Supervised Learning Methods: Decision Trees for Classification

1. Introduction
   - Overview on Decision Trees
   - An Example
2. Decision Tree Algorithm
   - Choosing the Nodes
3. Example: Playing Golf?

Decision Trees (DT)

- Decision tree is a classification method.

- It has two varieties: **classification tree** and **regression tree**. We focus on the first one.

What is Decision Tree?

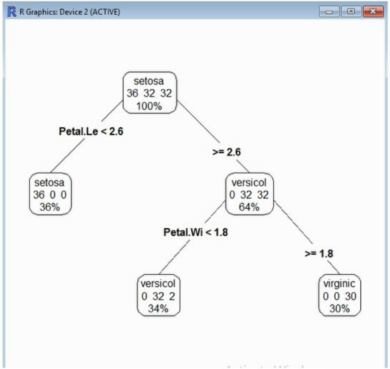
- A decision tree (also called prediction tree) uses a tree structure to specify sequences of decisions and consequences.

- Given a set of features $X = (x_1, x_2, ..., x_p)$, here, **each $\boldsymbol{x_i}$ is denoted for a feature**, the goal is to predict a response or output variable Y (categorical).

- Each member of the set $(x_1, x_2, ..., x_p)$ is called an input variable or a feature which could be categorical or continuous.

Decision Tree

Classification in R



Example of a decision tree

General idea of DT

- Prediction can be achieved by constructing a decision tree with test points and branches.

- Due to its flexibility and easy visualization, decision trees are commonly deployed in data mining applications for classification purposes.

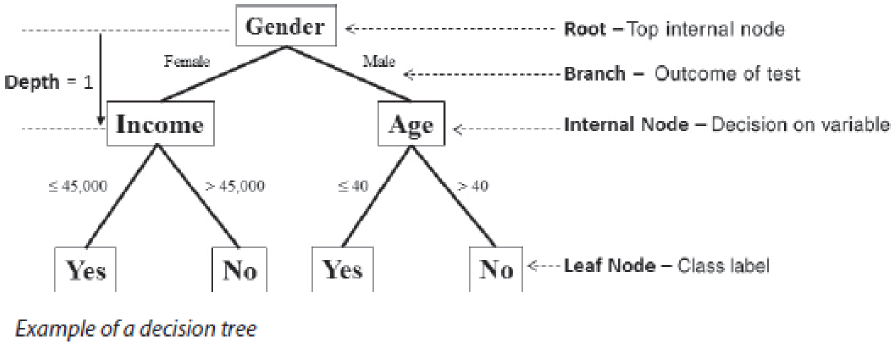
Some Components of a DT

- A decision tree employs a structure of test points, called **nodes**, and **branches**—which represent the decision being made.

- A node without further branches is called a **leaf node**.

- The leaf nodes return class labels (response) and, in some implementations, they return the probability scores.

An Example



Example of a decision tree. Source: *Data Science & Big Data Analytics*

Some Notes on the Trees

- 'Branch' refers to the outcome of a decision and is visualized as a line connecting two nodes.
- If a decision is numerical, the "greater than" branch is usually placed on the right, and the "less than" branch is placed on the left.
- Depending on the nature of the variable, one of the branches may need to include an "equal to" component.

Some Notes on the Trees

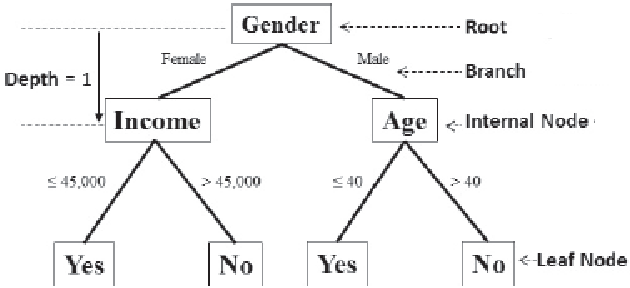
- Sometimes decision trees may have more than two branches stemming from a node.

- For example, suppose an input variable `Weather` is categorical and has three choices: Sunny, Rainy, and Snowy.

- Then the corresponding node `Weather` in the decision tree may have three branches labelled as Sunny, Rainy, and Snowy, respectively.

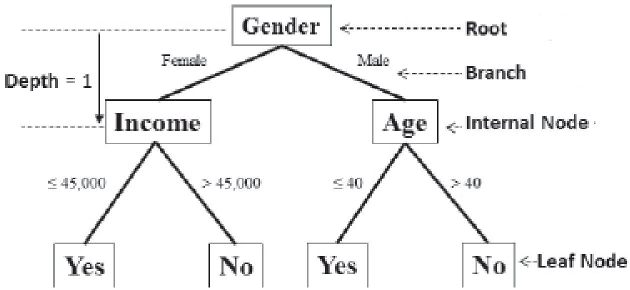
Example of a Decision Tree

- **Internal nodes** are the decision or test points.
- Each internal node refers to an input variable or an attribute.
- The top internal node is called the **root**.
- The decision tree on the right is a binary tree in that each internal node has no more than two branches.
- The branching of a node is referred to as a **split**.



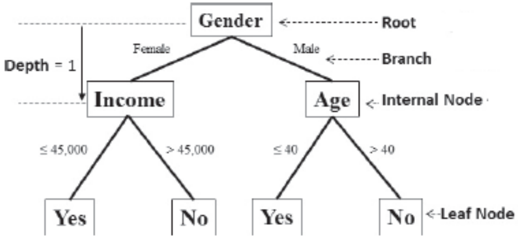
Example of a Decision Tree

- The **depth of a node** is the minimum number of steps required to reach the node from the root.
- In the decision tree on the right, nodes `Income` and `Age` have a depth of one, and the four nodes on the bottom of the tree have a depth of two.



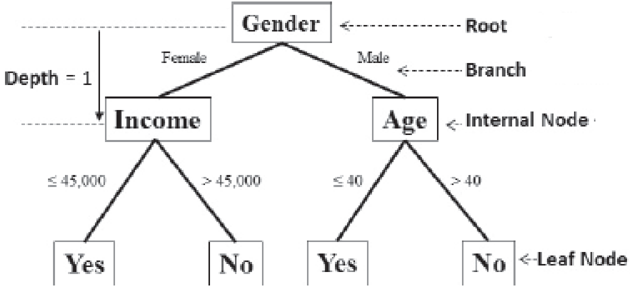
Example of a Decision Tree

- The root node splits into two branches with a `Gender` test. The right branch contains all records with `Gender = Male`, and the left branch contains all those records with `Gender = Female`, to create the depth 1 internal nodes.
- Each internal node effectively acts as the root of a sub-tree.



Example of a Decision Tree

- The left-hand side (LHS) internal node splits on a question based on the `Income` to create leaf nodes at depth 2, whereas the RHS splits on a question on the `Age`.
- This DT shows that **females with Income** $\leq \$45,000$ **and males** ≤ 40 **years old are classified as people who would purchase the product**.
- In traversing this tree, age does not matter for females; and income does not matter for males.



Applications of DT

- To classify animals, questions like cold-blooded or warm-blooded, mammal or not mammal, etc. are answered to arrive at a certain classification.

- Build a checklist of symptoms during medical evaluation of a patient.

- The artificial intelligence (AI) engine of a video game commonly uses decision trees to control the autonomous actions of a character in response to various scenarios.

Examples of DT

- Retailers can use decision trees to segment customers or predict response rates to marketing and promotions.

- Financial institutions can use decision trees to help decide if a loan application should be approved or denied. In the case of loan approval, computers can use the logical if-then statements to predict whether the customer will default on the loan.

- For customers with a clear (strong) outcome, no human interaction is required; for observations that may not generate a clear response, a human is needed for the decision.

Some Questions

- Question 1: Why `Gender` was selected as the root? Why not `Age` or `Income` be selected?

- Question 2: How do we implement/run DT in R when a data set is given?

Who would subscribe to a term deposit?

- Our first example of decision trees in R 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.

- The data set `bank-sample.csv` contains records of 2000 customers.

'bank-sample.csv' Data Set

- The variables include (1) job, (2) marital status, (3) education level, (4) if the credit is in default, (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.

- 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.

'bank-sample.csv' Data Set

```
> bankdata = read.csv("C:/Data/bank-sample.csv", header = TRUE)
> head(bankdata[,2:8])
           job  marital education default balance housing loan
1   management   single  tertiary      no       0     yes   no
2 entrepreneur  married  tertiary      no    1752     yes  yes
3     services divorced secondary      no    4329      no   no
4   management  married  tertiary      no    1108     yes   no
5   management  married secondary      no    1410     yes   no
6   management   single  tertiary      no     499     yes   no
```

‘bank-sample.csv’ Data Set

```
> head(bankdata[,c(9,16,17)])
   contact poutcome subscribed
1 cellular  unknown         no
2 cellular  unknown         no
3 cellular  unknown        yes
4 cellular  unknown         no
5  unknown  unknown         no
6  unknown  unknown         no
```

Some Features

```
> table(bankdata$job)

       admin.   blue-collar  entrepreneur     housemaid    management
          235           435            70            63           423
      retired self-employed      services       student    technician
           92            69           168            36           339
   unemployed       unknown
           60            10
> table(bankdata$marital)

divorced  married   single
     228     1201      571
>
```

Some Features

```
> table(bankdata$education)

  primary secondary  tertiary   unknown 
      335      1010       564        91 
> table(bankdata$default)

  no  yes 
1961   39 
```

```
> table(bankdata$housing)

  no  yes
 916 1084

> table(bankdata$loan)

  no  yes
1717  283

> table(bankdata$contact)

 cellular telephone   unknown
     1287       136       577

> table(bankdata$poutcome)

failure   other success unknown
    210      79      58    1653
```

Building The Decision Tree

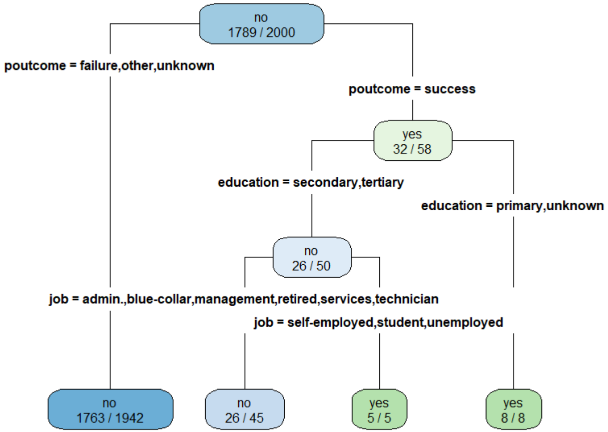
- We will build a decision tree to predict the response subscribed based on the features: job, marital, education, default, housing, loan, contact and poutcome.

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

To Visualize the tree

```
> library("rpart.plot")
> # To plot the fitted tree:
> rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE)#, faclen=0)
```

The Output Tree



Questions

- *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?

The Root Node

- The first step after identifying the modal category of the response is to choose the **most informative attribute**.

- A common way to identify the most informative attribute is to use entropy-based methods.

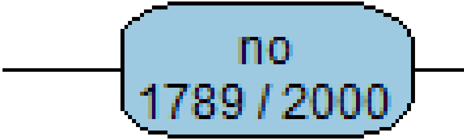
- The entropy methods select the most informative attribute based on two measurements:

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

(ii) *Information gain*, which measures the reduction in impurity (if a split is made)

The Purity

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



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 $\sum_{j=1}^{K} P(Y = y_j) = 1.$

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, $Y = 1$ for head; $Y = 0$ for tail.
- If the coin is a fair one, then $P(Y=0) = P(Y=1) = \frac{1}{2}$, then the entropy is

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

- If the coin is biased, suppose $P(Y=0) = \frac{3}{4}$, $P(Y=1) = \frac{1}{4}$, the entropy is now

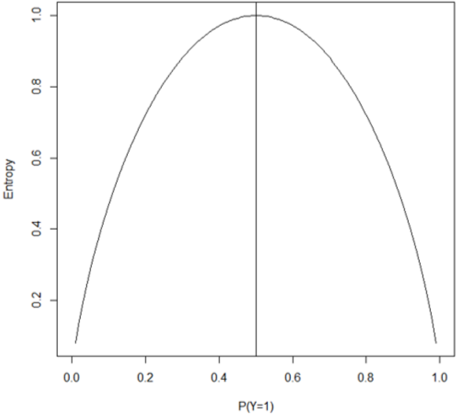
$$-\{0.25\log_2 0.25 + 0.75\log_2 0.75\} \approx 0.81.$$

Entropy

- 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.
- For a binary variable Y (0, 1), the entropy is largest when $P(Y = 1) = P(Y = 0) = 0.5$.

Entropy Plot

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



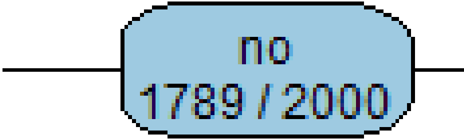
A good supplement

- A very simple example yet the general idea of Entropy is well explained.

https://www.youtube.com/watch?v=ZVR2Way4nwQ&t=11s

Bank Sample: Base Entropy

- The base entropy is defined as the entropy of the **output variable**.
- Recall: $P(\text{subscribed} = 0) = \dfrac{1789}{2000} \approx 89.45\%$ and $P(\text{subscribed} = 1) = 1 - \dfrac{1789}{2000} \approx 10.55\%$
- Let D denote for entropy, the base entropy is then $D_{\text{subscribed}} = -\{0.1055 log_2(0.1055) + 0.8945 log_2(0.8945)\} \approx 0.4862$.



Conditional Entropy

- Ideally, we would like to reduce the base entropy by leveraging on feature variables for prediction.

- Recall that lower entropy is associated with less "uncertainty" in predicting the outcome, which is something that we want.

- So, among many features, we want to select **the one that reduces the base entropy the most**.

Conditional Entropy

- Consider binary tree algorithm. Suppose a feature X has split values (x_1, x_2). The conditional entropy given feature X and the split points (x_1, x_2) is defined as

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

- We will illustrate the calculation of conditional entropy for the decision variable in the root node, `poutcome`.

Conditional Entropy

Assume that the split categories are $x_1 =$ `failure,other,unknown`; and $x_2 =$ `success`.

```
> x1=which(bankdata$poutcome!="success")
> # index of the rows where poutcome = x1
>
> length(x1) # 1942 rows that the value of poutcome = x1.
[1] 1942
> x2=which(bankdata$poutcome=="success")
> # index of the rows where poutcome = x2
>
> length(x2) # 58 rows that the value of poutcome = x2 = success
[1] 58
>
```

Conditional Entropy

- Probabilities of two categories of `poutcome`:

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

- However, what we need for calculating conditional entropy when `poutcome` is involved are $P(\text{Subscribed} = 1|\text{poutcome})$ and $P(\text{Subscribed} = 0|\text{poutcome})$.

Conditional Entropy

```
> table(bankdata$subscribed[x1])

  no  yes
1763  179

> # among 1942 customers with poutcome = x1, 179 subscribed (179 yes), and 176
> #
> table(bankdata$subscribed[x2])

 no yes
 26  32

> # among 58 customers with poutcome = x2, 32 subscribed (32 yes), and 26 no.
>
```

Conditional Entropy

- Conditional probabilities:

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

Conditional Entropy

- Therefore the conditional entropy for selecting `poutcome` as decision variable with the split at x_1 and x_2 is $D_{subscribed|poutcome}$, equal to

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

- Hence, 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**.

Entropy Reduction

- We can calculate the reduction for other 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 calculate the conditional entropy for splitting at the values x_1 : `other`, `success`, `unknown` and x_2: **failure**.
- We shall show that: this split is not chosen in the decision tree built earlier, because the amount of entropy reduction from it is less than 0.027.

Different split for poutcome

- Split `poutcome` at $x_1 =$ `other,success,unknown` and $x_2 =$ **failure**.

- Out of total 2000 customers, 1790 are x_1 and 210 are x_2 (failure).
- Probabilities of two categories:

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

Different split for poutcome

- Out of 1790 customers with poutcome $= x_1$, 190 have Subscribed = yes and 1600 no.
- Out of 210 customers with poutcome $= x_2$, 21 yes and 189 no.
- Hence, conditional probabilities are

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

Different split for poutcome

- The conditional entropy for selecting poutcome as decision variable with the split at $x_1 =$ **success, other, unknown** and $x_2 =$ **failure** is

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

$$= -\{0.895 \times [0.106log_2(0.106) + 0.894log_2(0.894)]$$

$$+0.105 \times [0.10log_2(0.10) + 0.90log_2(0.90)]\}$$

$$\approx 0.486$$

- There is a reduction of about $(0.4862 - 0.486) \approx 0.0002$ from the base entropy.

Why poutcome? Why not Education?

- Instead of the feature variable poutcome, let us calculate the entropy reduction if we choose education.
- Consider the split points for this variable with $x_1 =$ tertiary and $x_2 =$ secondary,primary,unknown.
- You also may try with other possible splits for education.
- We shall show that education gives a smaller reduction from the base entropy than poutcome.

If Education...

```
> table(bankdata$education)
  primary secondary  tertiary   unknown
      335      1010       564        91
```

-

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

If Education...

```
> table(bankdata$subscribed[x1])
  no  yes
1763  179
> table(bankdata$subscribed[x2])
 no yes
 26  32
```

-

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

If Education, then Information Gain is

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

$$
\begin{aligned}
&= -\sum_{i=1}^{2}\left\{P(X = x_i)\sum_{j=1}^{2}P(Y = y_j|X = x_i)log_2[P(Y = y_j|X = x_i)]\right\} \\
&= -\{0.282 \times [0.124log_2(0.124) + 0.876log_2(0.876)] \\
&+0.718 \times [0.098log_2(0.098) + 0.902log_2(0.902)]\} \\
&\approx 0.485
\end{aligned}
$$

- Therefore, there is a reduction of about $(0.4862 - 0.485) \approx 0.0012$ from the base entropy.

Conclusion

- 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 largest reduction from base 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.

Conclusion

- 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).

Gini Index

- Beside Information Gain, 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 $\sum_{j=1}^{K} P(Y = y_j) = 1.$

Example: Playing Golf?

- The goal of this illustrative example is to predict whether to play golf given factors such as weather outlook, temperature, humidity, and wind.
- Data set is `DTdata.csv` which contains five attributes: Play, Outlook, Temperature, Humidity, and Wind.
- Play would be the output variable (or the predicted class), and Outlook, Temperature, Humidity, and Wind would be the input variables.

Source: *The Straits Times*

Data Set

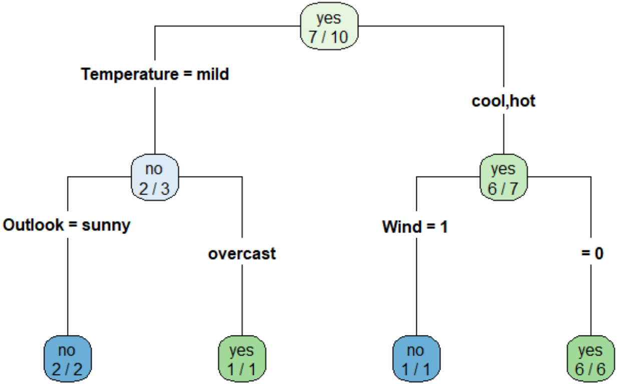
```
> library("rpart") # load libraries
> library("rpart.plot")
> play_decision <- read.table("C:/Data/DTdata.csv",header=TRUE,sep=",")
> head(play_decision)
  Play  Outlook Temperature Humidity  Wind
1  yes    rainy        cool   normal FALSE
2   no    rainy        cool   normal  TRUE
3  yes overcast         hot     high FALSE
4   no    sunny        mild     high FALSE
5  yes    rainy        cool   normal FALSE
6  yes    sunny        cool   normal FALSE
```

Aim

- We will build a *decision tree* to predict golf play based on feature variables such as weather outlook, temperature, humidity, and wind, using entropy reduction (or information gain) to determine the split variables.

```
> fit <- rpart(Play ~ Outlook + Temperature + Humidity + Wind,
+ method="class",
+ data=play_decision,
+ control=rpart.control(minsplit=1),
+ parms=list(split='information'))
> rpart.plot(fit, type=4, extra=2)
```

Output: The fitted decision tree



Prediction

```
> newdata <- data.frame(Outlook="rainy", Temperature="mild",
+ Humidity="high", Wind=FALSE)
> newdata
  Outlook Temperature Humidity  Wind
1   rainy        mild     high FALSE
```

- The decision tree can be used to predict outcomes for new data sets.
- Consider a testing set that contains the following record: Outlook='rainy', Temperature='mild', Humidity='high', Wind=FALSE.
- The goal is to predict the play decision of this record. The following code loads the data into R as a data frame `newdata`.

Prediction

```
> predict(fit,newdata=newdata,type="prob")
  no yes
1  1   0
> predict(fit,newdata=newdata,type="class")
 1
no
Levels: no yes
```

- High probability for `Play` to fall into category ‘no’ given the condition as in `newdata`.
- If to classify the decision, then the prediction should be in category ‘no’.