In [1]:

```python
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
import matplotlib.pyplot as plt
```

In [3]:

```python
data=pd.read_csv("C:\\Users\\USER\\Documents\\IRIS Flower.csv")
```

In [4]:

```python
data
```

Out[4]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [5]:

```python
data.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
```

In [6]:

```
data.isnull().sum()
```

Out[6]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [7]:

```
data.columns
```

Out[7]:

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [8]:

```
data.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]:

```
data.shape
```

Out[9]:

```
(150, 5)
```

In [10]:

```
data.drop_duplicates(subset ="species",)
```

Out[10]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
50	7.0	3.2	4.7	1.4	Iris-versicolor
100	6.3	3.3	6.0	2.5	Iris-virginica

In [11]:

```
data.value_counts("species")
```

Out[11]:

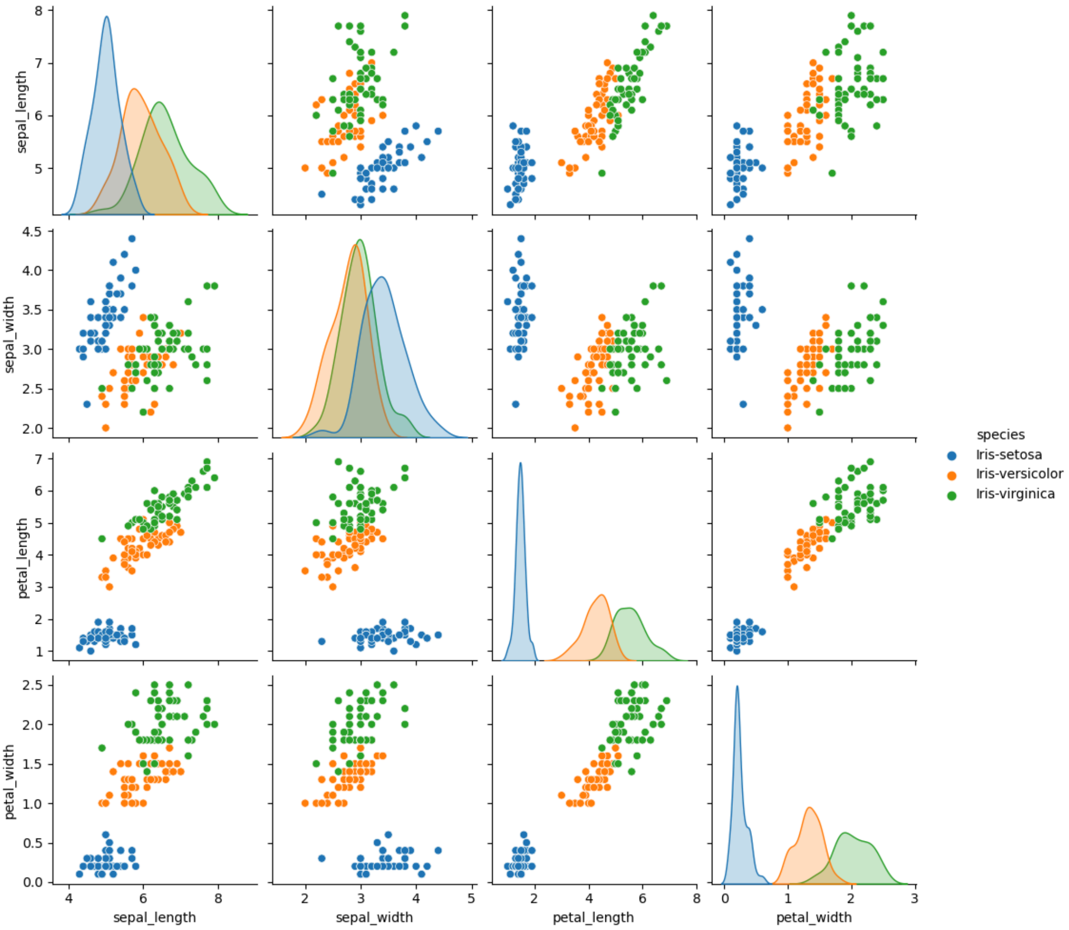
```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

In [12]:

```python
import seaborn as sns
```

In [13]:

```python
sns.pairplot(data, hue="species")
plt.show()
```



In [14]:

```python
X=data.iloc[:,:4]
X
```

Out[14]:

	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
...	...	...	...	...
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 [15]:

```
y=data.iloc[:,4]
y
```

Out[15]:

```
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 [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,ra
```

In [18]:

```
X_train.shape
```

Out[18]:

```
(120, 4)
```

In [19]:

```
X_test.shape
```

Out[19]:

```
(30, 4)
```

In [20]:

```
y_train.shape
```

Out[20]:

```
(120,)
```

In [21]:

```
y_test.shape
```

Out[21]:

```
(30,)
```

In [22]:

```
from sklearn.linear_model import LogisticRegression
```

In [23]:

```
model=LogisticRegression()
```

In [24]:

```
model.fit(X_train,y_train)
```

Out[24]:

```
▾ LogisticRegression
LogisticRegression()
```

In [25]:

```
y_pre=model.predict(X_test)
```

In [26]:

```
y_pre
```

Out[26]:

```
array(['Iris-versicolor', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setos
a',
       'Iris-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginic
a',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-virginica', 'Iris-setosa', 'Iris-virginic
a',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa'], dtype=object)
```

In [27]:

```
from sklearn.metrics import accuracy_score, classification_report
```

In [28]:

```
accuracy = accuracy_score(y_test, y_pre)
print('Accuracy of the model is',accuracy)
```

```
Accuracy of the model is 1.0
```

In [29]:

```
print(classification_report(y_test, y_pre))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       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
```

Conclusion

Mainly we focused on Logistic Regression

We took Iris Flowers dataset and performed a logistic regression algorithm

Finally, I got an accuracy of 1.0, which shows that the model we built is very accurate.

In [ ]:

```

```