Tutorial 3 Solutions

1. Possibility 1: Fold 1 is 2, 5, 7; Fold 2 is 1, 4, 8; Fold 3 is 3, 6.
   Possibility 2: Fold 1 is 1, 8; Fold 2 is 3, 5, 7; Fold 3 is 2, 4, 6.

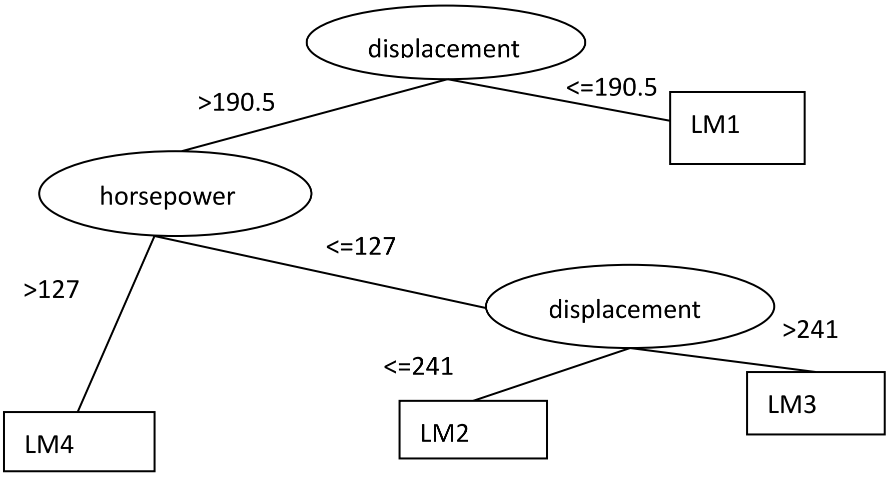
In 3-fold cross validation, the whole data set is randomly split into 3 folds (and usually stratified in each fold). The classification model is then built and tested, each time with 1 fold of data for testing and the rest of the 2 folds for training, for a total of 3 runs. The Cross validation error is the averaged error over the 3 runs. The CV error provides a reasonably good estimation of how the classifier would perform in a real-world application.

2.

a) The numbers represent categorical values. They could have been called "model a", "model b" etc. There is no "greater than" or "less than" relationship between the values. Numeric data type represents continuous values.

b) There is reason for coding cylinders either way. It can be enumerated to a small number of valves, but also an 8-cylinder car is more powerful than a 6-cylinder car.

c)



d) First check the value of displacement attribute. If it's less than 190.5, apply LM1 to predict the class value; else check the horsepower attribute. If horsepower is greater than 127, apply LM4; else check displacement value again, if greater than 241 then apply LM3, otherwise LM2.

e) 307 is greater than 190.5, so we need to check horsepower. 130 is greater than 127, so apply LM4. Note that "model" is equal to 70, therefore none of the expressions related to model will evaluate to true (i.e. 1).

Class = 0.0024*307-0.0278*130-0.0016*3504-0.3653*12+0.6228*0+0.1255*0+ … + 0.1634*0 + 0.2178 * 0 + 28.7128 = 15.8456

3.

Mean Absolute Error =MSE= $\frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{|30-25|+|25-30|+|30-50|+|45-40|}{4}$= 8.75

4.

Benign	Malignant	← Classified as
2	2	Benign
1	0	Malignant

5. Because it is numeric data set and not categorial/symbolic.

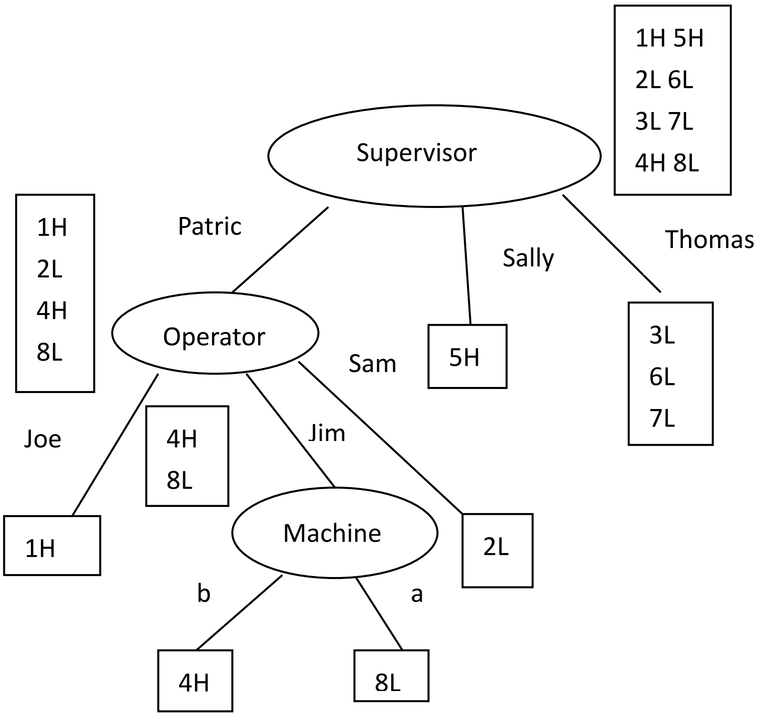
6.

```
Height = Short
|       Hair Colour = Blonde
|       |       Weight = Light: Not
|       |       Weight = Average: Sunburnt
|       Hair Colour = Dark: Not
Height = Average
|       Weight = Light: Sunburnt
|       Weight = Heavy
|               Hair = Red: Sunburnt
|               Hair = Dark: Not
Height = Tall: Not
```

8. Examples in the table are +, +, -.

The rule: If ‘A9’ is ‘f’, ‘A3’ is less than or equal to 0.165 and ‘A7’ equal to ‘n’, then classify the instance as ‘+’.

9. a)



b) First, choose the root node by calculating the disorder after splitting for every attribute. The attribute with the lowest disorder should be chosen.

supervisor			operator			machine			Overtime	
P	T	S	Joe	Sam	Jim	a	b	c	yes	no
2L:2H	3L:0H	0L:1H	1L:2H	2L:0H	2L:1H	1L:1H	2L:1H	2L:1H	2L:0H	3L:3H

$$\text{Disorder(supervisor)} = \frac{4}{8}\text{inf } o(P) + \frac{3}{8}\text{inf } o(T) + \frac{1}{8}\text{inf } o(S)$$

$$= \frac{4}{8}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) + 0 + 0$$

$$\text{Disorder(operator)} = \frac{3}{8}\text{inf } o(Joe) + \frac{2}{8}\text{inf } o(Sam) + \frac{3}{8}\text{inf } o(Jim)$$

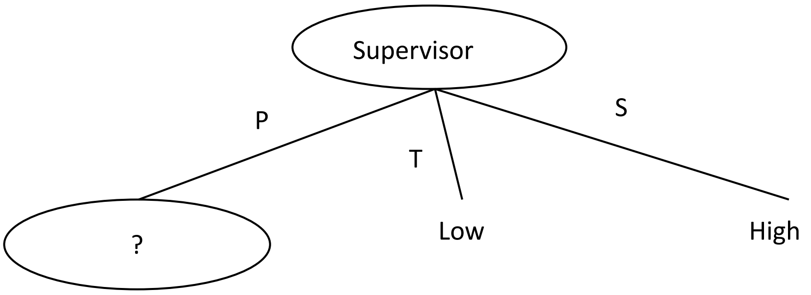
$$= \frac{3}{8}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) + 0 + \frac{3}{8}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3})$$

$$\text{Disorder(machine)} = \frac{2}{8}\text{inf } o(a) + \frac{3}{8}\text{inf } o(b) + \frac{3}{8}\text{inf } o(c)$$

$$= \frac{2}{8} * 1 + \frac{3}{8}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}) + \frac{3}{8}(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3})$$

$$\text{Disorder(overtime)} = \frac{2}{8}\text{inf } o(yes) + \frac{6}{8}\text{inf } o(no) = 0 + \frac{6}{8} * 1$$

After some calculation it can be found that Disorder(supervisor) is the lowest. Note that for values T and S, the leaf nodes consist of instances from only one class, therefore there is no need to split further on those two branches. We then get the following graph,



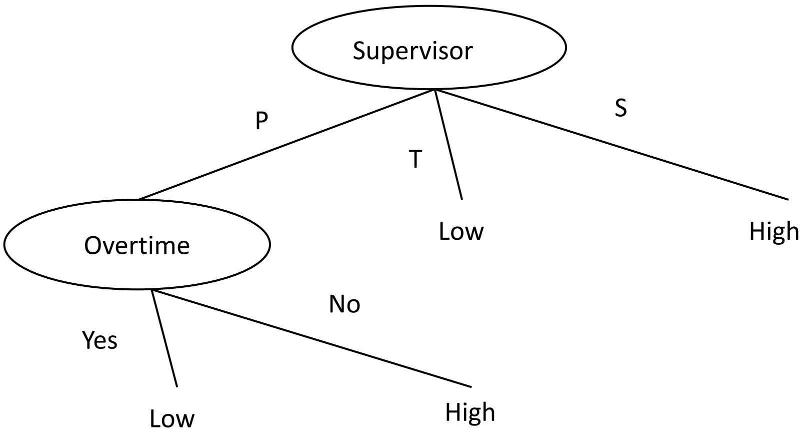
The branch for P has 3 instances from two classes on the leaf node, so we need to split further on that branch. The only instances that we need to consider when making the splitting decision is the set of those which conform to the rules along the path that lead to this branch – in this case, the rule is simply “the value of the attribute supervisor is Patrick”.

1	Patrick	Joe	A	No	High
2	Patrick	Sam	B	Yes	Low
4	Patrick	Jim	B	No	High
8	Patrick	Jim	A	Yes	Low

Just follow the same process we adopted to find the root node and continue recursively (Within the scope of this branch we are also trying to find the "root node"), but we don't consider supervisor as a candidate anymore as we've already used it along the path that lead to this branch.

operator			machine		Overtime	
Joe	Sam	Jim	a	b	yes	no
0L:1H	1L:0H	1L:1H	1L:1H	1L:1H	2L:0H	0L:2H

Apparently overtime is the attribute with lowest disorder. If we split on overtime then each leaf node will have instances that purely belong to one class. Thus we get the full tree,



c) We'll probably control the supervisor and overtime. Fire Thomas as this guy always leads to low productivity. Ask Patrick not to allow his team to work overtime.

d) The whole process is similar to what we did in b). However, instead of choosing the node that lead to *lowest* disorder after splitting, we choose the node with *highest* information gain after splitting. The information gain is the original disorder minus the after-split disorder, and for any attribute the original disorder is the same, so the two ideas are essentially the same thing, though viewed from different angles. The following example shows how to calculate the info gain for choosing supervisor as the root node.

InfoGain(supervisor) = OrigInfo – Disorder(supervisor)

The original information is the information contained in a distribution of three high and five low instances, so,

OrigInfo = $-3/8 * \log_2(3/8) -5/8 * \log_2(5/8)$

e) Both disorder (or interchangeably named information) and Information Gain favor attributes with a large number of possible values. Think about the extreme scenario where each instance has a distinct value for an attribute X, then the after-split disorder for X will always be 0, as each leaf node contains purely instances from one class. However this is definitely not a good attribute to split on – one rule for one instance is just not practical in a real world scenario. To fix this problem, info ratio is introduced, which is a fraction whose numerator is the information gain and the denominator the information of the attribute itself.

InfoRatio(supervisor) = infoGain(supervisor) / Info(supervisor)

Info(supervisor) =$-(4/8)*\log_2(4/8) -(3/8)*\log_2(3/8) -(1/8)*\log_2(1/8)$

For calculation of the rest of the formula refer to examples above.

10. Similar to Question 4. The major difference here is that we now have a numeric attribute, age. How to determine splits based on Age? First sort by age, keeping the link to lenses. The file is almost in this order. Possible split points are between 8 and 12, 12 and 10, 10 and 14, 14 and 15, 15 and 24, 56 and 60, 60 and 65, 65 and 72. Compute the information gain for each of these splits and choose the best. A refinement would be to use a binning procedure as in the OneR algorithm.

11 and 12 are optional. If interested, related discussions of these two questions could be found from the text book, Chap 6 Implementations: Real Machine Learning Schemes, Section 6.1 Decision Trees.