Iris Dataset

Hypothesis

In [7]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

Loading data in Pandas Dataframe

In [8]:

```
data=pd.read_csv('Iris.csv')
data
```

Out[8]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Feature Extraction and Test-Train-Split

In [9]:

```
data = data.replace(['Iris-versicolor','Iris-virginica','Iris-setosa'],[0, 1, 2])
data
```

Out[9]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	2
1	2	4.9	3.0	1.4	0.2	2
2	3	4.7	3.2	1.3	0.2	2
3	4	4.6	3.1	1.5	0.2	2

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
4	5	5.0	3.6	1.4	0.2	2
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	1
146	147	6.3	2.5	5.0	1.9	1
147	148	6.5	3.0	5.2	2.0	1
148	149	6.2	3.4	5.4	2.3	1
149	150	5.9	3.0	5.1	1.8	1

150 rows × 6 columns

In [10]:

```
X = data.drop("Species",axis=1)
X
```

Out[10]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

In [11]:

```
y  = data["Species"]
y
```

Out[11]:

```
0       2
1       2
2       2
3       2
4       2
       ..
145     1
146     1
147     1
148     1
149     1
Name: Species, Length: 150, dtype: int64
```

In [12]:

```
X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.33, random_state=42)
```

In [13]:

```
X_train
```

Out[13]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
96	**97**	**5.7**	**2.9**	**4.2**	**1.3**
105	**106**	**7.6**	**3.0**	**6.6**	**2.1**
66	**67**	**5.6**	**3.0**	**4.5**	**1.5**
0	**1**	**5.1**	**3.5**	**1.4**	**0.2**
122	**123**	**7.7**	**2.8**	**6.7**	**2.0**
...	**...**	**...**	**...**	**...**	**...**
71	**72**	**6.1**	**2.8**	**4.0**	**1.3**
106	**107**	**4.9**	**2.5**	**4.5**	**1.7**
14	**15**	**5.8**	**4.0**	**1.2**	**0.2**
92	**93**	**5.8**	**2.6**	**4.0**	**1.2**
102	**103**	**7.1**	**3.0**	**5.9**	**2.1**

100 rows × 5 columns

Importing Models (Algorithms)

In [14]:

```
labels_names = ['I.setosa', 'I.versicolor', 'I.virginica']
```

In [15]:

```
labels_names
```

Out[15]:

```
['I.setosa', 'I.versicolor', 'I.virginica']
```

In [16]:

```
clf=SVC()
clf.fit(X_train, y_train)
```

Out[16]:

```
SVC()
```

In [17]:

```
pred = clf.predict(X_test)
pred
```

Out[17]:

```
array([0, 2, 1, 0, 0, 2, 0, 1, 0, 0, 1, 2, 2, 2, 2, 0, 1, 0, 0, 1, 2, 1,
       2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 0, 2, 2, 1, 0, 2, 2, 2, 1, 0, 0, 2,
       2, 0, 1, 1, 0, 1])
```

In [18]:

```
metrics.accuracy_score(y_test, pred)
```

Out[18]:

```
1.0
```

In [19]:

```
metrics.accuracy_score(y_test, pred)
```

Out[19]:

Out[19]:

```
array([[15,  0,  0],
       [ 0, 16,  0],
       [ 0,  0, 19]])
```

In [20]:

```
print(metrics.classification_report(
    y_test, pred, target_names=labels_names))
```

```
              precision    recall  f1-score   support

    I.setosa       1.00      1.00      1.00        15
I.versicolor       1.00      1.00      1.00        16
 I.virginica       1.00      1.00      1.00        19

    accuracy                           1.00        50
   macro avg       1.00      1.00      1.00        50
weighted avg       1.00      1.00      1.00        50
```

In [ ]:

```

```

In [ ]:

```

```