What is a Decision Tree?

A Supervised Machine Learning Algorithm, used to build classification and regression models in the form of a tree structure.

A decision tree is a tree where each -

- Node - a feature(attribute)
- Branch - a decision(rule)
- Leaf - an outcome (categorical or continuous)

There are many algorithms to build decision trees, here we are going to discuss ID3 and CART algorithm with an example.

ID3 Algorithm

- ID3 stands for Iterative Dichotomiser 3
- It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain(IG) or minimum Entropy(H).

Entropy

- Entropy is a measure of the amount of uncertainty in the dataset S.
- **Mathematical Representation of Entropy is shown here –**

$$H(S) = \sum_{c \epsilon C} -p(c) \log_2 p(c)$$

Where,

- S - The current dataset for which entropy is being calculated (changes every iteration of the ID3 algorithm).
- C - Set of classes in S {example - C = {yes, no}}
- p(c) - The proportion of the number of elements in class c to the number of elements in set S.

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on that particular iteration.

Entropy = 0 implies it is of pure class, that means all are of same category.

Information Gain

- Information Gain IG(A) tells us how much uncertainty in S was reduced after splitting set S on attribute A.
- **Mathematical representation of Information gain is shown here –**

$$IG(A, S) = H(S) - \sum_{t \epsilon T} p(t) H(t)$$

Where,

- H(S) - Entropy of set S.

- T - The subsets created from splitting set S by attribute A
- p(t) - The proportion of the number of elements in t to the number of elements in set S.
- H(t) - Entropy of subset t.

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. **The attribute with the <u>largest information gain is used to split</u> the set S on that particular iteration.**

Steps in ID3 algorithm:

1. Calculate entropy for dataset.
2. For each attribute/feature
   a. Calculate entropy for all its categorical values.
   b. Calculate information gain for the feature.
   c. Find the feature with maximum information gain.
   d. Repeat it until we get the desired tree.

Q1. Use ID3 algorithm on a dataset given below and compute the decision tree.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
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

Solution:

Here, dataset is of binary classes (yes and no), where 9 out of 14 are "yes" and 5 out of 14 are "no".

Complete entropy of dataset is –

$H(S) = - p(yes) * \log_2(p(yes)) - p(no) * \log_2(p(no))$

$= - (9/14) * \log_2(9/14) - (5/14) * \log_2(5/14)$

$= - (-0.41) - (-0.53)$

$= 0.94$

For each attribute of the dataset, let's follow the step-2 of pseudocode: -

First Attribute – Outlook [Categorical values - sunny, overcast and rain]

$H(Outlook=sunny) = -(2/5)*\log_2(2/5)-(3/5)*\log_2(3/5) = 0.971$

$H(Outlook=rain) = -(3/5)*\log_2(3/5)-(2/5)*\log_2(2/5) = 0.971$

$H(Outlook=overcast) = -(4/4)*\log_2(4/4)-0 = 0$

Average Entropy Information for Outlook –

I(Outlook) = p(sunny) * H(Outlook=sunny) + p(rain) * H(Outlook=rain) + p(overcast) * H(Outlook=overcast)

= (5/14)*0.971 + (5/14)*0.971 + (4/14)*0

= 0.693

Information Gain = H(S) - I(Outlook)

= 0.94 - 0.693= 0.247

Now we do the same calculations for all the remaining attributes:

Second Attribute – Temperature [Categorical values - hot, mild, cool]

$H(Temperature=hot) = -(2/4)*\log_2(2/4)-(2/4)*\log_2(2/4) = 1$

$H(Temperature=cool) = -(3/4)*\log_2(3/4)-(1/4)*\log_2(1/4) = 0.811$

$H(Temperature=mild) = -(4/6)*\log_2(4/6)-(2/6)*\log_2(2/6) = 0.9179$

Average Entropy Information for Temperature –

I(Temperature) = p(hot)*H(Temperature=hot) + p(mild)*H(Temperature=mild) + p(cool)*H(Temperature=cool)

= (4/14)*1 + (6/14)*0.9179 + (4/14)*0.811

= 0.9108

Information Gain = H(S) - I(Temperature)

= 0.94 - 0.9108

= 0.0292

Third Attribute – Humidity [Categorical values - high, normal]

$H(Humidity=high) = -(3/7)*\log_2(3/7)-(4/7)*\log_2(4/7) = 0.983$

$H(Humidity=normal) = -(6/7)*\log_2(6/7)-(1/7)*\log_2(1/7) = 0.591$

Average Entropy Information for Humidity –

I(Humidity) = p(high)*H(Humidity=high) + p(normal)*H(Humidity=normal)

= (7/14)*0.983 + (7/14)*0.591

= 0.787

Information Gain = H(S) - I(Humidity)

= 0.94 - 0.787

= 0.153

Fourth Attribute – Wind [Categorical values - weak, strong]

$H(Wind=weak) = -(6/8)*\log_2(6/8)-(2/8)*\log_2(2/8) = 0.811$

$H(Wind=strong) = -(3/6)*\log_2(3/6)-(3/6)*\log_2(3/6) = 1$

Average Entropy Information for Wind –

$I(Wind) = p(weak)*H(Wind=weak) + p(strong)*H(Wind=strong)$

$= (8/14)*0.811 + (6/14)*1$

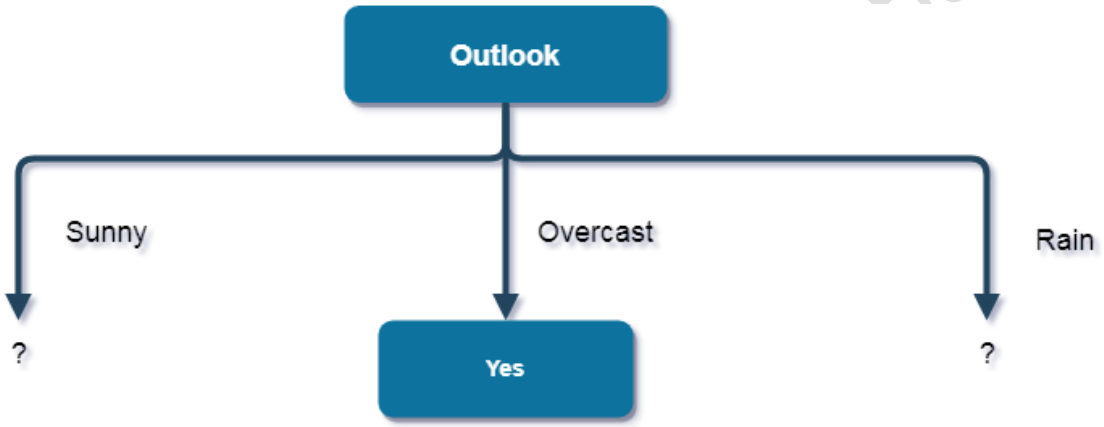
$= 0.892$

Information Gain = H(S) - I(Wind)

= 0.94 - 0.892

= 0.048

Here, the attribute with **maximum information gain is Outlook**. So, the decision tree built so far -



Here, when Outlook == overcast, it is of pure class(Yes).

Now, we have to repeat same procedure for the data with rows consist of Outlook value as Sunny and then for Outlook value as Rain.

Now, finding the best attribute for splitting the data with **Outlook=Sunny** values {Dataset rows = [1, 2, 8, 9, 11]}.

Complete entropy of Sunny is –

$H(Sunny) = - p(yes) * \log_2(p(yes)) - p(no) * \log_2(p(no))$

$= - (2/5) * \log_2(2/5) - (3/5) * \log_2(3/5)$

$= 0.971$

First Attribute – Temperature [Categorical values - hot, mild, cool]

$H(Sunny, Temperature=hot) = -0-(2/2)*\log_2(2/2) = 0$

$H(Sunny, Temperature=cool) = -(1)*\log_2(1)- 0 = 0$

$H(Sunny, Temperature=mild) = -(1/2)*\log_2(1/2)-(1/2)*\log_2(1/2) = 1$

Average Entropy Information for Temperature –

$I(Sunny, Temperature) = p(Sunny, hot)*H(Sunny, Temperature=hot) + p(Sunny,mild) *$

$H(Sunny, Temperature=mild) + p(Sunny, cool)*H(Sunny, Temperature=cool)$

= (2/5)*0 + (1/5)*0 + (2/5)*1

= 0.4

Information Gain = H(Sunny) - I(Sunny, Temperature)

= 0.971 - 0.4

= 0.571

Second Attribute – Humidity [Categorical values - high, normal]

H(Sunny, Humidity=high) = - 0 - (3/3)*$\log_2$(3/3) = 0

H(Sunny, Humidity=normal) = -(2/2)*$\log_2$(2/2)-0 = 0

Average Entropy Information for Humidity –

I(Sunny, Humidity) = p(Sunny, high)*H(Sunny, Humidity=high) + p(Sunny, normal) *

H(Sunny, Humidity=normal)

= (3/5)*0 + (2/5)*0

= 0

Information Gain = H(Sunny) - I(Sunny, Humidity)

= 0.971 – 0

= 0.971

Third Attribute – Wind [Categorical values - weak, strong]

H(Sunny, Wind=weak) = -(1/3)*$\log_2$(1/3)-(2/3)*$\log_2$(2/3) = 0.918

H(Sunny, Wind=strong) = -(1/2)*$\log_2$(1/2)-(1/2)*$\log_2$(1/2) = 1

Average Entropy Information for Wind –

I(Sunny, Wind) = p(Sunny, weak)*H(Sunny, Wind=weak) + p(Sunny, strong)*H(Sunny,

Wind=strong)

= (3/5)*0.918 + (2/5)*1

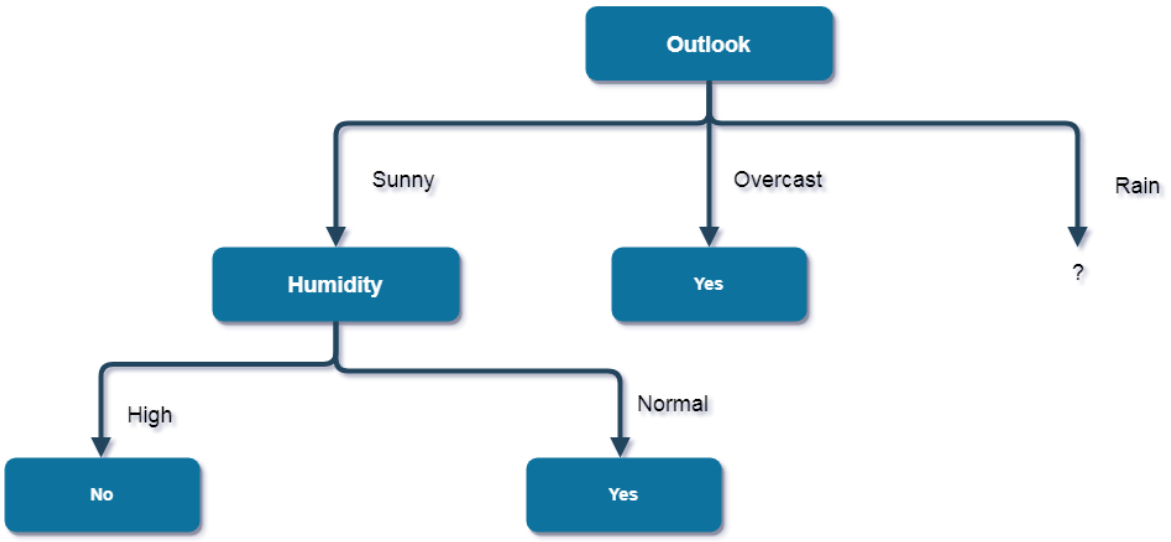
= 0.9508

Information Gain = H(Sunny) - I(Sunny, Wind)

= 0.971 - 0.9508

= 0.0202

Here, the attribute with **maximum information gain is Humidity**. So, the decision tree built so far –



Here, when Outlook = Sunny and Humidity = High, it is a pure class of category "no". And When Outlook = Sunny and Humidity = Normal, it is again a pure class of category "yes". Therefore, we don't need to do further calculations.

Now, finding the best attribute for splitting the data with **Outlook=Rain** values {Dataset rows = [4, 5, 6, 10, 14]}.
Complete entropy of Rain is –
$H(Rain) = - p(yes) * \log_2(p(yes)) - p(no) * \log_2(p(no))$
$= - (3/5) * \log_2(3/5) - (2/5) * \log_2(2/5)$
$= 0.971$

First Attribute – Temperature [Categorical values - mild, cool]
$H(Rain, Temperature=cool) = -(1/2)*\log_2(1/2)- (1/2)*\log_2(1/2) = 1$
$H(Rain, Temperature=mild) = -(2/3)*\log_2(2/3)-(1/3)*\log_2(1/3) = 0.918$
Average Entropy Information for Temperature –
$I(Rain, Temperature) = p(Rain, mild)*H(Rain, Temperature=mild) + p(Rain, cool)*H(Rain, Temperature=cool)$
$= (2/5)*1 + (3/5)*0.918$
$= 0.9508$
Information Gain = H(Rain) - I(Rain, Temperature)
= 0.971 - 0.9508
= 0.0202

Second Attribute – Wind [Categorical values - weak, strong]
$H(Wind=weak) = -(3/3)*\log_2(3/3)-0 = 0$
$H(Wind=strong) = 0-(2/2)*\log_2(2/2) = 0$
Average Entropy Information for Wind –
I(Wind) = p(Rain, weak)*H(Rain, Wind=weak) + p(Rain, strong)*H(Rain, Wind=strong)

= (3/5)*0 + (2/5)*0

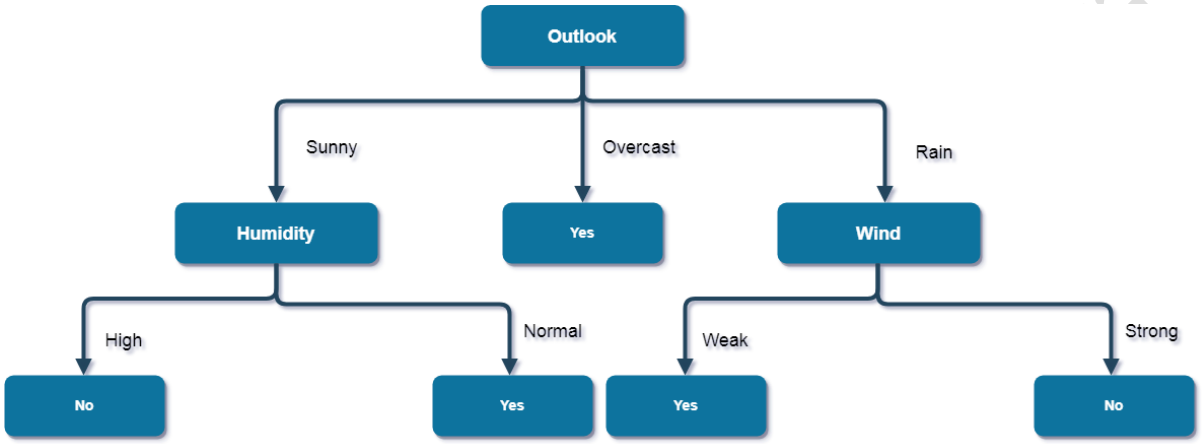
= 0

Information Gain = H(Rain) - I(Rain, Wind)

= 0.971 – 0

= 0.971

Here, the attribute with **maximum information gain is Wind**. So, the decision tree built so far –



Here, when Outlook = Rain and Wind = Strong, it is a pure class of category "no". And When Outlook = Rain and Wind = Weak, it is again a pure class of category "yes".

And this is our final desired tree for the given dataset.

Q2. Use ID3 algorithm on a dataset of car features given below and compute the decision tree.

Engine	SC/ Turbo	Weight	Fuel Eco	Fast
Small	No	Average	Good	No
Small	No	Light	Average	No
Small	Yes	Average	Bad	Yes
Medium	No	Heavy	Bad	Yes
Large	No	Average	Bad	Yes
Medium	No	Light	Bad	No
Large	Yes	Heavy	Bad	No
Large	No	Heavy	Bad	No
Medium	Yes	Light	Bad	Yes
Large	No	Average	Bad	Yes
Small	No	Light	Good	No
Small	No	Average	Average	No
Medium	No	Heavy	Bad	No
Small	Yes	Average	Average	No
Medium	No	Heavy	Bad	No

Solution:

Here, dataset is of binary classes (yes and no), where 5 out of 15 are "yes" and 10 out of 15 are "no".

Complete entropy of dataset is –

$H(S) = - p(yes) * \log_2(p(yes)) - p(no) * \log_2(p(no))$

$= - (5/15) * \log_2(5/15) - (10/15) * \log_2(10/15)$

$= - (-0.53) - (-0.39)$

$= 0.92$

For each attribute of the dataset, let's follow the step-2 of pseudocode: -

First Attribute – Engine [Categorical values – small, medium, large]

$H(Engine=small) = -(1/6)*\log_2(1/6)-(5/6)*\log_2(5/6) = 0.65$

$H(Engine=medium) = -(3/5)*\log_2(3/5)-(2/5)*\log_2(2/5) =0.971$

$H(Engine=large) = -(2/4)*\log_2(2/4)- (2/4)*\log_2(2/4) = 1$

Average Entropy Information for Outlook –

I(Engine) = p(small) * H(Engine=small) + p(medium) * H(Engine=medium) + p(large) * H(Engine=large)

$= (6/15)*0.65 + (5/15)*0.971 + (4/15)*1$

$= 0.85$

Information Gain = H(S) - I(Engine)

$= 0.92 - 0.85$

$= 0.07$

Now we do the same calculations for all the remaining attributes:

Second Attribute – SC/Turbo [Categorical values – yes, no]

$H(SC/Turbo = yes) = -(2/4)*\log_2(2/4) - (2/4)*\log_2(2/4) =1$

$H(SC/Turbo = no) = -(3/11)*\log_2(3/11) - (8/11)*\log_2(8/11) =0.85$

Average Entropy of SC/Turbo –

I(SC/Turbo) = p(yes)*H(SC/Turbo = yes) + p(no)*H(SC/Turbo = No)

$= (4/15)*1 + (11/15)*0.85$

$= 0.89$

Information Gain = H(S) – I(SC/Turbo)

$= 0.92 – 0.89$

$= 0.03$

Third Attribute – Weight [Categorical values – average, light, heavy]

$H(Weight = average) = -(3/6)*\log_2(3/6)-(3/6)*\log_2(3/6) = 1$

$H(Weight = light) = -(1/4)*\log_2(1/4)-(2/4)*\log_2(2/4) = 0.81$

$H(Weight = heavy) = -(1/5)*\log_2(1/5)- (4/5)*\log_2(4/5) = 0.72$

Average Entropy of Weight –

I(Weight) = p(average)*H(Weight=average) + p(light)*H(Weight=light) + p(heavy) * H(Weight=heavy)

= (6/15)*1 + (4/15)*0.81 + (5/15)* 0.72
= 0.86

Information Gain = H(S) – I(Weight)
= 0.92 – 0.86
= 0.06

Fourth Attribute – Fuel Economy [Categorical values – good, average, bad]

H(Fuel Economy = good) = $-(0/2)*\log_2(0/2)-(2/2)*\log_2(2/2) = 0$

H(Fuel Economy = average) = $-(0/3)*\log_2(0/3)-(3/3)*\log_2(3/3) =0$

H(Fuel Economy = bad) = $-(5/10)*\log_2(5/10)- (5/10)*\log_2(5/10) = 1$

Average Entropy Information for Outlook –

I(Fuel Economy) = p(good) * H(Fuel Economy=good) + p(average) * H(Fuel Economy = medium) + p(bad) * H(Fuel Economy = bad)

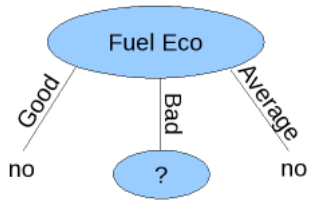
= (2/15)*0 + (3/15)*0 + (10/15)*1

= 0.67

Information Gain = H(S) - I(Fuel Economy)

= 0.92 - 0.67 = 0.25

Here, the attribute with **maximum information gain is Fuel Economy**. So, the decision tree built so far –



Here, when Fuel Eco == good, it is of pure class(no) and when Fuel Eco == average, it is of pure class (no).

Now, we have to repeat same procedure for the data with rows consist of Fuel Eco value as bad.

Now, finding the best attribute for splitting the data with Fuel Eco == bad values {Dataset rows = [3, 4, 5, 6, 7, 8, 9, 10, 13, 15]}.

Complete entropy of bad is –

H(bad) = - p(yes) * $\log_2$(p(yes)) - p(no) * $\log_2$(p(no))
= - (5/10) * $\log_2$(5/10) - (5/10) * $\log_2$(5/10)
= 1

First Attribute – Engine [Categorical values – small, medium, large]

H(Engine=small) = $-(1/1)*\log_2(1/1)-(0/1)*\log_2(0/1) = 0$

H(Engine=medium) = $-(2/5)*\log_2(2/5)-(3/5)*\log_2(3/5) =0.971$

H(Engine=large) = $-(2/4)*\log_2(2/4)- (2/4)*\log_2(2/4) = 1$

Average Entropy Information for Outlook –

I(Engine) = p(small) * H(Engine=small) + p(medium) * H(Engine=medium) + p(large) * H(Engine=large)

= (1/10)*0 + (5/10)*0.971 + (4/10)*1

= 0.86

Information Gain = H(bad) - I(Engine)

= 1 - 0.86

= 0.14

Second Attribute – SC/Turbo [Categorical values – yes, no]

$H(SC/Turbo = yes) = -(2/3)*\log_2(2/3) - (1/3)*\log_2(1/3) = 0.92$

$H(SC/Turbo = no) = -(3/7)*\log_2(3/7) - (4/7)*\log_2(4/7) = 0.96$

Average Entropy of SC/Turbo –

I(SC/Turbo) = p(yes)*H(SC/Turbo = yes) + p(no)*H(SC/Turbo = No)

= (3/10)*0.92 + (7/10)*0.96

= 0.95

Information Gain = H(bad) – I(SC/Turbo)

= 1 – 0.95

= 0.05

Third Attribute – Weight [Categorical values – average, light, heavy]

$H(Weight = average) = -(3/3)*\log_2(3/3)-(0/3)*\log_2(0/3) = 0$

$H(Weight = light) = -(1/2)*\log_2(1/2)-(1/2)*\log_2(1/2) = 1$

$H(Weight = heavy) = -(1/5)*\log_2(1/5)- (4/5)*\log_2(4/5) = 0.72$

Average Entropy of Weight –

I(Weight) = p(average)*H(Weight=average) + p(light)*H(Weight=light) + p(heavy) * H(Weight=heavy)

= (3/10)*0 + (2/10)*1 + (5/10)* 0.72

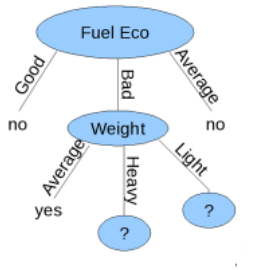
= 0.56

Information Gain = H(bad) – I(Weight)

= 1 – 0.56

= 0.44

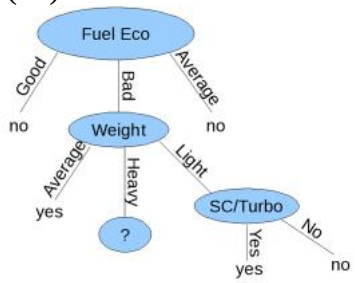
Here, the attribute with **maximum information gain is Weight**. So, the decision tree built so far –



Here, when Fuel Eco is Bad and Weight is Average, it is of pure class (yes).

Now, if we look at the original dataset given, there are only two items for SC/Turbo where Weight = Light and the result is consistent, i.e. when Fuel Eco is bad and Weight is Light and

SC/Turbo is yes, it is of pure class (yes) and when Fuel Eco is bad and Weight is Light and SC/Turbo is no, it is of pure class (no). So the decision tree now built will be –



Now, the updated dataset with Fuel Eco = Bad and Weight = Heavy is as shown below:

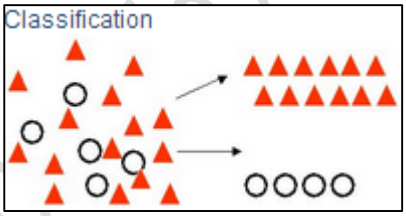
Engine	SC/ Turbo	Weight	Fuel Eco	Fast
Medium	No	Heavy	Bad	Yes
Large	Yes	Heavy	Bad	No
Large	No	Heavy	Bad	No
Medium	No	Heavy	Bad	No
Medium	No	Heavy	Bad	No

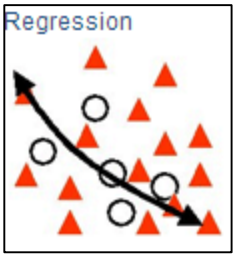
All the cars with large engines in this table are not fast, i.e. it is of pure class (No).
Due to inconsistent patterns in the data, there is no way to proceed since medium size engines may lead to either fast or not fast.

CART Algorithm

CART (Classification and Regression Tree) is an alternative decision tree building algorithm. It can handle both classification and regression tasks. This algorithm uses a new metric named gini index to create decision points for classification tasks. The CART or Classification & Regression Trees methodology was introduced in 1984 by Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone as an umbrella term to refer to the following types of decision trees:

- **Classification Trees**: In these trees, the target variable is categorical and the tree is used to identify the "class" within which a target variable would likely fall into.





- **Regression Trees**: In these trees, the target variable is continuous and tree is used to predict it's value.

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.

Gini (Attribute = value) = 1 – Σ $(P_i)^2$ for i=1 to number of classes

Gini (Attribute) = $\sum_{v=values} P_v \times Gini(Attribute = Value)$

Note: Select the attribute as the node of decision tree whose Gini Index is Low.

We will mention a step by step CART decision tree example by hand from scratch on the below given Golf Playing 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

There are 14 instances of golf playing decisions based on outlook, temperature, humidity and wind factors.

Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. We 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

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$

Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$

Gini(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.

Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = 0.342

Temperature

Similarly, temperature is a nominal feature and it could have 3 different values: Cool, Hot and Mild. Let’s summarize decisions for temperature feature.

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) x 0.5 + (4/14) x 0.375 + (6/14) x 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) x 0.489 + (7/14) x 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) = $1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$

Weighted sum for wind feature will be calculated next

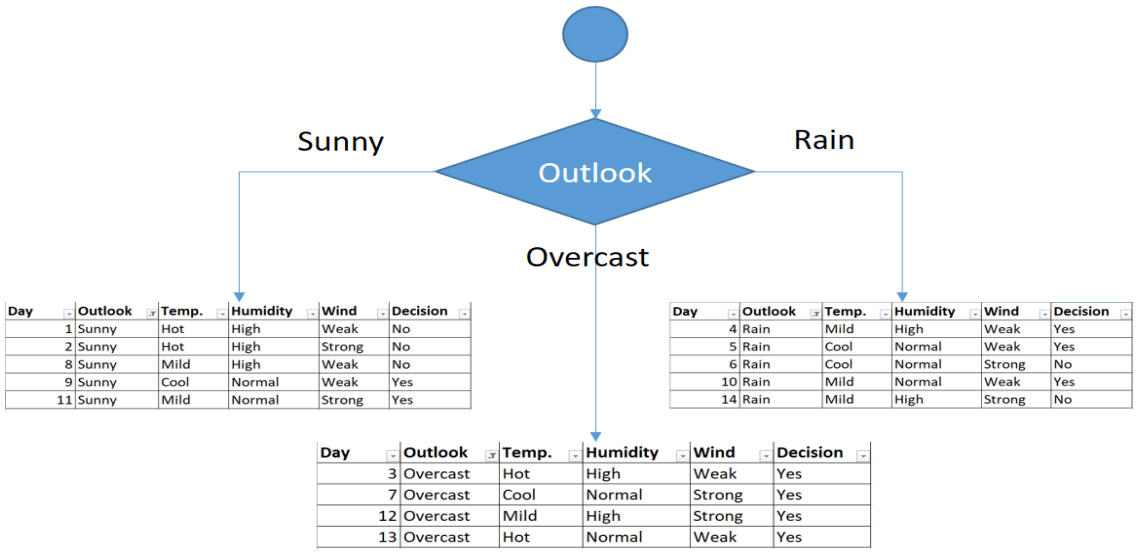
Gini(Wind) = (8/14) x 0.375 + (6/14) x 0.5 = 0.428

Time to decide

We've calculated gini index values for each feature. **The winner will be outlook feature because its cost is the lowest.**

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428

We'll put outlook decision at the top of the tree.

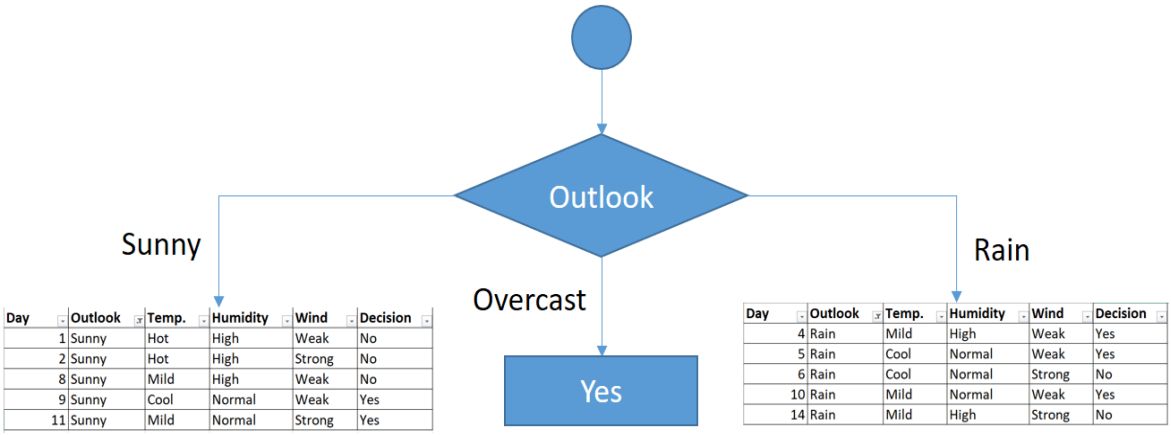


Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

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

You might realize that sub dataset in the overcast leaf has only yes decisions. This means that overcast leaf is over. Tree is over for overcast outlook leaf.



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

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

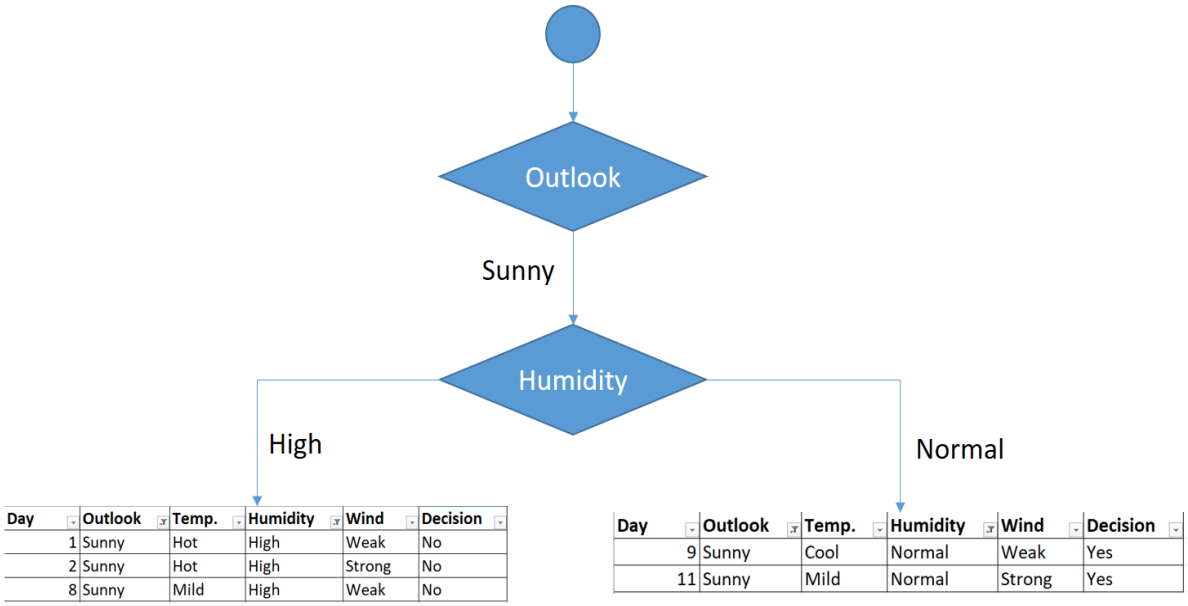
Gini(Outlook=Sunny and Wind) = (3/5)x0.266 + (2/5)x0.2 = 0.466

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

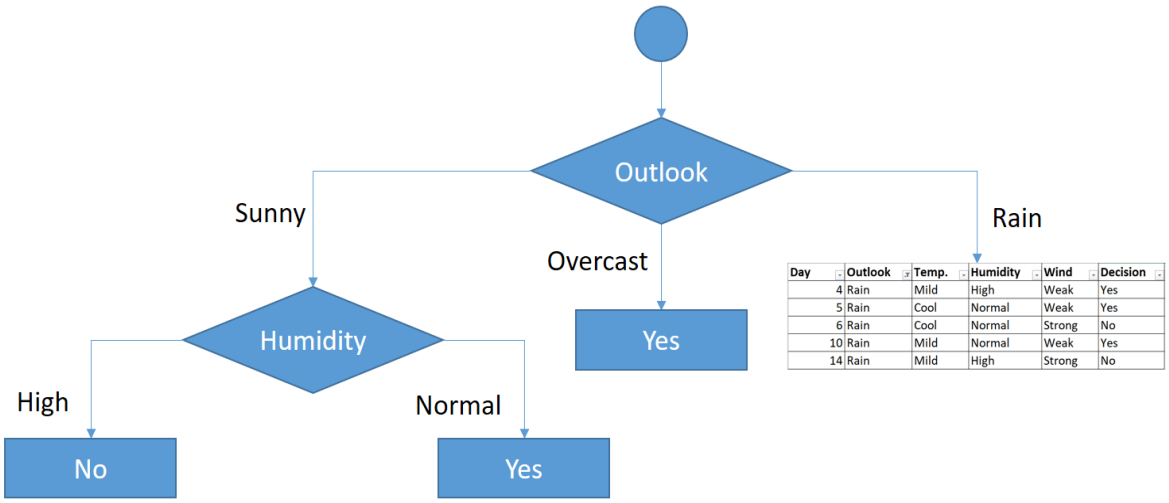
We'll put humidity check at the extension of sunny outlook.



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

As seen, decision is always no for high humidity and sunny outlook. On the other hand, decision will always be yes for normal humidity and sunny outlook. This branch is over.



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

Now, we need to focus on rain outlook.

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

We'll calculate gini index scores for temperature, humidity and wind features when outlook is rain.

Gini of temperature 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)x0.5 + (3/5)x0.444 = 0.466

Gini of wind for rain outlook

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

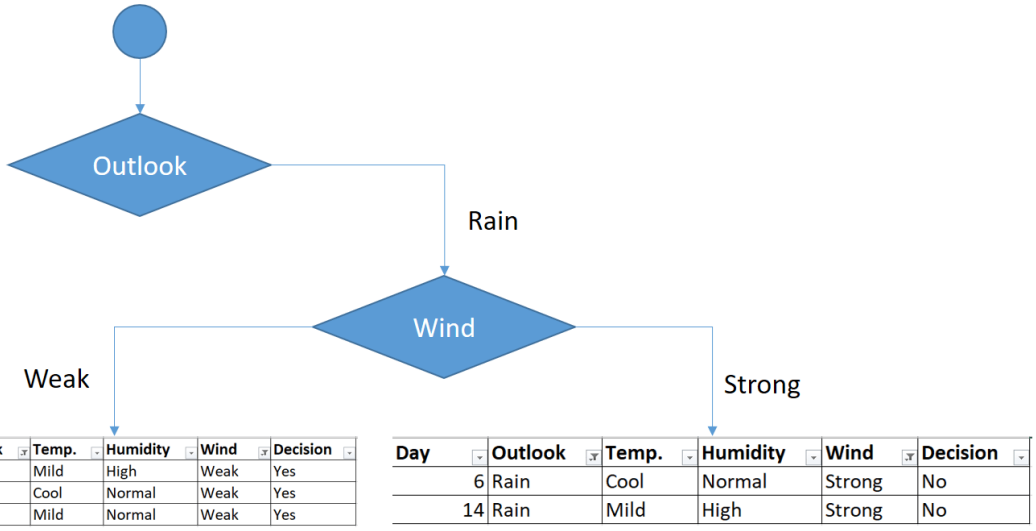
Gini(Outlook=Rain and Wind=Weak) = $1 – (3/3)^2 – (0/3)^2 = 0$
Gini(Outlook=Rain and Wind=Strong) = $1 – (0/2)^2 – (2/2)^2 = 0$
Gini(Outlook=Rain and Wind) = (3/5)x0 + (2/5)x0 = 0

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

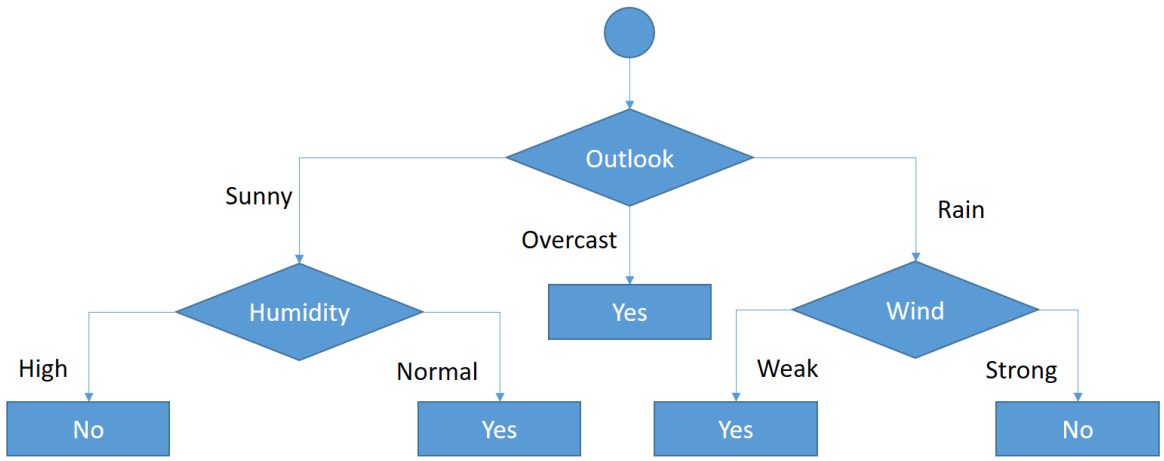
Put the wind feature for rain outlook branch and monitor the new sub data sets.



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

As seen, decision is always yes when wind is weak. On the other hand, decision is always no if wind is strong. This means that this branch is over.



Q2. For the following dataset of Employee Selection given below, compute gini index of each attribute and determine which attribute is the root attribute.

Tuple#	Age	Salary	Job	Performance	Select
1	Young	High	Private	Average	No
2	Young	High	Private	Excellent	No
3	Middle-aged	High	Private	Average	Yes
4	Old	Medium	Private	Average	Yes
5	Old	Low	Government	Average	Yes
6	Old	Low	Government	Excellent	No
7	Middle-aged	Low	Government	Excellent	Yes
8	Young	Medium	Private	Average	No
9	Young	Low	Government	Average	Yes
10	Old	Medium	Government	Average	Yes

11	Young	Medium	Government	Excellent	Yes
12	Middle-aged	Medium	Private	Excellent	Yes
13	Middle-aged	High	Government	Average	Yes
14	Old	Medium	Private	Excellent	No

Solution:
There are 14 instances of selection decisions based on age, salary, job and performance factors.

Age
Age is a nominal feature. It can be young, old or middle-aged. We will summarize the final decisions for age feature.

Age	Yes	No	Number of instances
Young	2	3	5
Middle-Aged	4	0	4
Old	3	2	5

Gini(Age=Young) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$
Gini(Age = Middle-aged) = $1 – (4/4)^2 – (0/4)^2 = 0$
Gini(Outlook=Old) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$
Then, we will calculate weighted sum of gini indexes for age feature.
Gini(Age) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = 0.342

Salary
Similarly, salary is a nominal feature and it could have 3 different values: High, Medium and Low. Let's summarize decisions for salary feature.

Salary	Yes	No	Number of instances
High	2	2	4
Low	3	1	4
Medium	4	2	6

Gini(Salary=High) = $1 – (2/4)^2 – (2/4)^2 = 0.5$
Gini(Salary=Low) = $1 – (3/4)^2 – (1/4)^2 = 1 – 0.5625 – 0.0625 = 0.375$
Gini(Salary=Medium) = $1 – (4/6)^2 – (2/6)^2 = 1 – 0.444 – 0.111 = 0.445$
We'll calculate weighted sum of gini index for salary feature
Gini(Salary) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = 0.439

Job
Humidity is a binary class feature. It can be private or government.

Job	Yes	No	Number of instances
Private	3	4	7
Govt.	6	1	7

Gini(Job=Private) = $1 – (3/7)^2 – (4/7)^2 = 1 – 0.183 – 0.326 = 0.489$
Gini(Job=Govt.) = $1 – (6/7)^2 – (1/7)^2 = 1 – 0.734 – 0.02 = 0.244$
Weighted sum for job feature will be calculated next
Gini(Job) = (7/14) x 0.489 + (7/14) x 0.244 = 0.367

Performance

Performance is a binary class similar to job. It can be average or excellent.

Performance	Yes	No	Number of instances
Average	6	2	8
Excellent	3	3	6

Gini(Performance=Average) = $1 – (6/8)^2 – (2/8)^2 = 1 – 0.5625 – 0.062 = 0.375$

Gini(Performance=Excellent) = $1 – (3/6)^2 – (3/6)^2 = 1 – 0.25 – 0.25 = 0.5$

Weighted sum for performance feature will be calculated next

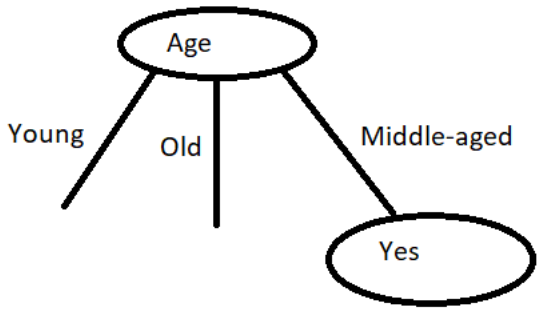
Gini(Performance) = (8/14) x 0.375 + (6/14) x 0.5 = 0.428

Time to decide

We've calculated gini index values for each feature. **The winner will be age feature because its cost is the lowest.**

Feature	Gini index
Age	0.342
Salary	0.439
Job	0.367
Performance	0.428

We'll put age decision at the top of the tree. From the dataset it is clear that for age = middle-aged, select = yes.



ID3 Vs CART

Sr. No.	**Features**	**ID3**	**CART**
1	Acronym	Iterative Dichotomiser 3	Classification and Regression Tree
2	Splitting Criteria	Information Gain	Gini Index
3	Attribute Type	Handles only categorical data	Handles both categorical and numeric value
4	Missing Values	Do not handle missing values	Handle missing values
5	Pruning Strategy	No pruning is done	Post pruning is done
6	Outlier Detection	Susceptible to outliers	Can handle outliers

Exercise Questions:

1. Consider the following Trading Dataset given below. Compute the decision tree using both ID3 and CART algorithm.

Past Trend	Open Interest	Trading Volume	Return
Positive	Low	High	Up
Negative	High	Low	Down
Positive	Low	High	Up
Positive	High	High	Up
Negative	Low	High	Down
Positive	Low	Low	Down
Negative	High	High	Down
Negative	Low	High	Down
Positive	Low	Low	Down
Positive	High	High	Up

2. Consider the following AllElectronics Customers Database. Compute the decision tree using both ID3 and CART algorithm.

Tuple#	**Age**	**Income**	**Student**	**Credit_Rating**	**Buys_Computer**
1	Young	High	No	Fair	No
2	Young	High	No	Excellent	No
3	Middle-aged	High	No	Fair	Yes
4	Senior	Medium	No	Fair	Yes
5	Senior	Low	Yes	Fair	Yes
6	Senior	Low	Yes	Excellent	No
7	Middle-aged	Low	Yes	Excellent	Yes
8	Young	Medium	No	Fair	No
9	Young	Low	Yes	Fair	Yes
10	Senior	Medium	Yes	Fair	Yes
11	Young	Medium	Yes	Excellent	Yes
12	Middle-aged	Medium	No	Excellent	Yes
13	Middle-aged	High	Yes	Fair	Yes
14	Senior	Medium	No	Excellent	No

3. Consider the following database of Transportation. Compute the decision tree using ID3 and CART algorithm.

Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

4. Consider the following database for Vegetation Classification. Compute the decision tree using ID3 algorithm.

ID	Stream	Slope	Elevation	Vegetation
1	False	Steep	High	Chaparral
2	True	Moderate	Low	Riparian
3	True	Steep	Medium	Riparian
4	False	Steep	Medium	Chaparral
5	False	Flat	High	Conifer
6	True	Steep	Highest	Conifer
7	True	Steep	High	Chaparral

5. Consider the database for classification of mammal. Compute the decision tree using both ID3 and CART algorithm.

	toothed	**hair**	**breathes**	**legs**	**Species**
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile