CART

Gini Index

- Many alternative measures to Information Gain
- Most popular altermative: Gini index
  - used in e.g., in CART (Classification And Regression Trees)
  - impurity measure (instead of entropy)

$$Gini(S) = 1 - \sum_i p_i^2$$

  - average Gini index (instead of average entropy / information)

$$Gini(S, A) = \sum_i \frac{|S_i|}{|S|} \cdot Gini(S_i)$$

  - Gini Gain
    - could be defined analogously to information gain
    - but typically avg. Gini index is minimized instead of maximizing Gini gain

Dataset

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Gini index

Gini index is a metric for classification tasks in CART. It stores sum of squared probabilities of each class. We can formulate it as illustrated below.

$$\text{Gini} = 1 - \Sigma\, (P_i)^2 \text{ for } i=1 \text{ to number of classes}$$

Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. I will summarize the final decisions for outlook feature.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

$\text{Gini}(\text{Outlook=Sunny}) = 1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$

$\text{Gini}(\text{Outlook=Overcast}) = 1 - (4/4)^2 - (0/4)^2 = 0$

$\text{Gini}(\text{Outlook=Rain}) = 1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$

Then, we will calculate weighted sum of gini indexes for outlook feature.

$\text{Gini}(\text{Outlook}) = (5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

Gini(Temp=Hot) = $1 - (2/4)^2 - (2/4)^2 = 0.5$

Gini(Temp=Cool) = $1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$

Gini(Temp=Mild) = $1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$

We'll calculate weighted sum of gini index for temperature feature

Gini(Temp) = $(4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.445 = 0.142 + 0.107 + 0.190 = 0.439$

Humidity

Humidity is a binary class feature. It can be high or normal.

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

Gini(Humidity=High) = $1 - (3/7)^2 - (4/7)^2 = 1 - 0.183 - 0.326 = 0.489$

Gini(Humidity=Normal) = $1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$

Weighted sum for humidity feature will be calculated next

Gini(Humidity) = $(7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$

Wind

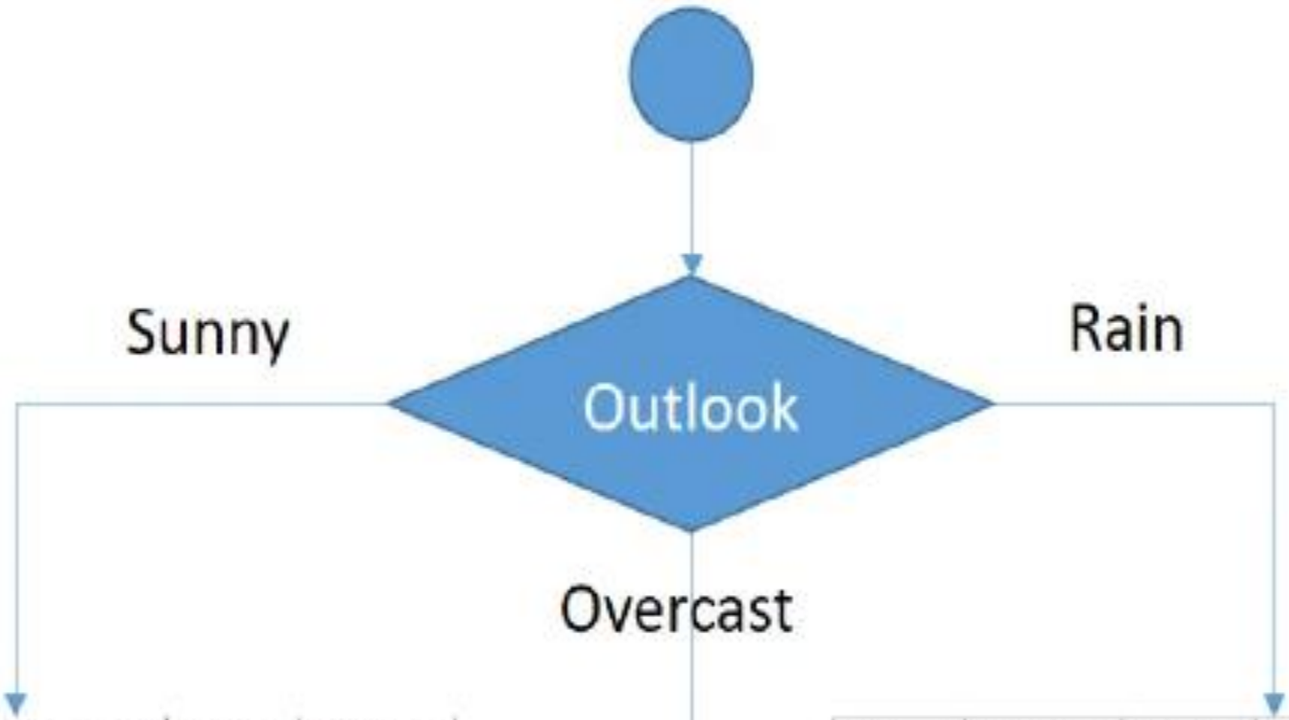
Wind is a binary class similar to humidity. It can be weak and strong.

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

Gini(Wind=Weak) = $1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$

Gini(Wind=Strong) = $1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$

Gini(Wind) = $(8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

We will apply same principles to those sub datasets in the following steps.

Focus on the sub dataset for sunny outlook. We need to find the gini index scores for temperature, humidity and wind features respectively.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Gini of temperature for sunny outlook

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini(Outlook=Sunny and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini(Outlook=Sunny and Temp.=Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$

Gini(Outlook=Sunny and Temp.=Mild) = $1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$

Gini(Outlook=Sunny and Temp.) = $(2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2$

Gini of humidity for sunny outlook

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini(Outlook=Sunny and Humidity=High) = $1 - (0/3)^2 - (3/3)^2 = 0$

Gini(Outlook=Sunny and Humidity=Normal) = $1 - (2/2)^2 - (0/2)^2 = 0$

Gini(Outlook=Sunny and Humidity) = (3/5)x0 + (2/5)x0 = 0

Gini of wind for sunny outlook

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

Gini(Outlook=Sunny and Wind=Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.266$

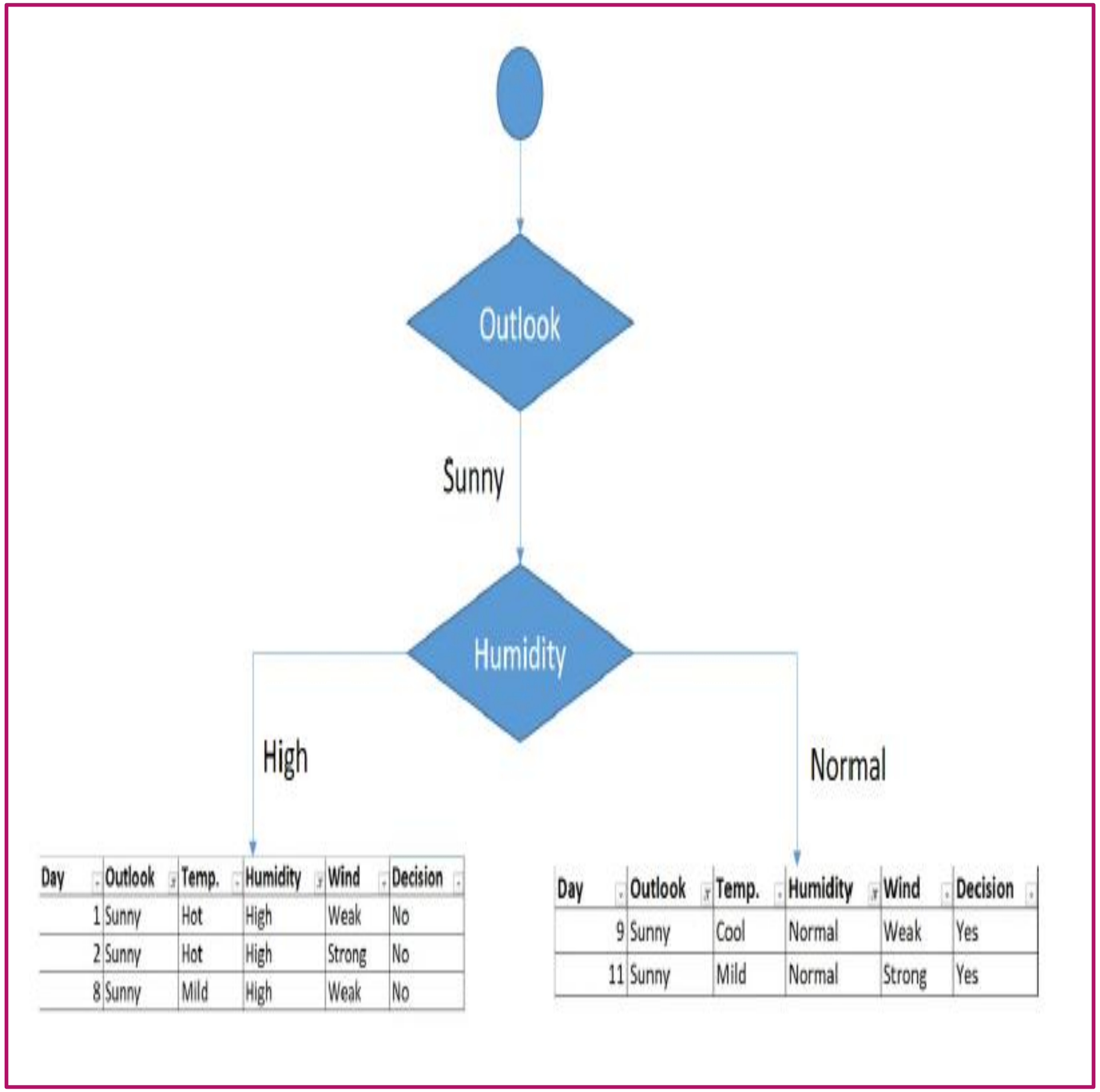
Gini(Outlook=Sunny and Wind=Strong) = $1- (1/2)^2 - (1/2)^2 = 0.2$

Gini(Outlook=Sunny and Wind) = (3/5)x0.266 + (2/5)x0.2 = 0.466

Decision for sunny outlook

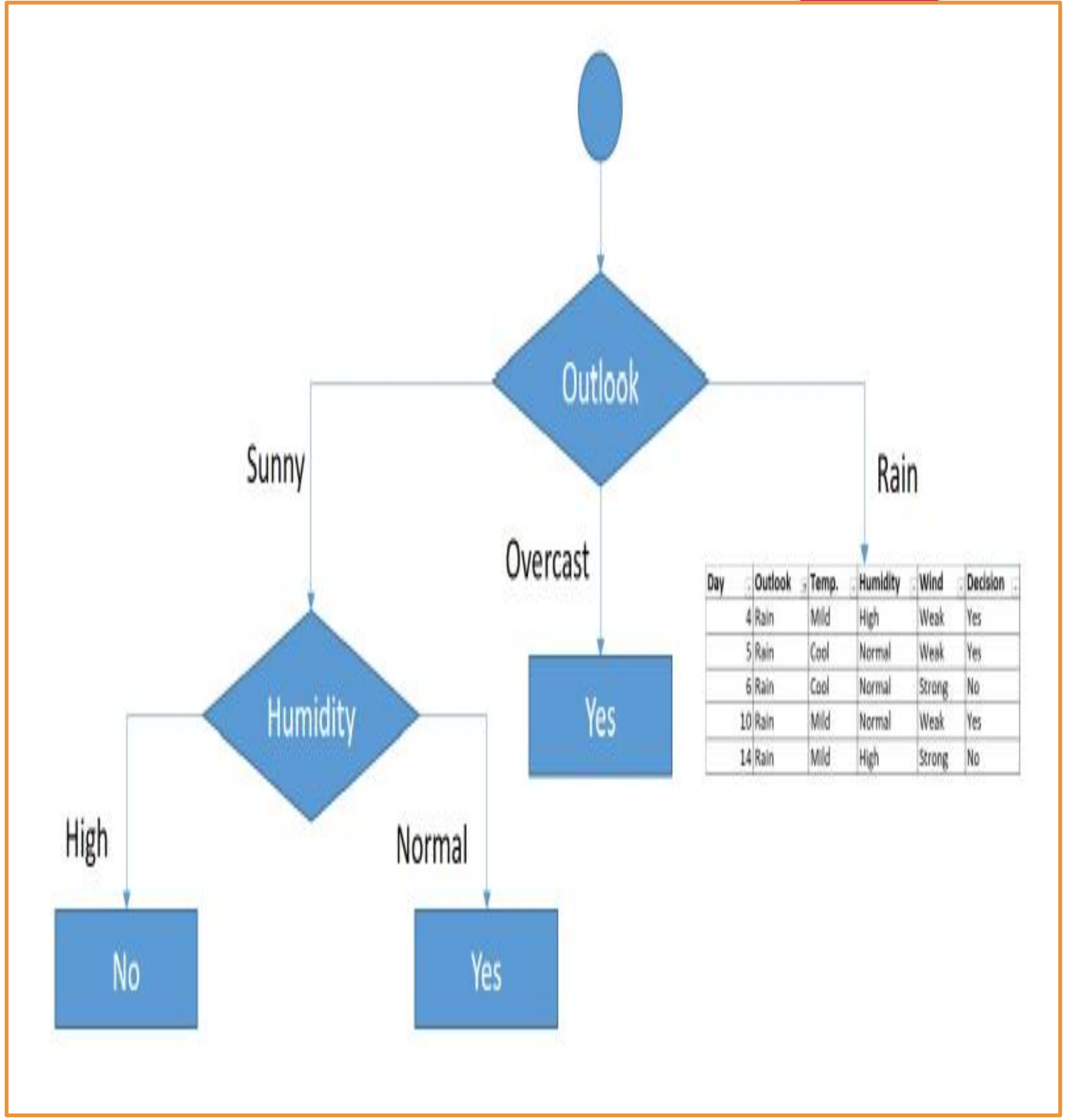
We've calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Rain outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Gini of temprature for rain outlook

Temperature	Yes	No	Number of instances
Cool	1	1	2
Mild	2	1	3

Gini(Outlook=Rain and Temp.=Cool) = $1 – (1/2)^2 – (1/2)^2 = 0.5$

Gini(Outlook=Rain and Temp.=Mild) = $1 – (2/3)^2 – (1/3)^2 = 0.444$

Gini(Outlook=Rain and Temp.) = (2/5)x0.5 + (3/5)x0.444 = 0.466

Gini of humidity for rain outlook

Humidity	Yes	No	Number of instances
High	1	1	2
Normal	2	1	3

Gini(Outlook=Rain and Humidity=High) = $1 - (1/2)^2 - (1/2)^2 = 0.5$

Gini(Outlook=Rain and Humidity=Normal) = $1 - (2/3)^2 - (1/3)^2 = 0.444$

Gini(Outlook=Rain and Humidity) = $(2/5)\text{x}0.5 + (3/5)\text{x}0.444 = 0.466$

Gini of wind for rain outlook

Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

Gini(Outlook=Rain and Wind=Weak) = $1 - (3/3)^2 - (0/3)^2 = 0$

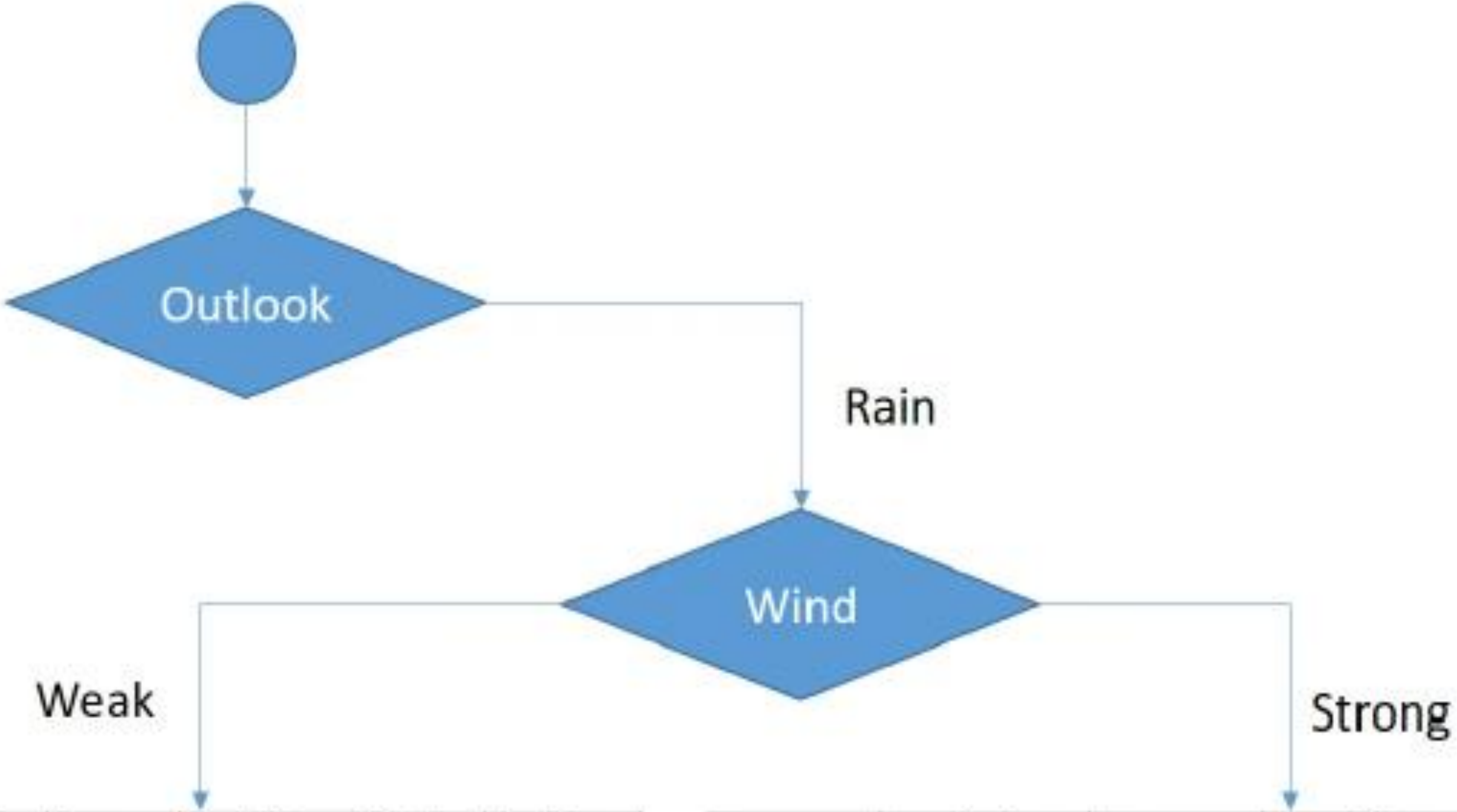
Gini(Outlook=Rain and Wind=Strong) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini(Outlook=Rain and Wind) = $(3/5)\text{x}0 + (2/5)\text{x}0 = 0$

Decision for rain outlook

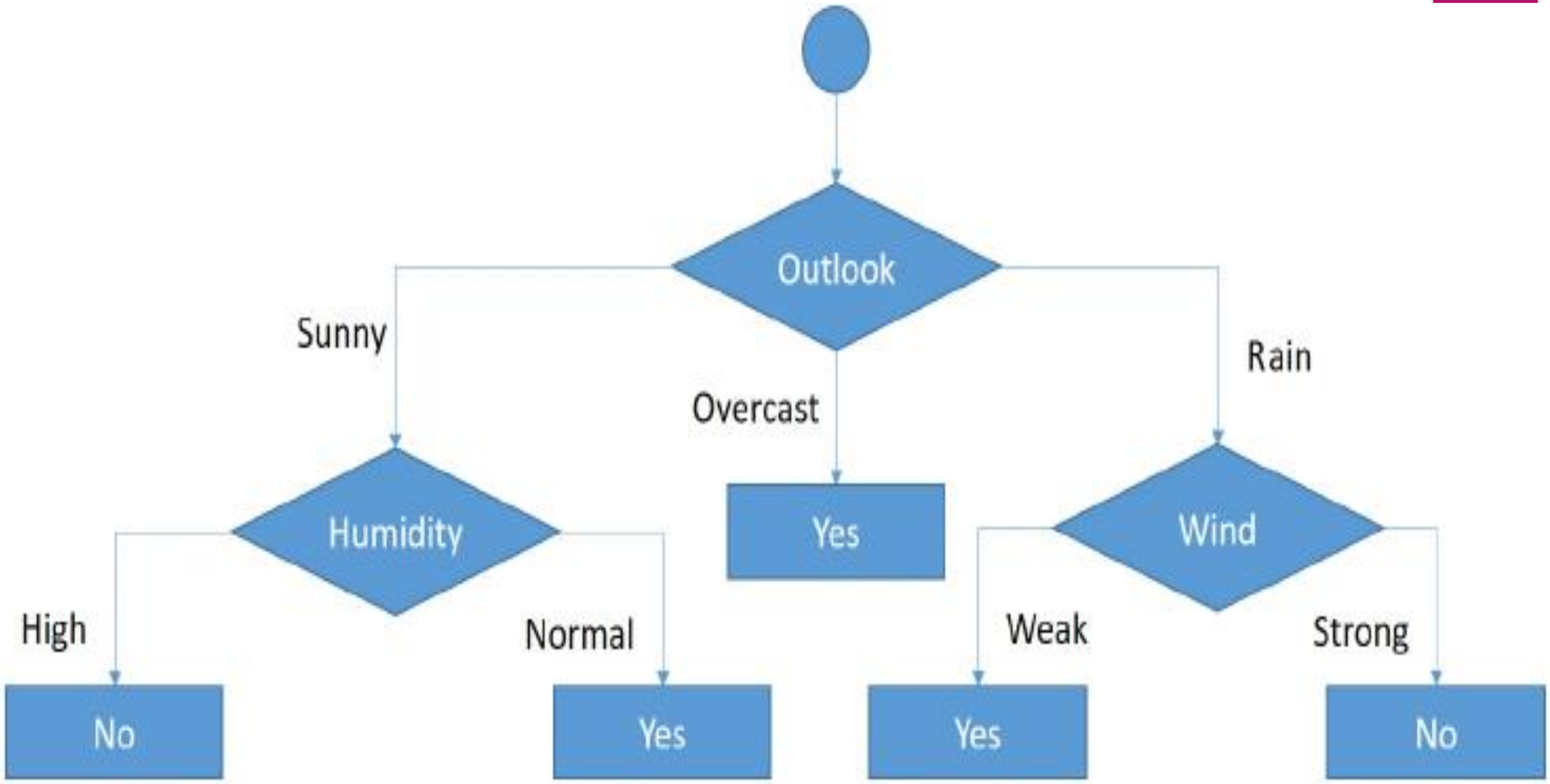
The winner is wind feature for rain outlook because it has the minimum gini index score in features.

Feature	Gini index
Temperature	0.466
Humidity	0.466
Wind	0



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Weak
Strong
No
Yes
Yes
No

Pros & Cons

Advantages :

- No preprocessing needed on data.
- No assumptions on distribution of data.
- Handles co linearity efficiently.
- Decision trees can provide understandable explanation over the prediction.

Disadvantages :

- Chances for overfitting the model if we keep on building the tree to achieve high purity. decision tree pruning can be used to solve this issue.
- Prone to outliers.
- Tree may grow to be very complex while training complicated datasets.
- Looses valuable information while handling continuous variables.

Decision tree vs naive Bayes :

- Decision tree is a discriminative model, whereas Naive bayes is a generative model.
- Decision trees are more flexible and easy.
- Decision tree pruning may neglect some key values in training data, which can lead the accuracy.
- A major advantage to Naive Bayes classifiers is that they are not prone to overfitting, thanks to the fact that they “ignore” irrelevant features.
- Naive Bayes classifiers are easily implemented and highly scalable, with a linear computational complexity with respect to the number of data entries.

Decision Theory- Naïve Bayes

Supervised Learning

Naïve Bayesian Classifier

According to Bayes' theorem, the probability that we want to compute $P(H|\mathbf{X})$ can be expressed in terms of probabilities $P(H)$, $P(\mathbf{X}|H)$, and $P(\mathbf{X})$ as

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)\ P(H)}{P(\mathbf{X})},$$

and these probabilities may be estimated from the given data.

Naive Bayesian Classifier

The naive Bayesian classifier works as follows:

Let T be a training set of samples, each with their class labels. There are k classes, $C_1, C_2, \ldots, C_k$. Each sample is represented by an n-dimensional vector, $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$, depicting n measured values of the n attributes, $A_1, A_2, \ldots, A_n$, respectively.

That is $\mathbf{X}$ is predicted to belong to the class C_i if and only if

$$P(C_i|\mathbf{X}) > P(C_j|\mathbf{X}) \qquad \text{for } 1 \leq j \leq m,\ j \neq i.$$

Thus we find the class that maximizes $P(C_i|\mathbf{X})$. The class C_i for which $P(C_i|\mathbf{X})$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)\ P(C_i)}{P(\mathbf{X})}.$$

As $P(\mathbf{X})$ is the same for all classes, only $P(\mathbf{X}|C_i)P(C_i)$ need be maximized. If the class a priori probabilities, $P(C_i)$, are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_1) = P(C_2) = \ldots = P(C_k)$, and we would therefore maximize $P(\mathbf{X}|C_i)$. Otherwise we maximize $P(\mathbf{X}|C_i)P(C_i)$. Note that the class a priori probabilities may be estimated by $P(C_i) = \mathrm{freq}(C_i, T)/|T|$.

$$P(\mathbf{X}|C_i) \approx \prod_{k=1}^{n} P(x_k|C_i).$$

The probabilities $P(x_1|C_i), P(x_2|C_i), \ldots, P(x_n|C_i)$ can easily be estimated from the training set. Recall that here x_k refers to the value of attribute A_k for sample $\mathbf{X}$.

(a) If A_k is categorical, then $P(x_k|C_i)$ is the number of samples of class C_i in T having the value x_k for attribute A_k, divided by freq(C_i, T), the number of sample of class C_i in T.

(b) If A_k is continuous-valued, then we typically assume that the values have a Gaussian distribution with a mean μ and standard deviation σ defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp -\frac{(x-\mu)^2}{2\sigma^2},$$

$$p(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

We need to compute μ_{C_i} and σ_{C_i}, which are the mean and standard deviation of values of attribute A_k for training samples of class C_i.

In order to predict the class label of $\mathbf{X}$, $P(\mathbf{X}|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of $\mathbf{X}$ is C_i if and only if it is the class that maximizes $P(\mathbf{X}|C_i)P(C_i)$.

Problem Statement

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

X = (age = youth, income = medium, student = yes, credit = fair)   Class label=?

Solution

We need to maximize $P(\mathbf{X}|C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the a priori probability of each class, can be estimated based on the training samples:

$$P(buy = yes) = \frac{9}{14}$$

$$P(buy = no) = \frac{5}{14}$$

To compute $P(\mathbf{X}|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$P(age = youth|buy = yes) = \frac{2}{9}$$

$$P(age = youth|buy = no) = \frac{3}{5}$$

$$P(income = medium|buy = yes) = \frac{4}{9}$$

$$P(income = medium | buy = no) = \frac{2}{5}$$

$$P(student = yes | buy = yes) = \frac{6}{9}$$

$$P(student = yes | buy = no) = \frac{1}{5}$$

$$P(credit = fair | buy = yes) = \frac{6}{9}$$

$$P(credit = fair | buy = no) = \frac{2}{5}$$

Using the above probabilities, we obtain

$$\begin{aligned} P(\mathbf{X} | buy = yes) &= P(age = youth | buy = yes) \\ &\quad P(income = medium | buy = yes) \\ &\quad P(student = yes | buy = yes) \\ &\quad P(credit = fair | buy = yes) \\ &= \frac{2}{9}\frac{4}{9}\frac{6}{9}\frac{6}{9} = 0.044. \end{aligned}$$

$$P(\mathbf{X}|buy = no) = \frac{3}{5}\frac{2}{5}\frac{1}{5}\frac{2}{5} = 0.019$$

To find the class that maximizes $P(\mathbf{X}|C_i)P(C_i)$, we compute

$$P(\mathbf{X}|buy = yes)P(buy = yes) = 0.028$$

$$P(\mathbf{X}|buy = no)P(buy = no) = 0.007$$

Thus the naive Bayesian classifier predicts $buy = yes$ for sample $\mathbf{X}$.

It will be Continued….