Decision Tree Classifier

In [ ]:
```
# Import libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree
from sklearn.metrics import accuracy_score
```

In [ ]:
```
# Load dataset
df = sns.load_dataset('iris')
df.head()
```

Out[ ]:

	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 [ ]:
```
# Split data
X = df.iloc[:, :-1]
y = df.iloc[:,-1:]
```

In [ ]:
```
X.head()
```

Out[ ]:

	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 [ ]:
```
y.head()
```

Out[ ]:

	species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa

In [ ]:
```
# # Train and plot results
# model = DecisionTreeClassifier().fit(X, y)
# plot_tree(model, filled=True)
# plt.title('Decision tree trained model on Iris data')

# # How to save plot in tiff, png and pdf format
# # # save in png format
# # plt.savefig('DecisionTree.png', dpi=300)

# # Save in tiff format
# plt.savefig('tiff_dt.tiff', dpi=600, format='tiff',
#             facecolor='white', edgecolor='none',
#             pil_kwargs={'compression': 'tiff_lzw'})
# plt.show()
```

Accuracy score on 80/20 split

In [ ]:
```
# 80/20 data split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Train model
model = DecisionTreeClassifier().fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Accuracy score
acc_score = accuracy_score(y_test, y_pred)
print(f'Model accuracy on 80/20 split: {acc_score}')
```

```
Model accuracy on 80/20 split: 1.0
```

Accuracy score on 70/30 split

In [ ]:
```
# 70/30 data split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Train model
model = DecisionTreeClassifier().fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Accuracy score
acc_score = accuracy_score(y_test, y_pred)
print(f'Model accuracy on 70/20 split: {acc_score}')
```

```
Model accuracy on 70/20 split: 0.9777777777777777
```

Accuracy score on 90/10 split

In [ ]:
```
# 90/10 data split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=0)

# Train model
model = DecisionTreeClassifier().fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Accuracy score
acc_score = accuracy_score(y_test, y_pred)
print(f'Model accuracy on 90/10 split: {acc_score}')
```

```
Model accuracy on 90/10 split: 1.0
```

Make Predictions on Unknown Samples

In [ ]:
```
# Create 10 random samples
np.random.seed(0) # for reproducibility
p1 = [0, 2, 3, 1]
p2 = [4, 4, 4, 4]
p3 = [6, 5, 7, 8]
rand_samples = np.random.triangular(left=p1, mode=p2, right=p3, size=(10, 4))
rand_samples
```

Out[ ]:
```
array([[3.62925944, 4.07564514, 4.8166907 , 4.43022818],
       [3.18868549, 3.96859459, 4.4021252 , 6.25920823],
       [5.33966155, 3.51678908, 5.41908269, 4.36806632],
       [3.69229867, 4.52754885, 3.53305181, 2.35266969],
       [0.69659281, 4.2913813 , 5.36839987, 6.09221074],
       [5.49346284, 4.22377561, 4.45790487, 5.52105202],
       [1.68481044, 3.95947088, 3.75724048, 6.75530312],
       [3.53897721, 3.57733054, 4.02925386, 5.48574928],
       [3.30871697, 3.84678198, 3.27415179, 4.72796606],
       [3.83279237, 3.9239553 , 6.17840213, 5.0151999 ]])
```

In [ ]:
```
# Make predictions
model.predict(rand_samples)
```

```
c:\users\awon\miniconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but
DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

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

In [ ]:
```

```