**CPSC 483 PROJECT 1 100 points**

**DUE DATE: Nov 27, 11:59 PM**

**Additionally if you submit the complete project by original deadline, viz. Nov 20, 11:59 PM, you will get *extra credit* (undergrads only).**

**THIS IS AN INDIVIDUAL PROJECT, NOT A GROUP PROJECT.**

PACKAGES LIKE decision-tree-id3 (python), data.tree (R) or any other such package **ARE NOT TO BE USED.**

Helper packages like numpy, math or any other such package for data storage and manipulation and facilitating log operations, etc., even drawing the final tree **ARE TOTALLY ALLOWED.**

**The idea is to CODE the ID3 algorithm yourself, by duly calculating the entropy and information gain at each step/stage; selecting the features for successive nodes, etc. and also recursively (or iteratively, whatever your logic might be) calling this ID3 algorithm.**

You are coding the ID3 algorithm, not just tailoring it to this particular dataset. It is quite possible you may not be able to take care of “all” possibilities. The idea is that the code should work for “most” “different” datasets, not just this dataset. **Specifically**, **In addition to the data set given below, your program should run successfully on the Outlook, Humidity, Wind, etc. example dataset on slide 5 in decision trees slides on Titanium.**

The table below shows a training set provided by politicians intending to solve problems with U.S. health care insurance.

Our interest is apolitical, we merely want to determine the ID3 decision tree.

|  |  |  |  |
| --- | --- | --- | --- |
| **HAS a JOB** | **HAS an INSURANCE** | **VOTES** | **ACTION** |
| yes | yes | yes | leave-alone |
| yes | no | yes | leave-alone |
| yes | no | no | force-into |
| no | no | yes | leave-alone |
| no | no | no | force-into |
| yes | yes | yes | leave-alone |
| yes | no | yes | leave-alone |
| yes | no | no | force-into |
| no | no | yes | leave-alone |
| no | no | no | force-into |

1. Derive the ID3 tree. ID3 algorithm was presented and demonstrated with example in class slides and is also summarized below for reference. **Show complete working: calculations, formulas with relevant values plugged in (at every step), intermediate results for values of entropy, and information gain at every step.** Write a program to calculate the above, and show the results in the output. The program should calculate and **generate** the complete working (described in red font above) in the output, clearly identified. Submit the program and output.

**Output**

HAS a JOB HAS an INSURANCE VOTES ACTION

0 yes yes yes leave-alone

1 yes no yes leave-alone

2 yes no no force-into

3 no no yes leave-alone

4 no no no force-into

5 yes yes yes leave-alone

6 yes no yes leave-alone

7 yes no no force-into

8 no no yes leave-alone

9 no no no force-into

-( 4 / 10 ) \* log\_2( 4 / 10 ) = 0.5287712379549449

-( 6 / 10 ) \* log\_2( 6 / 10 ) = 0.9709505944546686

Entropy = 0.9709505944546686 for HAS a JOB HAS an INSURANCE VOTES ACTION

0 yes yes yes leave-alone

1 yes no yes leave-alone

2 yes no no force-into

3 no no yes leave-alone

4 no no no force-into

5 yes yes yes leave-alone

6 yes no yes leave-alone

7 yes no no force-into

8 no no yes leave-alone

9 no no no force-into With attribute

HAS a JOB

-( 2 / 4 ) \* log\_2( 2 / 4 ) = 0.5

-( 2 / 4 ) \* log\_2( 2 / 4 ) = 1.0

-( 2 / 6 ) \* log\_2( 2 / 6 ) = 0.5283208335737187

-( 4 / 6 ) \* log\_2( 4 / 6 ) = 0.9182958340544896

weighted entropy is 0.9509775004326937

Information Gaim is 0.9709505944546686 - 0.9509775004326937 = 0.01997309402197489

Information Gain is: HAS a JOB ACTION 0.01997309402197489

-( 4 / 10 ) \* log\_2( 4 / 10 ) = 0.5287712379549449

-( 6 / 10 ) \* log\_2( 6 / 10 ) = 0.9709505944546686

Entropy = 0.9709505944546686 for HAS a JOB HAS an INSURANCE VOTES ACTION

0 yes yes yes leave-alone

1 yes no yes leave-alone

2 yes no no force-into

3 no no yes leave-alone

4 no no no force-into

5 yes yes yes leave-alone

6 yes no yes leave-alone

7 yes no no force-into

8 no no yes leave-alone

9 no no no force-into With attribute

HAS an INSURANCE

-( 4 / 8 ) \* log\_2( 4 / 8 ) = 0.5

-( 4 / 8 ) \* log\_2( 4 / 8 ) = 1.0

-( 2 / 2 ) \* log\_2( 2 / 2 ) = 0.0

weighted entropy is 0.8

Information Gaim is 0.9709505944546686 - 0.8 = 0.17095059445466854

Information Gain is: HAS an INSURANCE ACTION 0.17095059445466854

-( 4 / 10 ) \* log\_2( 4 / 10 ) = 0.5287712379549449

-( 6 / 10 ) \* log\_2( 6 / 10 ) = 0.9709505944546686

Entropy = 0.9709505944546686 for HAS a JOB HAS an INSURANCE VOTES ACTION

0 yes yes yes leave-alone

1 yes no yes leave-alone

2 yes no no force-into

3 no no yes leave-alone

4 no no no force-into

5 yes yes yes leave-alone

6 yes no yes leave-alone

7 yes no no force-into

8 no no yes leave-alone

9 no no no force-into With attribute

VOTES

-( 4 / 4 ) \* log\_2( 4 / 4 ) = 0.0

-( 6 / 6 ) \* log\_2( 6 / 6 ) = 0.0

weighted entropy is 0.0

Information Gaim is 0.9709505944546686 - 0.0 = 0.9709505944546686

Information Gain is: VOTES ACTION 0.9709505944546686

Best info gain = VOTES With the value of

0.9709505944546686

We have a pure subset. Leaf Node is creates

VOTES for ACTION

0 yes

1 yes

3 yes

5 yes

6 yes

8 yes

Name: VOTES, dtype: object this is the data values

We have a pure subset. Leaf Node is creates

VOTES for ACTION

2 no

4 no

7 no

9 no

Name: VOTES, dtype: object this is the data values

1. Draw the tree. This can be done in MS Word or any other similar program, after you have derived it in step 1) above.

VOTES

Yes No

Leave Alone

Force Into

1. What will the decision tree result, for the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **HAS a JOB** | **HAS an INSURANCE** | **VOTES** | **ACTION** |
| no | yes | yes | Leave Alone |
| yes | no | yes | Leave Alone |
| no | yes | no | Force Into |

Submit item 1), 2) and 3) above, in Titanium by **Nov 27, 11:59 PM**.

**Summary of ID3 algorithm:**

1. Calculate the entropy of every [attribute](https://en.wikipedia.org/wiki/Feature_(machine_learning)) a {\displaystyle a} ***a*** of the data set ***S***.
2. Partition ("split") the set, ***S*** into subsets using the attribute for which the resulting entropy after splitting is minimized; or, equivalently, information gain is maximum.
3. Make a decision tree node containing that attribute.
4. Perform the steps recursively on subsets using the remaining attributes, adding branches and connecting nodes.