Assignment 2 Report

Name: Sai Saketh Aluru
Roll No. 16CS30030

Part -1

a.) Using the given toy dataset and training a decision tree classifier using
I. Information Gain
II. Gini index

The given dataset is:

	price	maintenance	capacity	airbag	profitable
0	low	low	2	no	yes
1	low	med	4	yes	no
2	low	high	4	no	no
3	med	med	4	no	no
4	med	med	4	yes	yes
5	med	high	2	yes	no
6	high	med	4	yes	yes
7	high	high	2	yes	no
8	high	high	5	yes	yes
9	med	high	5	no	yes
10	low	low	4	no	yes

Where the first 0-8 columns are training data and the last two are test data.

The obtained decision tree using information gain as the criteria, printed in the specified format is:

```
maintenance = low : yes
maintenance = high
| capacity = 2 : no
| capacity = 4 : no
| capacity = 5 : yes
maintenance = med
| price    = low : no
| price    = high : yes
| price    = med
          | airbag  = yes : yes
          | airbag  = no : no
```

The class labels for the test data obtained using this decision tree to predict are:

1. data: prediction: yes
   price med
   maintenance high
   capacity 5
   airbag no

2. data: prediction: yes
   price low
   maintenance low
   capacity 4
   airbag no

The accuracy of this prediction, using the given ground truth values is 1.0, or 100%.

ii. The obtained decision tree using Gini index as the splitting criteria is:

```
maintenance = med
| price    = med
        | airbag  = no : no
        | airbag  = yes : yes
| price    = high : yes
| price    = low : no
maintenance = high
| capacity = 2 : no
| capacity = 4 : no
| capacity = 5 : yes
maintenance = low : yes
```

1. data: prediction: yes
   price med
   maintenance high
   capacity 5
   airbag no

2. data: prediction: yes
   price low
   maintenance low
   capacity 4
   airbag no

The accuracy obtained using this classifier on the test data is 1.0, or 100%.

The values of the Information gain and GINI index of the root node using my model is:
i. Information gain - 0.18606356007860825 for the feature maintenance
ii. Gini Index - 0.38888888888888895 for the feature maintenance

The decision trees are then built using scikit learn library using Information gain and Gini index as the criteria. The obtained accuracies on the trees are:

i. Information gain - 1.0 or 100%.
ii. Gini index - 1.0 or 100%.

The predictions on the test data by the scikit learn model are:
Using both information gain and gini index criteria.

1. data: prediction: yes

price	med
maintenance	high
capacity	5
airbag	no

2. data: prediction: yes

price	low
maintenance	low
capacity	4
airbag	no

The value of the information gain and gini index of the root node using scikit learn's model is:

i. Information gain:

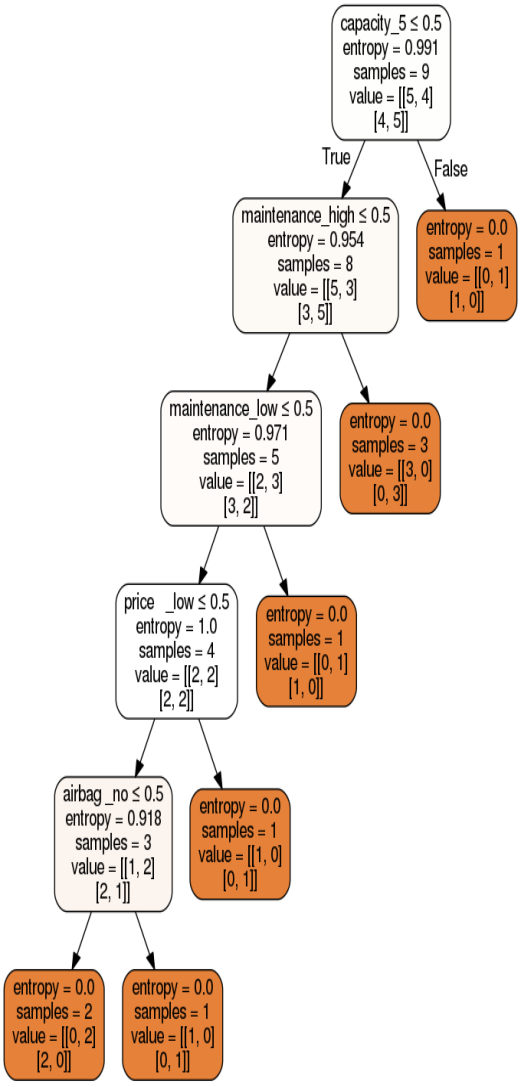
Entropy of parent node	-	0.991	Sample size	-	9
Entropy of child 1	-	0.954	Sample size	-	8
Entropy of child 2	-	0.0	Sample size	-	1

Information gain - 0.991 - 0.954*8/9 = 0.143

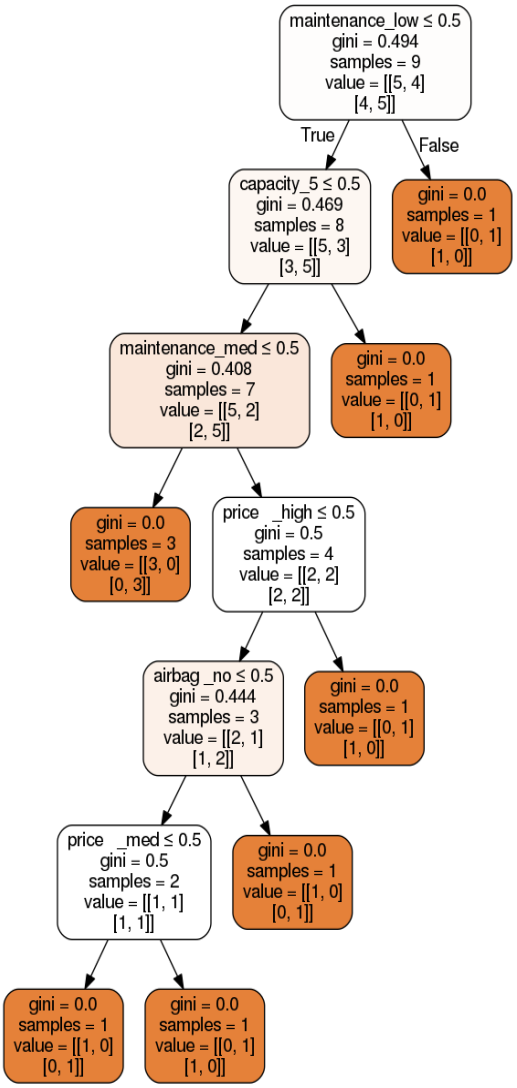
ii. Gini index: 0.49382716049382713

The trees obtained are:

Information gain:



Gini index:



Part - 2

Implementing a decision tree classifier for the simplified 20 newsgroup dataset.

Given data:

	archive	name	atheism	resources	alt	last	modified	december	version	atheist	...	illustrator	dps	toward	eastman	hackers	jennifer	schaertel
0	1	1	1	1	1	1	1	1	1	1	...	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

The accuracies obtained on training and test data with increasing depth of the decision tree is:

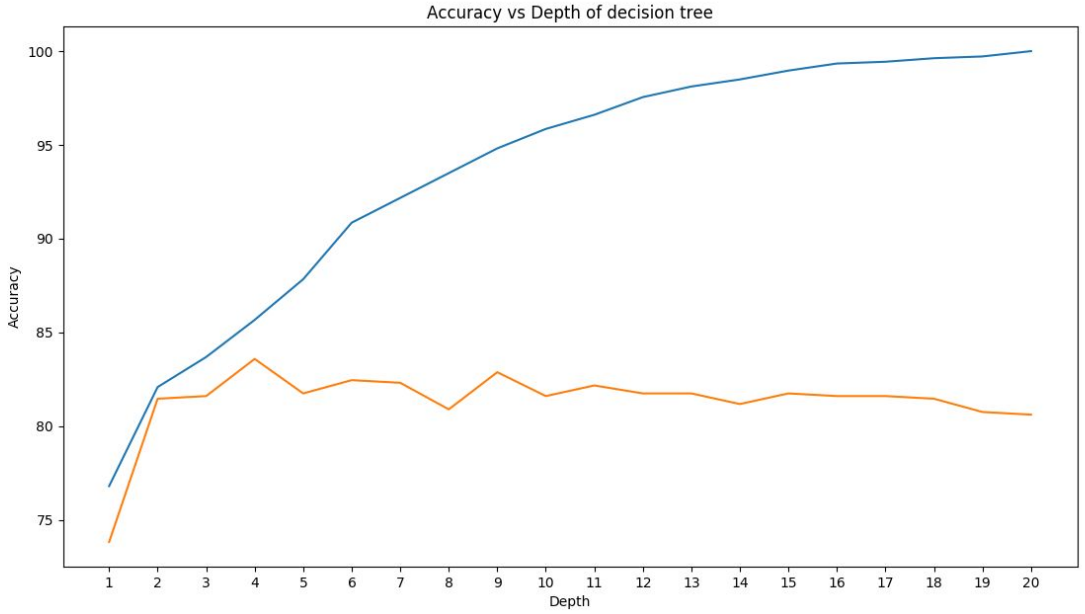
Depth of the tree	Training accuracy	Test accuracy
1	73.83309759547383,	73.83309759547383
2	82.09236569274269	81.47100424328147
3	83.69462770970782	81.61244695898161
4	85.67389255419415	83.5926449787836
5	87.8416588124411	81.75388967468176
6	90.85768143261075	82.46110325318246
7	92.17719132893497	82.31966053748232
8	93.49670122525919	80.90523338048091
9	94.81621112158341	82.88543140028288

10	95.85296889726673	81.61244695898161
11	96.60697455230914	82.17821782178218
12	97.54948162111216	81.75388967468176
13	98.11498586239397	81.75388967468176
14	98.49198868991517	81.18811881188118
15	98.96324222431669	81.75388967468176
16	99.34024505183788	81.61244695898161
17	99.43449575871819	81.61244695898161
18	99.62299717247879	81.47100424328147
19	99.71724787935909	80.76379066478076
20	100.0	80.62234794908062

We observe that the training accuracy increases steadily as the depth increases till a depth of 20 after which we reach the maximum tree size and a fully trained decision tree with training accuracy 100%. But we can see that initially, test accuracy increases with respect to the depth as the classifier is becoming better but after a depth of 4, we can see the test accuracy decreasing or varying unsteadily, which is due to the decision overfitting the data.

The best test accuracy occurs at a height of 4, at which the training accuracy is 85.67% and the test accuracy is 83.59%.

The obtained plot of the training and test accuracies w.r.t the depth of the decision tree is:

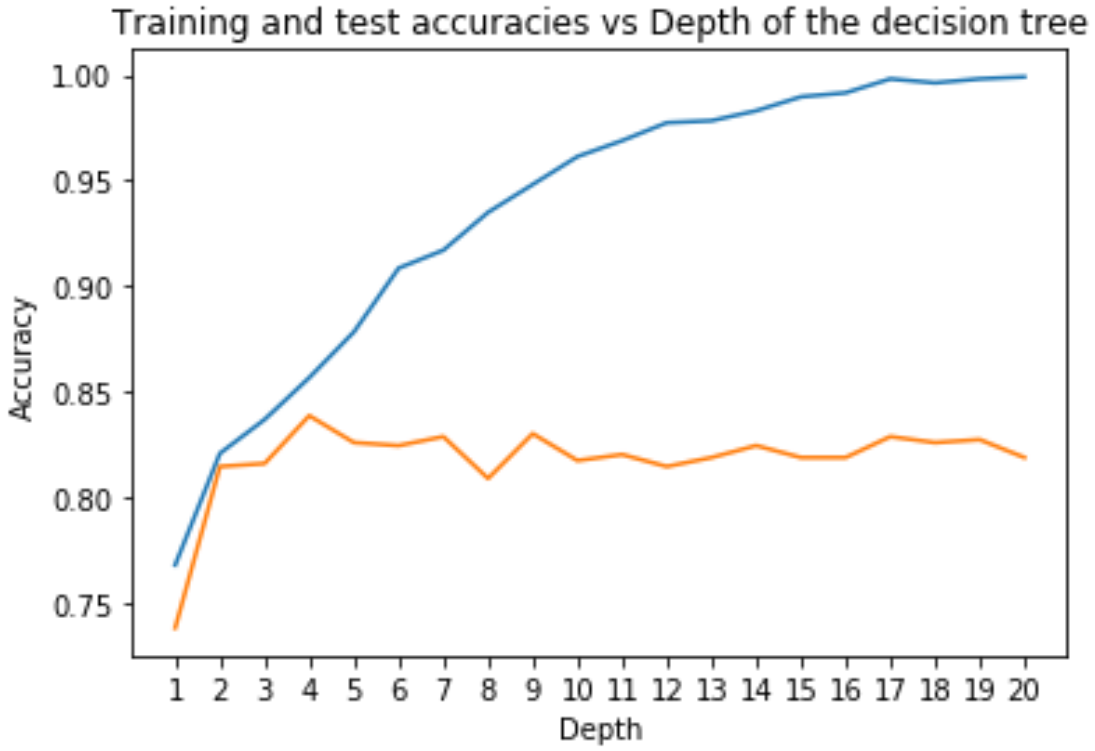


Using scikit-learn's decision trees and Information gain as the split criteria, the obtained accuracy values are:

Depth	Training Accuracy	Test Accuracy
1	76.81432610744581	73.83309759547383
2	82.0923656927427	81.47100424328148
3	83.69462770970783	81.61244695898161
4	85.67389255419415	83.87553041018387
5	87.84165881244109	82.6025459688826
6	90.85768143261075	82.46110325318247
7	91.70593779453345	82.88543140028288
8	93.49670122525919	80.9052333804809
9	94.81621112158342	83.02687411598303
10	95.94721960414703	81.75388967468176
11	96.60697455230914	82.03677510608204
12	97.92648444863337	81.47100424328148
13	97.92648444863337	81.89533239038189
14	98.49198868991518	82.46110325318247
15	98.96324222431668	81.89533239038189
16	99.52874646559849	81.89533239038189
17	99.52874646559849	82.88543140028288
18	99.8114985862394	82.6025459688826
19	99.90574929311969	82.74398868458275
20	100	81.89533239038189

The best accuracy occurs at a height of 4, similar to our decision tree from before. Here as well, we can see that the test accuracy increases first and then decreases, signifying overfitting.

The obtained plot of the training and test accuracies with respect to the depth of the decision tree is:



In the plot as well, we can see that overfitting occurs after a height of 4 as the test accuracy decreases and is varying unsteadily.

The tree formed at height 10 is

```
writes = 0
| god = 0
        | that = 0
                | bible = 0
                        | atheist = 0
                                | keith = 0 : 2
                                | keith = 1 : 1
                        | atheist = 1 : 1
                | bible = 1 : 1
        | that = 1
                | wrote = 0
                        | people = 0
                                | religious = 0 : 2
                                | religious = 1 : 1
                        | people = 1
                                | windows = 0 : 1
                                | windows = 1 : 2
                | wrote = 1
                        | ve = 0
                                | came = 0 : 1
                                | came = 1 : 2
                        | ve = 1 : 2
| god = 1
        | use = 0 : 1
        | use = 1
                | archive = 0 : 2
                | archive = 1 : 1
writes = 1
| graphics = 0
        | image = 0
                | that = 0
                        | god = 0
                                | keith = 0 : 2
                                | keith = 1 : 1
                        | god = 1 : 1
                | that = 1
                        | program = 0
                                | comp = 0 : 1
                                | comp = 1 : 2
                        | program = 1 : 2
        | image = 1 : 2
| graphics = 1 : 2
```

Here, we can see the features that are selected by this tree. Words like ‘atheist’, ‘god’, etc are a good indication of the alt.atheism class and words like ‘image’, ‘graphics’, ‘program’, etc are good indication of comp.graphics class labels. We can also see that some feature words such as ‘that’, ‘ve’, etc are not good words as they are very common and can occur in both documents.