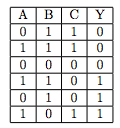
**Instructions**: Please work in your preassigned groups to complete and submit your work to the appropriate folder in LumiNUS. Please submit two documents:

1. Completed word file named as Group-X-Activity-Week-3.docx and
2. Your working ipynb file named as Group-X-Activity-Week-3.ipynb
3. I mentioned in lecture that the number of possible decision trees is very large. How many decision trees exist with n binary attributes? Here is way to think about the problem.

* Suppose you have one binary attribute. Then there are 2^1=2 possible values for the attribute and each of those values can be mapped to 2 outputs, so there are 4 decision trees.
* Suppose you have two binary attributes. Then there are 2^2=4 possible values for the pair of attributes, and each value can be mapped to 2 outputs, so there are 2^4=16 decision trees.
* Now suppose you have n attributes. How many possible decision trees are there? Please justify your answer.

2^(2^n)



1. Consider the following training set with features A, B, C, and target/label Y.
2. What is the entropy of the output Y?

**Solution**: Entropy= 

1. Using the information gain criterion, what is the first node you would split at? Explain clearly why?

**Solution: Information Gain** = Entropy(parent) – [Average Entropy(children)]

Child EntropyA\_0 =1/3\*1.585+1/3\*0.585=0.918

Child EntropyA\_1 =1/3\*1.585+2/3\*0.585=0.918

Child EntropyA =0.5\*0.918+0.5\*0.918=0.918

**Information Gain A = 1-0.918=0.082**

By the same way with B and C, we get

**Information Gain B =0**

**Information Gain C =0.082**

**We must follow the rules that the higher the information gain, the more information that**

**attribute contains about the label, so we can choose either A or C as the first node to split**

1. Using the information gain criterion, complete the learning of the decision tree for this dataset. Draw the decision tree and comment if the tree is unique.

**Solution**: When choosing C as the first node to split

**If c = 0 or 1, child entropy A = 2/3, child entropy B = 0**, **then information gain A = 0.251 < information gain B = 0.918**

**So we choose B as the next node to split**

1. In this problem, we will look at the Breast Cancer Wisconsin (Diagnostic) Data Set available UCI Machine Learning Repository. Please use the wdbc.data dataset from:

https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Compute the performance of the DT algorithm on this dataset for predicting the whether the cancer is malignant or benign. Use a random train/test data split of 70%/30%. Repeat this process 20 times and compute the average performance.

(0.923+0.918+0.900+0.935+0.912+0.953+0.929+0.906+0.918+0.935+0.947+0.929+0.935+0.941+0.923+0.889+0.953+0.900+0.923+0.935)/20=0.923

Please evaluate the following algorithms:

* DT1: DT with Information Gain (IG)
* DT2: DT with IG & limited tree size, vary the number of levels and try to beat DT1

Please compute the following metrics and fil in the table below.

* Training Accuracy and Test Accuracy
* Precision and Recall (which are important metrics that complement Accuracy)
* You can read about performance metrics at: <https://en.wikipedia.org/wiki/Confusion_matrix>
* SKLearn contains functions to compute these metrics:

<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | | Precision | Recall |
|  | Train | Test |  |  |
| DT1 |  |  |  |  |
| DT2 |  |  |  |  |