Lab Assignment No-6
Data Analytics III
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 the given dataset.

#Problem Analysis
To implement the Naive Bayes Classification, we shall use a very famous Iris Flower Dataset that consists of 3 classes of flowers.
In this, there are 4 independent variables namely the, sepal_length, sepal_width, petal_length and petal_width.
The dependent variable is the species which we will predict using the four independent features of the flowers.
Using the various features of the flower (independent variables),
we have to classify a given flower using Naive Bayes Classification model.

#Step 1: Importing the Libraries

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

#Step 2: Importing the dataset

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

#Step 3:Describe dataset,See dimention and Exlpore dataset

```
dataset.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

dataset.head()

	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

dataset.shape
(150, 5)

```
X = dataset.iloc[:,:4].values
X
[5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
       [5.8, 2.8, 5.1, 2.4],
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
```

```
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]])


y = dataset['species'].values
y
array(['setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
       'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
```

```
'setosa', 'setosa', 'setosa', 'setosa', 'setosa', 'setosa',
'setosa', 'setosa', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'versicolor', 'versicolor', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica', 'virginica', 'virginica',
'virginica', 'virginica', 'virginica'], dtype=object)
```

```
#Step 4: Splitting the dataset into the Training set and Test set
#Here, we have the test_size=0.2,
#which means that 20% of the dataset will be used for testing purpose as the test set and
#the remaining 80% will be used as the training set for training the Naive Bayes classification
model.


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)
```

```
#Step 5: 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.

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

```
array([[ 1.19801497,  0.10940701,  0.88731911,  1.14499694],
       [-0.11433338, -0.81191515,  0.03640041, -0.03608873],
       [-0.47224657, -1.04224569,  0.32003998, -0.03608873],
       [-0.23363777, -0.58158461,  0.37676789,  0.09514301],
       [ 1.67523255, -0.35125407,  1.39787033,  0.75130172],
       [ 0.4821886 , -0.81191515,  0.60367954,  0.75130172],
       [ 0.95940618, -0.12092353,  0.66040746,  0.62006997],
       [ 0.60149299,  0.33973754,  0.37676789,  0.35760649],
       [-1.54598612,  0.33973754, -1.38179742, -1.34840614],
       [-1.54598612,  1.2610597 , -1.60870908, -1.34840614],
       [ 1.07871058, -0.12092353,  0.94404703,  1.14499694],
       [ 0.95940618,  0.10940701,  0.32003998,  0.22637475],
       [-0.11433338,  2.18238185, -1.49525325, -1.34840614],
       [-1.06876854,  1.03072916, -1.2683416 , -0.82347917],
       [ 1.55592816, -0.12092353,  1.11423077,  0.48883823],
       [-1.30737733,  0.80039862, -1.2683416 , -1.34840614],
       [-0.83015975, -0.81191515,  0.03640041,  0.22637475],
       [ 0.12427541, -1.96356784,  0.09312832, -0.29855221],
       [ 1.31731937,  0.33973754,  0.49022372,  0.22637475],
       [ 0.3628842 ,  0.80039862,  0.88731911,  1.40746042],
       [-0.47224657,  2.64304293, -1.38179742, -1.34840614],
       [-0.11433338, -0.81191515,  0.71713537,  0.88253346],
       [-1.78459491,  0.33973754, -1.43852534, -1.34840614],
       [ 0.00497102,  0.33973754,  0.54695163,  0.75130172],
       [ 0.4821886 ,  0.80039862,  1.00077494,  1.53869216],
       [-0.59155096,  0.80039862, -1.21161368, -1.34840614],
       [ 0.95940618,  0.57006808,  1.05750285,  1.6699239 ],
       [ 0.4821886 , -0.35125407,  1.00077494,  0.75130172],
       [-1.06876854, -0.12092353, -1.2683416 , -1.34840614],
       [ 0.3628842 , -1.96356784,  0.37676789,  0.35760649],
       [ 0.95940618, -1.27257622,  1.11423077,  0.75130172],
       [-0.23363777, -0.12092353,  0.20658415, -0.03608873],
       [-0.47224657, -1.27257622,  0.09312832,  0.09514301],
       [-1.06876854,  0.57006808, -1.38179742, -1.34840614],
```

[1.07871058, -0.58158461, 0.54695163, 0.22637475],
[-1.06876854, 1.2610597 , -1.38179742, -1.34840614],
[0.4821886 , -1.7332373 , 0.32003998, 0.09514301],
[1.19801497, 0.33973754, 1.05750285, 1.40746042],
[2.03314574, -0.12092353, 1.56805407, 1.14499694],
[0.60149299, -0.35125407, 0.26331206, 0.09514301],
[-0.94946415, 1.03072916, -1.38179742, -1.21717439],
[0.72079739, -0.12092353, 1.11423077, 1.27622868],
[-1.78459491, -0.35125407, -1.38179742, -1.34840614],
[-0.11433338, -0.81191515, 0.71713537, 0.88253346],
[2.15245013, -0.58158461, 1.62478199, 1.0137652],
[-0.83015975, 0.80039862, -1.38179742, -1.34840614],
[-1.54598612, 0.10940701, -1.32506951, -1.34840614],
[-1.30737733, 0.80039862, -1.09815786, -1.34840614],
[-0.94946415, -1.27257622, -0.47415081, -0.16732047],
[0.95940618, -0.12092353, 0.77386328, 1.40746042],
[-0.23363777, -1.27257622, 0.66040746, 1.0137652],
[0.3628842 , -0.35125407, 0.26331206, 0.09514301],
[-0.59155096, 1.49139024, -1.32506951, -1.34840614],
[-1.06876854, -2.42422892, -0.19051124, -0.29855221],
[-0.47224657, -1.50290676, -0.0203275 , -0.16732047],
[-0.71085536, 1.49139024, -1.32506951, -1.34840614],
[1.19801497, 0.10940701, 0.60367954, 0.35760649],
[-0.94946415, 1.03072916, -1.38179742, -1.34840614],
[-1.18807294, 0.10940701, -1.32506951, -1.47963788],
[-0.83015975, 1.03072916, -1.32506951, -1.34840614],
[-1.06876854, 1.03072916, -1.43852534, -1.21717439],
[-0.35294217, -0.12092353, 0.37676789, 0.35760649],
[-1.18807294, -1.27257622, 0.37676789, 0.62006997],
[1.43662376, -0.12092353, 1.17095868, 1.14499694],
[0.00497102, -0.12092353, 0.20658415, 0.35760649],
[0.72079739, 0.33973754, 0.71713537, 1.0137652],
[0.12427541, -0.81191515, 0.71713537, 0.48883823],
[-0.23363777, -0.58158461, 0.14985624, 0.09514301],
[0.00497102, -0.12092353, 0.71713537, 0.75130172],
[-0.11433338, -0.81191515, 0.14985624, -0.29855221],
[1.79453695, -0.58158461, 1.28441451, 0.88253346],
[-1.18807294, 0.10940701, -1.32506951, -1.47963788],
[-0.23363777, -0.35125407, 0.20658415, 0.09514301],
[-0.23363777, 3.103704 , -1.32506951, -1.08594265],
[0.12427541, -1.96356784, 0.66040746, 0.35760649],
[-1.42668173, 0.33973754, -1.43852534, -1.34840614],
[0.12427541, -0.12092353, 0.54695163, 0.75130172],
[-0.23363777, -1.04224569, -0.19051124, -0.29855221],

[-0.23363777, 1.72172077, -1.21161368, -1.21717439],
[-0.94946415, 1.72172077, -1.09815786, -1.08594265],
[0.72079739, -0.12092353, 0.77386328, 1.0137652],
[1.55592816, 1.2610597 , 1.28441451, 1.6699239],
[-1.06876854, 0.33973754, -1.49525325, -1.34840614],
[0.24357981, -0.58158461, 0.49022372, -0.03608873],
[0.24357981, -0.58158461, 0.09312832, 0.09514301],
[-0.94946415, 1.72172077, -1.2683416 , -1.34840614],
[1.55592816, 0.33973754, 1.22768659, 0.75130172],
[0.12427541, -0.35125407, 0.37676789, 0.35760649],
[-0.35294217, -0.12092353, 0.14985624, 0.09514301],
[2.15245013, 1.72172077, 1.62478199, 1.27622868],
[-1.06876854, -1.7332373 , -0.30396707, -0.29855221],
[-0.94946415, 0.57006808, -1.21161368, -0.95471091],
[1.19801497, 0.10940701, 0.71713537, 1.40746042],
[2.39105892, 1.72172077, 1.45459825, 1.0137652],
[-1.18807294, 0.10940701, -1.32506951, -1.47963788],
[0.4821886 , -0.58158461, 0.71713537, 0.35760649],
[-1.42668173, 0.33973754, -1.2683416 , -1.34840614],
[1.07871058, 0.33973754, 1.17095868, 1.40746042],
[-0.47224657, 1.03072916, -1.43852534, -1.34840614],
[0.95940618, 0.10940701, 1.00077494, 1.53869216],
[0.4821886 , -1.27257622, 0.66040746, 0.88253346],
[-1.18807294, -1.50290676, -0.30396707, -0.29855221],
[-0.35294217, -0.35125407, -0.13378333, 0.09514301],
[0.84010178, -0.12092353, 0.32003998, 0.22637475],
[0.60149299, -0.58158461, 1.00077494, 1.27622868],
[-1.06876854, 0.80039862, -1.32506951, -1.34840614],
[-0.94946415, 0.80039862, -1.32506951, -1.34840614],
[2.15245013, -1.04224569, 1.73823781, 1.40746042],
[0.24357981, -0.12092353, 0.60367954, 0.75130172],
[2.15245013, -0.12092353, 1.28441451, 1.40746042],
[-0.59155096, 1.95205131, -1.43852534, -1.08594265],
[0.3628842 , -0.58158461, 0.54695163, 0.75130172],
[0.4821886 , 0.57006808, 1.22768659, 1.6699239],
[0.4821886 , -1.27257622, 0.60367954, 0.35760649],
[0.4821886 , 0.57006808, 0.49022372, 0.48883823],
[-0.47224657, -1.50290676, -0.07705542, -0.29855221],
[-0.35294217, -0.58158461, 0.60367954, 1.0137652],
[0.24357981, -1.04224569, 1.00077494, 0.22637475],
[-1.30737733, -0.12092353, -1.38179742, -1.21717439],
[-1.06876854, 0.80039862, -1.2683416 , -1.08594265]])

X_test

```
array([[-1.66529052, -1.7332373 , -1.43852534, -1.21717439],
       [-1.90389931, -0.12092353, -1.55198116, -1.47963788],
       [ 0.24357981, -0.12092353,  0.4334958 ,  0.22637475],
       [-1.54598612,  0.80039862, -1.38179742, -1.21717439],
       [-0.11433338, -1.04224569,  0.09312832, -0.03608873],
       [-0.59155096,  0.80039862, -1.32506951, -1.08594265],
       [ 0.84010178, -0.35125407,  0.4334958 ,  0.09514301],
       [-0.47224657, -1.7332373 ,  0.09312832,  0.09514301],
       [-0.35294217, -1.27257622,  0.03640041, -0.16732047],
       [-0.35294217, -0.81191515,  0.20658415,  0.09514301],
       [ 0.72079739, -0.58158461,  0.4334958 ,  0.35760649],
       [ 0.95940618,  0.10940701,  0.49022372,  0.35760649],
       [-0.11433338, -0.58158461,  0.71713537,  1.53869216],
       [ 0.60149299,  0.33973754,  0.8305912 ,  1.40746042],
       [-0.94946415,  1.72172077, -1.32506951, -1.21717439],
       [ 0.72079739, -0.12092353,  0.94404703,  0.75130172],
       [-0.94946415,  1.49139024, -1.32506951, -1.08594265],
       [-0.59155096, -0.12092353,  0.37676789,  0.35760649],
       [-1.78459491, -0.12092353, -1.43852534, -1.34840614],
       [-0.59155096,  1.95205131, -1.21161368, -1.08594265],
       [ 0.60149299,  0.10940701,  0.94404703,  0.75130172],
       [-1.30737733,  0.10940701, -1.2683416 , -1.34840614],
       [ 0.12427541,  0.80039862,  0.37676789,  0.48883823],
       [-1.18807294, -0.12092353, -1.38179742, -1.34840614],
       [ 0.60149299, -0.81191515,  0.8305912 ,  0.88253346],
       [ 0.24357981, -0.35125407,  0.49022372,  0.22637475],
       [ 0.60149299, -0.58158461,  1.00077494,  1.14499694],
       [-0.83015975,  2.41271239, -1.32506951, -1.47963788],
       [-1.30737733, -0.12092353, -1.38179742, -1.47963788],
       [ 0.95940618,  0.57006808,  1.05750285,  1.14499694]])
```

#Step 6: Training the Naive Bayes Classification model on the Training Set
#n this step, we introduce the class GaussianNB that is used from the sklearn.naive_bayes library.
#Here, we have used a Gaussian model, there are several other models such as Bernoulli, Categorical and Multinomial.

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
```

```
classifier.fit(X_train, y_train)


#Step 6: Predicting the Test set results
y_pred = classifier.predict(X_test)
y_pred


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


#Step 7: Confusion Matrix and Accuracy
#This is a step that is mostly used in classification techniques.
#In this, we see the Accuracy of the trained model and plot the confusion matrix.




from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
array([[12,  0,  0],
       [ 0, 11,  0],
       [ 0,  0,  7]])


from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(y_test, y_pred))
Accuracy :  1.0




#Step 8: Comparing the Real Values with Predicted Values


df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df
```

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

#Conclusion:
From the above confusion matrix, we infer that, out of 30 test set data,
29 were correctly classified and only 1 was incorrectly classified