: Decision Trees

Decision Trees

- Recall the *decision tree* example from week 6 which concerns a bank that wants to market its term deposit products (such as Certificates of Deposit) to the appropriate customers.
- Given the demographics of clients and their reactions to previous campaign phone calls, the bank's goal is to predict which clients would subscribe to a term deposit.
- We will look into the decision tree algorithm in greater detail, including how the decision variables at each node are selected.

Decision Trees

- The dataset ‘bank-sample.csv’ which has been posted to IVLE contains records of 2000 customers
- The variables include (1) job, (2) marital status, (3) education level, (4) if the credit is in default, thiếu tiền (5) if there is a housing loan, (6) if the customer currently has a personal loan, (7) contact type, (8) result of the previous marketing campaign contact (poutcome), and finally (9) if the client actually subscribed to the term deposit.

Decision Trees

- Attributes (1) through (8) are the input variables or features
- (9) is considered the (binary) outcome: The outcome subscribed is either yes (meaning the customer will subscribe to the term deposit) or no (meaning the customer won't subscribe).
- All the variables listed earlier are categorical.

Decision Trees

- Preliminary look at the dataset
-

```
> bankdata = read.csv("bank-sample.csv", header=TRUE)
> head(bankdata)
  age          job  marital education default balance
1  31   management   single  tertiary      no       0
2  45 entrepreneur  married  tertiary      no    1752
3  46     services divorced secondary      no    4329
4  35   management  married  tertiary      no    1108
5  39   management  married secondary      no    1410
6  31   management   single  tertiary      no     499
```

Decision Trees

- Preliminary look at the dataset
-

```
  housing loan  contact day month duration campaign
1     yes   no cellular  15   apr      185        2
2     yes  yes cellular  20   nov       56        2
3      no   no cellular  21   nov      534        2
4     yes   no cellular  17   nov       52        1
5     yes   no  unknown  23   may       55        1
6     yes   no  unknown   9   jun      122        2
  pdays previous poutcome subscribed
1    -1        0  unknown         no
2    -1        0  unknown         no
3    -1        0  unknown        yes
4    -1        0  unknown         no
5    -1        0  unknown         no
6    -1        0  unknown         no
```

Decision Trees

- In R, the package `rpart` contains functions for modeling decision trees
- The optional package `rpart.plot` enables the plotting of a tree.
- We will show how to use decision trees in R to predict which clients would subscribe to a term deposit.
-

```
install.packages("rpart")
install.packages("rpart.plot")
library("rpart")
library("rpart.plot")
```

Decision Trees

- We will build a decision tree to predict `subscribed` based on the features: `job`, `marital`, `education`, `default`, `housing`, `loan`, `contact` and `poutcome`.
- We will study how the decision tree is fitted in more detail after the recess week
-

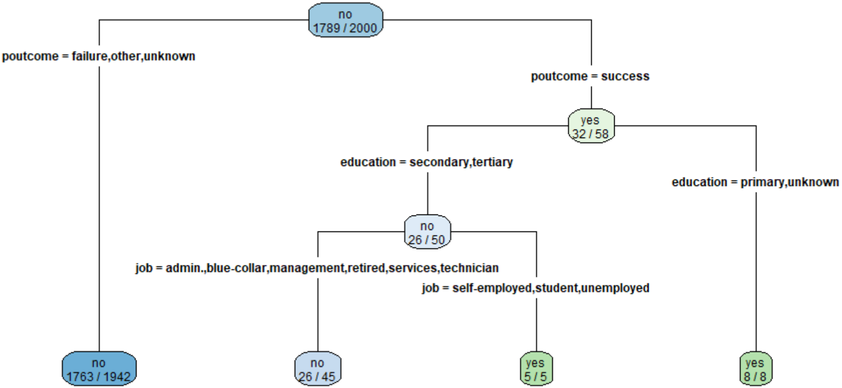
```
fit <- rpart(subscribed ~job + marital + education
+ default + housing + loan + contact + poutcome,
method="class",
data=bankdata,
control=rpart.control(minsplit=1),
parms=list(split='information'))
```

Decision Trees

- We can visualize the resulting fitted decision tree using `rpart.plot`:

```
rpart.plot(fit, type=4, extra=2, clip.right.labs=
    FALSE, varlen=0, faclen=0)
```

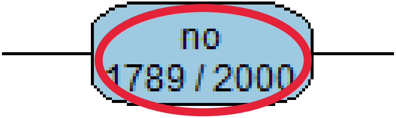
Decision Trees



Decision Tree algorithm

- *Question*: Why is the variable `poutcome` selected as the decision variable at the root node?
- *Question*: Traversing down the tree, how are the subsequent decision variables at each node selected?

Classification methods: Decision Trees



- The *purity* of a node is defined as its probability of the corresponding class
- For example, in the root node of the decision tree built earlier, $P(\text{subscribed} = 0) = \frac{1789}{2000} \approx 89.45\%$
- Therefore, the root is 89.45% pure on the subscribed $= 0$ class and 10.55% pure on the subscribed $= 1$ class

Decision Tree algorithm

- The first step in constructing a decision tree is to choose the most informative attribute.
- A common way to identify the most informative attribute is to use entropy-based methods, which are used by decision tree learning algorithms such as ID3 (or Iterative Dichotomiser 3) and C4.5.
- The entropy methods select the most informative attribute based on two basic measures:

(i) *Entropy*, which measures the impurity of an attribute

(ii) *Information gain*, which measures the purity of an attribute

Decision Tree algorithm: entropy

- Given variable Y and and the set of possible categorical values it can take, $(y_1, y_2, ..., y_K)$, the entropy of Y is defined as

$$D_Y = -\sum_{j=1}^{K} P(Y = y_j)(\log_2 P(Y = y_j)),$$

where $P(Y = y_j)$ denotes the purity or the probability of the class $Y = y_j$, and $\sum_{j=1}^{K} P(Y = y_j) = 1$.

Decision Tree algorithm: entropy

- If the variable Y is binary and only take on two values 0 or 1, the entropy of Y is

$$-\{P(Y=1)(\log_2 P(Y=1)) + P(Y=0)(\log_2 P(Y=0))\}.$$

- For example, let Y denote the outcome of a coin toss, where $Y=1$ for head and $Y=0$ for tail.
- If the coin if a fair one, then $P(Y=0)=P(Y=1)=\frac{1}{2}$, so that the entropy is calculated as

$$-\{0.5\log_2 0.5 + 0.5\log_2 0.5\} = 1$$

- On the other hand, if the coin is biased, then suppose $P(Y=0)=\frac{3}{4}$, $P(Y=1)=\frac{1}{4}$, so that the entropy is now

$$-\{0.25\log_2 0.25 + 0.75\log_2 0.75\} \approx 0.81$$

Decision Tree algorithm: entropy

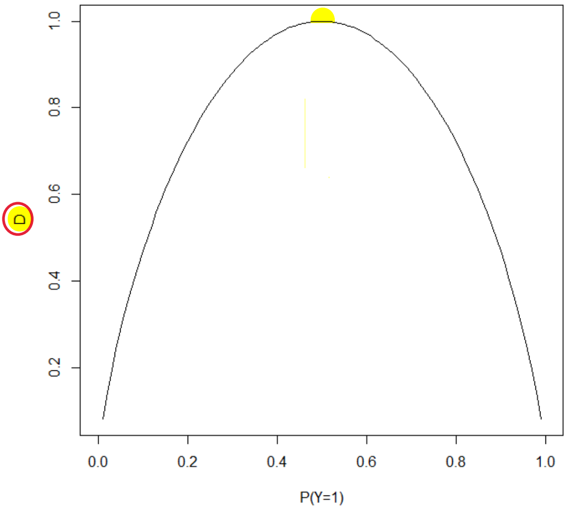
Practically Less entropy, lower uncertainty.

- Heuristically, entropy is a measure of unpredictability
- When the coin is biased, we have less "uncertainty" in predicting the outcome of its next toss, so that the entropy is lower
- When the coin is fair, we are much more less able to predict the next toss, and so the entropy is at its highest value 1
- For a binary variable Y, we can plot in R its entropy:

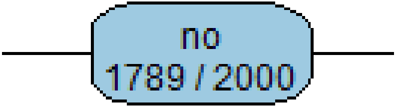
Decision Tree algorithm: entropy

```
p=seq(0,1,0.01)
D=-(p*log2(p)+(1-p)*log2(1-p))
plot(p,D,ylab="D", xlab="P(Y=1)", type="l")
```

Decision Trees



Classification methods: Decision Trees



- For the bank marketing example, the output Y variable is `subscribed`
- The base entropy is defined as the entropy of the output variable at the root node
- $P(\text{subscribed} = 0) = \frac{1789}{2000} \approx 89.45\%$ and $P(\text{subscribed} = 1) = 1 - \frac{1789}{2000} \approx 10.55\%$

Decision Tree algorithm: entropy

- Therefore, the base entropy is $D_{\text{subscribed}} = -\{0.1055 log_2(0.1055) + 0.8945 log_2(0.8945)\} \approx 0.4862$.
- Ideally, we would like to reduce this base entropy by leveraging on feature variables X for prediction.
- Recall that lower entropy is associated with less "uncertainty" in predicting the outcome, which is something that we want.
- So we select the feature that reduces entropy the most.

Decision Tree algorithm: entropy

- We consider binary tree algorithm
- Suppose we have a feature variable X and the split values (x_1, x_2). The conditional entropy given feature X and the split points (x_1, x_2) is defined as

x_1 x_2

$$D_{Y|X} = \sum_{i=1}^{2} P(X = x_i) D(Y|X = x_i)$$

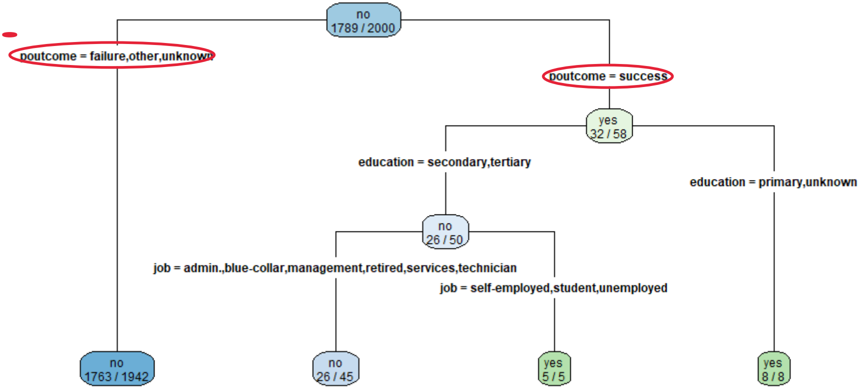
$$= -\sum_{i=1}^{2} \left\{ P(X = x_i) \sum_{j=1}^{K} P(Y = y_j|X = x_i) log_2[P(Y = y_j|X = x_i)] \right\}$$

- We will illustrate with several examples shortly

Decision Tree algorithm: entropy

- We will illustrate the calculation of conditional entropy for the decision variable in the root node, `poutcome`.
- Recall that the split categories are x_1 : `failure,other,unknown` and x_2: `success`.

Decision Trees



Decision Tree algorithm: entropy

```
> length(bankdata$poutcome)
[1] 2000
> table(bankdata$poutcome)

failure    other  success  unknown
    210       79       58     1653
```

	poutcome (X)	
	x_1 : failure,other,unknown	x_2: success
$P(X = x_i)$	$\frac{210+79+1653}{2000} = 0.971$	$\frac{58}{2000} = 0.029$

Decision Tree algorithm: entropy

```
> x1=which(bankdata$poutcome!="success")
> x2=which(bankdata$poutcome=="success")
> table(bankdata$subscribed[x1])

  no   yes
1763   179
> table(bankdata$subscribed[x2])

 no  yes
 26   32
```

failure + other + unknown = 1942

P(Y=0)=1763/1942 given in the tree

P(Y=1)=(1942-1763)/1942

	poutcome (X)	
	x_1: failure,other,unknown	x_2: success
$\boxed{P(X = x_i)}$	$\frac{210+79+1653}{2000} = \frac{1942}{2000} = 0.971$	$\frac{58}{2000} = 0.029$
$P(Y = 1\boxed{X = x_i})$	$\frac{179}{1942} \approx 0.092$	$\frac{32}{58} \approx 0.552$
$P(Y = 0\boxed{X = x_i})$	$\frac{1763}{1942} \approx 0.908$	$\frac{26}{58} \approx 0.448$

Decision Tree algorithm: entropy

- Therefore the conditional entropy for selecting `poutcome` as decision variable with the split at x_1 : `failure,other,unknown` and x_2: `success` is

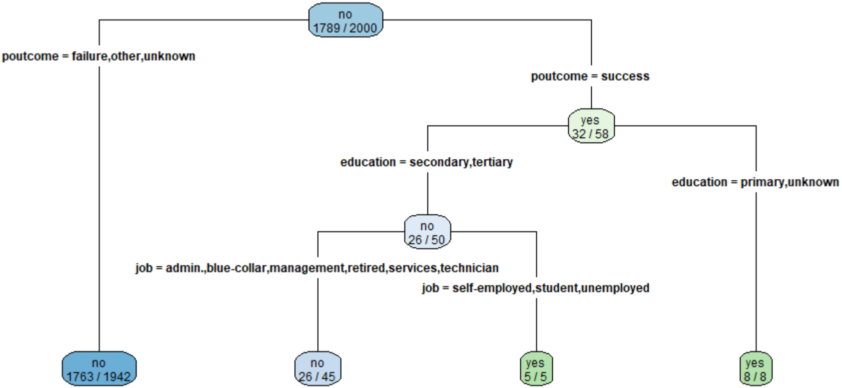
$$
\begin{aligned}
&D_{subscribed|poutcome}\\
&= -\sum_{i=1}^{2}\left\{\boxed{P(X=x_i)}\boxed{\sum_{j=1}^{2}P(Y=y_j|X=x_i)log_2[P(Y=y_j|X=x_i)]}\right\}\\
&= -\{0.971 \times [0.092log_2(0.092) + 0.908log_2(0.908)]\\
&+0.029 \times [0.552log_2(0.552) + 0.448log_2(0.448)]\}\\
&\approx 0.459
\end{aligned}
$$

- Therefore, there is a reduction of about $0.4862 - 0.459 \approx 0.027$ from the base entropy.
- This reduction in entropy is also known as information gain.

Decision Tree algorithm: entropy

- We can calculate the reduction for other feature variables and /or split points and show that they are all less than the entropy reduction of approximately 0.027.
- For example, using the same feature variable `poutcome`, let us instead calculate the conditional entropy for splitting at the values x_1 : `other,success,unknown` and x_2: `failure`
- We shall show why this split is not the one in the decision tree built earlier, in terms of entropy reduction

Decision Trees



Decision Tree algorithm: entropy

```
> length(bankdata$poutcome)
[1] 2000
> table(bankdata$poutcome)

failure   other  success  unknown
    210      79       58     1653
```

-

	poutcome (X)	
	x_1 : success,other,unknown	x_2: failure
$P(X = x_i)$	$\frac{58+79+1653}{2000} = 0.895$	$\frac{210}{2000} = 0.105$

Decision Tree algorithm: entropy

```
> x1=which(bankdata$poutcome!="failure")
> x2=which(bankdata$poutcome=="failure")
> table(bankdata$subscribed[x1])

  no  yes
1600  190
> table(bankdata$subscribed[x2])

 no yes
189  21
```

	`poutcome` (X)	
	x_1 : `success,other,unknown`	x_2: `failure`
$P(X = x_i)$	$\frac{58+79+1653}{2000} = \frac{1790}{2000} = 0.895$	$\frac{210}{2000} = 0.105$
$P(Y = 1 \mid X = x_i)$	$\frac{190}{1790} \approx 0.106$	$\frac{21}{210} = 0.10$
$P(Y = 0 \mid X = x_i)$	$\frac{1600}{1790} \approx 0.894$	$\frac{189}{210} = 0.90$

Decision Tree algorithm: entropy

- Therefore the conditional entropy for selecting `poutcome` as decision variable with the split at x_1 : `success,other,unknown` and x_2: `failure` is

$$
\begin{aligned}
& D_{subscribed|poutcome} \\
&= -\sum_{i=1}^{2}\left\{P(X = x_i)\sum_{j=1}^{2}P(Y = y_j|X = x_i)log_2[P(Y = y_j|X = x_i)]\right\} \\
&= -\{0.895 \times [0.106log_2(0.106) + 0.894log_2(0.894)] \\
&+0.105 \times [0.10log_2(0.10) + 0.90log_2(0.90)]\} \\
&\approx 0.486
\end{aligned}
$$

- Therefore, there is a reduction of about $0.4862 - 0.486 \approx 0.0002$ from the base entropy.
- This information gain is far less than splitting at x_1 : `failure,other,unknown` and x_2: `success`

Decision Tree algorithm: entropy

- We can calculate the reduction for other feature variables and /or split points and show that they are all less than the entropy reduction of approximately 0.027.
- For example, instead of the feature variable `poutcome`, let us calculate the conditional entropy for choosing feature variable `education` at the split points x_1 : `tertiary` and x_2 : `secondary,primary,unknown`
- We shall show why `education` is not the decision variable for the root node.

Decision Tree algorithm: entropy

```
> length(bankdata$education)
[1] 2000
>
> table(bankdata$education)

  primary secondary  tertiary   unknown
      335      1010       564        91
```

-

	education (X)	
	x_1 : tertiary	x_2: secondary,primary,unknown
$P(X = x_i)$	$\frac{564}{2000} = 0.282$	$\frac{335+1010+91}{2000} = 0.718$

Decision Tree algorithm: entropy

```
> x1=which(bankdata$education=="tertiary")
> x2=which(bankdata$education!="tertiary")
> table(bankdata$subscribed[x1])

 no  yes
494   70
> table(bankdata$subscribed[x2])

  no   yes
1295   141
```

-

	education (X)	
	x_1 : tertiary	x_2: secondary,primary,unknown
$P(X = x_i)$	$\frac{564}{2000} = 0.282$	$\frac{335+1010+91}{2000} = \frac{1436}{2000} = 0.718$
$P(Y = 1 \mid X = x_i)$	$\frac{70}{564} \approx 0.124$	$\frac{141}{1436} = 0.098$
$P(Y = 0 \mid X = x_i)$	$\frac{494}{564} \approx 0.876$	$\frac{1295}{1436} = 0.902$

Decision Tree algorithm: entropy

- Therefore the conditional entropy for selecting `education` as decision variable with the split at x_1 : `tertiary` and x_2: `secondary,primary,unknown` is

$$D_{subscribed|poutcome}$$

$$= -\sum_{i=1}^{2}\left\{P(X = x_i)\sum_{j=1}^{2} P(Y = y_j|X = x_i)log_2[P(Y = y_j|X = x_i)]\right\}$$

$$= -\{0.282 \times [0.124log_2(0.124) + 0.876log_2(0.876)]$$

$$+0.718 \times [0.098log_2(0.098) + 0.902log_2(0.902)]\}$$

$$\approx 0.485$$

- Therefore, there is a reduction of about $0.4862 - 0.485 \approx 0.0012$ from the base entropy.
- This information gain is far less than selecting `poutcome` as decision variable splitting at x_1 : `failure,other,unknown` and x_2: `success`

Decision Tree algorithm: entropy

- Therefore, the decision tree algorithm proceeds at the root node by calculating the conditional entropy for (i) each feature variable X and (ii) its different split points
- Then, the decision variable and its split points are selected based on the largest information gain (or decrease from base entropy)

Decision Tree algorithm: entropy

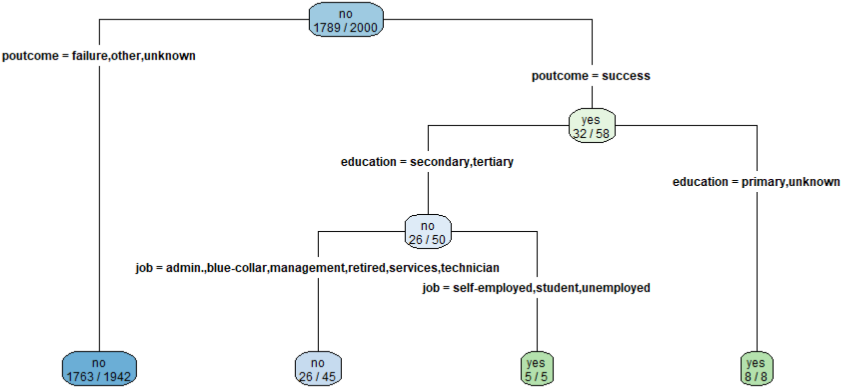
- At internal nodes, the decision tree algorithm proceeds similarly by calculating the conditional entropy for (i) each feature variable X and (ii) its different split points.
- However, the sample for calculating the base and conditional entropies is restricted to the one at the node.
- The tree is built recursively until a criteria is met, for example

(i) All the leaf nodes in the tree satisfy the minimum purity threshold.

(ii) The tree cannot be further split with the preset minimum purity threshold.

(iii) Any other stopping criterion is satisfied (such as the maximum depth of the tree).

Decision Trees



Decision Tree algorithm: Gini index for root node

- Another commonly used criteria for selecting decision variable and split points is the Gini index
- Given variable Y and and the set of possible categorical values it can take, $(y_1, y_2, ..., y_K)$, the Gini index of Y is defined as

$$G_Y = \sum_{j=1}^{K} P(Y = y_j)[1 - P(Y = y_j)],$$

where $P(Y = y_j)$ denotes the purity or the probability of the class $Y = y_j$, and $\sum_{j=1}^{K} P(Y = y_j) = 1$.

Decision Tree in R: Revisiting the wine recognition example

- Wines were grown in the same region in Italy but derived from 3 different cultivars.
- The task is to predict wine origin based on 13 attributes having continuous values

Decision Tree in R: Revisiting the wine recognition example

- The 13 features (X) of the dataset are:

1. Alcohol
2. Malic acid
3. Ash
4. Alkalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavonoids phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

Decision Tree in R: Revisiting the wine recognition example

- The data set is available from `https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data`
- The CSV file wine.csv has also been posted to IVLE
- Task is to predict label Y of origin: 1,2 or 3
-

```
wine_df <- read.csv("wine.csv", header = TRUE)
```

Decision Tree in R: Revisiting the wine recognition example

- ```
> head(wine_df)
  Wine Alcohol Malic.acid Ash Acl Mg Phenols
1   1   14.23   1.71 2.43 15.6 127   2.80
2   1   13.20   1.78 2.14 11.2 100   2.65
3   1   13.16   2.36 2.67 18.6 101   2.80
4   1   14.37   1.95 2.50 16.8 113   3.85
5   1   13.24   2.59 2.87 21.0 118   2.80
6   1   14.20   1.76 2.45 15.2 112   3.27
  Flavanoids Nonflavanoid.phenols Proanth Color.
     int Hue
1   3.06   0.28   2.29
    5.64 1.04
2   2.76   0.26   1.28
    4.38 1.05
3   3.24   0.30   2.81
    5.68 1.03
4   3.49   0.24   2.18
    7.80 0.86
5   2.69   0.39   1.82
    4.32 1.04
6   3.39   0.34   1.97
```
```

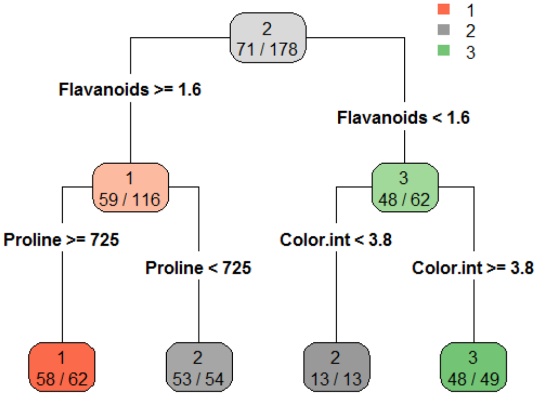
Decision Tree in R: Revisiting the wine recognition example

- We will build a classification tree for wine origin using the entropy criteria, with a maximum depth of 4.
-

```
wine_df <- read.csv("wine.csv", header = TRUE)
fit <- rpart(Wine ~.,
method="class",
data=wine_df,
control=rpart.control(maxdepth=4),
parms=list(split='information'))

rpart.plot(fit, type=4, extra=2, clip.right.labs=
    FALSE, varlen=0, faclen=0)
```

Decision Tree in R: Revisiting the wine recognition example



Decision Tree in R: Revisiting the wine recognition example

- We can build another classification tree for wine origin using the Gini index criteria, with a maximum depth of 4.
-

```
wine_df <- read.csv("wine.csv", header = TRUE)
fit <- rpart(Wine ~.,
method="class",
data=wine_df,
control=rpart.control(maxdepth=4),
parms=list(split='gini'))

rpart.plot(fit, type=4, extra=2, clip.right.labs=
    FALSE, varlen=0, faclen=0)
```

Decision Tree in R: Revisiting the wine recognition example