EECS 349 PS1, Fall 2014
Hangbin Li, HLL932

Problem 1

S_0: $\{\langle\varnothing, \varnothing, \varnothing, \varnothing, \varnothing\rangle\}$
G_0: $\{\langle ?, ?, ?, ?, ?\rangle\}$

S_1: $\{\langle Japan, Honda, Blue, 1980, Economy\rangle\}$
G_1: $\{\langle ?, ?, ?, ?, ?\rangle\}$

S_2: $\{\langle Japan, Honda, Blue, 1980, Economy\rangle\}$
G_2: $\{\langle ?, Honda, ?, ?, ?\rangle, \langle ?, ?, Blue, ?, ?\rangle, \langle ?, ?, ?, 1980, ?\rangle, \langle ?, ?, ?, ?, Economy\rangle\}$

S_3: $\{\langle Japan, ?, Blue, ?, Economy\rangle\}$
G_3: $\{\langle ?, ?, Blue, ?, ?\rangle \langle ?, ?, ?, ?, Economy\rangle\}$

S_4: $\{\langle Japan, ?, Blue, ?, Economy\rangle\}$
G_4: $\{\langle ?, ?, Blue, ?, ?\rangle, \langle Japan, ?, ?, ?, Economy\rangle\}$

S_5: $\{\langle Japan, ?, ?, ?, Economy\rangle\}$
G_5: $\{\langle Japan, ?, ?, ?, Economy\rangle\}$

Problem 2

A)

Supposing there are n features $f_1, f_2, f_3, \cdots, f_n$. Let the value of person A's feature f_i be $f_i(A)$. Then we define the distance $d_i(A, B)$ of feature i between A and B as following.
For every nominal feature,

$$d_i(A,B) = \begin{cases} 0 & if \;\; f_i(A) = f_i(B) \\ 1 & if \;\; f_i(A) \neq f_i(B) \end{cases}$$

For example feature "city", if feature city $= \{Evanston, Skokie, Aurora\}$, $f_i(A) = Evanston$ and $f_i(B) = Skokie$, then $d_i(A, B) = 1$.
For numerical features, directly scale them linearly into real number in range [0,1] with minimum value in database to be 0 and maximum value in database to be 1. $d_i(A, B) = abs(f_i(A) - f_i(B))$. For example, if the minimum and maximum ages are 5 and 95 in the database, and A and B are of age 20 and 25, then $d_i(A, B) = Abs((20 - 25)/(95 - 5)) = 0.056$.

Then, we can define

$$d(A,B) = \sum_{i=1}^{n} d_i(A,B)$$

Assertion: The measurement above is a metric.

Proof. $d_i^2(A, B) \geq 0$, so $d(A, B) \geq 0$.

$d(A, B) = 0 \iff \forall i, d_i(A, B) = 0$, i.e., all features of A and B are the same.

We know $d(A, B) = \sum_{i=1}^{n} d_i(A, B)$ and $d(B, A) = \sum_{i=1}^{n} d_i(B, A)$, since $\forall i, d_i(A, B) = d_i(B, A)$ according to the definition above, $d(A, B) = d(B, A)$.

For nominal features, if $f_i(A) = f_i(B)$ and $f_i(B) = f_i(C)$, then $f_i(A) = f_i(C)$, so $d_i(A, B) + d_i(B, C) = d_i(A, C) = 0$, $d_i(A, B) + d_i(B, C) \geq d_i(A, C)$ holds.
If $f_i(A) \neq f_i(B)$ or $f_i(B) \neq f_i(C)$, then $d_i(A, B) + d_i(B, C) \geq 1$ and $d_i(A, C) \leq 1$, so $d_i(A, B) + d_i(B, C) \geq d_i(A, C)$.

For numerical features, we know that since it's one dimensional, $d_i(A, C) = Abs(d_i(A, B) - d_i(B, C))$ or $d_i(A, C) = d_i(A, B) + d_i(B, C)$, so $d_i(A, B) + d_i(B, C) \geq d_i(A, C)$ holds.

Since $\forall i, d_i(A, B) + d_i(B, C) \geq d_i(A, C)$, $d(A, B) + d(B, C) \geq d(A, C)$ is true.

To conclude, this measurement is a metric. □

B)

The height is usually in range $(1, 8)$ in foot, if we take infant into consideration .
The weight is usually in range $(1, 200)$ in kg, also, infant is considered.
The number of hairs is usually in range $(0, 10^6)$.

No, it doesn't make much sense. Because those features varies a lot in scale, thus the distance cannot be measured properly using Euclidean distance if the numerical values are equally treated. Also, the relation between features are hard to justify and not as representative since the feature height and weight resembles each other a lot, and hair count can be manually altered by will regardless of age, thus not a good feature.

Designed metric: first, scale the values of height and weight into the same range linearly, and values of hair count into the same range exponentially, then calculate distance in Eulidean 3-space distance.
For those with the lower half range of height and weight, they belong to "child".
As the weight and height increases, it is more likely to be in "teen".
For other parts, the area of a little more height and some more weight, and a little smaller hair count, are more likely to be "middle-aged".
While with similar height to "middle-aged", but a little less in weight, and much less in hair count, is most likely to be "elderly".

C)

Define the distance between two strands of DNA as follows. The distance is the minimum numbers of operations (including adding, deleting and altering one character of DNA strand) performed to change one strand of DNA into the other one.

The metric can still be used.

Proof. First, the distance cannot be negative number since the operations is at least 0.
Second, the operations to change from DNA strand A to B, always have a corresponding operation

sequence with same amount of steps to change DNA strand B to A with following rules. If deleting char from A goes to B, then adding char from B will result in A, and vice versa. If altering char from A goes to B, then altering the char of same position of B will result in B. Thus the distance bewteen A and B or B and A are interchangeable.
Third, the steps to change from A to C is always less than or equal to the sum of steps to change from A to B and B to C, and they are equal iff the minimum-step procedure only includes deletion or addition of chars or altering different chars. Thus it is a metric. □

Problem 3

A)

Define size of T as $|T|$. For each instance in training set T, it can have $|L|$ possible result, so for all instances, the combination would be $|L|^{|T|}$.
Thus the size of D_T is $|L|^{|T|}$.

B)

The total number of possible different functions is at most $|L|^{|X|}$. And among all of them, $|L|^{|X|-|T|}$ could become indistinguishable from the real function. So the probability is $|L|^{|X|-|T|}/|L|^{|X|} = \frac{1}{|L|^{|T|}}$. Since in this case $|L| = 2$ and $|T| = 100$, so the probability is $\frac{1}{2^{100}}$.

C)

If the learner has found the hypothesis indistinguishable from the target function based on the training set T, then when expanding this to the set of all possible example X, the number of possible cases in total is $|L|^{|X|-|T|}$, thus the probability that it will still hold is $\frac{1}{|L|^{|X|-|T|}}$.

Problem 4

A)

root
$Entropy(S) = Entropy([+3, -2]) = -0.4\log_2 0.4 - 0.6\log_2 0.6 = 0.97095$

$Entropy(Origin, Japan) = Entropy([+3, -1]) = -0.25\log_2 0.25 - 0.75\log_2 0.75 = 0.81128$
$Entropy(Origin, USA) = Entropy([+0, -0]) = 0$
$InfoGain(Origin) = Entropy(S) - 0.8Entropy(Origin, Japan) - 0.2Entropy(Origin, USA) = 0.32193$

$Entropy(Manufacturer, Honda) = Entropy([+2, -0]) = 0$
$Entropy(Manufacturer, Toyota) = Entropy([+1, -1]) = -0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$
$Entropy(Manufacturer, Chrysler) = Entropy([+0, -1]) = 0$
$InfoGain(Manufacturer) = Entropy(S)-0.4Entropy(Manufacturer, Honda)-0.4Entropy(Manufacturer, To$
$0.2Entropy(Manufacturer, Chrysler) = 0.57075$

$Entropy(Decade, 1980) = Entropy([+2, -1]) = -\frac{1}{3}\log_2 \frac{1}{3} - \frac{2}{3}\log_2 \frac{2}{3} = 0.91830$
$Entropy(Decade, 1970) = Entropy([+0, -1]) = 0$

$Entropy(Decade, 1990) = Entropy([+1, -0]) = 0$
$InfoGain(Decade) = Entropy(S) - 0.6Entropy(Decade, 1980) - 0.2Entropy(Decade, 1970) - 0.2Entropy(Decade, 1990) = 0.41997$

$Entropy(Color, Blue) = Entropy([+2, -0]) = 0$
$Entropy(Color, Green) = Entropy([+0, -1]) = 0$
$Entropy(Color, Red) = Entropy([+0, -1]) = 0$
$Entropy(Color, White) = Entropy([+1, -0]) = 0$
$InfoGain(Color) = Entropy(S) - 0.4Entropy(Color, Blue) - 0.2Entropy(Color, Green) - 0.2Entropy(Color, Blue$
$0.2Entropy(Color, While) = 0.97075$

$Entropy(Type, Economy) = Entropy([+3, -1]) = -0.75 \log_2 0.75 - 0.25 \log_2 0.25 = 0.81128$
$Entropy(Type, Sports) = Entropy([+0, -1]) = 0$
$InfoGain(Type) = Entropy(S) - 0.8Entropy(Type, Economy) - 0.2Entropy(Type, Sports) = 0.32193$

$InfoGain(Color)$ is the largest, so choose Color to be the root.

Children of root *Color*:
child *Blue*: all 2 instances under this node are classified as 1.
child *Green*: all 1 instances under this node are classified as 0.
child *Red*: all 1 instances under this node are classified as 0.
child *White*: all 1 instances under this node are classified as 1.

B)

The logic function that captures "*JapaneseEconomyCar*" is $(Color = Blue) \vee (Color = White)$. This decision tree cannot capture the concept of "*JapaneseEconomyCar*" because thsi decision tree don't consider *Origin* or *Type*. In order to capture this concept, we need to manually select the instances classified by this decision tree and combine all of the branches into the logic function. So it's best the samples are large enough in order to eliminate possible overfitting.

Problem 5

A)

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

Id3

IsRich  = true
|  GoodLetters  = true
|  |  GoodGrades  = true : true
|  |  GoodGrades  = false
|  |  |  GoodSAT  = true : true
|  |  |  GoodSAT  = false : false
|  GoodLetters  = false
|  |  GoodGrades  = true
|  |  |  SchoolActivities = true : true
|  |  |  SchoolActivities = false : false
|  |  GoodGrades  = false : false
IsRich  = false
|  HasScholarship  = true
|  |  GoodSAT  = true
|  |  |  GoodLetters  = true : true
|  |  |  GoodLetters  = false : false
|  |  GoodSAT  = false : false
|  HasScholarship  = false : false

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          60               96.7742 %
Incorrectly Classified Instances         2                3.2258 %
Kappa statistic                          0.9355
Mean absolute error                      0.0323
Root mean squared error                  0.1796
Relative absolute error                  6.448  %
Root relative squared error             35.8995 %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.065      0.939     1         0.969      0.968    true
                 0.935     0          1         0.935     0.967      0.968
                    false
Weighted Avg.    0.968     0.032      0.97      0.968     0.968      0.968

=== Confusion Matrix ===

  a  b   <-- classified as
 31  0 |  a = true
  2 29 |  b = false
```

Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. In this case, 31 true instances are classified as true, 2 false instances are classified as true, and 29 false instances are classified as false.

B)

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

J48 pruned tree
------------------

IsRich  = true
|   GoodLetters  = true : true (25.0/1.0)
|   GoodLetters  = false
|   |   GoodGrades  = true
|   |   |   SchoolActivities = true : true (3.0)
|   |   |   SchoolActivities = false : false (4.0)
|   |   GoodGrades  = false : false (6.0)
IsRich  = false
|   HasScholarship  = true
|   |   GoodSAT  = true : true (5.0/1.0)
|   |   GoodSAT  = false : false (6.0)
|   HasScholarship  = false : false (13.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          59               95.1613 %
Incorrectly Classified Instances         3                4.8387 %
Kappa statistic                          0.9032
Mean absolute error                      0.0715
Root mean squared error                  0.2174
Relative absolute error                 14.2982 %
Root relative squared error             43.46   %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.968     0.065      0.938     0.968     0.952      0.942    true
                 0.935     0.032      0.967     0.935     0.951      0.942
                    false
Weighted Avg.    0.952     0.048      0.952     0.952     0.952      0.942

=== Confusion Matrix ===

  a  b   <-- classified as
 30  1 |  a = true
  2 29 |  b = false
```

No, C4.5 didn't outperform ID3 in this case.
Usually C4.5 outperforms ID3 when the data is noisier, i.e., often data collected from realistic environment. C4.5 uses prune to filter those noise in order to avoid overfitting, which leads to a better result compared with ID3 in those cases. While in this case, the data is ideal and small, and doesn't have those unfitted noise, thus ID3 can perform well compared to C4.5 in this case.

C)

```
=== Summary ===

Correctly Classified Instances          34               53.125  %
Incorrectly Classified Instances        30               46.875  %
Kappa statistic                         -0.0169
Mean absolute error                      0.4688
Root mean squared error                  0.6847
Relative absolute error                103.3999 %
Root relative squared error            143.9527 %
Total Number of Instances               64

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
```

```
                 0.364     0.381      0.333     0.364     0.348      0.491    true
                 0.619     0.636      0.65      0.619     0.634      0.491
                     false
Weighted Avg.    0.531     0.549      0.541     0.531     0.536      0.491

=== Confusion Matrix ===

  a  b   <-- classified as
  8 14 |  a = true
 16 26 |  b = false
```

There is a huge difference in performance. ID3 performes a lot better on IvyLeague.txt than on MajorityRule.txt, with accuray of 97% compared to 53%.

The main strength and perhaps the problem of ID3 is relying mainly on entropy to select the best attribute for each node of decision tree. In this case, rhe data in MajorityRule.txt tends to split equally in half in every branch, thus the entropy for every attribute is approximately the same, which is hard for ID3 to choose. Even ID3 is finally able to choose one attribute for a specific node, the result will not be a lot better than choosing other attributes. Meanwhile, data in IveLeague.txt tends to be more random, biased, instead of approximately equally split in each level, which is easier for ID3 algorithm to choose the perfect attribute to split every time.

D)

No, I don't expect Reduced Error Pruning to have a large positive effect on the performance of an ID3 on data MajorityRule.txt.

The method of Reduced Error Pruning is to check if the subtree could be removed based on the accuracy on the validation set. If so, remove the subtree to increase the accuracy. Therefore, the effect still relies on the structure of the subtrees and the decision tree.

In this case, due to the specially structure of instances on data MajorityRule.txt, the decision tree generated by ID3 algorithm will be largely balanced. When performing the pruning, the new result will not be greatly better due to the still balanced data and structure, i.e., still tends to split into two halves of approximately equal size and similar concept distribution. Thus, even though the tree could be simpler after pruning, the result cannot be dramatically better.