```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
```

```
# Load the Iris dataset
iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target)
```

```
# Split into training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train Decision Tree model
model = DecisionTreeClassifier(criterion="entropy", max_depth=4, random_state=42)
model.fit(X_train, y_train)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=4, random_state=42)
```

```
# Predictions
y_pred = model.predict(X_test)
```

```
# @title

# Evaluation
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))
```

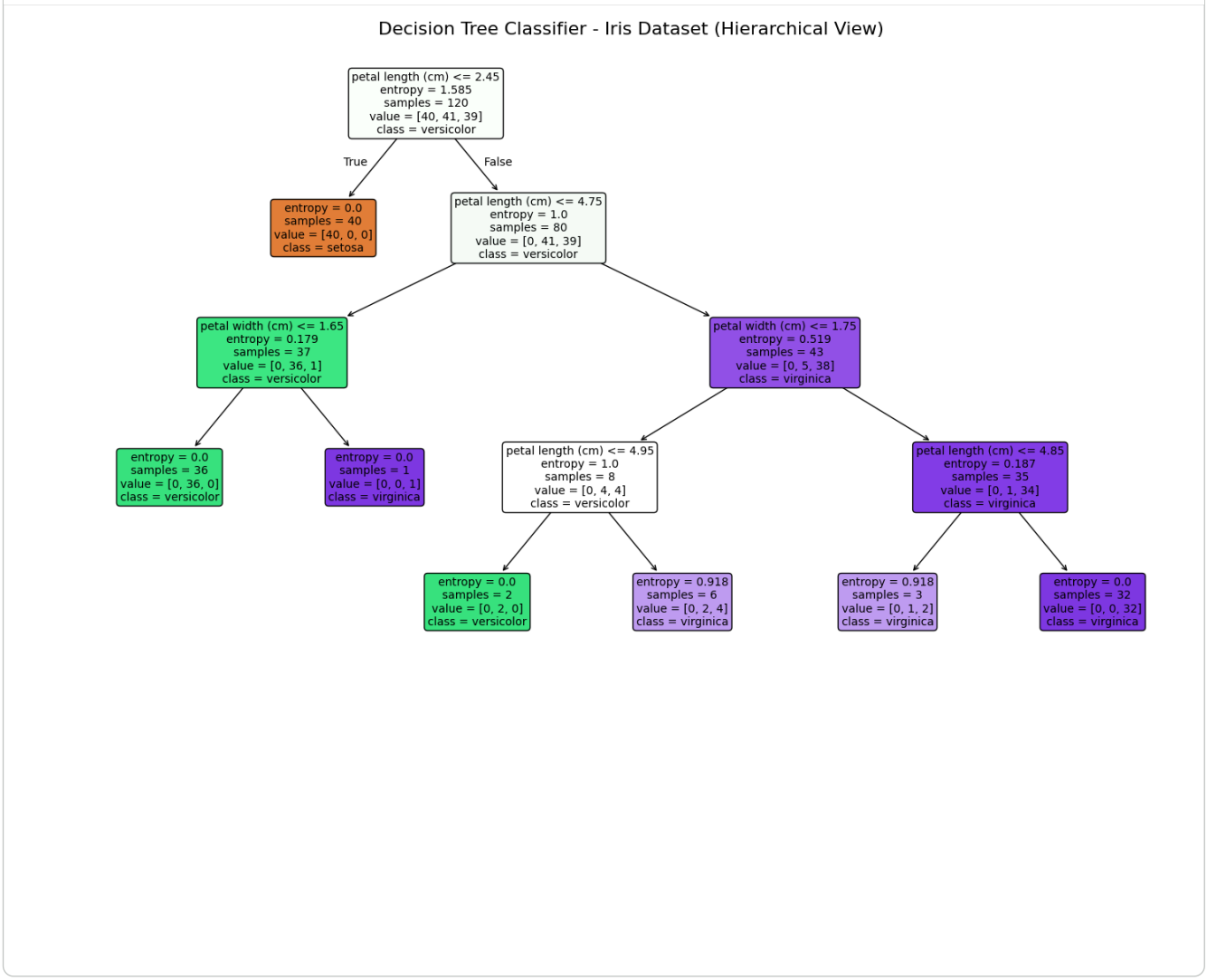
```
Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

```
# === Plot hierarchical Decision Tree ===
plt.figure(figsize=(18,10))
plot_tree(
    model,
    filled=True,
    feature_names=iris.feature_names,
    class_names=iris.target_names,
    rounded=True,
    fontsize=10
)
plt.title("Decision Tree Classifier - Iris Dataset (Hierarchical View)", fontsize=16)
plt.show()
```


Decision Tree Classifier - Iris Dataset (Hierarchical View)
petal length (cm) <= 2.45
entropy = 1.585
samples = 120
value = [40, 41, 39]
class = versicolor
True
False
entropy = 0.0
samples = 40
value = [40, 0, 0]
class = setosa
petal length (cm) <= 4.75
entropy = 1.0
samples = 80
value = [0, 41, 39]
class = versicolor
petal width (cm) <= 1.65
entropy = 0.179
samples = 37
value = [0, 36, 1]
class = versicolor
petal width (cm) <= 1.75
entropy = 0.519
samples = 43
value = [0, 5, 38]
class = virginica
entropy = 0.0
samples = 36
value = [0, 36, 0]
class = versicolor
entropy = 0.0
samples = 1
value = [0, 0, 1]
class = virginica
petal length (cm) <= 4.95
entropy = 1.0
samples = 8
value = [0, 4, 4]
class = versicolor
petal length (cm) <= 4.85
entropy = 0.187
samples = 35
value = [0, 1, 34]
class = virginica
entropy = 0.0
samples = 2
value = [0, 2, 0]
class = versicolor
entropy = 0.918
samples = 6
value = [0, 2, 4]
class = virginica
entropy = 0.918
samples = 3
value = [0, 1, 2]
class = virginica
entropy = 0.0
samples = 32
value = [0, 0, 32]
class = virginica