NAIVE BAYES CLASSIFICATION

```
NAIVE BAYES CLASSIFICATION
Assignment 6 Data Analytics III 1. Implement Simple Naïve Bayes classification algorithm using Python/R on
iris.csv dataset. II. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision,
Recall on the given dataset
```

In [1]:
```
# IMPORTING LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
dataset = pd.read_csv('https://raw.githubusercontent.com/mk-gurucharan/Classification/master/IrisDataset.csv'
```

In [3]:
```
dataset
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [4]:
```
dataset.head()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [5]:
```
gk=dataset.groupby('species')
```

In [8]:
```
gk.first()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width
species				
setosa	5.1	3.5	1.4	0.2
versicolor	7.0	3.2	4.7	1.4
virginica	6.3	3.3	6.0	2.5

In [9]:
```
X = dataset.iloc[:,:4].values
y = dataset['species'].values
```

In [10]:
```
# split the dataset into training and testing data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =0.2)
```

```
# feature scaling operation
Feature Scaling The dataset is scaled down to a smaller range using the Feature Scaling option. In this, both
the X_train and X_test values are scaled down to smaller values to improve the speed of the program.
```

In [11]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [12]:
```
#Training the Naive Bayes Classification model on the Training Set
from sklearn.naive_bayes import GaussianNB
```

```
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[12]:

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [13]:

```
# predicting the test
y_pred = classifier.predict(X_test)
y_pred
```

Out[13]:

```
array(['virginica', 'setosa', 'virginica', 'versicolor', 'virginica',
       'versicolor', 'setosa', 'setosa', 'virginica', 'versicolor',
       'virginica', 'versicolor', 'setosa', 'versicolor', 'versicolor',
       'virginica', 'virginica', 'setosa', 'versicolor', 'versicolor',
       'virginica', 'versicolor', 'virginica', 'setosa', 'setosa',
       'virginica', 'setosa', 'setosa', 'setosa', 'virginica'],
      dtype='<U10')
```

In [17]:

```
# finding the confussion matrix and accuracy
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(y_test, y_pred))
cm
```

```
Accuracy :  0.9666666666666667
```

Out[17]:

```
array([[10,  0,  0],
       [ 0,  8,  0],
       [ 0,  1, 11]], dtype=int64)
```

In [18]:

```
# comparing the real values and the predicted values
df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df
```

Out[18]:

	Real Values	Predicted Values
0	virginica	virginica
1	setosa	setosa
2	virginica	virginica
3	versicolor	versicolor
4	virginica	virginica
5	versicolor	versicolor
6	setosa	setosa
7	setosa	setosa
8	virginica	virginica
9	versicolor	versicolor
10	virginica	virginica
11	versicolor	versicolor
12	setosa	setosa
13	virginica	versicolor
14	versicolor	versicolor
15	virginica	virginica
16	virginica	virginica
17	setosa	setosa
18	versicolor	versicolor
19	versicolor	versicolor
20	virginica	virginica
21	versicolor	versicolor
22	virginica	virginica
23	setosa	setosa
24	setosa	setosa
25	virginica	virginica
26	setosa	setosa
27	setosa	setosa
28	setosa	setosa
29	virginica	virginica

In [ ]: