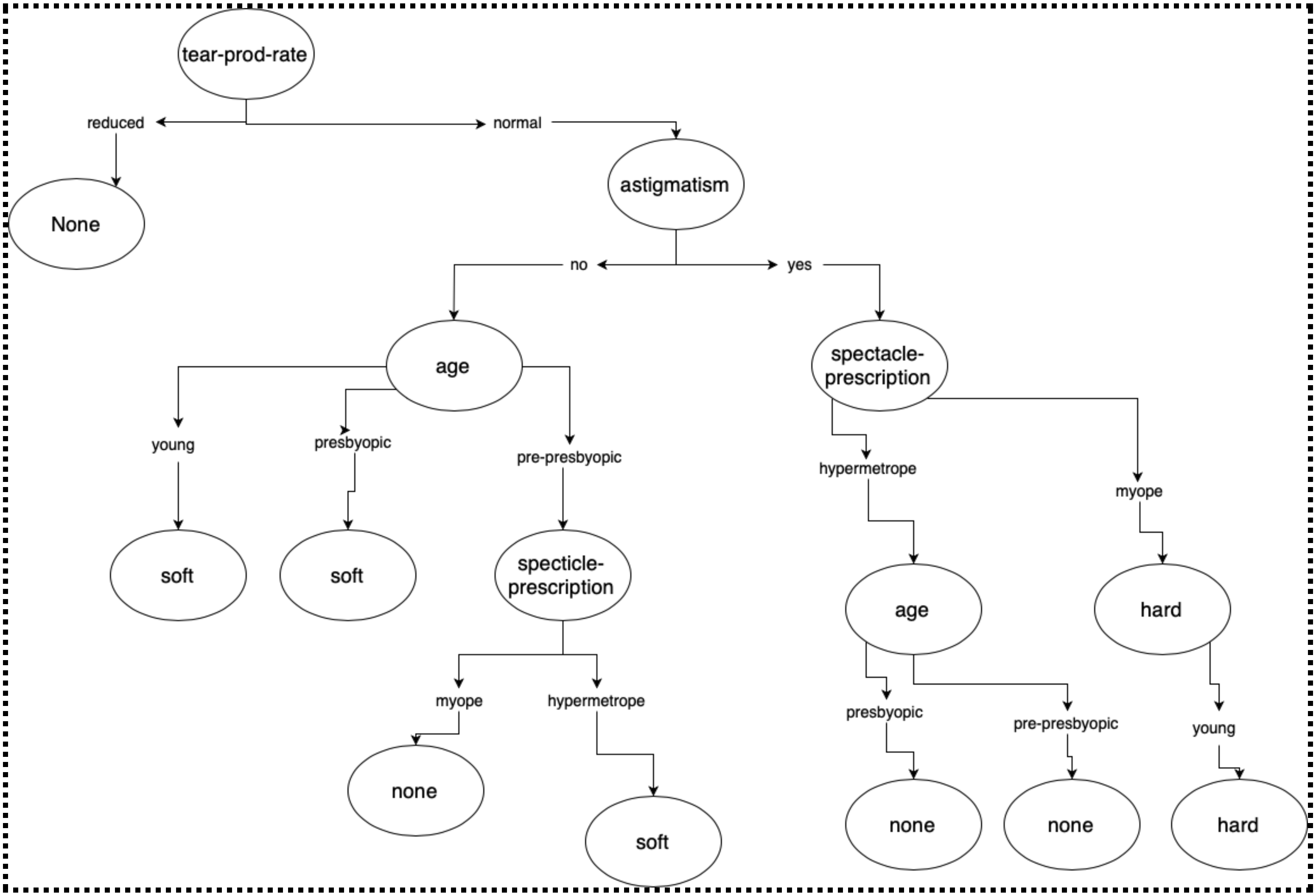
Data Mining - CM3104

Q1) Output from Weka:

```
------------------------------------------------------------------------------------------------------------------------
-----
=== Run information ===

Scheme:       weka.classifiers.trees.Id3
Relation:     contact-lenses
Instances:    24
Attributes:   5
              age
              spectacle-prescrip
              astigmatism
              tear-prod-rate
              contact-lenses
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

Id3


tear-prod-rate = reduced: none
tear-prod-rate = normal
|  astigmatism = no
|  |  age = young: soft
|  |  age = pre-presbyopic: soft
|  |  age = presbyopic
|  |  |  spectacle-prescrip = myope: none
|  |  |  spectacle-prescrip = hypermetrope: soft
|  astigmatism = yes
|  |  spectacle-prescrip = myope: hard
|  |  spectacle-prescrip = hypermetrope
|  |  |  age = young: hard
|  |  |  age = pre-presbyopic: none
|  |  |  age = presbyopic: none

Time taken to build model: 0 seconds

=== Evaluation on test set ===
Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances          24              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0      %
Root relative squared error              0      %
Total Number of Instances               24

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     soft
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     hard
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     none
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  0  4  0 |  b = hard
  0  0 15 |  c = none
```

c1634427


tear-prod-rate
reduced
normal
None
astigmatism
no
yes
age
spectacle-prescription
young
presbyopic
pre-presbyopic
hypermetrope
myope
soft
soft
specticle-prescription
age
hard
myope
hypermetrope
presbyopic
pre-presbyopic
young
none
soft
none
none
hard

b)
Cross Validation is a way of evaluating the performance of a machine learning algorithm. It is a method which improves on the holdout method. The holdout method is where you use a set amount of the data to use for training but then hold the rest of the data to not be used in training, so it can later be used as a testing set..

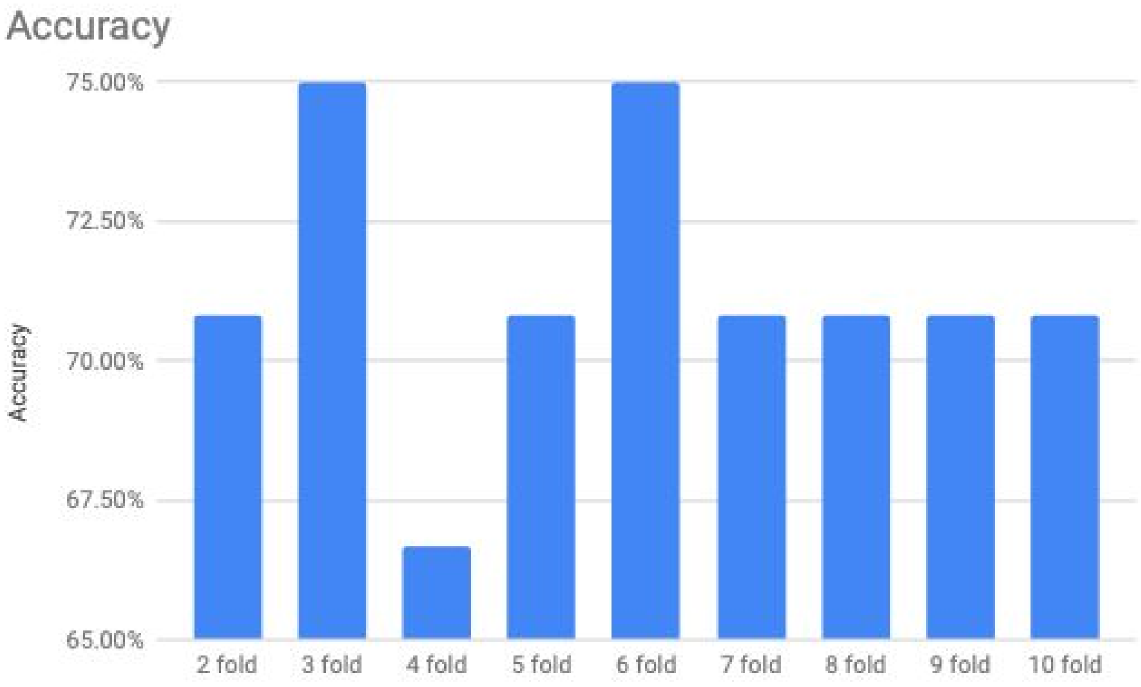
Cross Validation is a way of systematically implementing the holdout method which aims to reduce the variance of the estimation. Cross Validation as it addresses the issue with the holdout method which is data is split into training and testing data so there is data that won't be used for training and data that won't be used for testing where Cross Validation allows all entries to be used for both training and testing

Cross Validation divides the dataset into k bins, it then holds out 1 bin at a time to use for testing, the remaining k-1 bins are used for training. This repeats k times with a different bin used for testing each time. The results of each iteration is then averaged.

Weka uses Stratified Cross Validation which is a further improvement to by ensuring that each fold has approximately the correct proportion of each class value. As we don't have much data it is better to use cross validation as all entries will be used for both training and testing it makes the most use out of the data available. Cross Validation reduces the variance compared to the holdout method, and Stratified Cross Validation reduces the variance further than Cross Validation.

c)

2 fold	17	70.8333%
3 fold	18	75%
4 fold	16	66.6667%
5 fold	17	70.8333%
6 fold	18	75%
7 fold	17	70.8333%
8 fold	17	70.8333%
9 fold	17	70.8333%
10 fold	17	70.8333%



Which method (cross validation or using training set) is better for testing your derived tree and why?

Cross validation is the better method for testing the derived tree as there are only 24 entries so it matters that every data point gets to be in the test set and gets to be in the training set k-1 times to ensure the most options can be tested. By using cross validation the variance of the accuracy is reduced. This gives a more reliable accuracy so you know how good the classifications are.

Also the training set will be biased as the data will have been seen before as that set was used to train the set. In cross validation when there is always a part of the data set which is unseen to the classifier. This allows the data to be tested i

The data above shows that with more folds the variance of the accuracy decreases. Cross Validation is the better method for testing the derived tree as every data entry gets to be in the training set and also the test set where it is an unseen row of data. With the training set, all of the data is used for training and all of the data is used for testing, this gives a biased result where the accuracy will always be 100% accuracy as it is being tested on the same data which it was trained on.

The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch *k* times, which means it takes *k* times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set *k* different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.