LGM - Data Science Internship December 2023

Task-3 Prediction using Decision Tree Algorithm

In [1]:
```
# # Importing the Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# # Import the Dataset
dataset = pd.read_csv("C:/Users/MANOJ S/Downloads/8836201-6f9306ad21398ea43
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

In [22]:
```
# # Splitting the Dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25,
```

In [4]:
```
print(X_train)
```

```
[[5.9 3.  4.2 1.5]
 [5.8 2.6 4.  1.2]
 [6.8 3.  5.5 2.1]
 [4.7 3.2 1.3 0.2]
 [6.9 3.1 5.1 2.3]
 [5.  3.5 1.6 0.6]
 [5.4 3.7 1.5 0.2]
 [5.  2.  3.5 1. ]
 [6.5 3.  5.5 1.8]
 [6.7 3.3 5.7 2.5]
 [6.  2.2 5.  1.5]
 [6.7 2.5 5.8 1.8]
 [5.6 2.5 3.9 1.1]
 [7.7 3.  6.1 2.3]
 [6.3 3.3 4.7 1.6]
 [5.5 2.4 3.8 1.1]
 [6.3 2.7 4.9 1.8]
 [6.3 2.8 5.1 1.5]
 [4.9 2.5 4.5 1.7]
```

In [5]:
```
print(y_train)
```

```
['Versicolor' 'Versicolor' 'Virginica' 'Setosa' 'Virginica' 'Setosa'
 'Setosa' 'Versicolor' 'Virginica' 'Virginica' 'Virginica' 'Virginica'
 'Versicolor' 'Virginica' 'Versicolor' 'Versicolor' 'Virginica'
 'Virginica' 'Virginica' 'Virginica' 'Versicolor' 'Virginica' 'Versicolor'
 'Setosa' 'Virginica' 'Versicolor' 'Versicolor' 'Versicolor' 'Versicolor'
 'Virginica' 'Setosa' 'Setosa' 'Virginica' 'Versicolor' 'Setosa' 'Setosa'
 'Versicolor' 'Setosa' 'Virginica' 'Versicolor' 'Setosa' 'Versicolor'
 'Virginica' 'Versicolor' 'Setosa' 'Virginica' 'Virginica' 'Virginica'
 'Virginica' 'Setosa' 'Setosa' 'Virginica' 'Virginica' 'Setosa'
 'Virginica' 'Setosa' 'Virginica' 'Virginica' 'Setosa' 'Setosa'
 'Virginica' 'Setosa' 'Setosa' 'Setosa' 'Versicolor' 'Virginica'
 'Virginica' 'Setosa' 'Setosa' 'Setosa' 'Versicolor' 'Versicolor' 'Setosa'
 'Setosa' 'Versicolor' 'Setosa' 'Virginica' 'Versicolor' 'Virginica'
 'Versicolor' 'Setosa' 'Virginica' 'Setosa' 'Virginica' 'Setosa' 'Setosa'
 'Virginica' 'Setosa' 'Virginica' 'Versicolor' 'Versicolor' 'Versicolor'
 'Virginica' 'Virginica' 'Versicolor' 'Versicolor' 'Setosa' 'Versicolor'
 'Virginica' 'Virginica' 'Setosa' 'Versicolor' 'Versicolor' 'Versicolor'
 'Versicolor' 'Setosa' 'Setosa' 'Setosa' 'Virginica' 'Versicolor'
 'Virginica' 'Setosa']
```

In [6]:
```
print(X_test)
```

```
[[5.8 2.8 5.1 2.4]
 [6.  2.2 4.  1. ]
 [5.5 4.2 1.4 0.2]
 [7.3 2.9 6.3 1.8]
 [5.  3.4 1.5 0.2]
 [6.3 3.3 6.  2.5]
 [5.  3.5 1.3 0.3]
 [6.7 3.1 4.7 1.5]
 [6.8 2.8 4.8 1.4]
 [6.1 2.8 4.  1.3]
 [6.1 2.6 5.6 1.4]
 [6.4 3.2 4.5 1.5]
 [6.1 2.8 4.7 1.2]
 [6.5 2.8 4.6 1.5]
 [6.1 2.9 4.7 1.4]
 [4.9 3.6 1.4 0.1]
 [6.  2.9 4.5 1.5]
 [5.5 2.6 4.4 1.2]
 [4.8 3.  1.4 0.3]
 [5.4 3.9 1.3 0.4]
 [5.6 2.8 4.9 2. ]
 [5.6 3.  4.5 1.5]
 [4.8 3.4 1.9 0.2]
 [4.4 2.9 1.4 0.2]
 [6.2 2.8 4.8 1.8]
 [4.6 3.6 1.  0.2]
 [5.1 3.8 1.9 0.4]
 [6.2 2.9 4.3 1.3]
 [5.  2.3 3.3 1. ]
 [5.  3.4 1.6 0.4]
 [6.4 3.1 5.5 1.8]
 [5.4 3.  4.5 1.5]
 [5.2 3.5 1.5 0.2]
 [6.1 3.  4.9 1.8]
 [6.4 2.8 5.6 2.2]
 [5.2 2.7 3.9 1.4]
 [5.7 3.8 1.7 0.3]
 [6.  2.7 5.1 1.6]]
```

In [7]:
```
print(y_test)
```

```
['Virginica' 'Versicolor' 'Setosa' 'Virginica' 'Setosa' 'Virginica'
 'Setosa' 'Versicolor' 'Versicolor' 'Versicolor' 'Virginica' 'Versicolor'
 'Versicolor' 'Versicolor' 'Versicolor' 'Setosa' 'Versicolor' 'Versicolor'
 'Setosa' 'Setosa' 'Virginica' 'Versicolor' 'Setosa' 'Setosa' 'Virginica'
 'Setosa' 'Setosa' 'Versicolor' 'Versicolor' 'Setosa' 'Virginica'
 'Versicolor' 'Setosa' 'Virginica' 'Virginica' 'Versicolor' 'Setosa'
 'Versicolor']
```

In [8]:
```
# # Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [9]:
```
print(X_train)
```

```
[[ 1.54399532e-02 -1.19254753e-01  2.25126850e-01  3.55797625e-01]
 [-9.98450310e-02 -1.04039491e+00  1.13559562e-01 -2.98410911e-02]
 [ 1.05300481e+00 -1.19254753e-01  9.50314227e-01  1.12707506e+00]
 [-1.36797986e+00  3.41315328e-01 -1.39259884e+00 -1.31530348e+00]
 [ 1.16828980e+00  1.11030287e-01  7.27179649e-01  1.38416753e+00]
 [-1.02212490e+00  1.03217045e+00 -1.22524790e+00 -8.01118523e-01]
 [-5.60984968e-01  1.49274053e+00 -1.28103155e+00 -1.31530348e+00]
 [-1.02212490e+00 -2.42210516e+00 -1.65358660e-01 -2.86933568e-01]
 [ 7.07149859e-01 -1.19254753e-01  9.50314227e-01  7.41436341e-01]
 [ 9.37719827e-01  5.71600368e-01  1.06188152e+00  1.64126001e+00]
 [ 1.30724937e-01 -1.96153508e+00  6.71396005e-01  3.55797625e-01]
 [ 9.37719827e-01 -1.27067995e+00  1.11766516e+00  7.41436341e-01]
 [-3.30414999e-01 -1.27067995e+00  5.77759173e-02 -1.58387330e-01]
 [ 2.09056967e+00 -1.19254753e-01  1.28501609e+00  1.38416753e+00]
 [ 4.76579890e-01  5.71600368e-01  5.04045072e-01  4.84343863e-01]
 [-4.45699984e-01 -1.50096499e+00  1.99227301e-03 -1.58387330e-01]
 [ 4.76579890e-01 -8.10109874e-01  6.15612361e-01  7.41436341e-01]
 [ 4.76579890e-01 -5.79824834e-01  7.27179649e-01  3.55797625e-01]
 [-1.13740989e+00 -1.27067995e+00  3.92477783e-01  6.12890102e-01]
```

In [10]:
```
print(X_test)
```

```
[[-0.09984503 -0.57982483  0.72717965  1.51271377]
 [ 0.13072494 -1.96153508  0.11355956 -0.28693357]
 [-0.44569998  2.64416573 -1.33681519 -1.31530348]
 [ 1.62942973 -0.34953979  1.39658338  0.74143634]
 [-1.0221249   0.80188541 -1.28103155 -1.31530348]
 [ 0.47657989  0.57160037  1.22923245  1.64126001]
 [-1.0221249   1.03217045 -1.39259884 -1.18675724]
 [ 0.93771983  0.11103029  0.50404507  0.35579762]
 [ 1.05300481 -0.57982483  0.55982872  0.22725139]
 [ 0.24600992 -0.57982483  0.11355956  0.09870515]
 [ 0.24600992 -1.04039491  1.00609787  0.22725139]
 [ 0.59186487  0.34131533  0.39247778  0.35579762]
 [ 0.24600992 -0.57982483  0.50404507 -0.02984109]
 [ 0.70714986 -0.57982483  0.44826143  0.35579762]
 [ 0.24600992 -0.34953979  0.50404507  0.22725139]
 [-1.13740989  1.26245549 -1.33681519 -1.44384972]
 [ 0.13072494 -0.34953979  0.39247778  0.35579762]
 [-0.44569998 -1.04039491  0.33669414 -0.02984109]
 [-1.25269487 -0.11925475 -1.33681519 -1.18675724]
 [-0.56098497  1.95331061 -1.39259884 -1.058211  ]
 [-0.330415   -0.57982483  0.61561236  0.99852882]
 [-0.330415   -0.11925475  0.39247778  0.35579762]
 [-1.25269487  0.80188541 -1.05789697 -1.31530348]
 [-1.71383481 -0.34953979 -1.33681519 -1.31530348]
 [ 0.36129491 -0.57982483  0.55982872  0.74143634]
 [-1.48326484  1.26245549 -1.55994977 -1.31530348]
 [-0.90683992  1.72302557 -1.05789697 -1.058211  ]
 [ 0.36129491 -0.34953979  0.28091049  0.09870515]
 [-1.0221249  -1.73125004 -0.27692595 -0.28693357]
 [-1.0221249   0.80188541 -1.2252479  -1.058211  ]
 [ 0.59186487  0.11103029  0.95031423  0.74143634]
 [-0.56098497 -0.11925475  0.39247778  0.35579762]
 [-0.79155494  1.03217045 -1.28103155 -1.31530348]
 [ 0.24600992 -0.11925475  0.61561236  0.74143634]
 [ 0.59186487 -0.57982483  1.00609787  1.25562129]
 [-0.79155494 -0.81010987  0.05777592  0.22725139]
 [-0.21513002  1.72302557 -1.16946426 -1.18675724]
 [ 0.13072494 -0.81010987  0.72717965  0.48434386]]
```

In [11]:
```
# # Training the Decision Tree Classification Model on the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0
classifier.fit(X_train, y_train)
```

Out[11]:
```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

In [12]:
```
# # Predicting the Test Result set

y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_t
```

```
[['Virginica' 'Virginica']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Setosa' 'Setosa']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Virginica' 'Virginica']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Setosa' 'Setosa']
 ['Setosa' 'Setosa']
 ['Versicolor' 'Versicolor']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Virginica' 'Virginica']
 ['Virginica' 'Virginica']
 ['Versicolor' 'Versicolor']
 ['Setosa' 'Setosa']
 ['Virginica' 'Versicolor']]
```

In [13]:
```
# # Making the Confusion Matrix

from sklearn.metrics import confusion_matrix , accuracy_score
cm = confusion_matrix(y_test,y_pred)
print(cm)
accuracy_score(y_test,y_pred)
```

```
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
```

Out[13]: 0.9736842105263158

In [14]:
```
# Accuracy is 100 %

# # Visualising things
import seaborn as sns
```

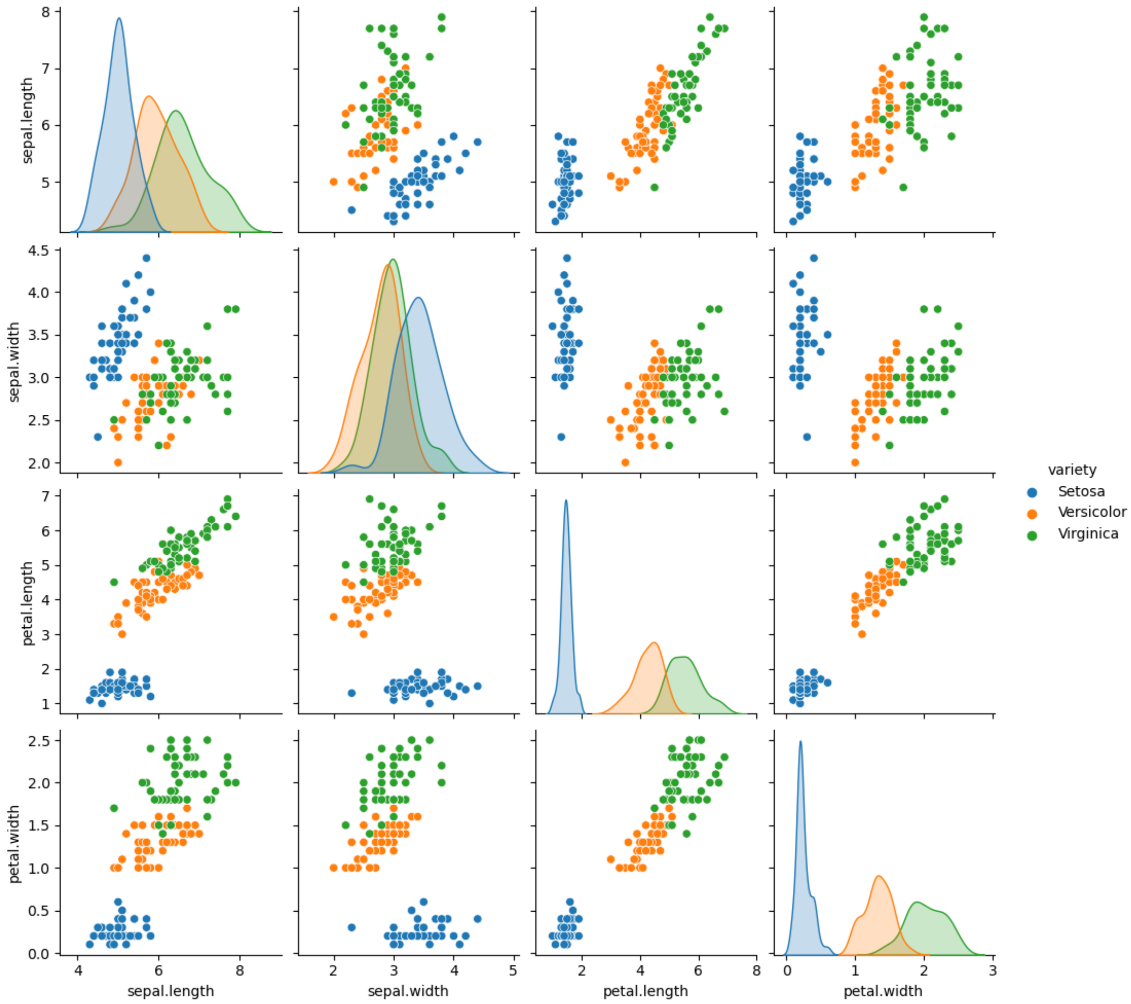
In [15]:
```
dataset
```

Out[15]:

	sepal.length	sepal.width	petal.length	petal.width	variety
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

In [19]:
```
sns.pairplot(data=dataset , hue='variety')
plt.show()
```



In [21]:
```
col = dataset.columns[:-1]
classes = dataset['variety'].unique().tolist()
from sklearn.tree import plot_tree
plt.figure(figsize=(16,10))
plot_tree(classifier, feature_names=col, class_names=classes,filled=True)
```

Out[21]:
```
[Text(0.4, 0.9, 'petal.width <= -0.544\nentropy = 1.581\nsamples = 112\nva
lue = [37, 34, 41]\nclass = Virginica'),
 Text(0.3, 0.7, 'entropy = 0.0\nsamples = 37\nvalue = [37, 0, 0]\nclass =
Setosa'),
 Text(0.5, 0.7, 'petal.length <= 0.644\nentropy = 0.994\nsamples = 75\nval
ue = [0, 34, 41]\nclass = Virginica'),
 Text(0.2, 0.5, 'petal.width <= 0.549\nentropy = 0.414\nsamples = 36\nvalu
e = [0, 33, 3]\nclass = Versicolor'),
 Text(0.1, 0.3, 'entropy = 0.0\nsamples = 32\nvalue = [0, 32, 0]\nclass =
Versicolor'),
 Text(0.3, 0.3, 'sepal.width <= 0.111\nentropy = 0.811\nsamples = 4\nvalue
= [0, 1, 3]\nclass = Virginica'),
 Text(0.2, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = Vi
rginica'),
 Text(0.4, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = Ve
rsicolor'),
 Text(0.8, 0.5, 'petal.width <= 0.677\nentropy = 0.172\nsamples = 39\nvalu
e = [0, 1, 38]\nclass = Virginica'),
 Text(0.7, 0.3, 'petal.width <= 0.549\nentropy = 0.811\nsamples = 4\nvalue
= [0, 1, 3]\nclass = Virginica'),
 Text(0.6, 0.1, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0, 3]\nclass = Vi
rginica'),
 Text(0.8, 0.1, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]\nclass = Ve
rsicolor'),
 Text(0.9, 0.3, 'entropy = 0.0\nsamples = 35\nvalue = [0, 0, 35]\nclass =
Virginica')]
```

In [ ]: