Solutions to tutorial02

1.

A) The *training set* is used to build the classification model. After training you can get the *training error*, which is the error of classification when testing the model on the instances from the training set.

B) The test set is used to test the classification model. Since the classification model is not built on this set, i.e. the instances from the test set are *unknown* to the model; the error on this set is more likely to reflect the true performance of the model when applied to a real-world scenario.

C) 100, 000, 000 instances is usually more than what we need to build a model. We might do stratified sampling (sample set with the same class distribution as the original set) to get a subset of the documents first, e.g. 100,000 instances, and use it for training. For most classifiers, i.e. those which can classify an example reasonably fast, we would use any examples not used in the training as test data.

With a dataset of a small size, we typically use most of the instances for training but only a small portion for testing. This is because labeled data set is generally hard to get, and while more training instances usually contributes to a more accurate classification models, the number of test instances doesn't. That said, a test set with too small a number of instances is not desirable either, as the variance would be huge. For example, for a test set of 10 instances, a misclassification of one instance would result in 10% loss in accuracy, which makes it difficult to accurately evaluate how good the performance of the model is.

2.

a) When the petal-length attribute of the test instance holds a value less than 2.45, the instance will be classified as Iris-setosa. If the value is greater or equal to 2.45 but less than 4.95, it will be classified as Iris-versicolor. Otherwise, classify it as Iris-virginica.

b) The petal-length is 1.3, less than 2.45, thus the instance is classified as Iris-setosa.

c) The first part of the output was describing the performance of the classification model when evaluation was carried out on the full training set, where the model was built from. The second part however, was describing the model's performance in cross validation. 139 is the total number of correctly classified instances over the 10 CV runs.

d) 7 Iris-versicolor instances were classified as c (Iris-virginica), which are incorrect.

3.

Body temp		Mammal	reptile	fish	amphibian	Bird
	Warm-blooded	5	0	0	0	1
	Cold-blooded	0	3	3	2	0

More frequent class is mammal for warm-blooded and either reptile or fish for cold-blooded.

Error is 3+2+1=6.

Skin cover		Mammal	reptile	fish	amphibian	Bird
	Hair	3	0	0	0	0
	Scale	0	3	3	0	0
	None	0	0	0	2	0
	Feather	0	0	0	0	2
	Quil	1	0	0	0	0
	fur	1	0	0	0	0

More frequent class in hair is mammal=3, in scale is either fish or reptile =3, none is amphibian =2, feather is bird=2, quil is mammal=1, and fur is mammal=1.
Error= 3

Gives birth		Mammal	reptile	fish	amphibian	Bird
	Yes	5	0	1	0	0
	no	0	3	2	2	2

More frequent class for yes is mammal=5, and for no is reptile=3. Error= 1+2+2+2=7.

Now, we need to pick attribute with the smallest error which is skin cover. Then we construct the rule based on this attribute.

The rule is:

If skin cover= hair then label is mammal,
Else if skin cover= scale then label is reptile,
Else if skin cover= none then label is amphibian,
Else if skin cover= feather then label is bird,
Else if skin cover = quil then label is mammal,
Else if skin cover = fur then label is mammal.

4. To get a rule for Height:
First sort the height column, keeping track of the associated class.

```
50.75 51.70 51.92 52.93 53.02 53.23 53.61 53.75 54.19 55.13 55.43 55.53 55.67 56.11
57.07
girl  girl  girl   girl  boy   girl   boy   girl  girl   boy    boy  girl  girl   girl
boy
57.04 57.65 58.00 58.81 58.96
boy    boy   boy    boy    boy
```

Fix the bin size. Assume a bin size of 5. So mark off groups of 5 and make a rule for each group.

```
50.75 51.70 51.92 52.93 53.02|53.23 53.61 53.75 54.19 55.13| 55.43 55.53 55.67 56.11
57.04|
girl girl  girl   girl  boy   |girl  boy    girl  girl  boy | boy   girl  girl  girl
boy  |
57.07 57.65 58.00 58.81 58.96
boy     boy    boy     boy     boy
```

Rule: If height <= 53.23 then girl
else   if height > 53.23 and height <=55.13 then girl
else   if height > 55.13 and height <=57.04 then girl
else   If height > 57.04 then boy

Number of errors: 1+2+2+0 = 5

Repeat for weight. Choose the rule with the smallest rule.
For different bin sizes the rules will be different.

An alternative approach would be to discretize the numeric attributes and then use the OneR algorithm for symbolic attributes.

5. Yes it is possible. Disregarding the difference in similarity between different symbolic values, we could arbitrarily set the distance between any two different symbolic values to be 1, whereas the difference between any two instances whose attribute values are the same is set to 0. Of course heuristics can be made use of to consider such similarities where applicable.

6. Yes there will be a problem applying the nearest neighbour technique to this data as the range of values for the "Age" attribute is significantly larger than the range of values for the "WhiteBloodCell" attribute. The nearest neighbour classifier relies on a measure of distance between data points and assumes that all attributes have the same range of values. If we were to use the data values in this example then the larger "Age" values will be over-emphasised. To fix this problem we need to "normalise" the data values by standardising both sets of attribute values. This is usually done using the following formula:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x_{new} is a new normalised value between [0,1], x is the attribute value to normalise, x_{min} is the smallest value of the attribute range and x_{max} is the largest.

7. There is significant overfitting. The model would not generalize well on unseen data.

8. The classifier has barely learnt anything. A random guess would have got 50%. But there is no overfitting.

9. It is possible, but quite rare. Since instances are allocated to the test set at random, by bad luck many ‘hard’ cases could end up in the test set, particularly if the test set is small.