Untitled3

October 29, 2024

[1]:
```
import pandas as pd
import seaborn as sns
```

[2]:
```
import warnings
warnings.filterwarnings("ignore")
```

[3]:
```
df=sns.load_dataset('iris')
```

[4]:
```
df.head()
```

[4]:

	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

[5]:
```
df.size
```

[5]:
```
750
```

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

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

[7]:
```
df.species.value_counts()
```

[7]:
```
species
setosa        50
versicolor    50
virginica     50
Name: count, dtype: int64
```

[8]:
```
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
```

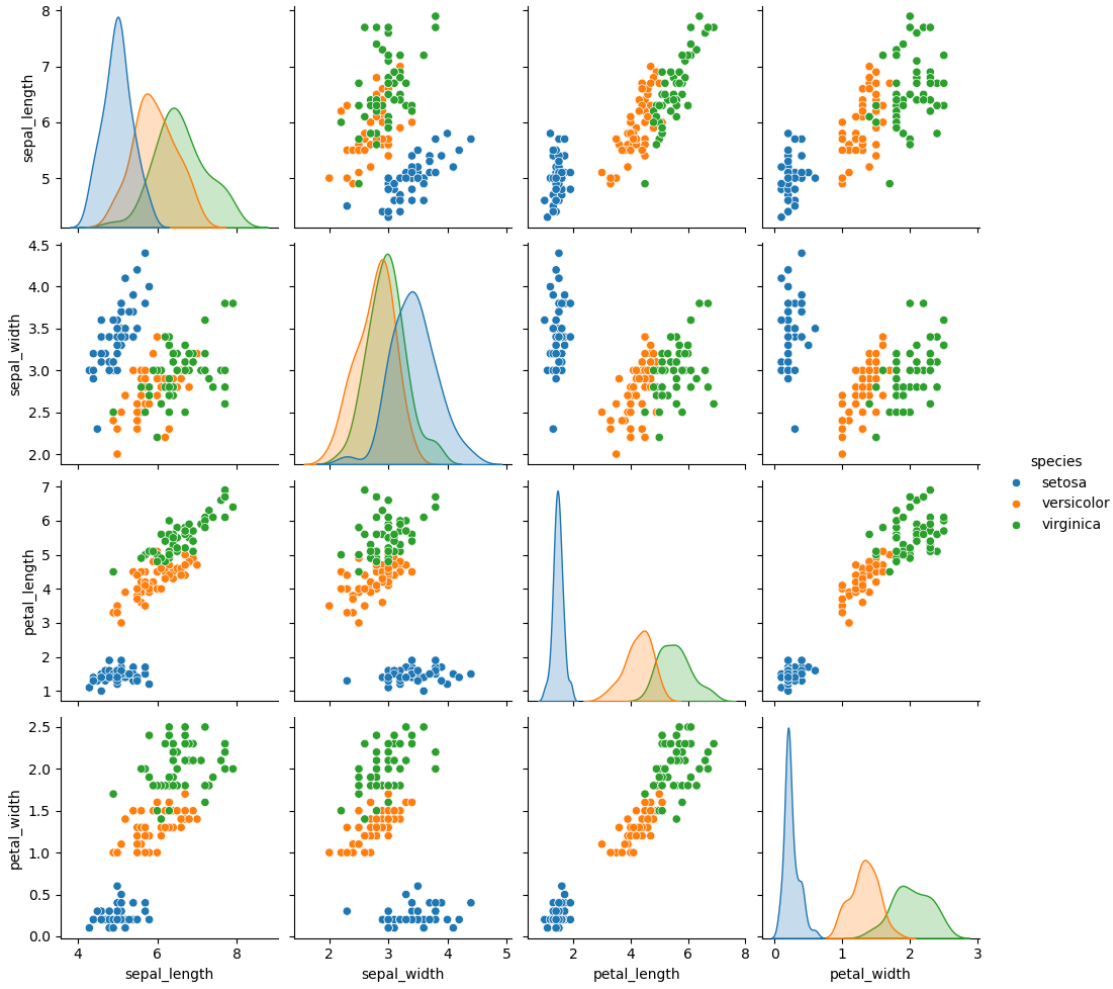
[9]:
```
df.describe()
```

[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

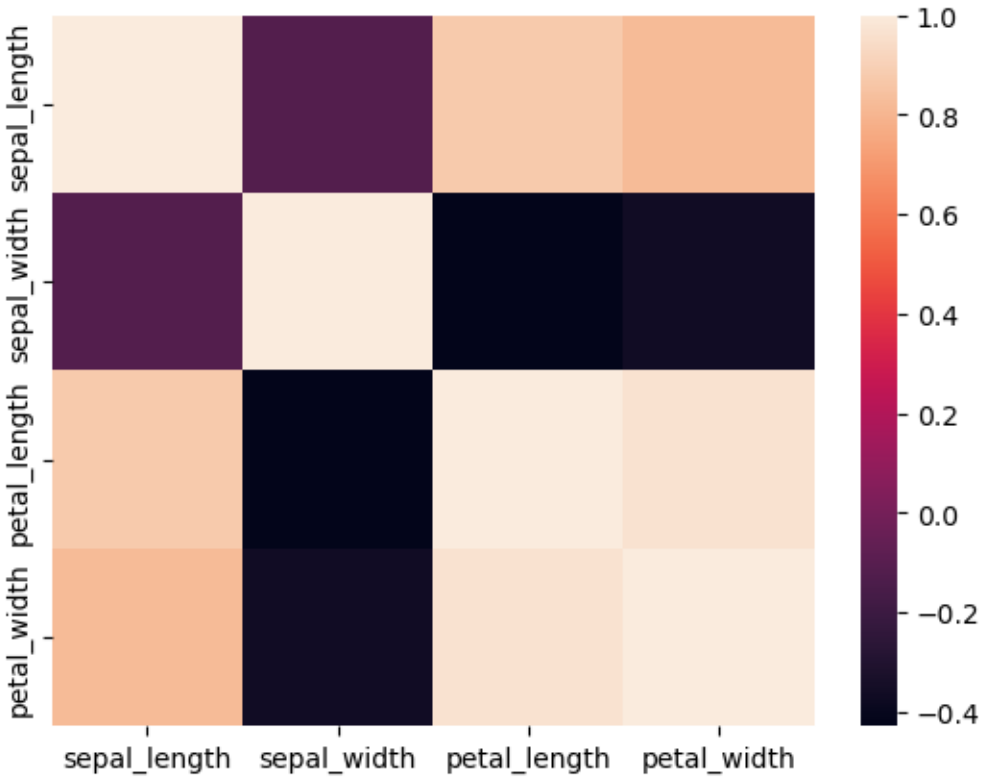
[10]:
```
sns.pairplot(data=df,hue='species')
```

[10]:
```
<seaborn.axisgrid.PairGrid at 0x1c0903dcb00>
```



[11]:
```
sns.heatmap(df.drop('species',axis=1).corr())
```

[11]: <Axes: >



[12]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(df['species'])
```

[13]:
```
y[0:5]
```

[13]:
```
array([0, 0, 0, 0, 0])
```

[14]:
```
X=df.drop('species',axis=1)
```

[15]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
↪2,random_state=42)
```

[16]:
```
from sklearn.tree import DecisionTreeClassifier
dtree=DecisionTreeClassifier()
dtree.fit(X_train,y_train)
y_predicted=dtree.predict(X_test)
```

[17]:
```
from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,y_predicted))
```

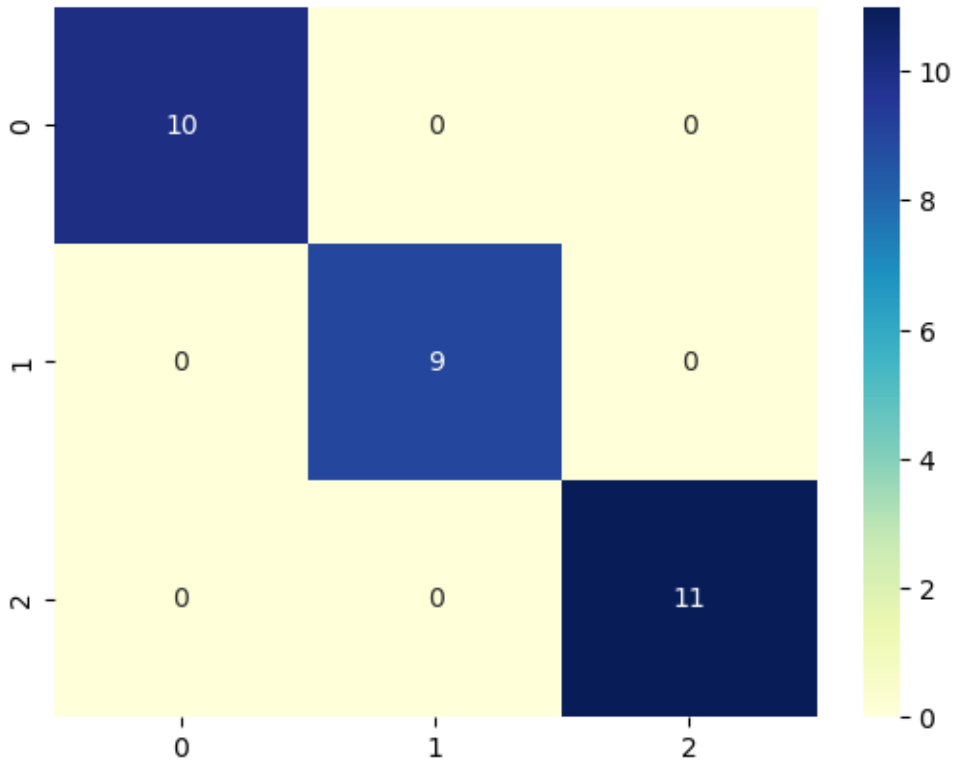
```
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

[18]:
```
print(classification_report(y_test,y_predicted))
```

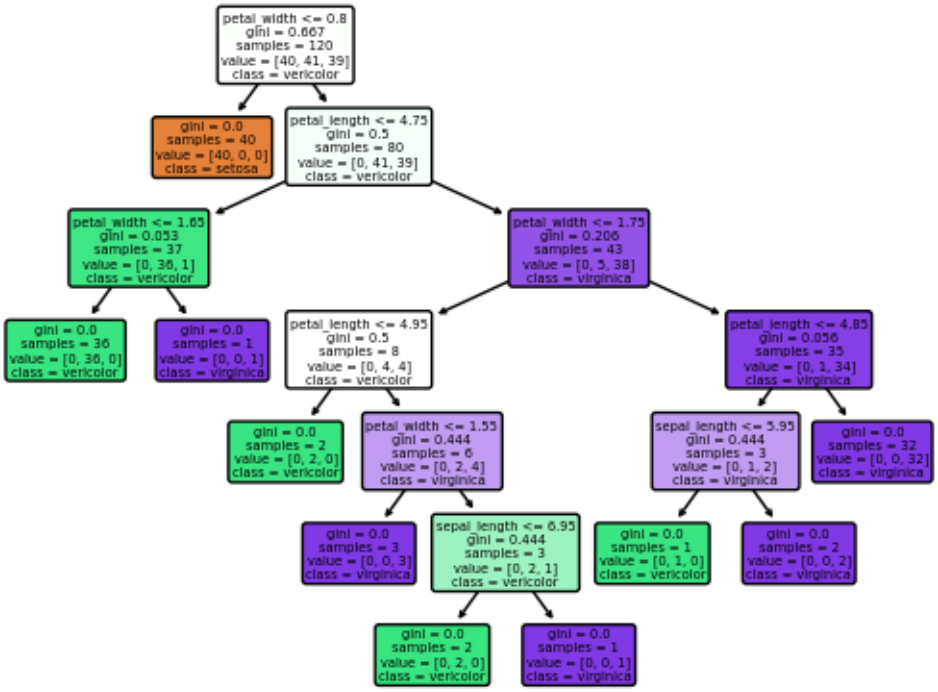
```
              precision    recall  f1-score   support

           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
```

[20]:
```
sns.heatmap(pd.
↪DataFrame(confusion_matrix(y_test,y_predicted)),annot=True,cmap='YlGnBu')
```

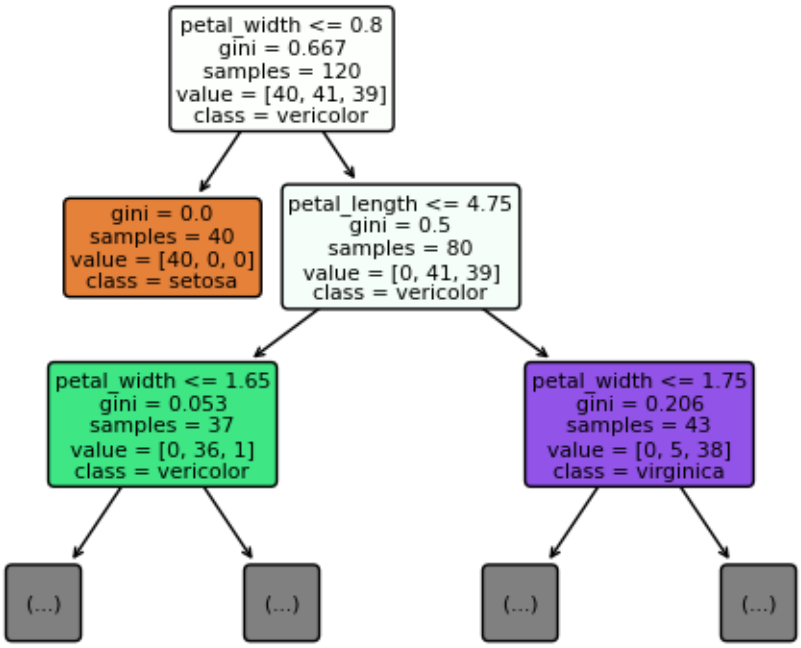
[20]: <Axes: >



[21]:
```
from sklearn.tree import plot_tree
plot=plot_tree(decision_tree=dtree,feature_names=df.
  ↪columns,class_names=("setosa","vericolor","virginica"),filled=True,rounded=True)
```



[22]:
```
plot=plot_tree(decision_tree=dtree,feature_names=df.
  ↪columns,class_names=("setosa","vericolor","virginica"),max_depth=2,filled=True,rounded=True
```


petal_width <= 0.8
gini = 0.667
samples = 120
value = [40, 41, 39]
class = vericolor
gini = 0.0
samples = 40
value = [40, 0, 0]
class = setosa
petal_length <= 4.75
gini = 0.5
samples = 80
value = [0, 41, 39]
class = vericolor
petal_width <= 1.65
gini = 0.053
samples = 37
value = [0, 36, 1]
class = vericolor
petal_width <= 1.75
gini = 0.206
samples = 43
value = [0, 5, 38]
class = virginica
(...)
(...)
(...)
(...)

[ ]: