In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [4]:
```
iris = pd.read_csv("IRIS.csv")
```

In [5]:
```
iris.head()
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [6]:
```
iris.isnull().all()
```

Out[6]:
```
sepal_length    False
sepal_width     False
petal_length    False
petal_width     False
species         False
dtype: bool
```

In [7]:
```
iris.dtypes
```

Out[7]:
```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

In [8]:
```
iris.describe()
```

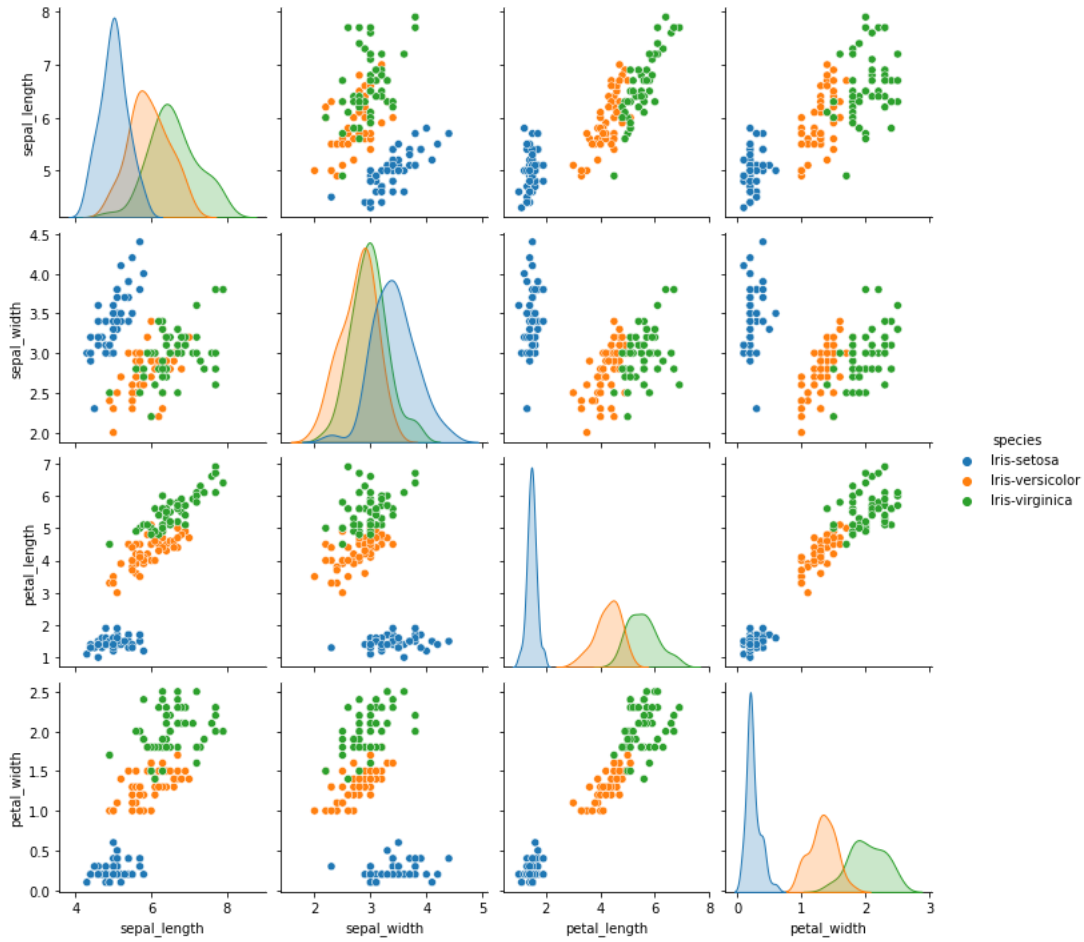
Out[8]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
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

In [9]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report , confusion_matrix
```

In [10]:
```
sns.pairplot(iris , hue ='species')
```

Out[10]: <seaborn.axisgrid.PairGrid at 0x228204d88b0>



In [11]:
```
x = iris[['sepal_length' , 'sepal_width' , 'petal_length' , 'petal_width']].values
y = iris['species'].values
```

In [16]:
```
x_train , x_test , y_train , y_test  = train_test_split(x , y , random_state = 1 , test_size = 0.3)
```

In [17]:
```
dtc = DecisionTreeClassifier()
dtc.fit(x,y)
```

Out[17]: DecisionTreeClassifier()

In [19]:
```
predictions = dtc.predict(x_test)
```

In [20]:
```
print(classification_report(y_test , predictions ))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        14
Iris-versicolor       1.00      1.00      1.00        18
 Iris-virginica       1.00      1.00      1.00        13

       accuracy                           1.00        45
      macro avg       1.00      1.00      1.00        45
   weighted avg       1.00      1.00      1.00        45
```

In [21]:
```
print(confusion_matrix(y_test , predictions ))
```

```
[[14  0  0]
 [ 0 18  0]
 [ 0  0 13]]
```

In [22]:
```
dtc.score(x_test , y_test)
```

Out[22]: 1.0

Without sepal width

In [23]:
```
x = iris [['sepal_length', 'petal_length' , 'petal_width']]
y = iris['species']
```

In [24]:
```
print(x,y)
```

```
     sepal_length  petal_length  petal_width
0             5.1           1.4          0.2
1             4.9           1.4          0.2
2             4.7           1.3          0.2
3             4.6           1.5          0.2
4             5.0           1.4          0.2
..            ...           ...          ...
145           6.7           5.2          2.3
146           6.3           5.0          1.9
147           6.5           5.2          2.0
148           6.2           5.4          2.3
149           5.9           5.1          1.8

[150 rows x 3 columns] 0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: species, Length: 150, dtype: object
```

In [25]:
```
x_train , x_test , y_train , y_test  = train_test_split(x , y , random_state = 30 , test_size = 0.3)
```

In [26]:
```
dtc = DecisionTreeClassifier()
dtc.fit(x,y)
dtc.score(x_test , y_test)
```

Out[26]:
```
1.0
```

In [27]:
```
predictions2 = dtc.predict(x_test)
```

In [28]:
```
print(confusion_matrix(y_test , predictions2))
```

```
[[13  0  0]
 [ 0 13  0]
 [ 0  0 19]]
```

In [ ]:

In [ ]: