DATA MINING

LA2

- Submitted To: **Mrs. Vani K.S.**

GROUP 12

Implement Decision with Gini and entropy. Compare the accuracies using both and plot the trees.

AMAY DEEPAK NAYAK	(1NT19IS015)
AMAAN MOHIB	(1NT19IS012)
AMIT KUMAR	(1NT19IS016)
ADITYA RANJAN	(1NT19IS008)
ADITYA NARAYAN	(1NT19IS007)

Implementation using Python

```
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import seaborn as sns


# Function to make predictions
def prediction(X_test, clf_object):
    # Predicton on test with giniIndex
    y_pred = clf_object.predict(X_test)
    print("Predicted values: ")
    print(y_pred)
    return y_pred


df = sns.load_dataset('iris')
df.info()
```

```
df.isnull().any()
df.shape
target = df['species']
df1 = df.copy()
df1 = df1.drop('species', axis=1)
df1.shape
df1.head()
# Defining the attributes
X = df1
target
# label encoding
le = LabelEncoder()
target = le.fit_transform(target)
target
y = target
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.5, random_state=42)

print("Training split input: ", X_train.shape)
print("Testing split input: ", X_test.shape)

# Creating Decision Tree Classifier
dtree = DecisionTreeClassifier(criterion='entropy')
dtree.fit(X_train, y_train)
gini_dtree = DecisionTreeClassifier(criterion='gini')
gini_dtree.fit(X_train, y_train)

# accuracy
# entropy
y_pred = prediction(X_test, dtree)
entropy_accuracy = accuracy_score(y_test, y_pred)
print("Accuracy using entopry: ", entropy_accuracy)

# gini
y_pred_gini = prediction(X_test, gini_dtree)
gini_accuracy = accuracy_score(y_test, y_pred_gini)
print("Accuracy using gini: ", gini_accuracy)

if entropy_accuracy > gini_accuracy:
    print("\nAccuracy with Entropy is higher\n")
else:
    print("\nAccuracy with Gini is higher\n")

# Decision Tree plotting
plt.figure(figsize=(20, 20))
```

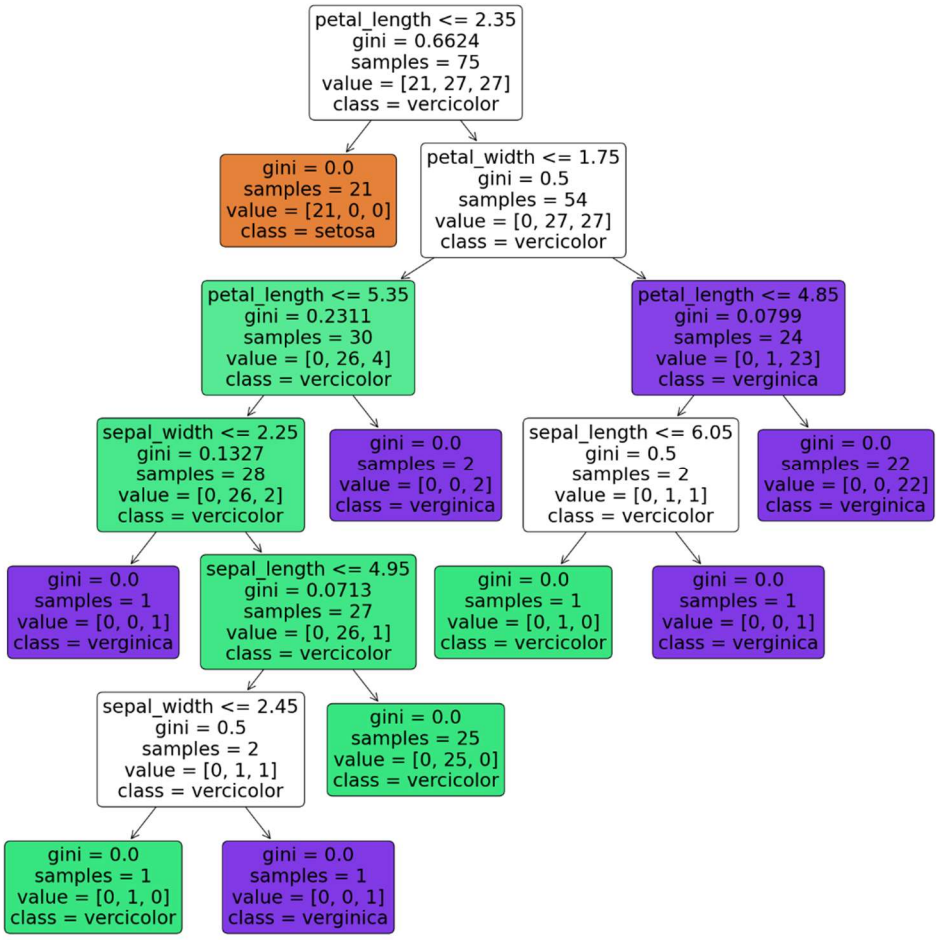
```
dec_tree = plot_tree(decision_tree=dtree, feature_names=df1.columns,
                     class_names=["setosa", "vercicolor", "verginica"],
filled=True, precision=4, rounded=True)

plt.savefig("IrisTree_Entropy.png")

plt.figure(figsize=(20, 20))
gini_dec_tree = plot_tree(decision_tree=gini_dtree,
feature_names=df1.columns,
                          class_names=["setosa", "vercicolor", "verginica"],
filled=True, precision=4, rounded=True)

plt.savefig("IrisTree_Gini.png")

plt.show()
```

Output:

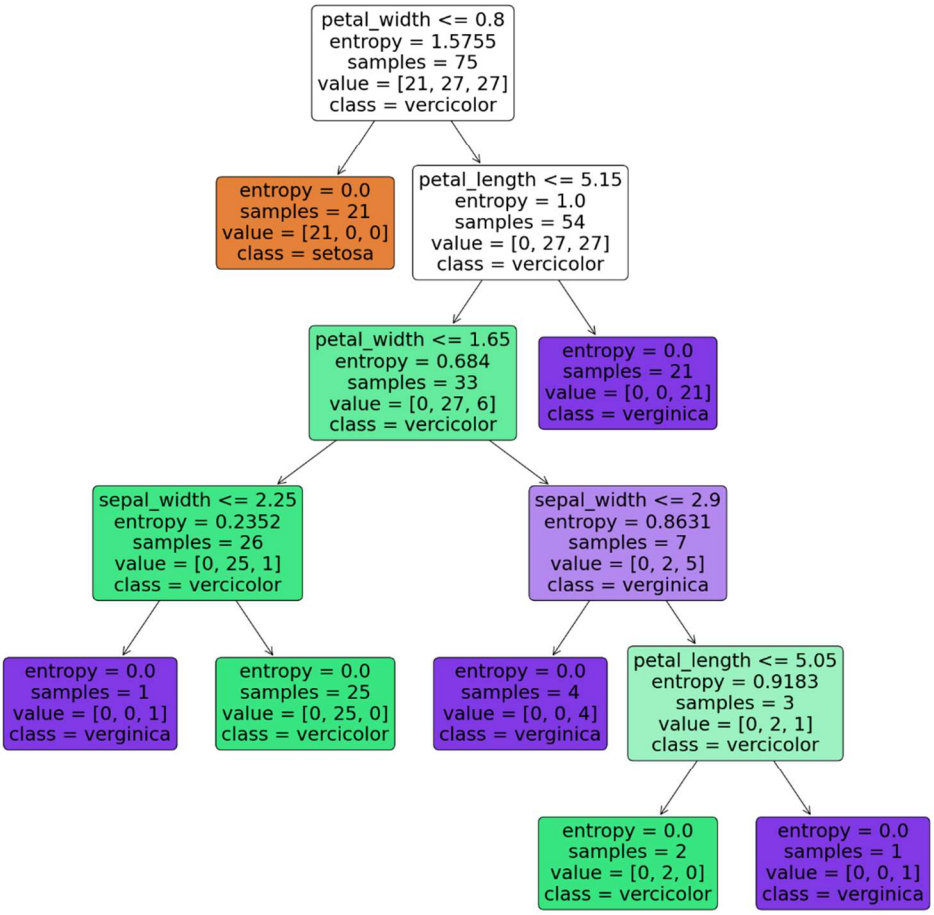
```
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
Training split input:  (75, 4)
Testing split input:  (75, 4)
Predicted values:
[1 0 2 1 1 0 1 2 2 1 2 0 0 0 0 1 2 1 1 2 0 1 0 2 2 2 2 2 0 0 0 0 1 0 0 2 1
 0 0 0 2 1 1 0 0 1 1 2 1 2 1 2 1 0 2 1 0 0 0 2 2 0 0 0 1 0 1 2 0 1 2 0 1 2
 2]
Accuracy using entopry:  0.92
Predicted values:
[1 0 2 1 1 0 1 2 2 1 2 0 0 0 0 1 2 1 1 2 0 2 0 2 2 2 2 2 0 0 0 0 1 0 0 2 1
 0 0 0 2 1 1 0 0 1 1 2 1 2 1 2 1 0 2 1 0 0 0 2 2 0 0 0 1 0 1 2 0 1 2 0 2 2
 2]
Accuracy using gini:  0.9466666666666667

Accuracy with gini index is higher
```

Decision Trees:

Gini



Entropy


petal_width <= 0.8
entropy = 1.5755
samples = 75
value = [21, 27, 27]
class = vercicolor
entropy = 0.0
samples = 21
value = [21, 0, 0]
class = setosa
petal_length <= 5.15
entropy = 1.0
samples = 54
value = [0, 27, 27]
class = vercicolor
petal_width <= 1.65
entropy = 0.684
samples = 33
value = [0, 27, 6]
class = vercicolor
entropy = 0.0
samples = 21
value = [0, 0, 21]
class = verginica
sepal_width <= 2.25
entropy = 0.2352
samples = 26
value = [0, 25, 1]
class = vercicolor
sepal_width <= 2.9
entropy = 0.8631
samples = 7
value = [0, 2, 5]
class = verginica
entropy = 0.0
samples = 1
value = [0, 0, 1]
class = verginica
entropy = 0.0
samples = 25
value = [0, 25, 0]
class = vercicolor
entropy = 0.0
samples = 4
value = [0, 0, 4]
class = verginica
petal_length <= 5.05
entropy = 0.9183
samples = 3
value = [0, 2, 1]
class = vercicolor
entropy = 0.0
samples = 2
value = [0, 2, 0]
class = vercicolor
entropy = 0.0
samples = 1
value = [0, 0, 1]
class = verginica