Assignment 1 CS 484

Decision Tree Learning

Submission Deadline: 01/30 11:59 pm

Q1.
Inspect the dataset titled lab01_dataset_1.csv which has a mixture of numerical and categorical data. Your task will be to write a function my_ID3( ) which can create a decision tree for the given dataset using the ID3 algorithm. However, before doing that, you will be have to perform some data processing tasks. Here are all the required tasks in order –

1. ID3 cannot handle continuous numerical data. Perform necessary operations to handle all continuous-valued attributes. Do not forget to show the output i.e., the updated dataset after handling continuous-valued attributes. (2 marks)

2. Next, you will have to ensure the newly obtained dataset is optimal and free of errors. Take appropriate actions based on the outcomes.
    - Check if the dataset has any missing values. (1 mark)
    - Check if the dataset has any redundant or repeated input sample. (1 mark)
    - Check if the dataset has any contradicting <input, output> pairs. (1 mark)

3. Your function my_ID3( ) should operate in a manner such that after ever round of decision making, it will output the attributes and its associated gain, with a message stating “Attribute X with Gain = Y is chosen as the decision attribute”. Once your function completes, it should output the decision tree. The representation of the decision tree is upto you. You can choose either a textual representation or a graphical one; either is fine. (10 marks)

======> run python3 Q1.py in terminal

press “1” to updated the data

Press “2” to run the my_ID3 Algo

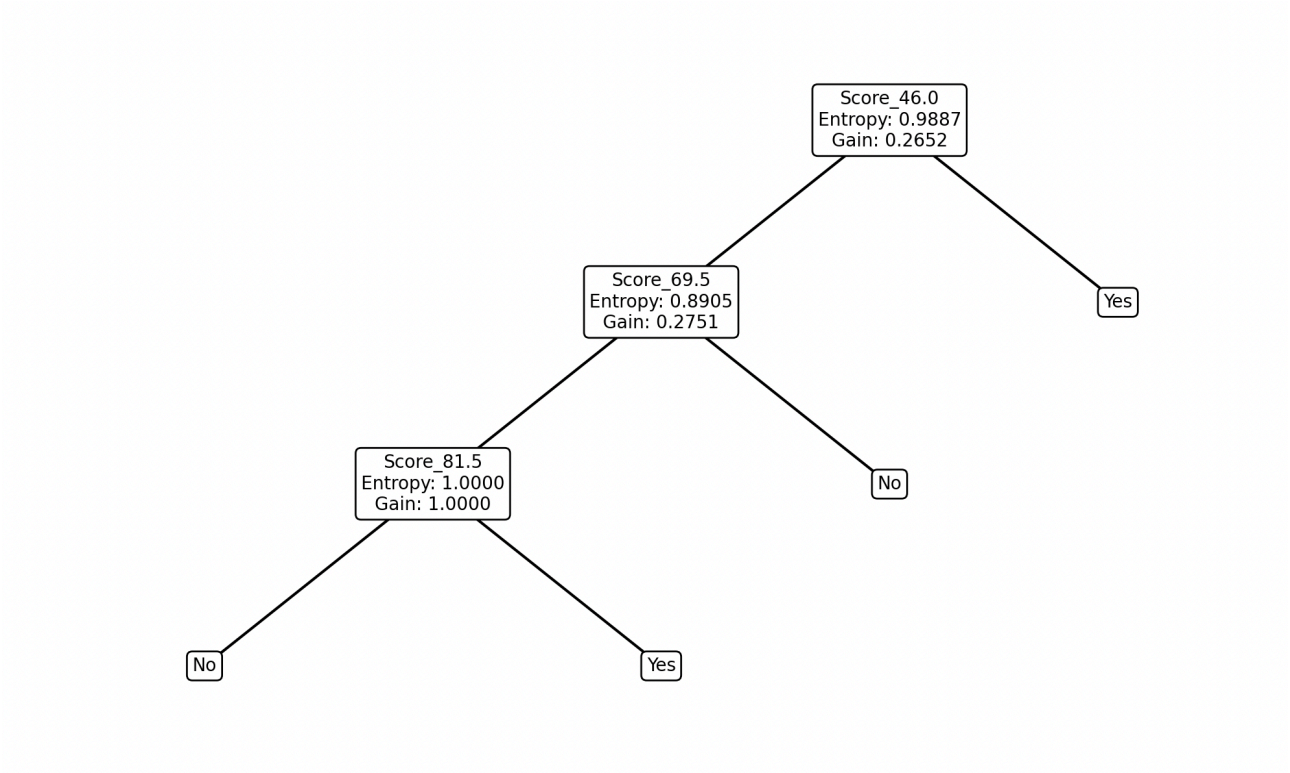
Press “q” to exit the program

=====> Output

```
Press '1' to update data, '2' to run the my_ID3, or 'q' to quit: 1
Dataset has redundant or repeated input samples.
Duplicate rows:
      Mood  Effort  Score_46.0  Score_69.5  Score_81.5 Output
9    Happy     Low        True        True        True    Yes
5  Neutral    High       False        True        True     No
1    Happy  Medium       False       False       False     No
Duplicate rows removed.
Modified and cleaned dataset saved to lab01_dataset_1_updated.csv.
Press '1' to update data, '2' to run the my_ID3, or 'q' to quit:
```

Mood	Effort	Score_46.0	Score_69.5	Score_81.5	Output
Happy	Low	True	True	True	Yes
Happy	High	True	True	True	Yes
Sad	Low	True	True	True	Yes
Neutral	Medium	False	True	True	No
Neutral	High	False	True	True	No
Happy	High	False	True	True	No
Happy	Low	False	True	True	No
Sad	Low	False	True	True	No
Sad	Medium	False	False	True	Yes
Sad	High	False	False	True	Yes
Neutral	Medium	False	False	True	Yes
Sad	Low	False	False	True	Yes
Neutral	Low	False	False	False	No
Happy	Medium	False	False	False	No
Sad	High	False	False	False	No
Sad	Medium	False	False	False	No

```
Press '1' to update data, '2' to run the my_ID3, or 'q' to quit: 2
Attribute Score_46.0 with Gain = 0.2652 and Entropy = 0.9887 is chosen as the decision attribute.
Attribute Score_69.5 with Gain = 0.2751 and Entropy = 0.8905 is chosen as the decision attribute.
Attribute Score_81.5 with Gain = 1.0000 and Entropy = 1.0000 is chosen as the decision attribute.
2024-01-30 11:10:22.130 Python[46677:2087693] WARNING: Secure coding is not enabled for restorable s
orableState: and returning YES.
```



Q2.Inspect the dataset titled lab01_dataset_2.csv which also has a mixture of numerical and categorical data. For this problem, you will use decision tree classifiers for supervised learning. In particular, you will be using the functionalities of the sklearn.tree library. The classification task using sklearn libraries work only on numerical-valued attributes, and not on categorical ones. (What to do now? Hint: Look up One-hot Encoding and Integer Encoding). Here are all the required tasks –

1. Restructure the dataset such that it has all numerical-valued attributes. (2 marks)

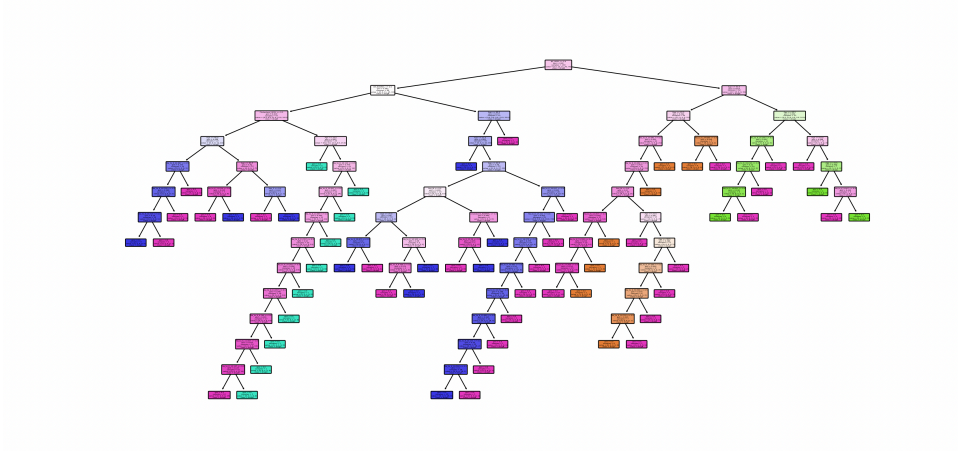
2. Perform supervised learning using decision tree classifiers. Employ the train-test split approach during the learning. (4 marks)

3. After the learning is complete, show the results by predicting the class of the test set. Display the results of the prediction and test set side-by-side. (2 marks)

4. Output the decision tree; it can be either a textual representation or a graphical representation. (2 marks)

=====> Run python3 Q2.py in terminal

=====> output



```
aniruddhkapileshwari@Aniruddhs-MacBook-Ai
     Actual Predicted
95   drugX     drugX
15   drugY     drugY
30   drugX     drugX
158  drugC     drugY
128  drugY     drugY
115  drugY     drugB
69   drugY     drugY
170  drugX     drugX
174  drugA     drugY
45   drugX     drugX
66   drugA     drugA
182  drugX     drugX
165  drugY     drugY
78   drugA     drugY
186  drugB     drugB
177  drugY     drugY
56   drugB     drugY
152  drugX     drugX
82   drugC     drugY
68   drugY     drugX
124  drugB     drugB
16   drugX     drugY
148  drugX     drugY
93   drugY     drugX
65   drugY     drugX
60   drugY     drugX
84   drugC     drugY
67   drugX     drugX
125  drugY     drugY
132  drugX     drugY
9    drugY     drugX
18   drugC     drugC
55   drugC     drugY
75   drugY     drugX
150  drugA     drugA
104  drugY     drugA
135  drugX     drugY
137  drugA     drugY
164  drugY     drugY
76   drugA     drugY
Accuracy: 0.475
```