**Assignment 2**

**Due**: Oct. 23, 11:59PM on Moodle  
TOTAL POINTS = 31 points

Show **all** your work for each question

This assignment is to be completed individually.

|  |  |
| --- | --- |
| 1. Calculate the entropy of a node that contains 20 samples. 5 samples belong to the **yes** class and 15 samples belong to the **no** class.  Ans) There are 20 samples, 5 are yes 15 are no.  Probability of yes = 5/20 = 0.25  Probability of no = 15/20 = 0.75  entropy = -(0.25 log2 0.25 + 0.75 log2 0.75)  = -(-0.5 + -0.311)  = 0.811 (ans) | (1 point) |
| 2. Calculate the mutual information of the attributes **favourite sport** and **favourite colour** in the dataset below.   |  |  |  | | --- | --- | --- | | **ID** | **Favourite sport** | **Favourite Colour** | | 0 | Basketball | Red | | 1 | Basketball | Red | | 2 | Basketball | Red | | 3 | Basketball | Blue | | 4 | Basketball | Blue | | 5 | Football | Red | | 6 | Football | Blue | | 7 | Football | Blue | | 8 | Football | Green | | 9 | Tennis | Red | | 10 | Tennis | Blue | | 11 | Tennis | Blue | | 12 | Tennis | Green | | 13 | Tennis | Green | | 14 | Tennis | Green | | 15 | Tennis | Green |     Ans) first we calculate individual probability of favourite sport and favourite color.  For favourite sport:  Basketball: 5/16 probability  Football: 4/16 probability  Tennis: 7/16 probability  Entropy for favourite sport is:   * ( log2 log2 log2   = 1.546  favourite color:  Red: 5/16 probability  Blue: 6/16 probability green: 5/16 probability  Entropy for favourite color is:   * ( log2 log2 log2   = 1.561  now we calculate joint probabilities:   |  |  |  |  | | --- | --- | --- | --- | | Favourite sport | red | blue | green | | Basketball | 3/16 | 2/16 | 0 | | Tennis | 1/16 | 2/16 | 4/16 | | football | 1/16 | 2/16 | 1/16 |   Now we calculate joint entropy using this table:  Total number of samples is 16.  Probability of (red, basketball) = 3/16  Probability of (blue, basketball) = 2/16  Probability of (red, football) = 1/16  Probability of (blue, football) = 2/16  Probability of (red, tennis) = 1/16  Probability of (blue, tennis) = 2/16  Probability of (green, tennis) = 4/16  Probability of (green, football) = 1/16  To calculate joint entropy, we use formula = -  After we substitute every joint probability, we sum the terms and the joint entropy calculated after using the formula would be 2.828  To finally calculate mutual information, we use the 3 entropies calculated above:  I(X,Y) = H(X) + H(Y) – H(X,Y)  I(sport, color) = 1.54 + 1.561 – 2.828  After substituting, we get joint probability of favourite sport and favourite color to be 0.279 (ans). | (3 points) |
| 3. Given the following data from a node in a growing decision tree, calculate if the node should be split based on **rain** or **temperature**. The dataset is used to determine if a person will go for a run or not. Use Gain Ratio and entropy to determine which attribute that the following node of 9 samples should split on. To clarify, use entropy in the calculation of the Gain.   |  |  |  | | --- | --- | --- | | **Rain** | **Temperature** | **Run?** | | Yes | Low | No | | Yes | Low | No | | Yes | High | Yes | | Yes | Medium | No | | No | Low | Yes | | No | Medium | Yes | | No | Medium | Yes | | No | High | No | | No | High | Yes | |  |  |  |   Ans) We will get the solution by calculating and comparing the gain ratio for each attribute using entropy.  Entropy of the node (run):  Probability of yes: 5/9  Probability of no: 4/9  Entropy = - ( log2  log2  = 0.657  Entropy after splitting on rain:  For rain = yes, we have 4 samples for run, 1 for yes and 3 for no.  So entropy where rain = yes is:   * ( log2  log2   = 0.811  Similarly, for rain = no, we get:   * ( log2  log2   = 0.722  Weighted entropy for rain:  = 0.761  We perform the same steps after splitting on temperature.  For Temperature = low, we get entropy:  (1 yes 2 no)  - ( log2  log2  = 0.918  For temperature = medium, we get:  (2 yes 1 no)  - ( log2  log2  = 0.918  For temperature = high, we get:  (2 yes 1 no)  = same as medium : 0.918  Weighted entropy for temperature:  =0.918  Calculate split information:  Splitinfo for rain: - ( log2  log2  = 0.990  Splitinfo for temperature: - ( log2  log2 log2  =0.918  Gain ratio:  Gain ratio for rain: gain(rain) / splitinfo (rain) = -0.105  Gain ratio for temperature = gain(temperature) / splitinfo(temp) = -0.284  If we had to choose between rain and temperature, we will pick rain as it has a lesser negative gain ratio than temperature (as temperature is more negative, it is worse than rain). | (3 points) |
| 4. Given the following dataset (which is used to determine if a given mammal is a cat or not), determine which value I should split the attribute **weight** on, based on the Gini Index and only using binary splits.   |  |  |  | | --- | --- | --- | | **Height (inches)** | **Weight**  **(lbs)** | **Cat?** | | 22 | 20 | Yes | | 5 | 6 | No | | 100 | 350 | No | | 22 | 40 | Yes | | 42 | 86 | no |   Ans) we want to check what value to split weight on, in sorted order, our weight values are:  6 20 40 86 350  The possible split values are between these weight values:  Split 1: (6+20) / 2 = 13  Split 1: (20+40) / 2 = 30  Split 1: (40+86) / 2 = 63  Split 1: (86+ 350) / 2 = 218  We then calculate gini index for each split:  Gini = 1-2  Where P(i) is the probability of each class.  Split 1:  Weight = 13  Left side (weight <= 13):  Only 1 sample exists which is for no. so P(No) = 1  Gini(left) = 0  Right side (weight > 13):  4 samples, 2 yes 2 no, P(yes) = 2/4 and P(no) = 2/4  Gini(right) = 0.5  Weighted gini for split 1: ((1/5 \* 0) + (4/5 \* 0.5 )) = 0.4  Split 2:  Weight = 30  Left side (weight <= 30):  2 samples, 1 yes 1 no. P(Yes) = ½ and P(No) = 1/2  Gini(left) = 0.5  Right side (weight > 30):  3 samples, 1 yes 2 no, P(yes) = 1/3 and P(no) = 2/3  Gini(right) = 0.444  Weighted gini for split 2: ((2/5 \* 0.5) + (3/5 \* 0.444)) = 0.471  Split 3:  Weight = 63  Left side (weight <= 63):  3 samples, 2 yes 1 no. P(Yes) = 2/3, P(No) = 1/3  Gini(left) = 0.444  Right side (weight > 63):  2 samples, 0 yes 2 no, P(yes) = 0 and P(no) = 1  Gini(right) = 0  Weighted gini for split 3 : ((3/5 \* 0.444) + (2/5 \* 0 )) = 0.266  Split 4:  Weight = 218  Left side (weight <= 218):  4 samples, 2 yes 2 no. P(Yes) = 2/4 , P(No) = 2/4  Gini(left) = 0.5  Right side (weight > 218):  1 Sample. 1 no, P(yes) = 0 and P(no) = 1  Gini(right) = 0  Weighted gini for split 4: ((4/5 \* 0.5) + (1/5 \* 0 )) = 0.4  So gini index split which we calculated is as shown:  Split 1 (weight 13) = 0.4  Split 2(weight 30) = 0.471  Split 3(weight = 63) = 0.266  Split 4 (weight = 218) = 0.4  The lowest gini is for split 3 (weight = 63) so that is our best split as after splitting the resulting groups are as pure as possible. (ans) | (3 points) |
| 5. What is the cosine similarity of the vectors [1,3,6,8] and [2,2,6,0]?  Ans) cosine similarity formula:  where A\*B is the dot product of vectors A and B while ||A|| and ||B|| are the magnitudes of vector A and B respectively.  A\* B = (1\*2) + (3\*2) + (6\*6) + (8\*0) = 44  Magnitude of A = = 10.488 Magnitude of B = = 6.633  cosine similarity = 44/69.52 = 0.633 (ans) | (2 points) |
| 6. What is the Euclidean distance between the vectors:  [1,3,6,8] and [2,2,6,0]  Ans) Euclidean distance for these vectors:  get :  or 8.124 as the answer. | (2 point) |
| 7. What is the Manhattan distance between the vectors:  [1,3,6,8] and [2,2,6,0]  Ans) Manhattan distance = | A1-B1| + |A2-B2| + |A3-B3| + |A4-B4|  = 1 + 1 + 0 + 8  = 10 is the answer. | (2 points) |
| 8. What is the Chebyshev distance between the vectors:  [1,3,6,8] and [2,2,6,0]  Ans) Chebyshev distance = Max( | A1-B1| , |A2-B2| , |A3-B3| , |A4-B4|) = max(1,1,0,8)  = 8 is the answer. | (1 points) |
| 9. What is the dissimilarity of 30 on a scale of [10-50]?  Ans) dissimilarity = (value – min) / (max – min)  = (30-10) / (50-10) = 20-40  = 0.5 is the answer. | (1 point) |
| 10. What is the dissimilarity of 5 on a scale of [0, ∞)  Ans) if scale is [0, ∞), we use another formula for proximity transformation:  Dissimilarity = value / (1+ value)  = 5 / 6  = 0.833 is the answer. | (1 point) |

Code (12 points)

Complete and submit the python script decision\_tree.py that reads in the file zoo\_modified.json. Your code will create a decision tree based on a given impurity metric. The function descriptions are given in the starter code and below:

### entropy(samples):

Returns the entropy the current node, which is given as a list of samples.

### gini(samples):

Returns the Gini Index of the current node, which is given as a list of samples.

### impurity\_of\_children(impurity\_0, impurity\_1, num\_0, num\_1):

Calculates the weighted impurity of children of a candidate split, which are given as arguments. num\_0 is the number of samples in node 0 and num\_1 is the number of samples in num\_1. impurity\_0 and impurity\_1 are the impurities of these nodes.

### get\_most\_frequent\_label(samples)

Return tuple of the most frequent label and the number of times it occurred

### Class DecisionTreeNode

DecisionTreeNode class representd a node that will be added to a decision tree. Each node in your decision tree will be an instance of the class DecisionTreeNode When assigning label to empty node, assign the label of the parent. This class has the following method:

### create\_children(self, impurity\_metric)

Based on a given impurity metric (gini or entropy), create the best binary split and create the two children nodes of the class DecisionTreeNodes. You can use the gini and entropy functions that you create earlier. Update all attributes of the new children nodes. The newly created children should be assigned to this node’s self.leftchild and self.rightchild attributes. Left child is when the value of the split attribute is 0 and right is when the value of the split attribute is 1. Check that all samples don't have the same label. Don't create children if all samples in the node have the same labels. Don’t forget to update the is\_leaf attribute for the parent node as well.

### tree\_classify(root\_node, sample)

Given a root node and a sample, predict the label of the sample. Return the predicted label.

# IMPORTANT for your code!

* To run the code, you will need to run from command line:
  + python .\decision\_tree.py zoo\_modified.json
* I will run the code in the same manner and therefore, it must work this way.
* The code submission must be a .py file and you don’t need to submit the data.csv file.
* You may **NOT** import any additional libraries or packages that are not already imported with the starter code.
* The only function from the math library that you can use is the log2 function.
* If you import additional libraries, your code will automatically be marked out of 50% and all code that uses the library will be marked as incorrect.
* Any attempt to modify the declaration of functions, such as the parameters that it uses, will be marked as incorrect.
* Any modification to the test code (all code below the line ###TESTS) will result in a **0** on the entire coding assignment.
* Code that can’t be run due to a syntax error will be marked out of 50%. If you are unable to get a function working, have it return a 0 for all expected int values and “none” for all expected string values. This should make your code not give a syntax error and not be marked out of 50% because of it.
* Your submitted code must **NOT** contain any additional print statements than what was given in the starter code.
* Have fun with the assignment and the data!