

Name : Syed Riaz Ali

Email : syedriazali1997@gmail.com

Whatsapp : 923002502513

Day_26 : Decision Tree Classification Assignment

In [41]:

```
# Importing libraries

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Loading data set

df = sns.load_dataset("Iris")
df.head()
```

Out[41]:

	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

In [42]:

```
# Selection of Input Variables

X = df[["sepal_length","sepal_width", "petal_length", "petal_width"]]
y = df["species"]
```

In [43]:

```
X.head()
```

Out[43]:

	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

In [44]:

```
y.head()
```

Out[44]:

```
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object
```

Modeling by using 100 % Data for Training

In [45]:

```
# Applying Machine Learning Algorithms
from sklearn.tree import DecisionTreeClassifier

# Creating and fit the model
model_100 = DecisionTreeClassifier().fit(X, y)
```

In [46]:

```
# Checking score

score_100 = model_100.score(X,y)
print("The score of model using 100 % data for training is : ", score_100)
```

The score of model using 100 % data for training is : 1.0

In [47]:

```
# Giving any random 2D array for predicting the unknown values

import numpy as np

pred_100 = np.array([[4.3, 2.3, 1.0, 0.8],
                    [4.8, 2.6, 6.7, 0.2],
                    [5.2, 2.0, 1.9, 2.2],
                    [5.4, 2.9, 5.4, 2.0],
                    [5.9, 3.4, 4.0, 1.8],
                    [6.2, 3.6, 3.3, 1.4],
                    [6.6, 3.7, 2.8, 1.3],
                    [6.8, 3.9, 1.4, 0.6],
                    [7.4, 4.0, 6.1, 2.4],
                    [7.0, 4.2, 5.0, 2.3]])
```

In [48]:

```
predict_values_100 = model_100.predict(pred_100)
predict_values_100
```

Out[48]:

```
array(['setosa', 'setosa', 'virginica', 'virginica', 'versicolor',
       'versicolor', 'versicolor', 'setosa', 'virginica', 'virginica'],
      dtype=object)
```

In [49]:

```
# Tree Graph

from sklearn.tree import plot_tree

plot_tree(model_100, filled = True)
plt.title("DTC Model using 100% data for Training")

plt.savefig('tiff_compressed.tiff', dpi = 600, format = "tiff",
           facecolor = "white", edgecolor = "none",
           pil_kwargs = {"compression": "tiff_lzw"})

plt.show()
```

DTC Model using 100% data for Training

Modeling by using 90 % Data for Training

In [50]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.1)

# Creating and fit the model
model_90 = DecisionTreeClassifier().fit(X_train, y_train)

# Prediction
pred_values = model_90.predict(X_test)
pred_values
```

Out[50]:

```
array(['setosa', 'versicolor', 'virginica', 'versicolor', 'versicolor',
       'versicolor', 'setosa', 'virginica', 'virginica', 'versicolor',
       'versicolor', 'setosa', 'setosa', 'virginica', 'virginica'],
      dtype=object)
```

In [51]:

```
# Checking Score

score_90 = accuracy_score(y_test, pred_values)
print("The score of model using 90 % data for training is : ", score_90)
```

The score of model using 90 % data for training is : 0.9333333333333333

In [52]:

```
X.max()
```

Out[52]:

```
sepal_length    7.9
sepal_width     4.4
petal_length     6.9
petal_width     2.5
dtype: float64
```

In [53]:

```
X.min()
```

Out[53]:

```
sepal_length    4.3
sepal_width     2.0
petal_length     1.0
petal_width     0.1
dtype: float64
```

In [54]:

```
# Giving any random 2D array for predicting the unknown values

import numpy as np

pred_90 = np.array([[4.3, 2.3, 1.0, 0.8],
                    [4.8, 2.6, 6.7, 0.2],
                    [5.2, 2.0, 1.9, 2.2],
                    [5.4, 2.9, 5.4, 2.0],
                    [5.9, 3.4, 4.0, 1.8],
                    [6.2, 3.6, 3.3, 1.4],
                    [6.6, 3.7, 2.8, 1.3],
                    [6.8, 3.9, 1.4, 0.6],
                    [7.4, 4.0, 6.1, 2.4],
                    [7.0, 4.2, 5.0, 2.3]])
```

In [55]:

```
predict_values_90 = model_90.predict(pred_90)
predict_values_90
```

Out[55]:

```
array(['setosa', 'virginica', 'setosa', 'virginica', 'virginica',
       'versicolor', 'versicolor', 'setosa', 'virginica', 'virginica'],
      dtype=object)
```

In [56]:

```
# Tree Graph

from sklearn.tree import plot_tree

plot_tree(model_90, filled = True)
plt.title("DTC Model using 90% data for Training")

plt.savefig('tiff_compressed.tiff', dpi = 600, format = "tiff",
           facecolor = "white", edgecolor = "none",
           pil_kwargs = {"compression": "tiff_lzw"})

plt.show()
```

DTC Model using 90% data for Training

Modeling by using 80 % Data for Training

In [57]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.2)

# Creating and fit the model
model_80 = DecisionTreeClassifier().fit(X_train, y_train)

# Prediction
pred_values = model_80.predict(X_test)
pred_values
```

Out[57]:

```
array(['setosa', 'versicolor', 'setosa', 'virginica', 'setosa',
       'versicolor', 'virginica', 'virginica', 'versicolor', 'virginica',
       'versicolor', 'setosa', 'virginica', 'setosa', 'virginica',
       'virginica', 'setosa', 'versicolor', 'virginica', 'virginica',
       'versicolor', 'setosa', 'setosa', 'virginica', 'setosa', 'setosa',
       'setosa', 'versicolor', 'setosa', 'virginica'],
      dtype=object)
```

In [58]:

```
# Checking Score

score_80 = accuracy_score(y_test, pred_values)
print("The score of model using 80 % data for training is : ", score_80)
```

The score of model using 80 % data for training is : 1.0

In [59]:

```
# Giving any random 2D array for predicting the unknown values

import numpy as np

pred_80 = np.array([[4.3, 2.3, 1.0, 0.8],
                    [4.8, 2.6, 6.7, 0.2],
                    [5.2, 2.0, 1.9, 2.2],
                    [5.4, 2.9, 5.4, 2.0],
                    [5.9, 3.4, 4.0, 1.8],
                    [6.2, 3.6, 3.3, 1.4],
                    [6.6, 3.7, 2.8, 1.3],
                    [6.8, 3.9, 1.4, 0.6],
                    [7.4, 4.0, 6.1, 2.4],
                    [7.0, 4.2, 5.0, 2.3]])
```

In [60]:

```
predict_values_80 = model_80.predict(pred_80)
predict_values_80
```

Out[60]:

```
array(['setosa', 'virginica', 'setosa', 'virginica', 'versicolor',
       'versicolor', 'versicolor', 'setosa', 'virginica', 'virginica'],
      dtype=object)
```

In [61]:

```
# Tree Graph

from sklearn.tree import plot_tree

plot_tree(model_80, filled = True)
plt.title("DTC Model using 80% data for Training")

plt.savefig('tiff_compressed.tiff', dpi = 600, format = "tiff",
           facecolor = "white", edgecolor = "none",
           pil_kwargs = {"compression": "tiff_lzw"})

plt.show()
```

DTC Model using 80% data for Training

Modeling by using 70 % Data for Training

In [62]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= 0.3)

# Creating and fit the model
model_70 = DecisionTreeClassifier().fit(X_train, y_train)

# Prediction
pred_values = model_70.predict(X_test)
pred_values
```

Out[62]:

```
array(['setosa', 'virginica', 'setosa', 'virginica', 'virginica',
       'virginica', 'setosa', 'virginica', 'setosa', 'setosa',
       'versicolor', 'virginica', 'setosa', 'setosa', 'setosa',
       'versicolor', 'virginica', 'setosa', 'versicolor', 'virginica',
       'versicolor', 'virginica', 'versicolor', 'versicolor',
       'versicolor', 'virginica', 'virginica', 'setosa', 'setosa',
       'setosa', 'setosa', 'versicolor', 'versicolor', 'setosa',
       'setosa', 'virginica', 'virginica', 'virginica', 'versicolor',
       'setosa'],
      dtype=object)
```

In [63]:

```
# Checking Score

score_70 = accuracy_score(y_test, pred_values)
print("The score of model using 70 % data for training is : ", score_70)
```

The score of model using 70 % data for training is : 0.9555555555555555

In [64]:

```
# Giving any random 2D array for predicting the unknown values

import numpy as np

pred_70 = np.array([[4.3, 2.3, 1.0, 0.8],
                    [4.8, 2.6, 6.7, 0.2],
                    [5.2, 2.0, 1.9, 2.2],
                    [5.4, 2.9, 5.4, 2.0],
                    [5.9, 3.4, 4.0, 1.8],
                    [6.2, 3.6, 3.3, 1.4],
                    [6.6, 3.7, 2.8, 1.3],
                    [6.8, 3.9, 1.4, 0.6],
                    [7.4, 4.0, 6.1, 2.4],
                    [7.0, 4.2, 5.0, 2.3]])
```

In [65]:

```
predict_values_70 = model_70.predict(pred_70)
predict_values_70
```

Out[65]:

```
array(['setosa', 'setosa', 'versicolor', 'virginica', 'versicolor',
       'versicolor', 'versicolor', 'setosa', 'virginica', 'virginica'],
      dtype=object)
```

In [66]:

```
# Tree Graph

from sklearn.tree import plot_tree

plot_tree(model_70, filled = True)
plt.title("DTC Model using 70% data for Training")

plt.savefig('tiff_compressed.tiff', dpi = 600, format = "tiff",
           facecolor = "white", edgecolor = "none",
           pil_kwargs = {"compression": "tiff_lzw"})

plt.show()
```

DTC Model using 70% data for Training

In []: