Decision_tree_16_09_2025

September 16, 2025

[69]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import␣
  ↪classification_report,confusion_matrix,accuracy_score
from sklearn.tree import plot_tree
```

[37]:

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

[37]:

```
   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
```

[38]:

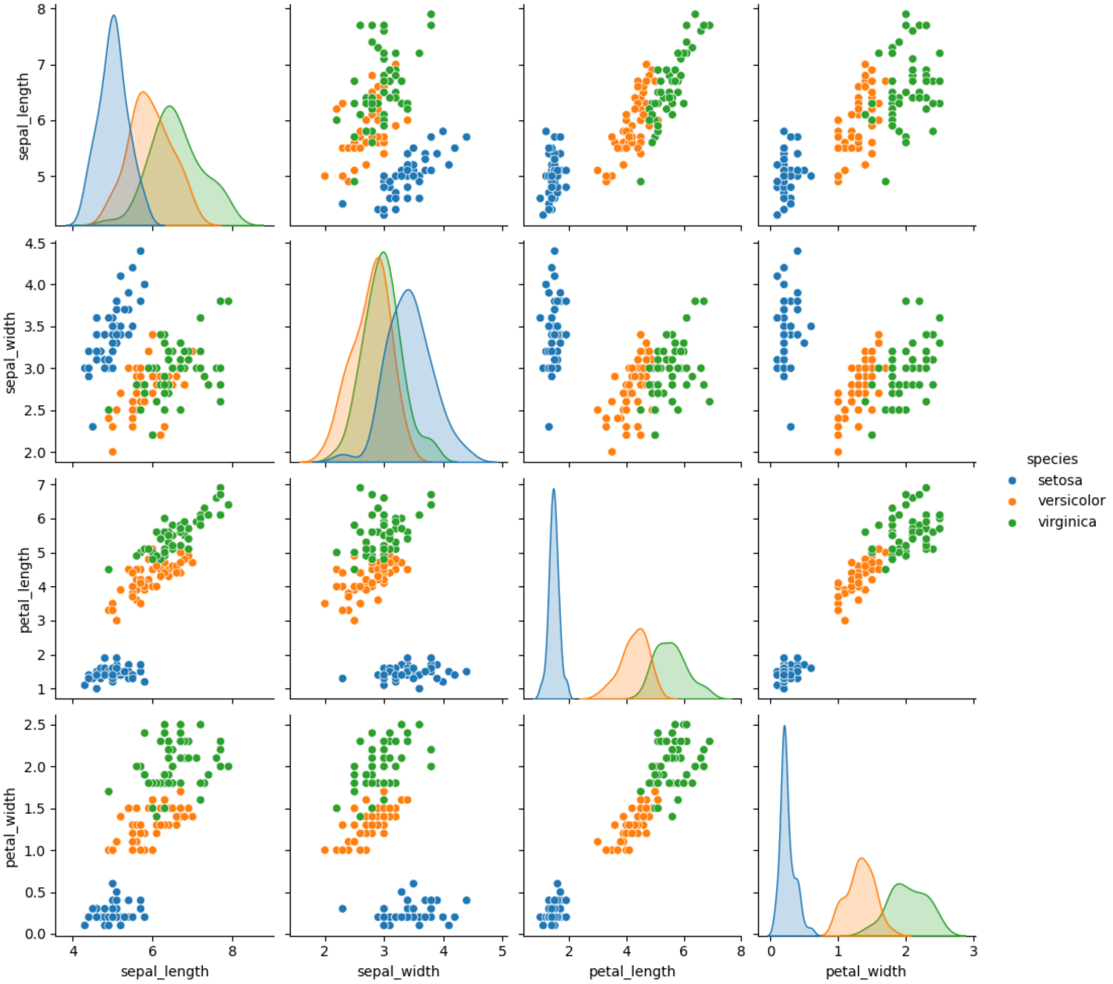
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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
```

[39]:

```
sns.pairplot(data = df ,hue='species')
```

```
[39]: <seaborn.axisgrid.PairGrid at 0x1e80e47a520>
```

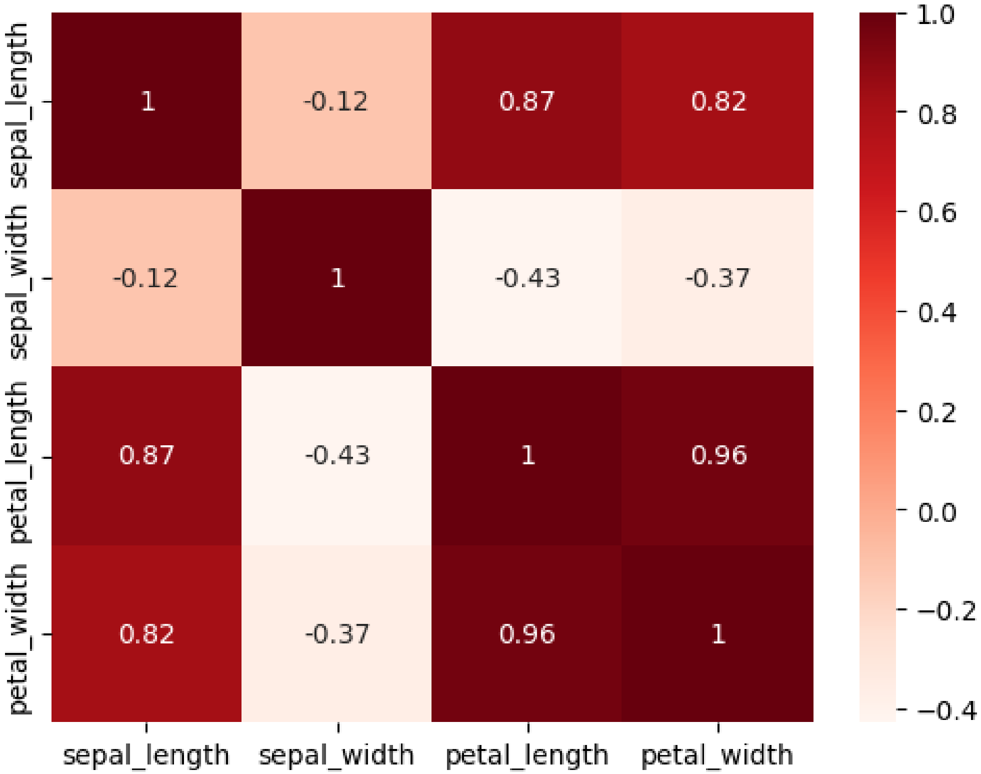


```
[35]: df.isnull().sum()
```

```
[35]: sepal_length    0
      sepal_width     0
      petal_length    0
      petal_width     0
      species         0
      dtype: int64
```

```
[30]: sns.heatmap(df.drop('species',axis = 1).corr(),annot = True,cmap = 'Reds')
```

```
[30]: <Axes: >
```



[42]:
```
traget = df['species']
df1 = df.copy()
df1.drop('species',axis =1, inplace=True)
```

[45]:
```
a = df1
print(a.head())
```

```
   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
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object
```

[52]:
```
le = LabelEncoder()
traget = le.fit_transform(traget)
```

```
traget
```

```
[52]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[60]: b = traget
      b
```

```
[60]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
[6]: x = df.iloc[:,1:4]
     x
```

```
[6]:      sepal_width  petal_length  petal_width
     0            3.5           1.4          0.2
     1            3.0           1.4          0.2
     2            3.2           1.3          0.2
     3            3.1           1.5          0.2
     4            3.6           1.4          0.2
     ..           …             …            …
     145          3.0           5.2          2.3
     146          2.5           5.0          1.9
     147          3.0           5.2          2.0
     148          3.4           5.4          2.3
     149          3.0           5.1          1.8

     [150 rows x 3 columns]
```

```
[7]: y = df.iloc[:,-1]
     y
```

```
[7]: 0         setosa
     1         setosa
     2         setosa
     3         setosa
     4         setosa
              …
```

```
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: species, Length: 150, dtype: object
```

[61]:

```
le = LabelEncoder()
y = le.fit_transform(y)
y
```

[61]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

[64]:

```
x_train,x_test,y_train,y_test = train_test_split(a,b,test_size = 0.
↪2,random_state = 42)#random_state is the produces the same sequence of the␣
↪numbers every time it is runned
```

[47]:

```
x_train
```

[47]:

```
     sepal_length  sepal_width  petal_length  petal_width
48            5.3          3.7           1.5          0.2
23            5.1          3.3           1.7          0.5
94            5.6          2.7           4.2          1.3
139           6.9          3.1           5.4          2.1
102           7.1          3.0           5.9          2.1
..            …            …            …            …
120           6.9          3.2           5.7          2.3
143           6.8          3.2           5.9          2.3
126           6.2          2.8           4.8          1.8
142           5.8          2.7           5.1          1.9
91            6.1          3.0           4.6          1.4

[120 rows x 4 columns]
```

[63]:

```
y_train
```

[63]:

```
array([0, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0, 0, 1, 2, 2, 1, 2, 1, 2,
       1, 0, 2, 1, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 1, 2, 0, 1, 2, 0, 2, 2,
       1, 1, 2, 1, 0, 1, 2, 0, 0, 1, 1, 0, 2, 0, 0, 1, 1, 2, 1, 2, 2, 1,
       0, 0, 2, 2, 0, 0, 0, 1, 2, 0, 2, 2, 0, 1, 1, 2, 1, 2, 0, 2, 1, 2,
       1, 1, 1, 0, 1, 1, 0, 1, 2, 2, 0, 1, 2, 2, 0, 2, 0, 1, 2, 2, 1, 2,
       1, 1, 2, 2, 0, 1, 2, 0, 1, 2])
```

[49]:

```
x_test
```

[49]:

```
     sepal_length  sepal_width  petal_length  petal_width
101           5.8          2.7           5.1          1.9
63            6.1          2.9           4.7          1.4
106           4.9          2.5           4.5          1.7
22            4.6          3.6           1.0          0.2
115           6.4          3.2           5.3          2.3
95            5.7          3.0           4.2          1.2
104           6.5          3.0           5.8          2.2
56            6.3          3.3           4.7          1.6
90            5.5          2.6           4.4          1.2
59            5.2          2.7           3.9          1.4
111           6.4          2.7           5.3          1.9
130           7.4          2.8           6.1          1.9
110           6.5          3.2           5.1          2.0
80            5.5          2.4           3.8          1.1
149           5.9          3.0           5.1          1.8
4             5.0          3.6           1.4          0.2
41            4.5          2.3           1.3          0.3
122           7.7          2.8           6.7          2.0
3             4.6          3.1           1.5          0.2
46            5.1          3.8           1.6          0.2
62            6.0          2.2           4.0          1.0
45            4.8          3.0           1.4          0.3
38            4.4          3.0           1.3          0.2
105           7.6          3.0           6.6          2.1
31            5.4          3.4           1.5          0.4
73            6.1          2.8           4.7          1.2
69            5.6          2.5           3.9          1.1
67            5.8          2.7           4.1          1.0
40            5.0          3.5           1.3          0.3
136           6.3          3.4           5.6          2.4
```

[50]:

```
y_test
```

[50]:

```
101     virginica
63     versicolor
106     virginica
22         setosa
115     virginica
95     versicolor
104     virginica
56     versicolor
90     versicolor
```

```
59     versicolor
111     virginica
130     virginica
110     virginica
80     versicolor
149     virginica
4          setosa
41         setosa
122     virginica
3          setosa
46         setosa
62     versicolor
45         setosa
38         setosa
105     virginica
31         setosa
73     versicolor
69     versicolor
67     versicolor
40         setosa
136     virginica
Name: species, dtype: object
```

[51]:
```
dc = DecisionTreeClassifier()
```

[65]:
```
dc.fit(x_train,y_train)
```

[65]:
```
DecisionTreeClassifier()
```

[66]:
```
y_pred = dc.predict(x_test)
```

[67]:
```
print("Classification report : ",classification_report(y_test,y_pred))
```

```
Classification report :                precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       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
```

[70]:
```
print("Accuracy Report : ",accuracy_score(y_test,y_pred))
```

```
Accuracy Report :  1.0
```

[72]:

```
cm=confusion_matrix
```

[73]:

```
c = cm(y_test,y_pred)
```

[74]:

```
c
```

[74]:

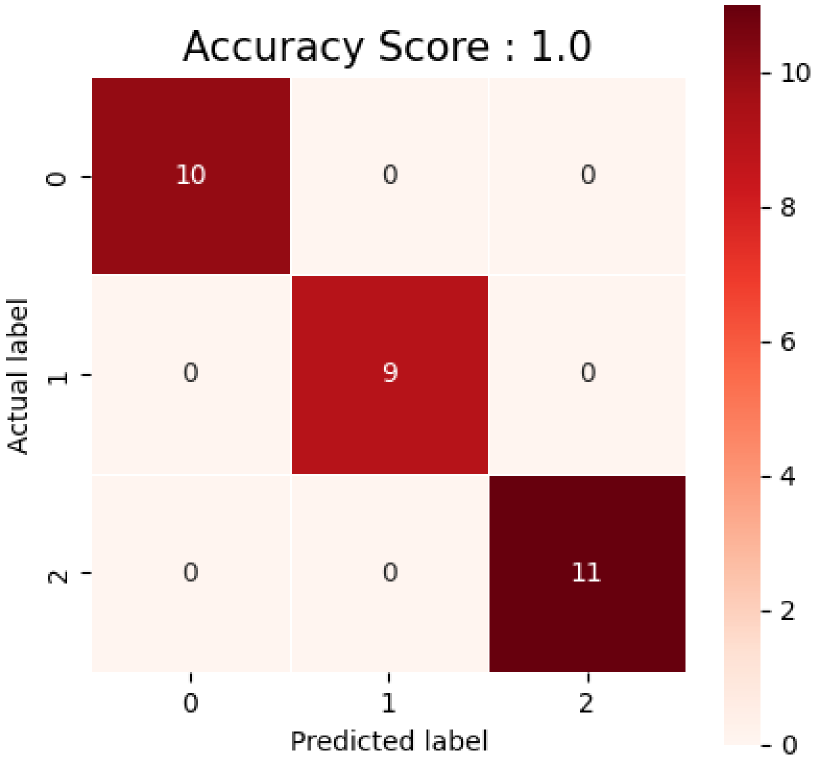
```
array([[10,  0,  0],
       [ 0,  9,  0],
       [ 0,  0, 11]])
```

[76]:

```
plt.figure(figsize=(5,5))
sns.heatmap(data = c,linewidth = .5,annot = True,square = True, cmap = 'Reds')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
all_sample_title = 'Accuracy Score : {0}'.format(dc.score(x_test,y_test))
plt.title(all_sample_title,size = 15)
```

[76]:

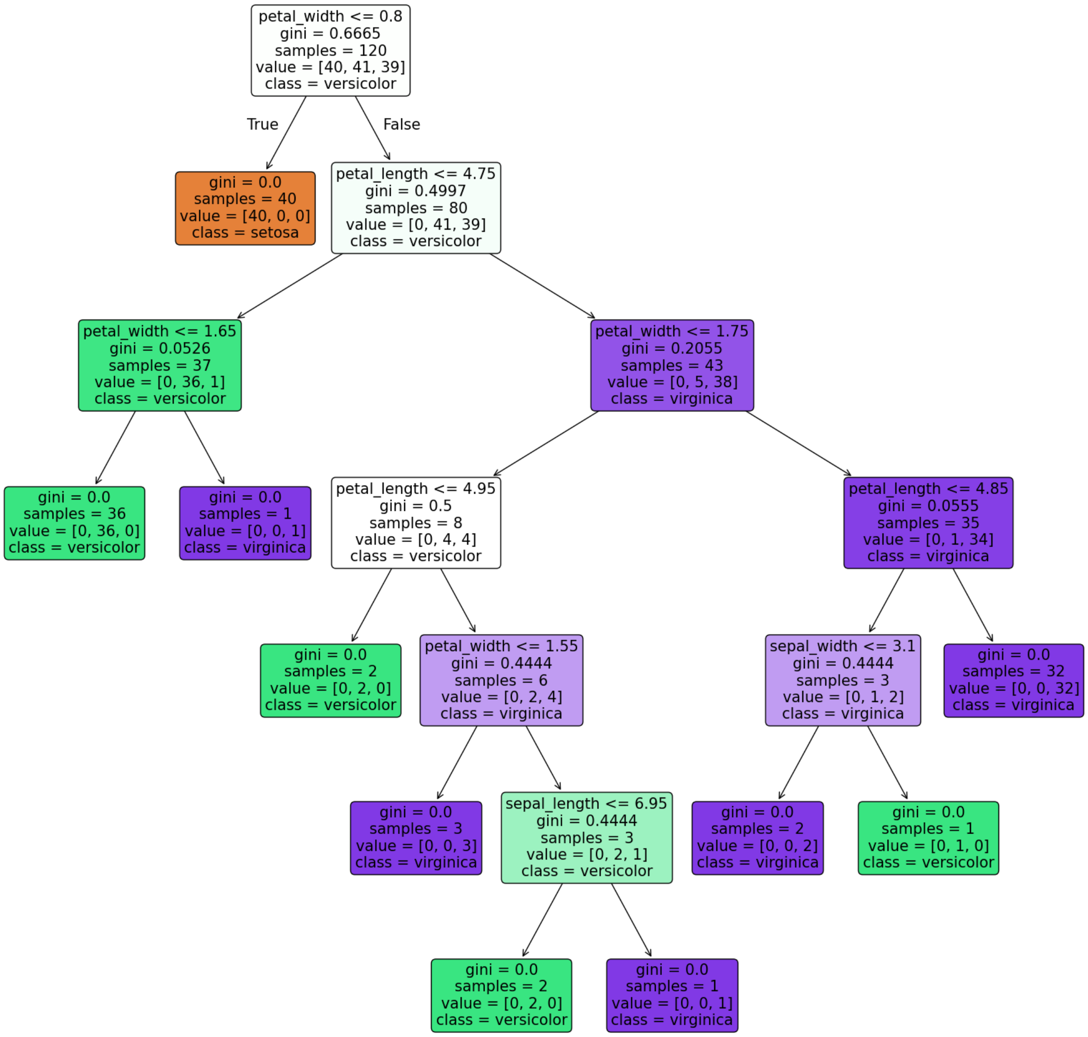
```
Text(0.5, 1.0, 'Accuracy Score : 1.0')
```



[77]:

```
plt.figure(figsize = (20,20))
```

```python
dec_tree = plot_tree(decision_tree=dc,feature_names = df1.columns,class_names = ['setosa','versicolor','virginica'],filled = True, precision = 4, rounded = True)
```



```
[ ]:
```