Data Analytics III the given dataset.

1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

In [2]:

```
dataset= pd.read_csv('/home/mca01/Downloads/Iris.csv')
```

In [3]:

```
dataset.head()
```

Out[3]:

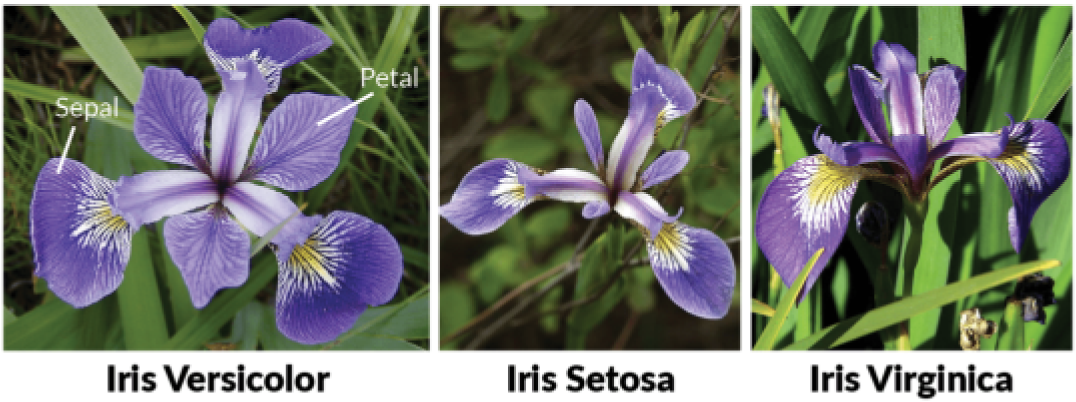
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Variety
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [5]:

```
%matplotlib inline
img=mpimg.imread('/home/mca01/Downloads/iris_types.png')
plt.figure(figsize=(10,30))
plt.axis('off')
plt.imshow(img)
```

Out[5]:

```
<matplotlib.image.AxesImage at 0x7f820859c070>
```



In [6]:

```
x=dataset.iloc[:,:4].values
y=dataset['Variety'].values
```

In [7]:

```
from pandas.core.common import random_state
from sklearn.model_selection import train_test_split
X_train, X_test , y_train, y_test=train_test_split(x,y, test_size = 0.20)
```

In [8]:

```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [9]:

```
from sklearn.naive_bayes import GaussianNB
nvclassifier=GaussianNB()
nvclassifier.fit(X_train,y_train)
```

Out[9]:

```
▼ GaussianNB ⓘ ?

GaussianNB()
```

In [10]:

```
y_pred= nvclassifier.predict(X_test)
print(y_pred)
```

```
['Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica']
```

In [11]:

```
y_compare=np.vstack((y_test,y_pred)).T
y_compare[:5,:]
```

Out[11]:

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

In [12]:

```
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[12  0  0]
 [ 0  8  0]
 [ 0  0 10]]
```

In [13]:

```
a=cm.shape
corrPred=0
falsePred=0
```

In [14]:

```
for row in range(a[0]):
    for c in range(a[1]):
        if row == c :
            corrPred += cm[row,c]
        else :
            falsePred += cm[row,c]
```

In [15]:

```
print('correct Prediction',corrPred)
print('false Prediction',falsePred)
print('\n')
print('\n Accuracy of Naive Bayes Classification',corrPred/(cm.sum()))
print('\n ErrorRate of Naive Bayes Classification',falsePred/(cm.sum()))
```

```
correct Prediction 30
false Prediction 0




 Accuracy of Naive Bayes Classification 1.0

 ErrorRate of Naive Bayes Classification 0.0
```

In [ ]:

```

```