School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies (Semester 1, 2017)
Workshop exercises: Week 10

1. A **confusion matrix** is an indication of the performance of a classifier over a set of test data, by counting the various output instances:

		Actual			
		a	b	c	d
Classified	a	10	2	3	1
	b	2	5	3	1
	c	1	3	7	1
	d	3	0	3	5

   (a) Calculate the classification **accuracy** of the system.

   - Accuracy is (more simply) defined as the fraction of correctly identified instances, out of all of the instances. In the case of a confusion matrix, the correct instances are the ones enumerated along the main diagonal (classified as a and actually a etc.):

$$\begin{aligned} \textit{Accuracy} &= \frac{\text{\# of correctly identified instances}}{\text{total \# of instances}} \\ &= \frac{10+5+7+5}{10+2+3+1+2+5+3+1+1+3+7+1+3+0+3+5} \\ &= \frac{27}{50} = 54\% \end{aligned}$$

   (b) Calculate the **precision**, **recall**, **F-score** (where $\beta = 1$), **sensitivity**, and **specificity** for class d. (Why can't we do this for the whole system? How can we consider the whole system?)

   - Precision for a given class is defined as the fraction of correctly identified instances of that class, from the times that class was attempted to be classified. We are interested in the true positives (TP) where we attempted to classify an item as an instance of said class (in this case, d) and it was actually of that class (d): in this case, there are 5 such instances. The false positives (FP) are those items that we attempted to classify as being of class d, but they were actually of some other class: there are $3+0+3=6$ of those.

$$\begin{aligned} \textit{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}} \\ &= \frac{5}{5+3+0+3} \\ &= \frac{5}{11} \approx 45\% \end{aligned}$$

   - Recall for a given class is defined as the fraction of correctly identified instance of that class, from the times that class actually occurred. This time, we are interested in the true positives, and the false negatives (FN): those items that were actually of class d, but we classified as being of some other class; there are $1+1+1=3$ of those.

$$\begin{aligned} \textit{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\ &= \frac{5}{5+1+1+1} \\ &= \frac{5}{8} \approx 62\% \end{aligned}$$

- F-score is a measure which attempts to combine precision (P) and recall (R) into a single score. In general, it is calculated as:

$$F_\beta = \frac{(1+\beta^2)P \cdot R}{(\beta^2 \cdot P) + R}$$

- By far, the most typical formulation is where the parameter β is set to 1: this means that precision and recall are equally important to the score, and that the score is a harmonic mean:

$$F_{\beta=1} = \frac{2 \cdot P \cdot R}{P + R}$$

- In this case, we have calculated the precision of class d to be $\frac{5}{11}$ and the recall to be $\frac{5}{8}$. The F-score where $(\beta = 1)$ of class d is then:

$$\begin{aligned} F_{\beta=1} &= \frac{2 \cdot P \cdot R}{P + R} \\ &= \frac{2 \cdot \frac{5}{11} \cdot \frac{5}{8}}{\frac{5}{11} + \frac{5}{8}} \\ &= \frac{50}{95} \approx 53\% \end{aligned}$$

- Sensitivity is defined the same way as recall: $\frac{TP}{TP+FN}$.
- Specificity is precision with respect to the negative instances:

$$\begin{aligned} \textit{Specificity} &= \frac{\text{TN}}{\text{TN} + \text{FP}} \\ &= \frac{10+2+3+2+5+3+1+3+7}{10+2+3+2+5+3+1+3+7+3+0+3} \\ &= \frac{36}{42} \approx 86\% \end{aligned}$$

2. How is **holdout** evaluation (like in the Project 2 data) different to **cross-validation** evaluation?

   - In a holdout evaluation strategy, we partition the data into a training set and a test set: we build the model on the former, and evaluate on the latter.
   - In a cross-validation evaluation strategy, we do the same as above, but a number of times, where each iteration uses one partition of the data as a test set and the rest as a training set (and the partition is different each time).

3. 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 a Decision Tree:

   (a) Using the Information Gain as a splitting criterion

	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
$P(Y)$	$\frac{1}{2}$	0	1	$\frac{2}{3}$	$\frac{1}{3}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{3}{4}$	0	0	1	1	1	0
$P(N)$	$\frac{1}{2}$	1	0	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{4}$	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		
$GINI$	0.5	0	0	0.4444	0.4444	0	0.5	0.5	0.5	0	0.375	0	0	0	0	0	0
GS			0.2778			0.1111			0		0.25				0.5		

- For 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);

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

- In this dataset, we have 6 instances total — 3 Y and 3 N. The entropy at the top level of our tree is $H(R) = -[\frac{3}{6}\log_2\frac{3}{6} + \frac{3}{6}\log_2\frac{3}{6}] = 1$.
- 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 $H(O = \texttt{s}) = -[0\log 0 + 1\log 1] = 0$. 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 $H(O = \texttt{r}) = -[\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}] \approx 0.9183$.
- 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 $IG(O) = H(R) - MI(O) = 1 - 0.4592 = 0.5408$.
- 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. At that point, we would be done, because each daughter is purely of a single class — however, we would be left with a completely useless classifier! (Because the IDs of the test instances won't have been observed in the training data.)
- Instead, let's take the second best attribute: *Outl*.
- There are now three branches from our root node: for `s`, for `o`, and for `r`. The first two are pure, so we can't improve them any more. Let's examine the third branch (*Outl*=`r`):
  - Three instances (D, E, and F) have the attribute value `r`; we've already calculated the entropy here to be 0.9183.
  - If we split now according to *Temp*, we observe that there is a single instance for the value `m` (of class N, the entropy is clearly 0); there are two instances for the value `c`, one of class Y and one of class N (so the entropy here is 1). The mean information

is $\frac{1}{3}(0) + \frac{2}{3}(1) \approx 0.6667$, and the information gain at this point is $0.9183 - 0.6667 \approx 0.2516$.

  - For *Humi*, we again have a single instance (with value `h`, class Y, $H = 0$), and two instances (of `n`) split between the two classes ($H = 1$). The mean information here will also be 0.6667, and the information gain 0.2516.
  - For *Wind*, there are two `F` instances, both of class Y ($H = 0$), and one `T` instance of class N ($H = 0$). Here, the mean information is 0 and the information gain is 0.9183.
  - *ID* would still look like a good attribute to choose, but we'll continue to ignore it.
  - All in all, we will choose to branch based on *Wind* for this daughter.
- All of the daughters of `r` are pure now, so our decision tree is complete. We can represent a decision tree over a 2-class problem like this as a pair of Boolean formulae, for example:
  - *Outl*=`o` $\cup$(*Outl*=`r` $\cap$ *Wind*=`F`) $\rightarrow$ Y (so we classify `G` as Y)
  - *Outl*=`s` $\cup$(*Outl*=`r` $\cap$ *Wind*=`T`) $\rightarrow$ N (so we classify `H` as N)

(b) Using the Gini Index as a splitting criterion

- For the Gini Index, at each level of the decision tree, we're going to choose the attribute that has the largest difference between the Gini Index of the class distribution at the parent node, and the averaged Ginis across its daughter nodes (weighted by the fraction of instances at each node); this is sometimes called *GINI-split*:

$$GS(A|R) \quad = \quad GINI(R) - \sum_{i \in A} P(A = i) GINI(A = i)$$

- Observe that this is the same formula as for Information Gain above.
- How do we calculate GINI for this dataset?

$$GINI(X) \quad = \quad 1 - [p(Y)^2 + p(N)^2]$$

- You might like to compare this with the formula to entropy to see why these values are closely correlated.
- Anyway, since the steps of the method are so similar to Information Gain, we've simply recorded the GINI values and GINI-split values in the table above. You can double-check that the same tree is produced as for Information Gain.