

```
In [6]: import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")

# Load the dataset
iris_data = pd.read_csv('iris.csv') # Update the path to your dataset

# Separate features and target variable
X = iris_data.drop('variety', axis=1) # Features
y = iris_data['variety'] # Target variable

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2)

# Create a decision tree classifier
dtc = DecisionTreeClassifier(criterion='entropy')

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

# Plot the decision tree
plt.figure(figsize=(20, 16))
plot_tree(dtc, feature_names=iris_data.columns[:-1], class_names=dtc.class_names)
plt.show()

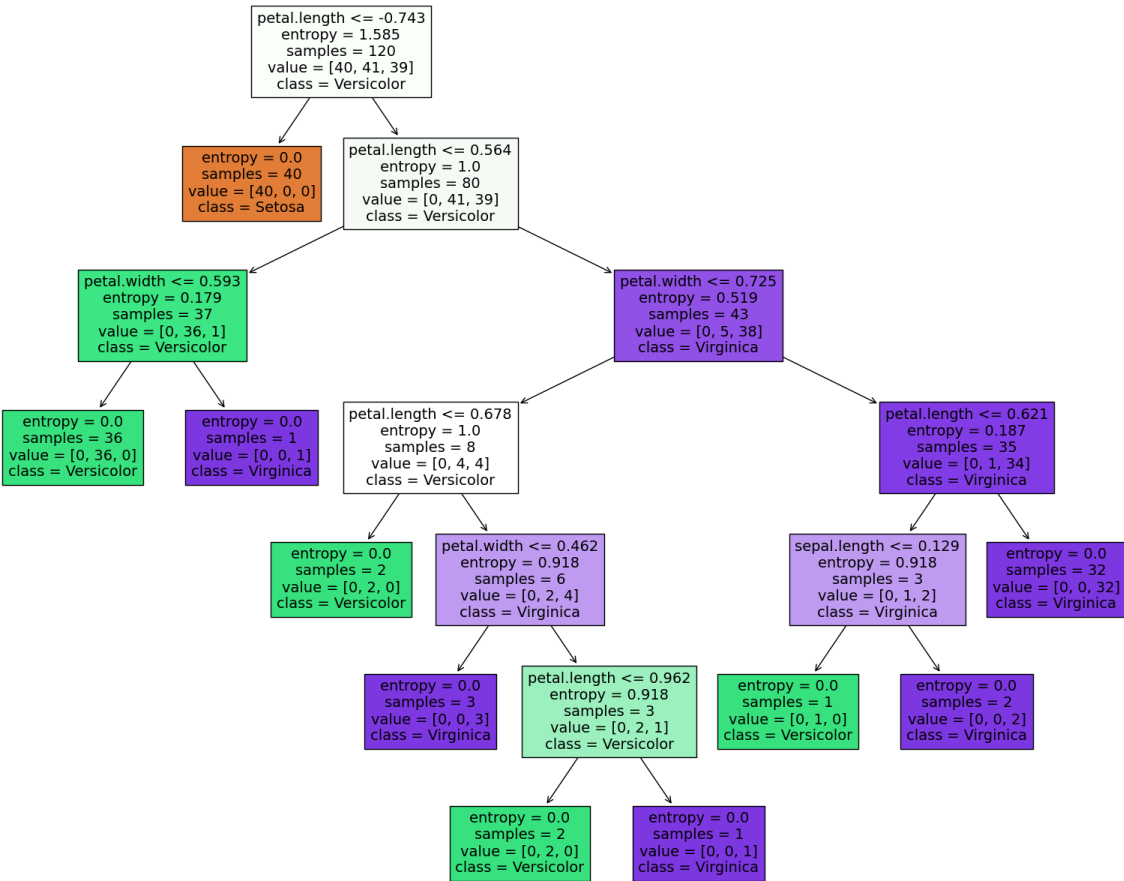
# Make predictions on the testing set
y_pred = dtc.predict(X_test)

# Calculate accuracy
train_accuracy = accuracy_score(y_train, dtc.predict(X_train))
test_accuracy = accuracy_score(y_test, y_pred)

print(f"Training Accuracy: {train_accuracy}")
print(f"Testing Accuracy: {test_accuracy}")

# Predict a single value (example)
sepal_length = float(input("Enter sepal length (cm): "))
sepal_width = float(input("Enter sepal width (cm): "))
petal_length = float(input("Enter petal length (cm): "))
petal_width = float(input("Enter petal width (cm): "))

# Create a list with the user's input
new_data = [[sepal_length, sepal_width, petal_length, petal_width]]
new_data_scaled = scaler.transform(new_data) # Scale the new data
prediction = dtc.predict(new_data_scaled)
print(f"\nPrediction for new data: {prediction[0]}")
```



Training Accuracy: 1.0
Testing Accuracy: 1.0
Prediction for new data: Virginica

```
In [ ]:
In [ ]:
```