Task 3- GRIP at Sparks Foundation

Decision Tree Algorithm on Iris Dataset

To create the decision tree classifier and visualizing it

In [1]:

```
# Importing libraries in Python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv("D://Dipali//Datasets//iris.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
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

In [4]:

```
df.tail()
```

Out[4]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
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

```
In [5]: df.columns
```

```
Out[5]: Index(['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width',
               'Species'],
              dtype='object')
```

```
In [6]: df.shape
```

```
Out[6]: (150, 5)
```

```
In [7]: df.isnull().any()
```

```
Out[7]: Sepal.Length    False
        Sepal.Width     False
        Petal.Length    False
        Petal.Width     False
        Species         False
        dtype: bool
```

```
In [8]: df.dtypes
```

```
Out[8]: Sepal.Length    float64
        Sepal.Width     float64
        Petal.Length    float64
        Petal.Width     float64
        Species          object
        dtype: object
```

In [9]:

```
df.describe()
```

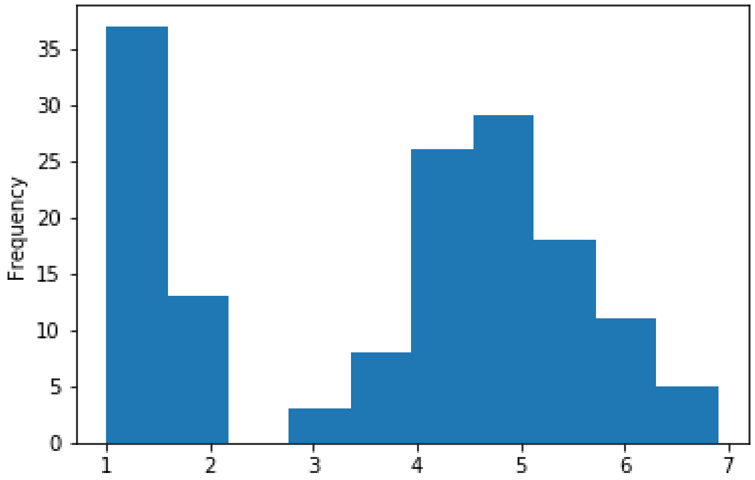
Out[9]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Exploratory Data Analysis

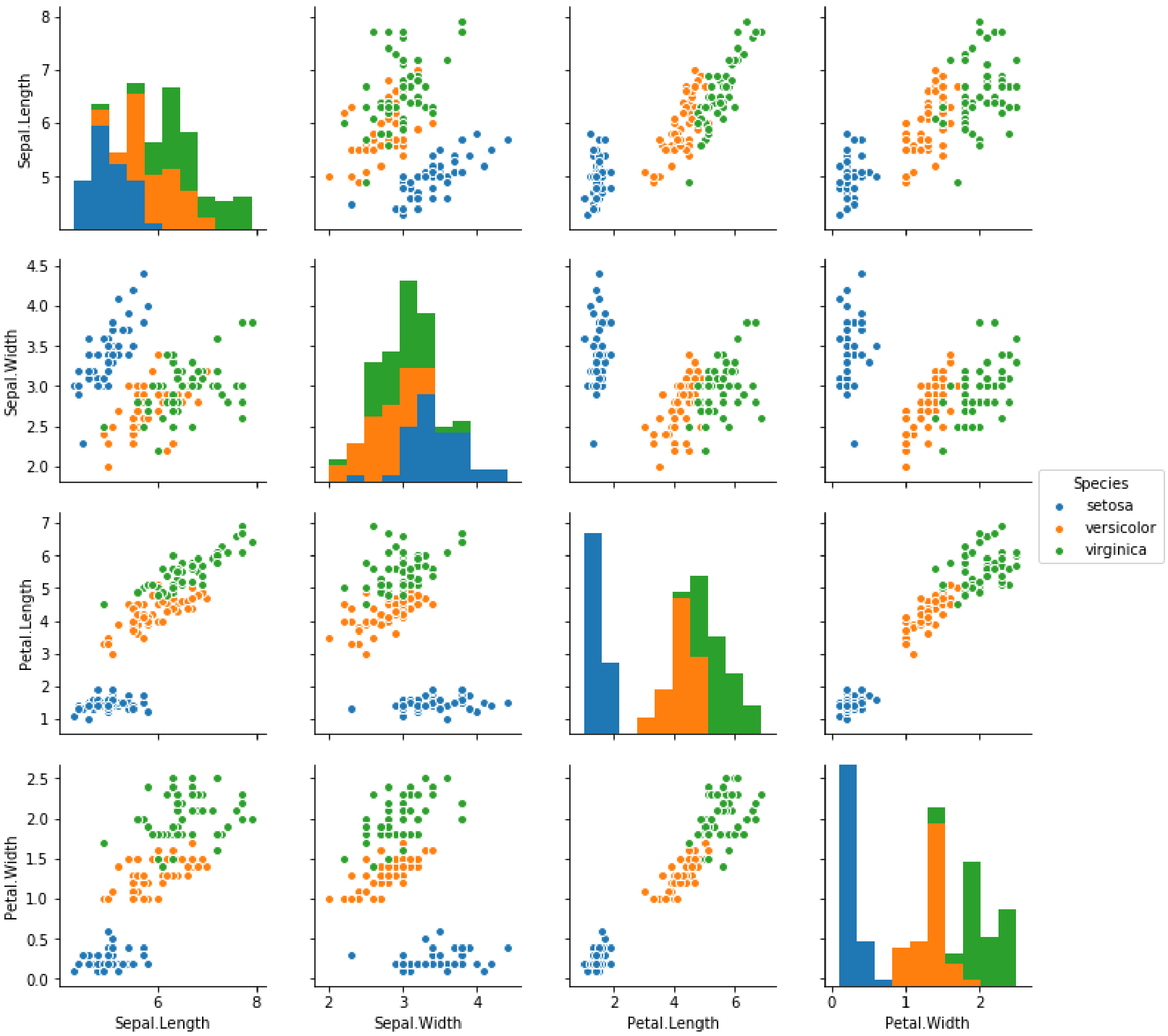
In [10]:

```
df['Petal.Length'].plot.hist()
plt.show()
```



```
In [11]: sns.pairplot(df, hue='Species')
```

```
Out[11]: <seaborn.axisgrid.PairGrid at 0x98a3438>
```



In [13]:

```
numeric_col=df.select_dtypes(exclude=['object'])
numeric_col
```

Out[13]:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
5	5.4	3.9	1.7	0.4
6	4.6	3.4	1.4	0.3
7	5.0	3.4	1.5	0.2
8	4.4	2.9	1.4	0.2
9	4.9	3.1	1.5	0.1
10	5.4	3.7	1.5	0.2
11	4.8	3.4	1.6	0.2
12	4.8	3.0	1.4	0.1
13	4.3	3.0	1.1	0.1
14	5.8	4.0	1.2	0.2
15	5.7	4.4	1.5	0.4
16	5.4	3.9	1.3	0.4
17	5.1	3.5	1.4	0.3
18	5.7	3.8	1.7	0.3
19	5.1	3.8	1.5	0.3
20	5.4	3.4	1.7	0.2
21	5.1	3.7	1.5	0.4

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
22	4.6	3.6	1.0	0.2
23	5.1	3.3	1.7	0.5
24	4.8	3.4	1.9	0.2
25	5.0	3.0	1.6	0.2
26	5.0	3.4	1.6	0.4
27	5.2	3.5	1.5	0.2
28	5.2	3.4	1.4	0.2
29	4.7	3.2	1.6	0.2
...	...	...	...	...
120	6.9	3.2	5.7	2.3
121	5.6	2.8	4.9	2.0
122	7.7	2.8	6.7	2.0
123	6.3	2.7	4.9	1.8
124	6.7	3.3	5.7	2.1
125	7.2	3.2	6.0	1.8
126	6.2	2.8	4.8	1.8
127	6.1	3.0	4.9	1.8
128	6.4	2.8	5.6	2.1
129	7.2	3.0	5.8	1.6
130	7.4	2.8	6.1	1.9
131	7.9	3.8	6.4	2.0
132	6.4	2.8	5.6	2.2

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
133	6.3	2.8	5.1	1.5
134	6.1	2.6	5.6	1.4
135	7.7	3.0	6.1	2.3
136	6.3	3.4	5.6	2.4
137	6.4	3.1	5.5	1.8
138	6.0	3.0	4.8	1.8
139	6.9	3.1	5.4	2.1
140	6.7	3.1	5.6	2.4
141	6.9	3.1	5.1	2.3
142	5.8	2.7	5.1	1.9
143	6.8	3.2	5.9	2.3
144	6.7	3.3	5.7	2.5
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [14]:

```
charcter_col =df.select_dtypes(include=['object'])
charcter_col
```

Out[14]:

	Species

	Species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa
5	setosa
6	setosa
7	setosa
8	setosa
9	setosa
10	setosa
11	setosa
12	setosa
13	setosa
14	setosa
15	setosa
16	setosa
17	setosa
18	setosa
19	setosa
20	setosa
21	setosa

	Species
22	setosa
23	setosa
24	setosa
25	setosa
26	setosa
27	setosa
28	setosa
29	setosa
...	...
120	virginica
121	virginica
122	virginica
123	virginica
124	virginica
125	virginica
126	virginica
127	virginica
128	virginica
129	virginica
130	virginica
131	virginica
132	virginica

	Species
133	virginica
134	virginica
135	virginica
136	virginica
137	virginica
138	virginica
139	virginica
140	virginica
141	virginica
142	virginica
143	virginica
144	virginica
145	virginica
146	virginica
147	virginica
148	virginica
149	virginica

150 rows × 1 columns

In [ ]:

```
# Plot normal probability plot
```

In [15]:

```
numeric_col.head()
charcter_col.head()
```

Out[15]:

	Species
0	setosa
1	setosa
2	setosa
3	setosa
4	setosa

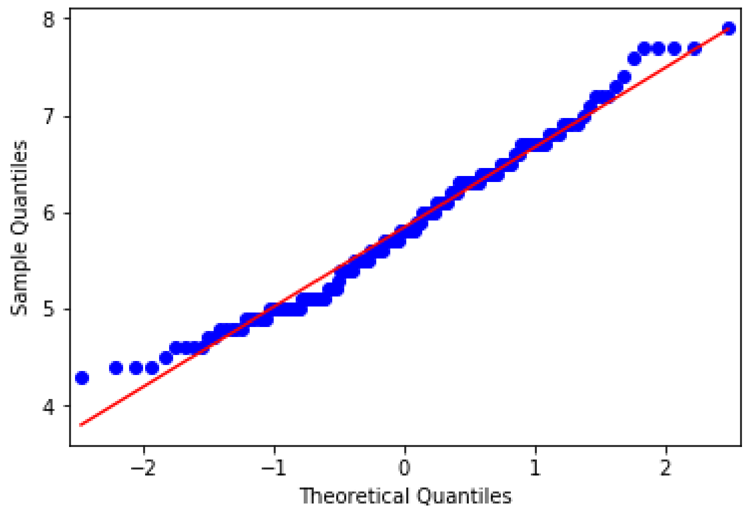
In [16]:

```
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot
```

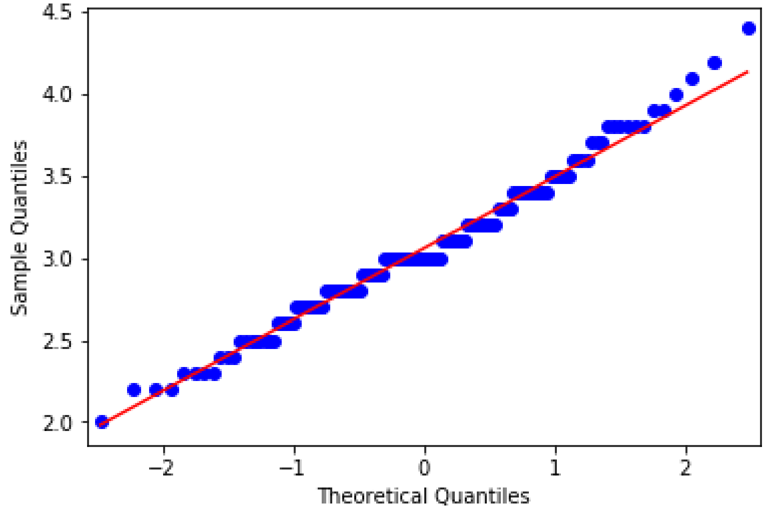
In [17]:

```
# plot one by one column in dataframe
for i in numeric_col:
    print("--------------------")
    print(i)
    # To check the normality by graph
    qqplot(df[i], line='s')
    pyplot.show()
```

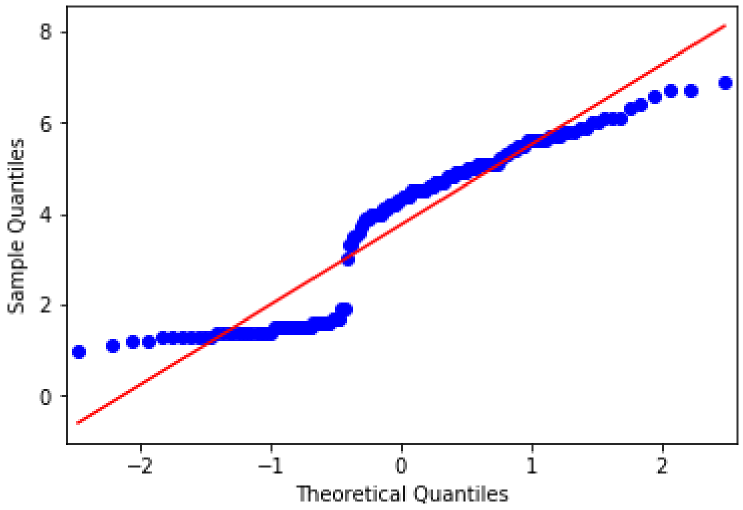
```
--------------------
Sepal.Length
```



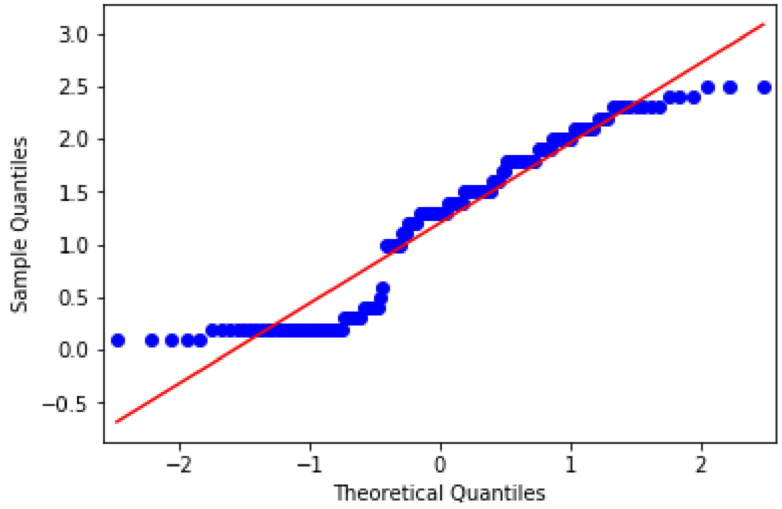
```
--------------------
Sepal.Width
```



```
--------------------
Petal.Length
```

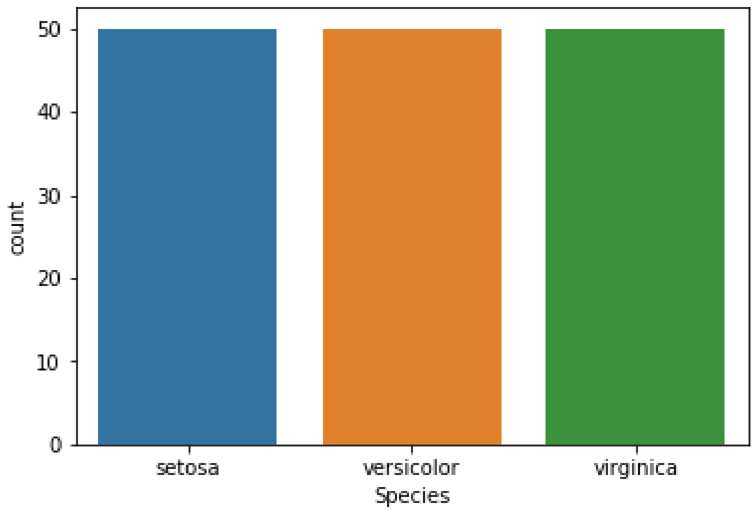

Sample Quantiles
8
6
4
2
0
−2
−1
0
1
2
Theoretical Quantiles

```
--------------------
Petal.Width
```


Sample Quantiles
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−2
−1
0
1
2
Theoretical Quantiles

In [18]:
```
sns.countplot(df['Species'])
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xb3c19b0>
```



Model Building and Evaluation

In [19]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [20]:
```
X=df.drop('Species',axis=1)
y=df['Species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2
)
```

In [21]:
```
tree=DecisionTreeClassifier()
tree.fit(X_train,y_train)
```

```
predic=tree.predict(X_test)
print("Decision Tree is ready")
```

```
Decision Tree is ready
```

Classification Report

In [22]:

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
print(classification_report(y_test,predic))
```

```
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00         8
  versicolor       1.00      0.75      0.86         8
   virginica       0.88      1.00      0.93        14

    accuracy                           0.93        30
   macro avg       0.96      0.92      0.93        30
weighted avg       0.94      0.93      0.93        30
```

In [23]:

```
print(confusion_matrix(y_test,predic))
```

```
[[ 8  0  0]
 [ 0  6  2]
 [ 0  0 14]]
```

In [24]:

```
print(accuracy_score(y_test,predic))
```

```
0.9333333333333333
```

In [0]:

```
# Install required libraries
!pip install pydotplus
!apt-get install graphviz -y
```

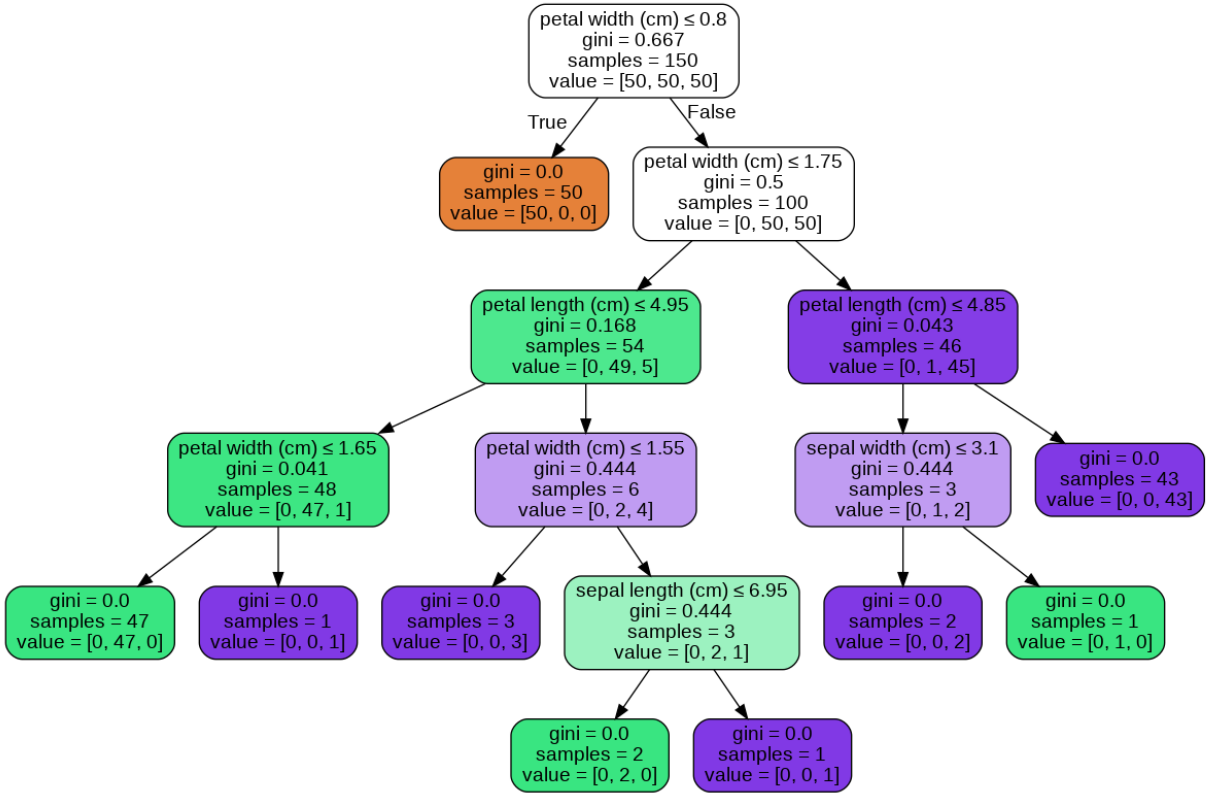
Decision tree visualization

In [0]:

```
# Import necessary libraries for graph viz
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus

# Visualize the graph
dot_data = StringIO()
export_graphviz(dtree, out_file=dot_data, feature_names=iris.feature_names,
                filled=True, rounded=True,
                special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

Out[0]:


petal width (cm) ≤ 0.8
gini = 0.667
samples = 150
value = [50, 50, 50]
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
petal length (cm) ≤ 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
petal length (cm) ≤ 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
petal width (cm) ≤ 1.65
gini = 0.041
samples = 48
value = [0, 47, 1]
petal width (cm) ≤ 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
sepal width (cm) ≤ 3.1
gini = 0.444
samples = 3
value = [0, 1, 2]
gini = 0.0
samples = 43
value = [0, 0, 43]
gini = 0.0
samples = 47
value = [0, 47, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 3
value = [0, 0, 3]
sepal length (cm) ≤ 6.95
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]