PA 2 Decision trees report

Sai Charan Thannir - 1001635048

Jahnavi Nuthalapati - 1001827251

1.Decision Tree Methods

A decision tree is a tree-like model of decisions and possible consequences, including chance event outcomes, resource costs, and utility. A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from the root to leaf represent classification rules.

To build a decision tree, first we need to find a root node from all the attributes. The root node is selected by calculating the information gain of each attribute and the attribute with the highest information gain is considered as a root node and the child nodes are also selected based on the information gain. For calculating information gain of an attribute, we can use Entropy or Gini index.

Entropy: Entropy is nothing but the measure of disorder. It aims to reduce the level of entropy, starting from the root node to the leave nodes. Information Gain is used to determining which feature/attribute gives us the maximum information about a class We calculate the Entropy and Information gain of an attribute using the below formulae:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

$$IG(Y,X) = E(Y) - E(Y|X)$$

Gini Index: Gini index measures the probability of a variable being wrongly classified when it is randomly chosen. While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

We calculate the Gini index of an attribute using the below formula:

$$\text{Gini} = 1 - \sum_{i=1}^{n} (p_i)^2$$

Naive Bayes classifier:

A Naive Bayes classifier is a model used for classification task. This classifier is based on the Bayes theorem. Bayes theorem is used to find the probability of an event A occurring given that event B has occurred. We need to find out independent probability of every feature variable in the dataset. And, the target class probability. So, when calculating the final prediction of an event happening or not, we can use the pre-computed data

Bayes theorem is represented as below:

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

2.Dataset Description:

In the provided cardio train dataset there are 13 attributes in total. This data can be used to predict and analyze if a person has cardio disease or not. The following are the attributes or features of the given dataset.

1.**ID :** It is a primary or a unique attribute of the dataset. Each person is given a unique Identification number.

2.**Age:** It is a numeric attribute which is the age of the person.

3.**Gender:** It is used to specify if the person is male or female. It is a categorical data. The possible values for this variable are ['1','2'], where 1&2 are used for ['Female' and 'male'].

4.**Height**: It is a numeric data which specifies the height of a person in inches.

5.**Weight**: It is numeric data which specifies the weight of a person in kilograms.

6. **ap_hi**: This attribute specifies the Systolic blood pressure of a person.

7.**ap_lo**: This attribute specifies the Diastolic blood pressure of a person.

8. **cholesterol:** This attribute is categorical data and the possible values for this feature are ['1,2&3'] where 1,2&3 represent ['normal', 'above normal', 'well above normal'].

9.**gluc:** It specifies the glucose level of a person and is represented using ['1,2&3'] where 1,2&3 represent ['normal', 'above normal', 'well above normal'].

10. **Smoke:** It is used to specify if a person smoke or not. It is represented by ['0','1'] where1,0 indicates ['smokes' or 'not smokes'] respectively.

11.**alco:** It is used to specify if a person drinks alcohol or not. It is represented by ['0','1'] where1,indicates ['drinks' or 'not drinks] respectively.

12. **active:** It is used to indicate if a person is physically active or not.

13. **cardio:** It is a target variable which specifies if a person has a cardiovascular disease or not.

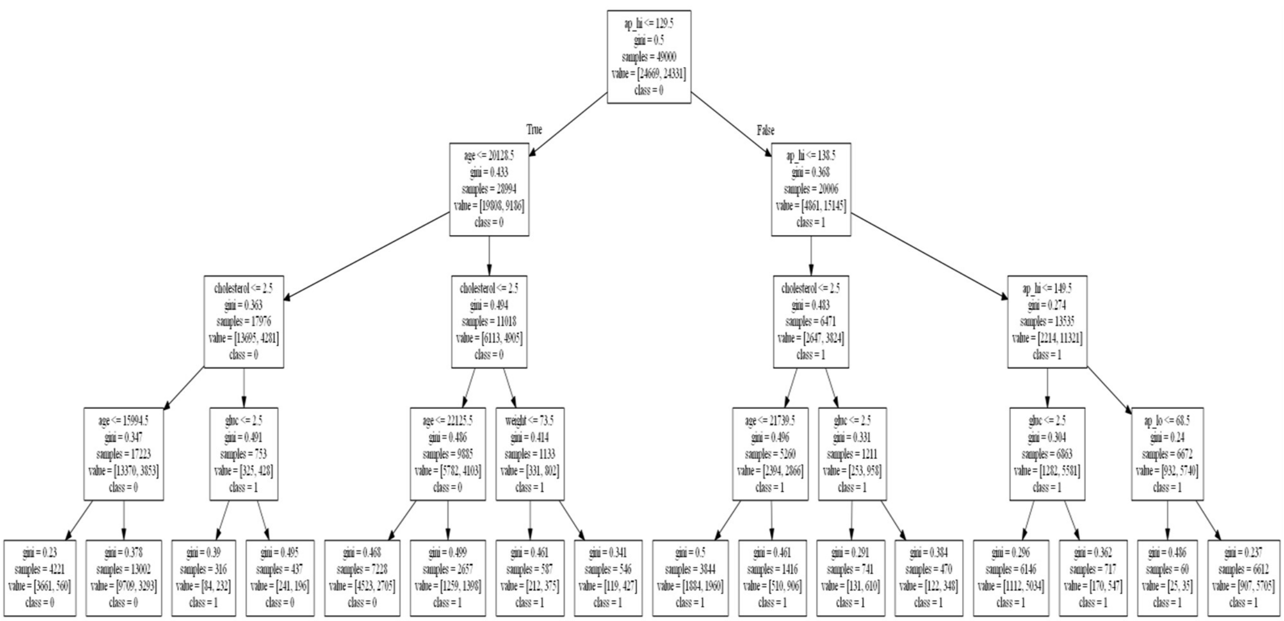
Preprocessing: For the data set, every detail is in the single column but to proceed with the further classification we are required to split the data for the classifier to understand it. Foe this we have split the single column into multiple columns with each attribute in each column by taking the split parameter as **';'**. We have split the dataset into training model and testing model i.e. 70% into training model and 30% into testing model.

For the training data we have applied Gini and entropy classification and got the decision trees.

3. Visualization of the decision tree:

Gini:

Here is the decision tree generated based on Gini index classification. This decision is obtained by calculating Gini index and information gain for each of the attributes in the dataset.



The confusion matrix for this is:

```
The Confusion Matrix is:  [[7587 2765]
 [2897 7751]]
```

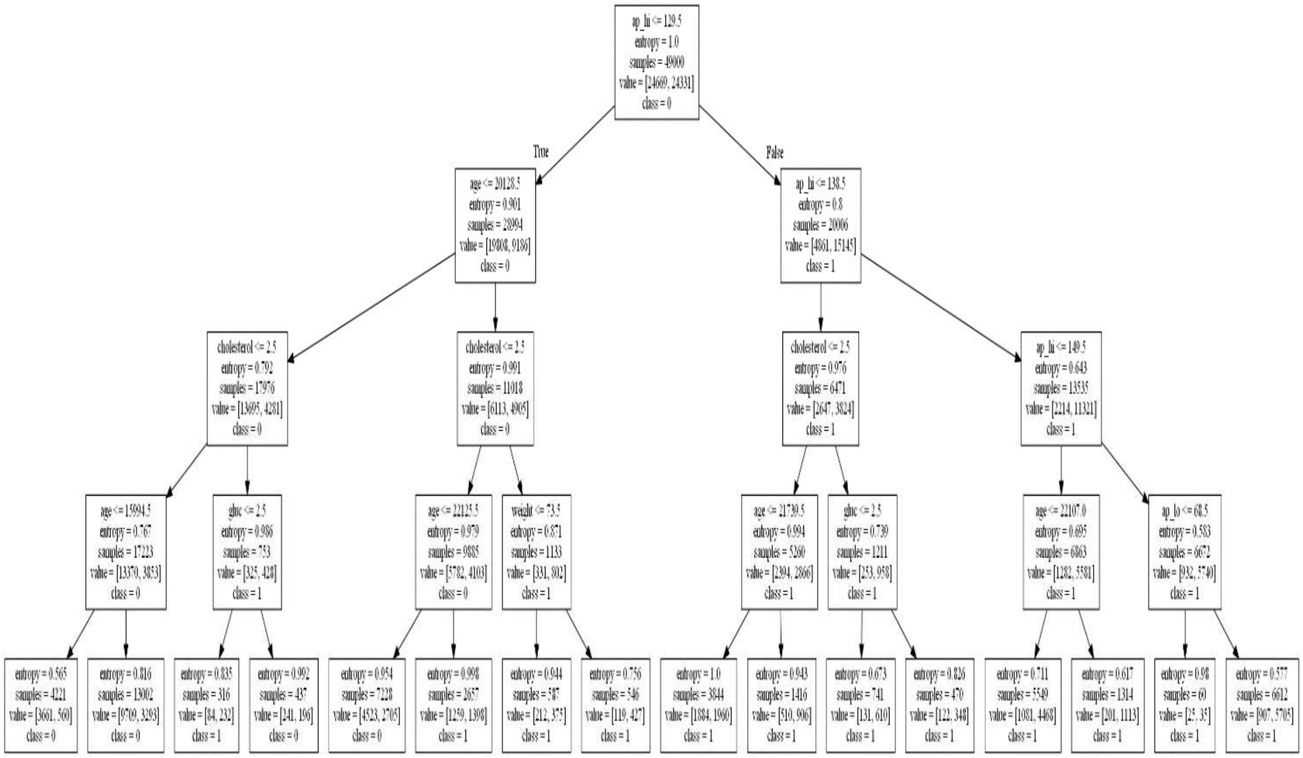
The classification report is:

```
Report :                    precision    recall  f1-score   support

0        0.72       0.73       0.73      10352
1        0.74       0.73       0.73      10648
     accuracy                               0.73
21000    macro avg       0.73       0.73       0.73
21000 weighted avg       0.73       0.73       0.73
21000
```

Entropy:

Here is the decision tree generated based on Entropy classification. This Decision Tree is constructed by using the values of entropy and Miscalculation Error for the attribute



The confusion matrix is:

```
Confusion Matrix is:  [[7587 2765]
 [2897 7751]]
```

The classification report is:

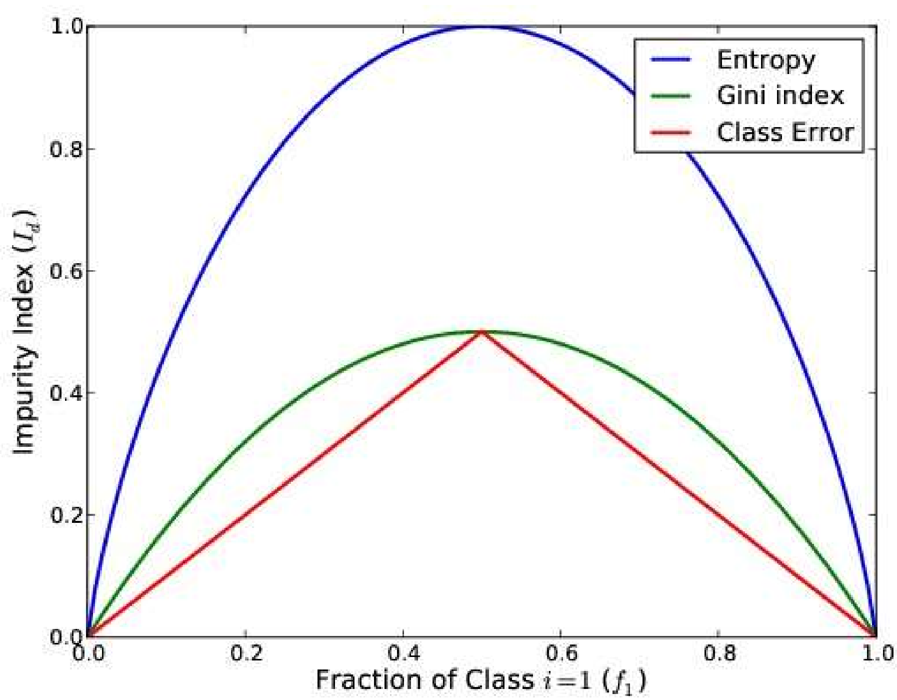
```
Report :                precision    recall  f1-score   support

0       0.72      0.73      0.73     10352
1       0.74      0.73      0.73     10648
     accuracy                          0.73
21000    macro avg       0.73      0.73      0.73
21000 weighted avg       0.73      0.73      0.73
21000
```

4.Interpretation and comparison of results:

The graph below shows that the Gini index and entropy are very similar impurity criterion. The maximum impurity index for entropy is 1 whereas for Gini index it is 0.5. The time taken to compute entropy takes longer when compared to the time taken by GINI index is because of the logarithmic function in entropy.

Gini shows that the data point that we have randomly chosen for the splitting from the dataset how it is incorrectly labelled. It gives a more accurate value and less than the entropy index, which is its best quality, the small values show less impurity. Entropy is more computationally complex because of the log in the equation. It gives larger values that are not much use for the splitting criteria



For the maximum depth which ranges from 1 to 50 the accuracy for both the Gini index and entropy have been calculated as below. From the below values the highest accuracy value is achieved by entropy which is 72.986(approximate) whereas for the Gini it is 72.952(approximate).

```
Accuracy values for the Gini index:

Accuracy :  1 71.52380952380952
Accuracy :  2 71.52380952380952
Accuracy :  3 72.75238095238096
Accuracy :  4 73.03809523809524 Accuracy
:  5 72.8142857142857
Accuracy :  6 72.93809523809523
Accuracy :  7 72.95238095238096 Accuracy
:  8 72.6
Accuracy :  9 72.41428571428571
Accuracy :  10 72.39523809523808
Accuracy :  11 72.15238095238095
Accuracy :  12 71.75714285714285
Accuracy :  13 71.24761904761905
Accuracy :  14 70.86190476190475
Accuracy :  15 70.03809523809524
Accuracy :  16 69.7952380952381
Accuracy :  17 69.06666666666666
Accuracy :  18 68.66190476190475
Accuracy :  19 68.34761904761905
Accuracy :  20 67.79047619047618
Accuracy :  21 67.21904761904763
Accuracy :  22 66.83333333333333
Accuracy :  23 66.23333333333333
Accuracy :  24 66.02380952380953
Accuracy :  25 65.77619047619048
Accuracy :  26 65.28571428571428
Accuracy :  27 64.91428571428571
Accuracy :  28 64.68095238095238
Accuracy :  29 64.3952380952381
Accuracy :  30 64.29047619047618
Accuracy :  31 63.98571428571429
Accuracy :  32 63.94761904761906
Accuracy :  33 63.628571428571426
Accuracy :  34 63.88571428571429
Accuracy :  35 63.714285714285715
Accuracy :  36 63.771428571428565
Accuracy :  37 63.62380952380953
Accuracy :  38 63.49523809523809
Accuracy :  39 63.81428571428571
Accuracy :  40 63.733333333333334
Accuracy :  41 63.60476190476191
Accuracy :  42 63.542857142857144
Accuracy :  43 63.58571428571429
Accuracy :  44 63.62380952380953
```

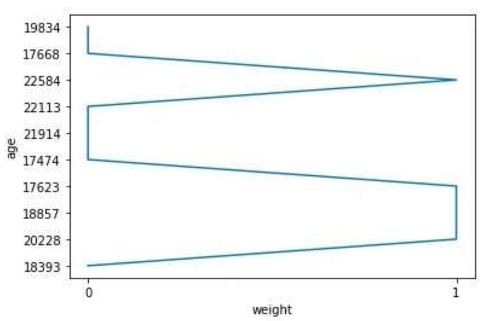
```
Accuracy :  45 63.48095238095238
Accuracy :  46 63.50476190476191
Accuracy :  47 63.50476190476191
Accuracy :  48 63.50476190476191
Accuracy :  49 63.50476190476191
Accuracy :  50 63.50476190476191
Accuracy values for the Entropy:

Accuracy :  1 71.52380952380952
Accuracy :  2 71.52380952380952
Accuracy :  3 72.75238095238096
Accuracy :  4 73.03809523809524
Accuracy :  5 72.82857142857144
Accuracy :  6 72.95238095238096
Accuracy :  7 72.98571428571428
Accuracy :  8 72.73809523809524
Accuracy :  9 72.36666666666667
Accuracy :  10 72.28571428571429
Accuracy :  11 72.05714285714285
Accuracy :  12 71.9952380952381
Accuracy :  13 71.50952380952381
Accuracy :  14 71.11904761904762
Accuracy :  15 70.76190476190476
Accuracy :  16 70.22380952380952
Accuracy :  17 69.7952380952381
Accuracy :  18 69.4095238095238
Accuracy :  19 68.74761904761904
Accuracy :  20 68.75238095238096
Accuracy :  21 68.12857142857143
Accuracy :  22 67.51904761904763
Accuracy :  23 67.13333333333334
Accuracy :  24 66.96190476190476
Accuracy :  25 66.4
Accuracy :  26 66.07142857142857
Accuracy :  27 65.9095238095238
Accuracy :  28 65.56666666666666
Accuracy :  29 65.05238095238096
Accuracy :  30 64.74761904761904
Accuracy :  31 64.74285714285715
Accuracy :  32 64.77142857142857
Accuracy :  33 64.54285714285714
Accuracy :  34 64.23333333333333
Accuracy :  35 64.24285714285715
Accuracy :  36 64.09523809523809
Accuracy :  37 64.24285714285715
Accuracy :  38 63.91428571428571
Accuracy :  39 64.06190476190477
Accuracy :  40 63.85238095238095
Accuracy :  41 63.800000000000004
```

```
Accuracy :  42 63.752380952380946
Accuracy :  43 63.661904761904765
Accuracy :  44 63.82380952380953
Accuracy :  45 63.800000000000004
Accuracy :  46 63.642857142857146
Accuracy :  47 63.81428571428571
Accuracy :  48 63.800000000000004
Accuracy :  49 63.72380952380953
Accuracy :  50 63.8095238095238
```

5.Visualize the dataset, for the target variable

```
Out[28]: Text(0, 0.5, 'age')
```




```
Out[29]: Text(0, 0.5, 'age')
```