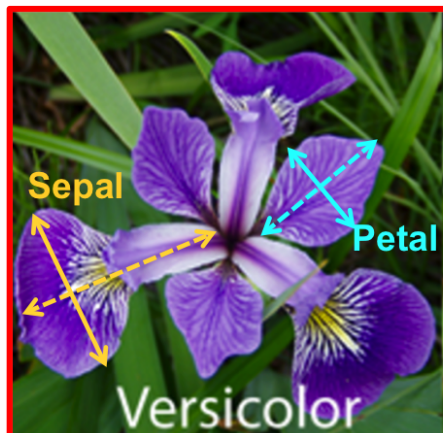


★ Classification Model to Identify Species of IRIS Flower



The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features.

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# import data
iris = pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/IRIS.csv')
```

```
# view data
iris.head()
```

	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	

```
# info of data
iris.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

```

```

# summary statistics
iris.describe()

```

	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



```

# check for missing value
iris.isna().sum()

```

```

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

```

```

# check for categories
iris.nunique()

```

```

sepal_length    35
sepal_width     23
petal_length    43
petal_width     22
species         3
dtype: int64

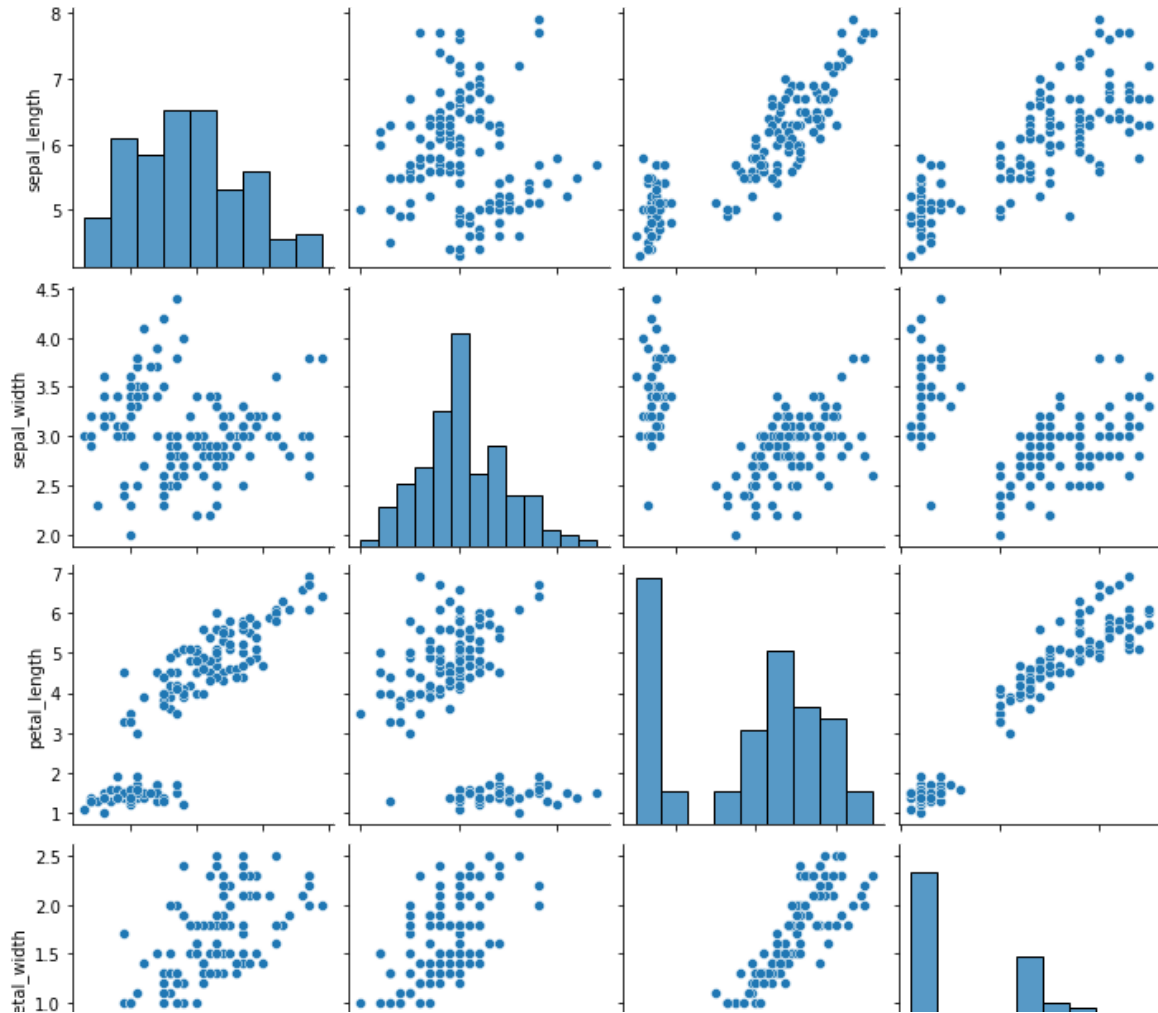
```

```

# visualize pairplot
sns.pairplot(iris)

```

<seaborn.axisgrid.PairGrid at 0x7f8d3473d890>



```
# column names
iris.columns
```

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

```
# define y
y = iris['species']
```

```
# define X
X = iris[['sepal_length', 'sepal_width',
          'petal_length', 'petal_width']]
```

```
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=.30, random_state=2529)
```

```
# verify shape
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((105, 4), (45, 4), (105,), (45,))
```

```
# select model
```

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

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

```
RandomForestClassifier()
```

```
# predict with model
y_pred = model.predict(X_test)
```

```
# model evaluation
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
# model accuracy
accuracy_score(y_test, y_pred)
```

```
1.0
```

```
# model confusion matrix
confusion_matrix(y_test, y_pred)
```

```
array([[14,  0,  0],
       [ 0,  9,  0],
       [ 0,  0, 22]])
```

```
# model classification report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	14
Iris-versicolor	1.00	1.00	1.00	9
Iris-virginica	1.00	1.00	1.00	22
accuracy			1.00	45
macro avg	1.00	1.00	1.00	45
weighted avg	1.00	1.00	1.00	45

```
# future prediction
sample = iris.sample()
sample
```

	sepal_length	sepal_width	petal_length	petal_width	species
53	5.5	2.3	4.0	1.3	Iris-versicolor



```
# define X_new
X_new = sample.loc[:,X.columns]
X_new
```



```
# predict for X_new  
model.predict(X_new)
```

```
array(['Iris-versicolor'], dtype=object)
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 5:01 PM

