assignment-4

February 27, 2024

[1]:
```
import pandas as pd
```

[2]:
```
data=pd.read_csv('/content/TVlikes.csv', header='infer')
```

[3]:
```
data
```

[3]:

	Name	comedy	doctors	lawyers	guns	likes
0	A1	False	True	False	False	False
1	A2	True	False	True	False	True
2	A3	False	False	True	True	True
3	A4	False	False	True	False	False
4	A5	False	False	False	True	False
5	A6	True	False	False	True	False
6	A7	True	False	False	False	True
7	A8	False	True	True	True	True
8	A9	False	True	True	False	False
9	A10	True	True	True	False	True
10	A11	True	True	False	True	False
11	A12	False	False	False	False	False
12	A13	True	True	False	False	True

[4]:
```
from sklearn import tree
```

[5]:
```
y= data["likes"]
```

[6]:
```
y
```

[6]:
```
0     False
1      True
2      True
3     False
4     False
5     False
6      True
7      True
8     False
9      True
```

```
10    False
11    False
12     True
Name: likes, dtype: bool
```

[7]:
```
X= data.drop(["Name", "likes"], axis= 1)
```

[8]:
```
X
```

[8]:
```
    comedy  doctors  lawyers   guns
0    False     True    False  False
1     True    False     True  False
2    False    False     True   True
3    False    False     True  False
4    False    False    False   True
5     True    False    False   True
6     True    False    False  False
7    False     True     True   True
8    False     True     True  False
9     True     True     True  False
10    True     True    False   True
11   False    False    False  False
12    True     True    False  False
```

[10]:
```
dtc= tree.DecisionTreeClassifier(criterion= "entropy", max_depth= 3)
```

[11]:
```
dtc
```

[11]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

[12]:
```
import pydotplus
```

[13]:
```
clf=dtc.fit(X,y)
```

[15]:
```
clf
```

[15]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

[16]:
```
from IPython.display import Image
```

[17]:
```
dot_data=tree.export_graphviz(clf, feature_names=X.columns,␣
 ↪class_names=['TRUE', 'FALSE'], filled=True, out_file=None)
```

[18]:
```
dot_data
```

[18]:
```
'digraph Tree {\nnode [shape=box, style="filled", color="black",
fontname="helvetica"] ;\nedge [fontname="helvetica"] ;\n0 [label="lawyers <=
```

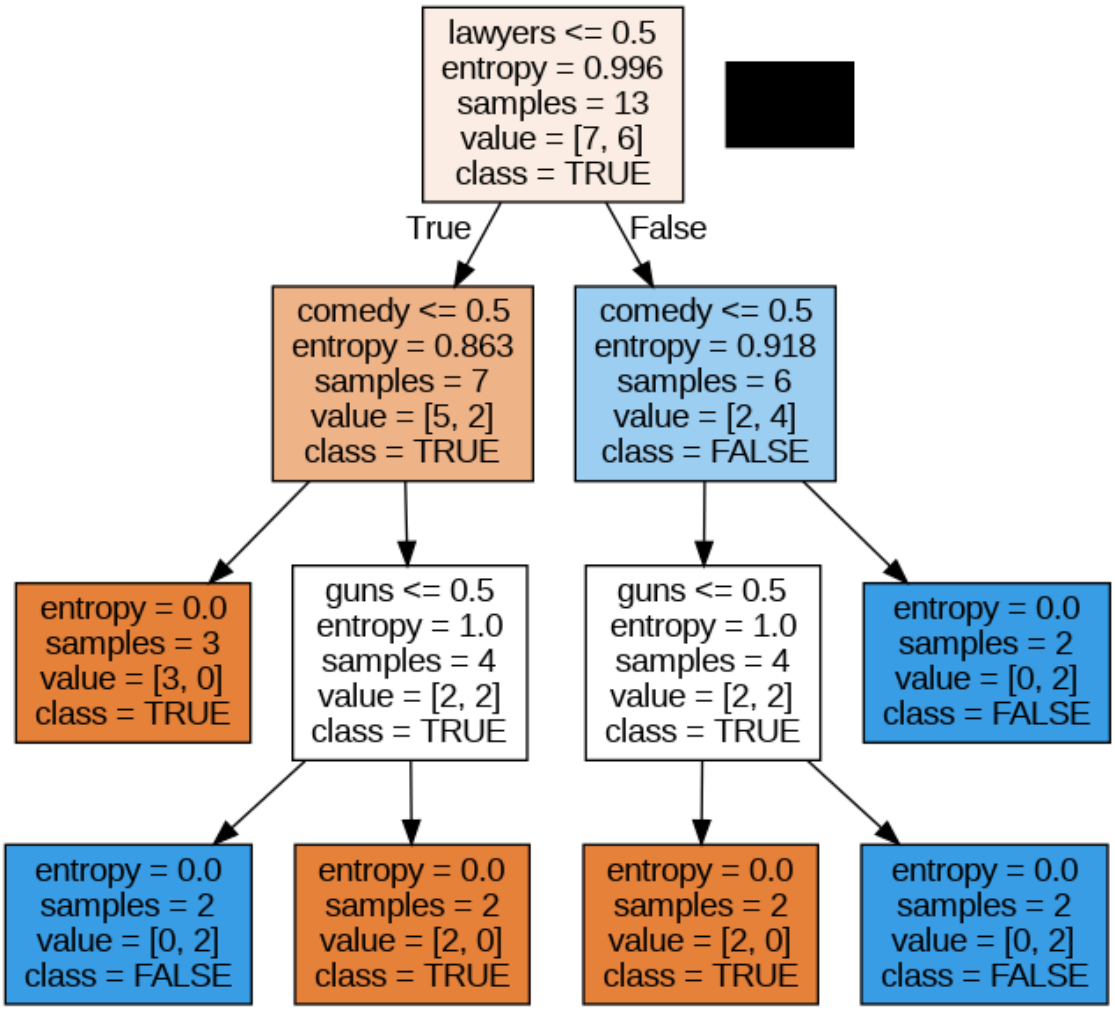
```
0.5\\nentropy = 0.996\\nsamples = 13\\nvalue = [7, 6]\\nclass = TRUE",
fillcolor="#fbede3"] ;\n1 [label="comedy <= 0.5\\nentropy = 0.863\\nsamples =
7\\nvalue = [5, 2]\\nclass = TRUE", fillcolor="#efb388"] ;\n0 -> 1
[labeldistance=2.5, labelangle=45, headlabel="True"] ;\n2 [label="entropy =
0.0\\nsamples = 3\\nvalue = [3, 0]\\nclass = TRUE", fillcolor="#e58139"] ;\n1 ->
2 ;\n3 [label="guns <= 0.5\\nentropy = 1.0\\nsamples = 4\\nvalue = [2,
2]\\nclass = TRUE", fillcolor="#ffffff"] ;\n1 -> 3 ;\n4 [label="entropy =
0.0\\nsamples = 2\\nvalue = [0, 2]\\nclass = FALSE", fillcolor="#399de5"] ;\n3
-> 4 ;\n5 [label="entropy = 0.0\\nsamples = 2\\nvalue = [2, 0]\\nclass = TRUE",
fillcolor="#e58139"] ;\n3 -> 5 ;\n6 [label="comedy <= 0.5\\nentropy =
0.918\\nsamples = 6\\nvalue = [2, 4]\\nclass = FALSE", fillcolor="#9ccef2"] ;\n0
-> 6 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;\n7 [label="guns <=
0.5\\nentropy = 1.0\\nsamples = 4\\nvalue = [2, 2]\\nclass = TRUE",
fillcolor="#ffffff"] ;\n6 -> 7 ;\n8 [label="entropy = 0.0\\nsamples = 2\\nvalue
= [2, 0]\\nclass = TRUE", fillcolor="#e58139"] ;\n7 -> 8 ;\n9 [label="entropy =
0.0\\nsamples = 2\\nvalue = [0, 2]\\nclass = FALSE", fillcolor="#399de5"] ;\n7
-> 9 ;\n10 [label="entropy = 0.0\\nsamples = 2\\nvalue = [0, 2]\\nclass =
FALSE", fillcolor="#399de5"] ;\n6 -> 10 ;\n}'
```

[19]:
```
graph= pydotplus.graph_from_dot_data(dot_data)
```

[20]:
```
graph

Image(graph.create_png())
```

[20]:



[21]:
```
testData= [["A1",False,True,False,False, 'FALSE'],
           ['A2',True,False,True,False, 'TRUE'],
           ['A5',False,False,False,True,'FALSE'],
            ['A12',False,False,False,False,'FALSE']]
```

[22]:
```
testData
```

[22]:
```
[['A1', False, True, False, False, 'FALSE'],
 ['A2', True, False, True, False, 'TRUE'],
 ['A5', False, False, False, True, 'FALSE'],
 ['A12', False, False, False, False, 'FALSE']]
```

[23]:
```
testData= pd.DataFrame(testData, columns= data.columns)
```

[24]:
```
testData
```

[24]:

	Name	comedy	doctors	lawyers	guns	likes
0	A1	False	True	False	False	FALSE
1	A2	True	False	True	False	TRUE
2	A5	False	False	False	True	FALSE
3	A12	False	False	False	False	FALSE

[25]:
```
testY = testData['likes']
```

[26]:
```
testY
```

[26]:
```
0    FALSE
1     TRUE
2    FALSE
3    FALSE
Name: likes, dtype: object
```

[27]:
```
testX= testData.drop(['Name', 'likes'], axis= 1)
```

[28]:
```
testX
```

[28]:

	comedy	doctors	lawyers	guns
0	False	True	False	False
1	True	False	True	False
2	False	False	False	True
3	False	False	False	False

[29]:
```
predY= clf.predict(testX)
```

[30]:
```
predY
```

[30]:
```
array([False,  True, False, False])
```

[31]:
```
3/4*100
```

[31]:
```
75.0
```

[32]:
```
predictions= pd.concat([testData['Name'],testData['likes'], pd.Series(predY,␣
↪name= 'Predicted likes')], axis=1)
```

[33]:
```
predictions
```

[33]:

	Name	likes	Predicted likes
0	A1	FALSE	False
1	A2	TRUE	True
2	A5	FALSE	False
3	A12	FALSE	False

[34]:
```
maxDepths=[2,3,4,5,6,7,8,9,10,15,20,25,30,35,40,45,50]
```

[35]:
```
maxDepths
```

[35]:
```
[2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50]
```

[36]:
```
import numpy as np
```

[41]:
```
trainAcc=np.zeros(len(maxDepths))
trainAcc
```

[41]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[42]:
```
trainAcc=np.zeros(len(maxDepths))
trainAcc
```

[42]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

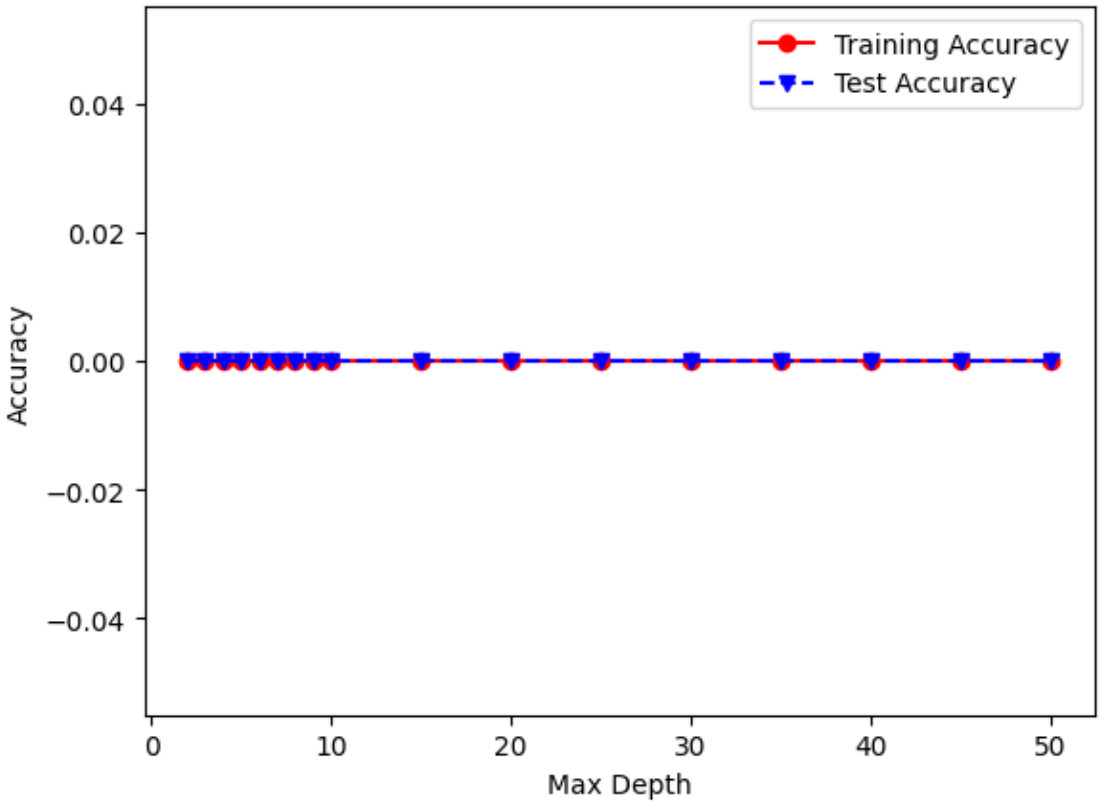
[44]:
```
testAcc=np.zeros(len(maxDepths))
testAcc
```

[44]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[40]:
```
import matplotlib.pyplot as plt
```

[45]:
```
plt.plot(maxDepths, trainAcc,'ro-', maxDepths,testAcc, 'bv--')
plt.legend(['Training Accuracy', 'Test Accuracy'])
plt.xlabel('Max Depth')
plt.ylabel('Accuracy')
```

[45]:
```
Text(0, 0.5, 'Accuracy')
```


Training Accuracy
Test Accuracy
0.04
0.02
0.00
−0.02
−0.04
Accuracy
0
10
20
30
40
50
Max Depth