Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exe20 - Decision Tree Exercise 2

Name: Phua Yan Han

IC Number: 050824070059

Date :

Introduction :

Conclusion :

Decision Tree

In [1]:
```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.metrics import accuracy_score
from matplotlib import pyplot as plt
iris = load_iris()
X = iris.data[:, 2:] # petal length and width
y = iris.target
```

DecisionTree Modeling

In [2]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random

# Create a decision tree classifier
clf = DecisionTreeClassifier()

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9555555555555556
```

Build decision tree in both entropy and GINI

In [3]:
```
clf_entropy = DecisionTreeClassifier(criterion='entropy')
# Train the classifier
clf_entropy.fit(X_train, y_train)
# Make predictions on the test set
y_pred_entropy = clf_entropy.predict(X_test)
# Calculate accuracy
accuracy_entropy = accuracy_score(y_test, y_pred_entropy)
print("Accuracy (Entropy):", accuracy_entropy)

# Create a decision tree classifier with Gini impurity criterion
clf_gini = DecisionTreeClassifier(criterion='gini')
# Train the classifier
clf_gini.fit(X_train, y_train)
# Make predictions on the test set
y_pred_gini = clf_gini.predict(X_test)
# Calculate accuracy
accuracy_gini = accuracy_score(y_test, y_pred_gini)
print("Accuracy (Gini impurity):", accuracy_gini)
print(y_pred_entropy,y_pred_gini)
```

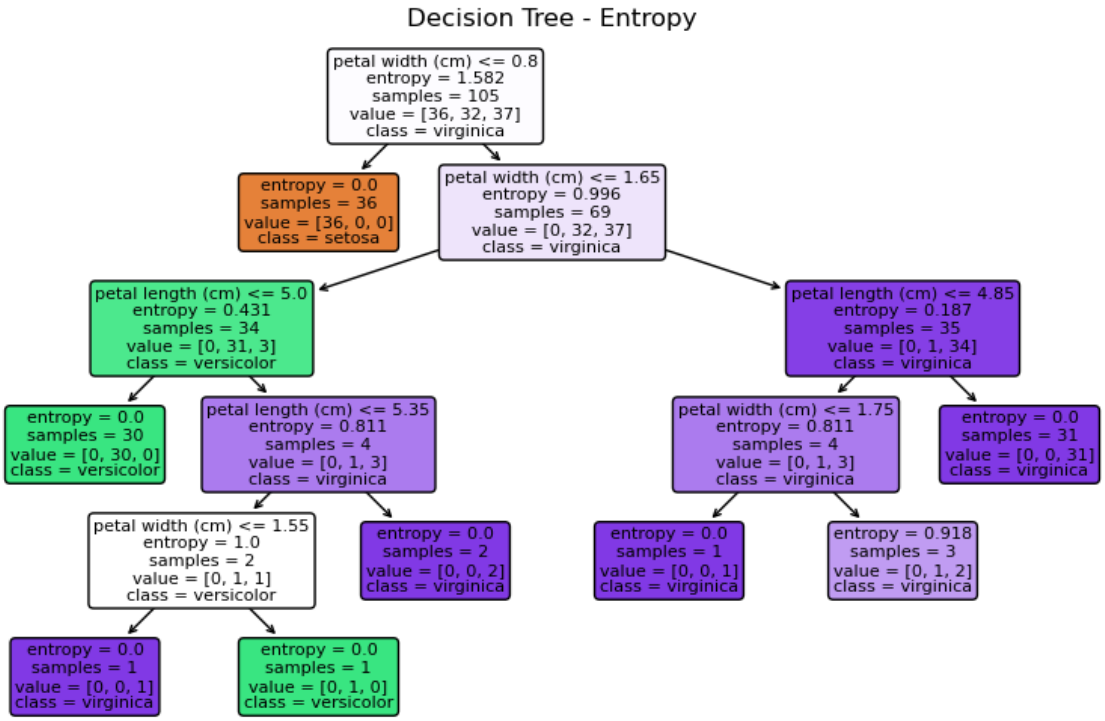
```
Accuracy (Entropy): 0.9555555555555556
Accuracy (Gini impurity): 0.9555555555555556
[0 1 1 0 2 1 2 0 0 2 1 0 2 1 1 0 1 1 0 0 1 1 2 0 2 1 0 0 1 2 1 2 1 2 2 0 1
 0 1 2 2 0 1 2 1] [0 1 1 0 2 1 2 0 0 2 1 0 2 1 1 0 1 1 0 0 1 1 2 0 2 1 0 0 1
2 1 2 1 2 2 0 1
 0 1 2 2 0 1 2 1]
```

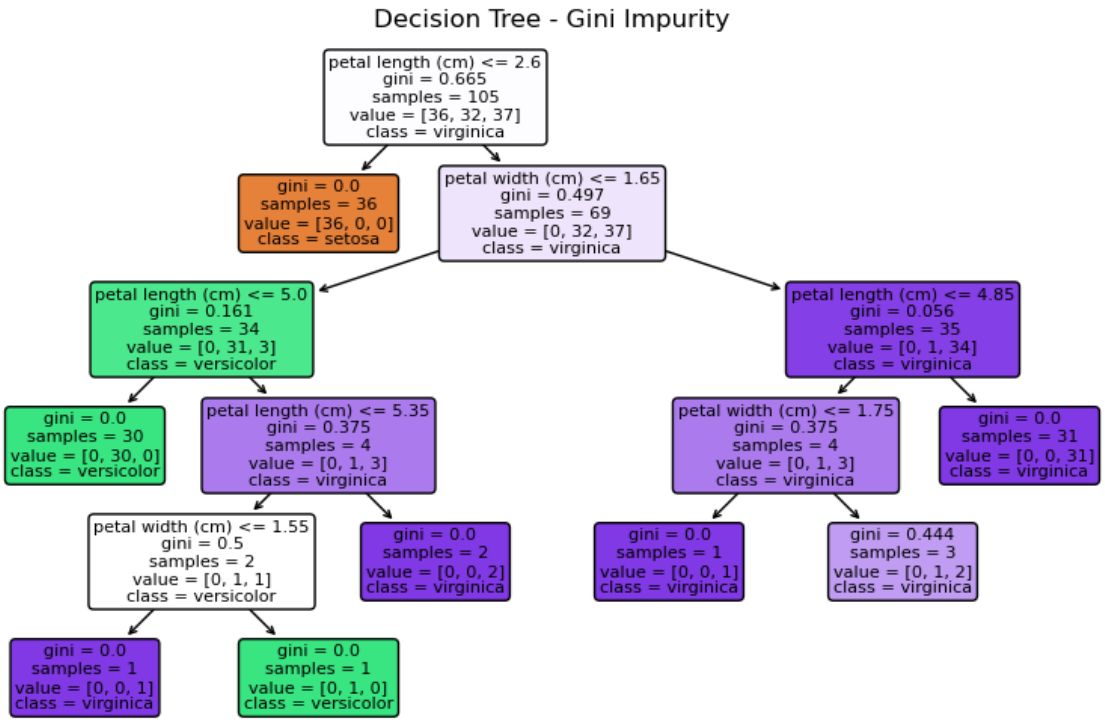
In [4]:
```
pip install graphviz
```

```
Requirement already satisfied: graphviz in c:\users\asus\anaconda3\envs\pytho
n-dscourse\lib\site-packages (0.20.1)
Note: you may need to restart the kernel to use updated packages.
```

Convert to Decision Tree Diagram

In [5]:
```
plt.figure(figsize=(10, 6))
tree.plot_tree(clf_entropy,
               feature_names=iris.feature_names[2:],
               class_names=iris.target_names.tolist(),
               filled=True,
               rounded=True)
plt.title("Decision Tree - Entropy")
plt.show()
plt.figure(figsize=(10, 6))
tree.plot_tree(clf_gini,
               feature_names=iris.feature_names[2:],
               class_names=iris.target_names.tolist(),
               filled=True,
               rounded=True)
plt.title("Decision Tree - Gini Impurity")
plt.show()
```




Decision Tree - Gini Impurity
petal length (cm) <= 2.6
gini = 0.665
samples = 105
value = [36, 32, 37]
class = virginica
gini = 0.0
samples = 36
value = [36, 0, 0]
class = setosa
petal width (cm) <= 1.65
gini = 0.497
samples = 69
value = [0, 32, 37]
class = virginica
petal length (cm) <= 5.0
gini = 0.161
samples = 34
value = [0, 31, 3]
class = versicolor
petal length (cm) <= 4.85
gini = 0.056
samples = 35
value = [0, 1, 34]
class = virginica
gini = 0.0
samples = 30
value = [0, 30, 0]
class = versicolor
petal length (cm) <= 5.35
gini = 0.375
samples = 4
value = [0, 1, 3]
class = virginica
petal width (cm) <= 1.75
gini = 0.375
samples = 4
value = [0, 1, 3]
class = virginica
gini = 0.0
samples = 31
value = [0, 0, 31]
class = virginica
petal width (cm) <= 1.55
gini = 0.5
samples = 2
value = [0, 1, 1]
class = versicolor
gini = 0.0
samples = 2
value = [0, 0, 2]
class = virginica
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
gini = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
gini = 0.0
samples = 1
value = [0, 1, 0]
class = versicolor

In [ ]: