School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 1, 2023)
Week 3: Solutions

1. Approximately 1% of women aged between 40 and 50 have breast cancer. 80% of mammogram screening tests detect breast cancer when it is there. 90% of mammograms DO NOT show breast cancer when it is **NOT** there. Based on this information, complete the following table.

Cancer	Probability
No	99%
Yes	1%

Cancer	Test	Probability
Yes	Positive	80%
Yes	Negative	?
No	Positive	?
No	Negative	90%

Based on the probability rule of sum for mutually exclusive events (events that cannot both happen at the same time), we know that the sum of positive and negative test results should sum up to 1 (or 100%).

Therefore, when we have a patient with cancer (`Cancer` = '`Yes`'), and we know that there is 80% probability that the test detects it (`Test` returns '`Positive`'), it means that there is 20% chance (1 – 0.80 = 0.20) that the test does not detect the cancer (`Test` returns '`Negative`' results). We call this a **False Negative** (wrong negative); you will learn more about it later in lectures.

Similarly, when a patient does not have cancer (`Cancer` = '`No`'), and we have that there is 90% chance that the test proves that (`Test` returns '`Negative`'), it means that there is 10% chance (1 – 0.9 = 0.1) that the test detects cancer (returns '`positive`' results) when it is not there! We call this a **False Positive** (wrong positive), and again you will learn more about it later in lectures when we talk about evaluations.

So, the filled table would be as follow:

Cancer	Test	Probability
Yes	Positive	80%
Yes	Negative	20%
No	Positive	10%
No	Negative	90%

2. Based on the results in question 2, calculate the **marginal probability** of 'positive' results in a Mammogram Screening Test.

According to the law of total probability, we know that

$$P(A) = \sum_{n} P(A|B_n)\, P(B_n)$$

So, to calculate the probability of '`positive`' result for Test, we will have:

$$P(Test = 'positive') = P(Test = 'positive'|Cancer = 'no').P(Cancer = 'no') + P(Test = 'positive'|Cancer = 'yes').P(Cancer = 'yes')$$

Based on the question definition, we know that the chance of having breast cancer (for females aged

between 40 and 50) is 1% . So `P(Cancer = 'yes') = 0.01` and `P(Cancer = 'no') = 0.99`.

From question 1, we know that the probability of a `positive` test result is 80% for a patient with cancer (`P(Test = 'positive'|Cancer= 'yes')=0.8`) and the probability of a `positive` test result is 10% for a patient with no cancer (`P(Test='positive'|Cancer='no')=0.1`).

So, we have:

$$P(Test = 'positive') = 0.1 \times 0.99 + 0.8 \times 0.01 = 0.107$$

We can show all these in a **Joint Probability Distribution** table as follow.

		Test		Total
		Positive	Negative	
Cancer	Yes	0.01 x 0.8 = 0.008	0.01 x 0.2 = 0.002	0.01
	No	0.99 x 0.1 = 0.099	0.99 x 0.9 = 0.891	0.99
Total		0.107	0.893	1

We call the totals (row and column) the **Marginal Probability**, because they are in the margin!

3. Based on the results in question 2, calculate *P(Cancer = 'Yes' | Test = 'Positive')*, using the Bayes Rule.

According to the Bayesian Rule, we know that we can calculate the probability that a person actually has breast cancer given that her mammography test results return positive, using the following formula:

$$P(Cancer = 'yes'|Test = 'positive') = \frac{P(Test = 'positive'|Cancer = 'yes').P(Cancer = 'yes')}{P(Test = 'positive')}$$

Based on the given information in the question text, we know that "80% of mammogram screening tests detect breast cancer when it is there", so `P(Test = 'positive' | Cancer = 'yes')` is `0.8 (80%)`.

Also, there `1%` chance of having breast cancer (for females aged between 40 and 50). So `P(Cancer = 'yes') = 0.01`.

Also, from Question2, we have the `P(Test = 'positive') = 0.107` (the expectation of `'positive'` results for a mammogram test).

So we can easily calculate the `P(Cancer= 'yes' | Test = 'positive')`:

$$P(Cancer = 'yes'|Test = 'positive') = \frac{0.8 \times 0.01}{0.107} \cong 0.075 = 7.5\%$$

This result shows that even if a mammography test result return positive, there is only a 7.5% chance that the person actually has Cancer! ☺

4. For the following dataset:

ID	*Outl*	*Temp*	*Humi*	*Wind*	PLAY
TRAINING INSTANCES					
A	s	h	h	F	N
B	s	h	h	T	N
C	o	h	h	F	Y

D	r	m	h	F	Y
E	r	c	n	F	Y
F	r	c	n	T	N
TEST INSTANCES					
G	o	c	n	T	?
H	s	m	h	F	?

Classify the test instances using the **ID3 Decision Tree** method and **Gain Ratio** as a splitting criterion.

For calculating the Gain Ration, we first need to calculate the Information Gain at each level of the decision tree. We're going to choose the attribute that has the largest difference between the entropy of the class distribution at the parent node, and the average entropy across its daughter nodes (weighted by the fraction of instances at each node);

$$IG(A|R) = H(R) - \sum_{i \in A} P(A = i)H(A = i)$$

In this dataset, we have 6 instances total — 3 Y and 3 N. The entropy at the top level of our tree is $H(R) = -\left[\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}\right]$

This is a very even distribution. We're going to hope that by branching the tree according to an attribute, that will cause the daughters to have an uneven distribution - which means that we will be able to select a class with more confidence - which means that the entropy will go down.

For example, for the attribute `Outl`, we have three attribute values: `s, o, r`.

- When `Outl=s`, there are 2 instances, which are both N. The entropy of this distribution is $H(O = s) = -\left[0\log_2 0 + \frac{2}{2}\log_2\frac{2}{2}\right] = 0$. Obviously, at this branch, we will choose N with a high degree of confidence.
- When `Outl=o`, there is a single instance, of class Y. The entropy here is going to be 0 as well.
- When `Outl=r`, there are 2 Y instances and 1 N instance. The entropy here is $H(o = r) = -\left[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right] \approx 0.9183$

To find the average entropy (the "mean information"), we sum the calculated entropy at each daughter multiplied by the fraction of instances at that daughter: $MI(O) = \frac{2}{6}(0) + \frac{1}{6}(0) + \frac{3}{6}(0.9183) \approx 0.4592$

The overall Information Gain here is $IG(O) = H(R) - MI(O) = 1 - 0.4592 = 0.5408$.

	R	Outl			Temp			H		Wind		ID					
		s	o	r	h	m	c	h	n	T	F	A	B	C	D	E	F
Y	3	0	1	2	1	1	1	2	1	0	3	0	0	1	1	1	0
N	3	2	0	1	2	0	1	2	1	2	1	1	1	0	0	0	1
Total	6	2	1	3	3	1	2	4	2	2	4	1	1	1	1	1	1
P(Y)	1/2	0	1	2/3	1/3	1	1/2	1/2	1/2	0	3/4	0	0	1	1	1	0
P(N)	1/2	1	0	1/3	2/3	0	1/2	1/2	1/2	1	1/4	1	1	0	0	0	1
H	1	0	0	0.9183	0.9183	0	1	1	1	0	0.8112	0	0	0	0	0	0
MI		0.4592			0.7924			1		0.5408		0					
IG		0.5408			0.2076			0		0.4592		1					
SI		1.459			1.459			0.9183		0.9183		2.585					
GR		0.3707			0.1423			0		0.5001		0.3868					

The table above lists the Mean Information and Information Gain, for each of the 5 attributes.

At this point, `ID` has the best information gain, so hypothetically we would use that to split the root node. But we are not supposed to use the Information Gain, we are going to calculate the Gain Ratio for each attribute. To do so, we're going to weight down (or up!) by the "split information"

— the entropy of the distribution of instances across the daughters of a given attribute.

For example, we found that, for the root node, `Outl` has an information gain of 0.5408. There are 2 (out of 6) instances at the s daughter, 1 at the o daughter, and 3 at the r daughter.

The split information for `Outl` is $SI(o) = -\left[\frac{2}{6}\log_2\frac{2}{6} + \frac{1}{6}\log_2\frac{1}{6} + \frac{3}{6}\log_2\frac{3}{6}\right] \approx 1.459.$

The Gain ratio is consequently $GR(o) = \frac{IG(o)}{SI(o)} \approx \frac{0.5408}{1.459} \approx 0.3707$

The values for split information and gain ratio for each attribute at the root node are shown in the table above. The best attribute (with the greatest gain ratio) at the top level this time is `Wind`.

`Wind` has two branches: T is pure, so we focus on improving `F` (which has 3 Y in- stances (C, D, E), and 1 N instance (A)). The entropy of this daughter is 0.8112.

- For `Outl`, we have a single instance at s (class N, H = 0), a single instance at `o` (class Y, H = 0), and 2 Y instances at `r` (H = 0). The mean information here is clearly 0; the information gain is 0.8112. The split information is $SI(o|(W=F)) = -\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{2}\log_2\frac{1}{2}\right] = 1.5$, so the gain ratio is $GR\big(o\big|(W=F)\big) = \frac{0.8112}{1.5} \approx 0.5408$
- For `Temp`, we have two `h` instances (one Y and one N, so H = 1), a single `m` instance (Y, H = 0), and a single `c` instance (Y, H = 0). The mean information is $\frac{2}{4}(1) + \frac{1}{4}(0) + \frac{1}{4}(0) = 0.5$, so the information gain is 0.8112-0.5 = 0.3112. The distribution of instances here is the same as Outl, so the split information is also 1.5, and the gain ratio is $GR\big(T\big|(W=F)\big) = \frac{0.3112}{1.5} \approx 0.2075$
- For `Humi`, we have 3 `h` instances (2 Y and 1 N, H = 0.9183), and 1 `n` instance (Y, H = 0): the mean information is $\frac{3}{4}$ (0.9183) + $\frac{1}{4}$(0) = 0.6887 and the information gain is 0.8112 − 0.6887 = 0.1225. The split information is $SI(H|(W=F)) = -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] \approx 0.8112$, so the gain ratio is $GR\big(H\big|(W=F)\big) = \frac{0.1225}{0.8112} \approx 0.1387.$
- For `ID`, the mean information is obviously still 0, so the information gain is 0.8112. The split information at this point is $-\left[\frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4}\right] = 2$, so the gain ratio is approximately 0.4056.

Of our four choices at this point, `Outl` has the best gain ratio. The resulting daughters are all pure, so the decision tree is finished:

- `Wind=F ∩(Outl=o ∪ Outl=r) → Y`
- `Wind=T ∪(Wind=F ∩ Outl=s) → N` (so we classify G and H as N)

Note also that `ID` attribute f is no longer the “best” attribute. In calculating the Gain Ratio, the split information pushed down its “goodness” to the point where we didn’t need to ignore it explicitly.

The final tree would have a structure as below: