import the libraries

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

load the data

In [2]:
```
github_url = "https://raw.githubusercontent.com/amankharwal/Website-data/mas
```

In [3]:
```
data = pd.read_csv(github_url)
```

In [4]:
```
print(data)
```

```
     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 x 5 columns]
```

data preprocessing

In [5]:
```
data.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]:
```
data.shape
```

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

In [7]:
```
data.describe()
```

Out[7]:

	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 [8]:
```
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 [9]:
```
data['species'].value_counts()
```

Out[9]:
```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

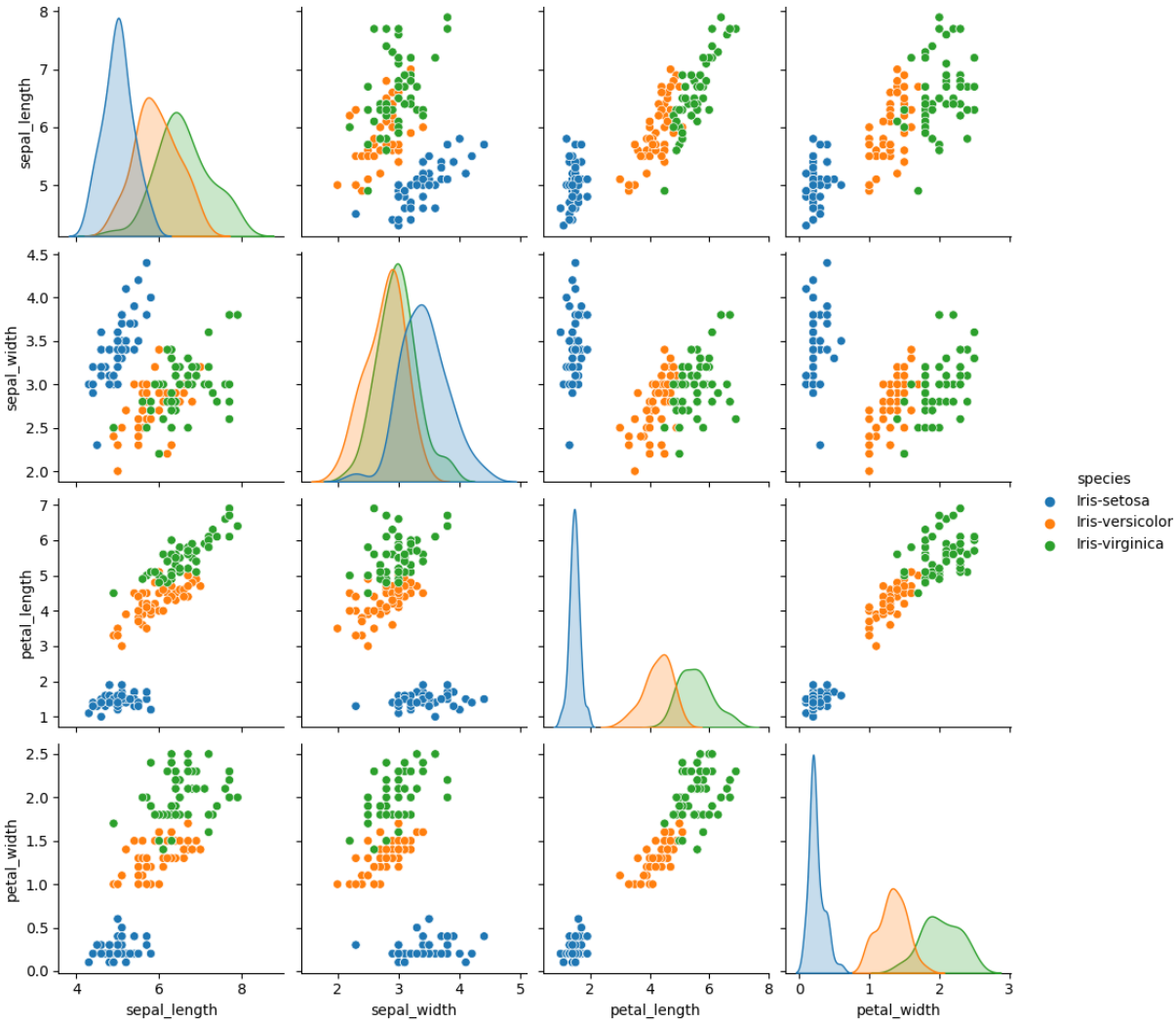
In [10]:
```
data.isnull().sum()
```

Out[10]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

choosing the suitable ml classifier

In [11]:
```
sns.pairplot(data, hue= "species")
```

Out[11]:
```
<seaborn.axisgrid.PairGrid at 0x7fd70e14fc50>
```



as we can see in these plots that there is no overlapping between the points in abundance and can easily classify these points by using k-nearest neighbor

In [12]:
```
from sklearn.neighbors import KNeighborsClassifier
```

In [13]:
```
x = data.drop(["species"], axis = 1)
```

In [14]:
```
print(x)
```

```
     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 x 4 columns]
```

In [15]:
```
y = data["species"]
```

In [16]:
```
print(y)
```

```
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
```

splitting the dataset into training and testing datasets

In [17]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state = 2,
```

In [18]:
```
print(x.shape, x_train.shape, x_test.shape)
```

```
(150, 4) (135, 4) (15, 4)
```

In [19]:
```
model = KNeighborsClassifier(n_neighbors = 7)
```

In [20]:
```
model.fit(x_train, y_train)
```

Out[20]:
```
▾        KNeighborsClassifier

KNeighborsClassifier(n_neighbors=7)
```

In [21]:
```
x_train_prediction = model.predict(x_train)
```

In [22]:
```
training_data_accuracy = accuracy_score(x_train_prediction, y_train)
```

In [23]:
```
training_data_accuracy
```

Out[23]: 0.9703703703703703

In [24]:
```
x_test_prediction = model.predict(x_test)
```

In [25]:
```
testing_data_accuracy = accuracy_score(x_test_prediction, y_test)
```

In [26]:
```
testing_data_accuracy
```

Out[26]: 1.0