Lab 13 19-10-2022

October 19, 2022

1 Decision Tree

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

1.1 Load Dataset

[2]:
```
df = pd.read_csv('iris.csv')
print(df)
```

```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            …            …             …            …          …
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

[3]:
```
df.species.value_counts()
```

[3]:
```
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

1.2 Preprocessing

[4]:
```
y=df['species']
print(y)
df=df.drop(['species'],axis=1)
x= df
print(x)
```

```
0          setosa
1          setosa
2          setosa
3          setosa
4          setosa
          …
145     virginica
146     virginica
147     virginica
148     virginica
149     virginica
Name: species, Length: 150, dtype: object
     sepal_length  sepal_width  petal_length  petal_width
0             5.1          3.5           1.4          0.2
1             4.9          3.0           1.4          0.2
2             4.7          3.2           1.3          0.2
3             4.6          3.1           1.5          0.2
4             5.0          3.6           1.4          0.2
..            …           …            …          …
145           6.7          3.0           5.2          2.3
146           6.3          2.5           5.0          1.9
147           6.5          3.0           5.2          2.0
148           6.2          3.4           5.4          2.3
149           5.9          3.0           5.1          1.8

[150 rows x 4 columns]
```

[5]:
```
x.describe()
```

[5]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

1.3 Test & Train

1.3.1 Split Dataset

[6]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.25,␣
 ↪random_state=0)
```

1.3.2 Feature Scaling

[7]:
```
from sklearn.preprocessing import MinMaxScaler
st_x= MinMaxScaler()
x_train= st_x.fit_transform(x_train)
x_test=st_x.fit_transform(x_test)
print(x_train)
```

```
[[0.44444444 0.41666667 0.53448276 0.58333333]
 [0.41666667 0.25       0.5        0.45833333]
 [0.69444444 0.41666667 0.75862069 0.83333333]
 [0.11111111 0.5        0.03448276 0.04166667]
 [0.72222222 0.45833333 0.68965517 0.91666667]
 [0.19444444 0.625      0.0862069  0.20833333]
 [0.30555556 0.70833333 0.06896552 0.04166667]
 [0.19444444 0.         0.4137931  0.375     ]
 [0.61111111 0.41666667 0.75862069 0.70833333]
 [0.66666667 0.54166667 0.79310345 1.        ]
 [0.47222222 0.08333333 0.67241379 0.58333333]
 [0.66666667 0.20833333 0.81034483 0.70833333]
 [0.36111111 0.20833333 0.48275862 0.41666667]
 [0.94444444 0.41666667 0.86206897 0.91666667]
 [0.55555556 0.54166667 0.62068966 0.625     ]
 [0.33333333 0.16666667 0.46551724 0.41666667]
 [0.55555556 0.29166667 0.65517241 0.70833333]
 [0.55555556 0.33333333 0.68965517 0.58333333]
 [0.16666667 0.20833333 0.5862069  0.66666667]
 [0.55555556 0.20833333 0.67241379 0.75      ]
 [0.75       0.5        0.62068966 0.54166667]
 [0.61111111 0.41666667 0.70689655 0.79166667]
 [0.47222222 0.58333333 0.5862069  0.625     ]
 [0.13888889 0.45833333 0.0862069  0.04166667]
 [0.41666667 0.29166667 0.68965517 0.75      ]
 [0.36111111 0.29166667 0.53448276 0.5       ]
 [0.36111111 0.375      0.43103448 0.5       ]
 [0.33333333 0.20833333 0.5        0.5       ]
 [0.5        0.41666667 0.60344828 0.54166667]
 [0.80555556 0.5        0.84482759 0.70833333]
 [0.27777778 0.70833333 0.06896552 0.04166667]
 [0.         0.41666667 0.         0.        ]
```

```
[0.58333333 0.29166667 0.72413793 0.75      ]
[0.38888889 0.41666667 0.53448276 0.45833333]
[0.30555556 0.58333333 0.10344828 0.04166667]
[0.38888889 1.         0.06896552 0.125     ]
[0.72222222 0.45833333 0.65517241 0.58333333]
[0.08333333 0.45833333 0.06896552 0.04166667]
[0.44444444 0.41666667 0.68965517 0.70833333]
[0.22222222 0.20833333 0.32758621 0.41666667]
[0.08333333 0.58333333 0.05172414 0.08333333]
[0.52777778 0.08333333 0.5862069  0.58333333]
[0.80555556 0.66666667 0.86206897 1.        ]
[0.38888889 0.375      0.53448276 0.5       ]
[0.13888889 0.41666667 0.05172414 0.        ]
[0.77777778 0.41666667 0.82758621 0.83333333]
[0.72222222 0.5        0.79310345 0.91666667]
[0.61111111 0.41666667 0.81034483 0.875     ]
[0.58333333 0.33333333 0.77586207 0.83333333]
[0.22222222 0.75       0.0862069  0.04166667]
[0.13888889 0.58333333 0.0862069  0.04166667]
[0.61111111 0.5        0.68965517 0.79166667]
[0.66666667 0.54166667 0.79310345 0.83333333]
[0.05555556 0.125      0.03448276 0.08333333]
[0.52777778 0.58333333 0.74137931 0.91666667]
[0.16666667 0.41666667 0.05172414 0.04166667]
[0.38888889 0.20833333 0.67241379 0.79166667]
[0.72222222 0.45833333 0.74137931 0.83333333]
[0.02777778 0.5        0.03448276 0.04166667]
[0.19444444 0.66666667 0.05172414 0.04166667]
[0.80555556 0.41666667 0.81034483 0.625     ]
[0.22222222 0.625      0.05172414 0.08333333]
[0.02777778 0.41666667 0.03448276 0.04166667]
[0.30555556 0.79166667 0.10344828 0.125     ]
[0.33333333 0.125      0.5        0.5       ]
[0.69444444 0.5        0.82758621 0.91666667]
[0.91666667 0.41666667 0.94827586 0.83333333]
[0.22222222 0.625      0.05172414 0.04166667]
[0.16666667 0.45833333 0.06896552 0.        ]
[0.25       0.58333333 0.05172414 0.04166667]
[0.38888889 0.33333333 0.5862069  0.5       ]
[0.63888889 0.41666667 0.56896552 0.54166667]
[0.19444444 0.5        0.01724138 0.04166667]
[0.22222222 0.54166667 0.10344828 0.16666667]
[0.58333333 0.375      0.55172414 0.5       ]
[0.30555556 0.58333333 0.06896552 0.125     ]
[0.94444444 0.25       1.         0.91666667]
[0.16666667 0.16666667 0.37931034 0.375     ]
[1.         0.75       0.9137931  0.79166667]
[0.66666667 0.45833333 0.56896552 0.54166667]
```

```
 [0.25       0.875      0.06896552 0.        ]
 [0.47222222 0.41666667 0.63793103 0.70833333]
 [0.41666667 0.83333333 0.01724138 0.04166667]
 [0.94444444 0.33333333 0.96551724 0.79166667]
 [0.22222222 0.75       0.06896552 0.08333333]
 [0.11111111 0.5        0.0862069  0.04166667]
 [0.86111111 0.33333333 0.86206897 0.75      ]
 [0.19444444 0.54166667 0.05172414 0.04166667]
 [0.55555556 0.58333333 0.77586207 0.95833333]
 [0.38888889 0.33333333 0.51724138 0.5       ]
 [0.41666667 0.29166667 0.48275862 0.45833333]
 [0.38888889 0.25       0.4137931  0.375     ]
 [0.58333333 0.5        0.72413793 0.91666667]
 [0.66666667 0.41666667 0.70689655 0.91666667]
 [0.55555556 0.20833333 0.65517241 0.58333333]
 [0.66666667 0.41666667 0.67241379 0.66666667]
 [0.19444444 0.41666667 0.0862069  0.04166667]
 [0.33333333 0.16666667 0.44827586 0.375     ]
 [0.66666667 0.45833333 0.77586207 0.95833333]
 [0.41666667 0.29166667 0.68965517 0.75      ]
 [0.22222222 0.58333333 0.06896552 0.04166667]
 [0.63888889 0.375      0.60344828 0.5       ]
 [0.36111111 0.41666667 0.51724138 0.5       ]
 [0.44444444 0.5        0.63793103 0.70833333]
 [0.55555556 0.125      0.56896552 0.5       ]
 [0.33333333 0.625      0.03448276 0.04166667]
 [0.22222222 0.70833333 0.06896552 0.125     ]
 [0.16666667 0.45833333 0.06896552 0.        ]
 [0.55555556 0.375      0.77586207 0.70833333]
 [0.41666667 0.29166667 0.51724138 0.375     ]
 [0.94444444 0.75       0.96551724 0.875     ]
 [0.08333333 0.5        0.05172414 0.04166667]]
```

1.3.3 Fitting to Decision Tree

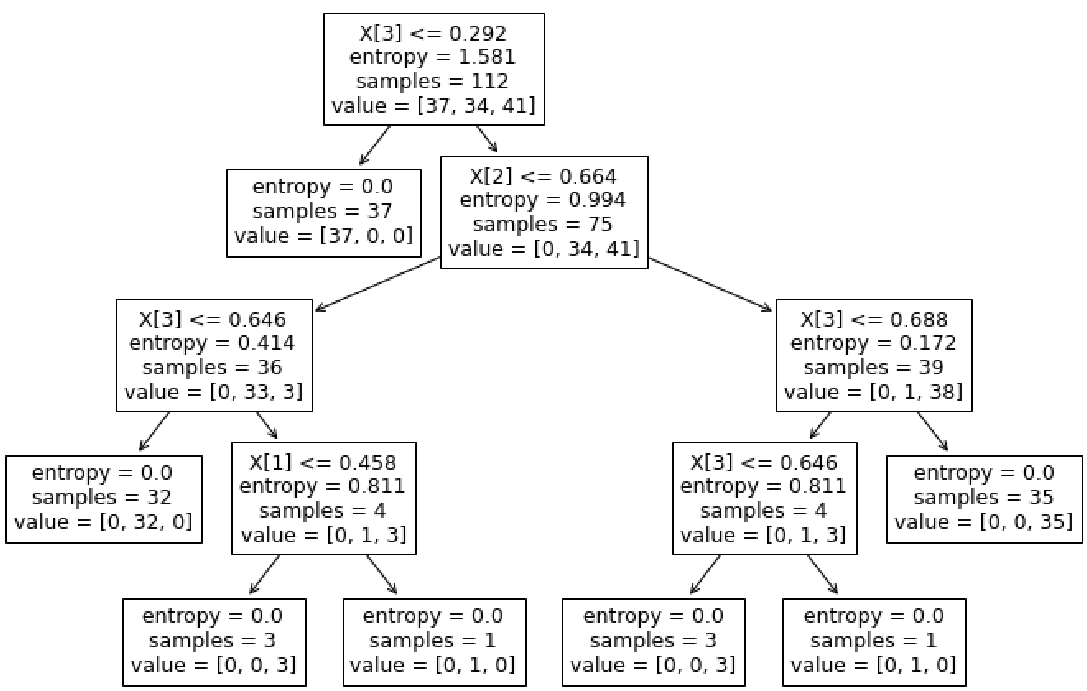
[8]:
```
from sklearn import tree
classifier = tree.DecisionTreeClassifier(criterion = 'entropy', random_state =␣
 ↪0)
classifier.fit(x_train, y_train)
```

[8]:
```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

1.4 Visualisation

[9]:
```
plt.figure(figsize = (12,8))
tree.plot_tree(classifier)
```

```
[9]: [Text(267.84000000000003, 391.392, 'X[3] <= 0.292\nentropy = 1.581\nsamples =
112\nvalue = [37, 34, 41]'),
 Text(200.88000000000002, 304.416, 'entropy = 0.0\nsamples = 37\nvalue = [37, 0,
0]'),
 Text(334.80000000000007, 304.416, 'X[2] <= 0.664\nentropy = 0.994\nsamples =
75\nvalue = [0, 34, 41]'),
 Text(133.92000000000002, 217.44, 'X[3] <= 0.646\nentropy = 0.414\nsamples =
36\nvalue = [0, 33, 3]'),
 Text(66.96000000000001, 130.464, 'entropy = 0.0\nsamples = 32\nvalue = [0, 32,
0]'),
 Text(200.88000000000002, 130.464, 'X[1] <= 0.458\nentropy = 0.811\nsamples =
4\nvalue = [0, 1, 3]'),
 Text(133.92000000000002, 43.488, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0,
3]'),
 Text(267.84000000000003, 43.488, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1,
0]'),
 Text(535.6800000000001, 217.44, 'X[3] <= 0.688\nentropy = 0.172\nsamples =
39\nvalue = [0, 1, 38]'),
 Text(468.72, 130.464, 'X[3] <= 0.646\nentropy = 0.811\nsamples = 4\nvalue = [0,
1, 3]'),
 Text(401.76000000000005, 43.488, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0,
3]'),
 Text(535.6800000000001, 43.488, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1,
0]'),
 Text(602.6400000000001, 130.464, 'entropy = 0.0\nsamples = 35\nvalue = [0, 0,
35]')]
```



1.4.1 Visualisation using Graphviz

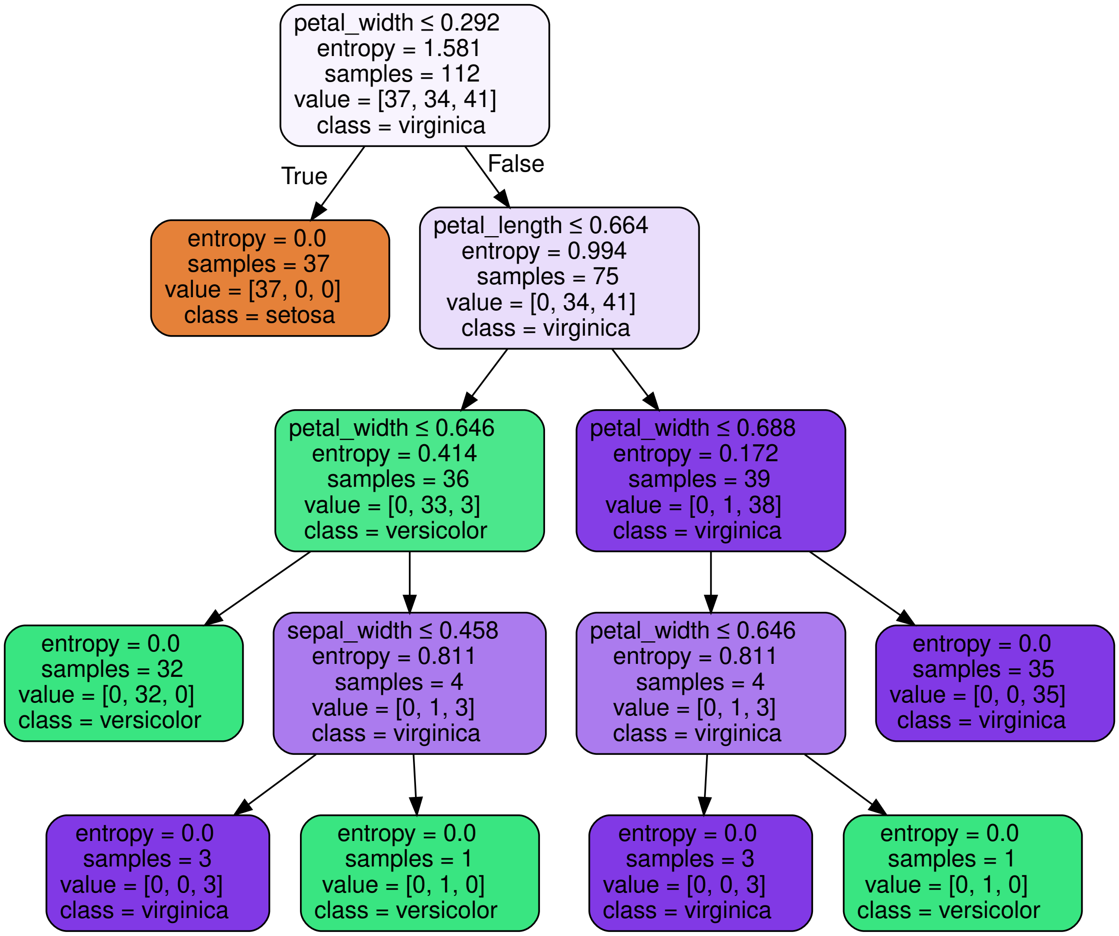
[10]:
```
!pip install graphviz
import graphviz
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: graphviz in
/home/student/.local/lib/python3.9/site-packages (0.20.1)
WARNING: You are using pip version 20.3.3; however, version 22.2.2 is
available.
You should consider upgrading via the '/opt/anaconda3/bin/python -m pip install
--upgrade pip' command.
```

[11]:
```
g_tree = tree.export_graphviz(classifier, out_file=None,
                              feature_names=df.columns,
                              class_names=['setosa','versicolor','virginica'],
                              filled=True, rounded=True,
                              special_characters=True)
```

[12]:
```
graph = graphviz.Source(g_tree)
graph
```

[12]:



[13]:
```python
y_pred= classifier.predict(x_test)
print(y_pred)
```

```
['virginica' 'versicolor' 'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'virginica' 'virginica' 'versicolor' 'virginica' 'versicolor'
 'virginica' 'virginica' 'virginica' 'setosa' 'versicolor' 'versicolor'
 'setosa' 'setosa' 'virginica' 'versicolor' 'setosa' 'setosa' 'virginica'
 'setosa' 'setosa' 'versicolor' 'versicolor' 'setosa' 'virginica'
 'versicolor' 'setosa' 'virginica' 'virginica' 'versicolor' 'setosa'
 'virginica']
```

1.5 Confusion Matrix

[14]:
```python
from sklearn.metrics import confusion_matrix
cm= confusion_matrix (y_test, y_pred,labels=classifier.classes_)
print(cm)
```

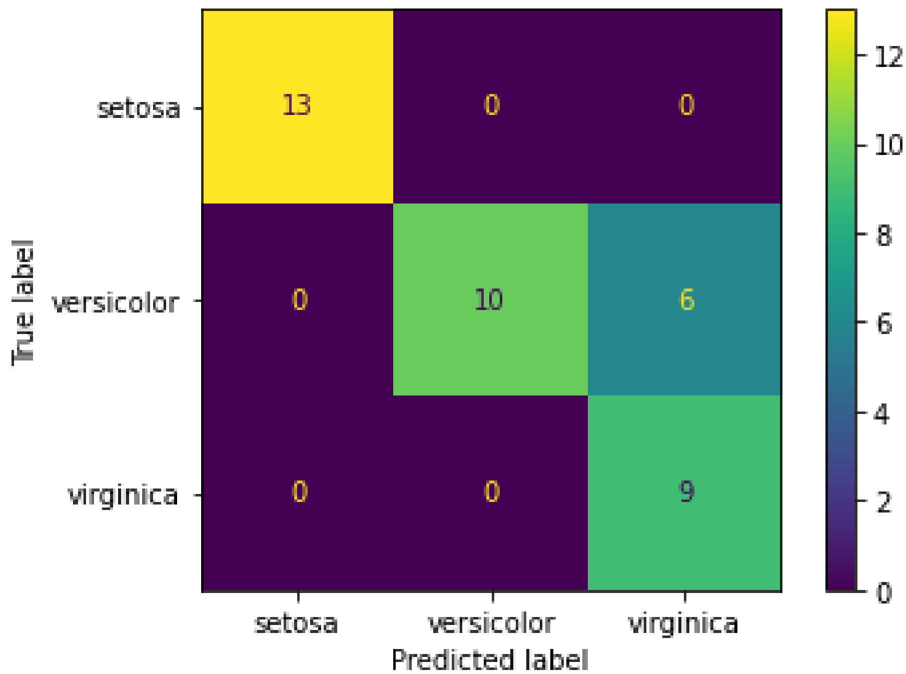
```
[[13  0  0]
 [ 0 10  6]
```

```
[ 0  0  9]]
```

1.5.1 CM Display

[15]:
```python
from sklearn.metrics import ConfusionMatrixDisplay
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=classifier.classes_)
disp.plot()
```

[15]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7ff610213a00>
```



1.6 Accuracy

1.6.1 Test

[16]:
```python
from sklearn.metrics import accuracy_score
print('Test accuracy score with criterion entropy: {0:0.4f}'.
 ↪format(accuracy_score(y_test,y_pred)))
```

```
Test accuracy score with criterion entropy: 0.8421
```

1.6.2 Train

[17]:
```
y_pred_train= classifier.predict(x_train)
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train,␣
↪y_pred_train)))
```

```
Training-set accuracy score: 1.0000
```