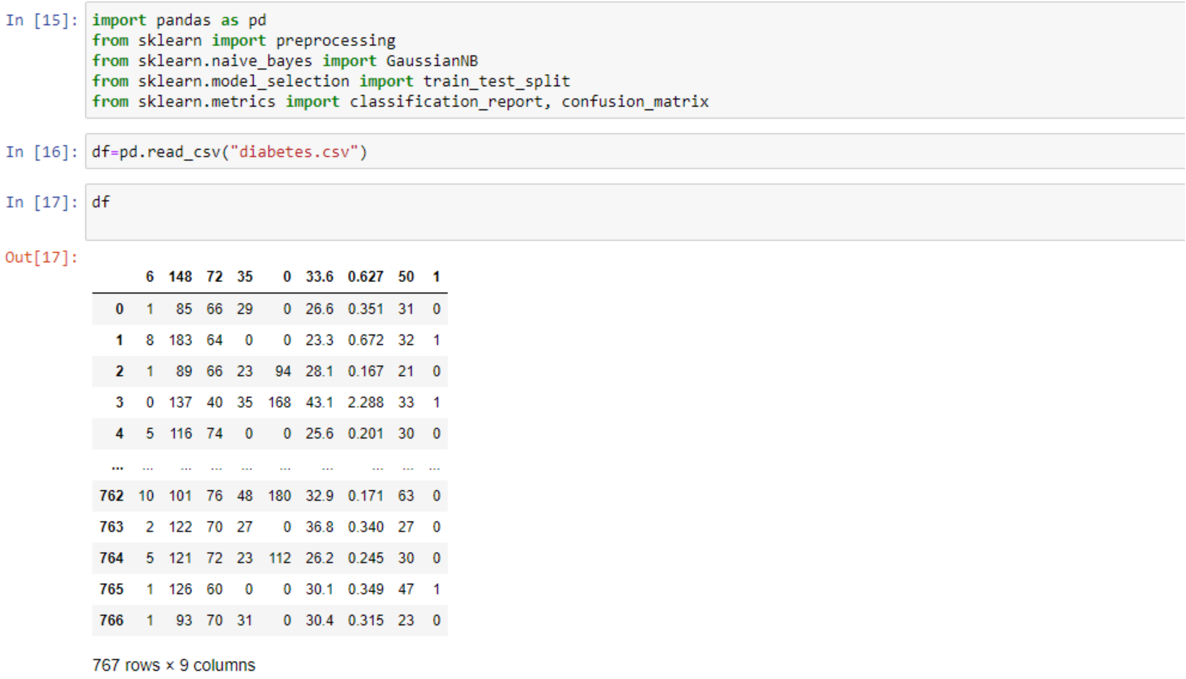
U18CSI6203L_DWDM_End_Sem_Lab_Practicals_23.06.2021

18BCS050

Charan A B

SET 2

1. Download a suitable dataset for classification from any Repository. List the attributes and its type in a word Doc.



```
In [15]: import pandas as pd
         from sklearn import preprocessing
         from sklearn.naive_bayes import GaussianNB
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import classification_report, confusion_matrix
```

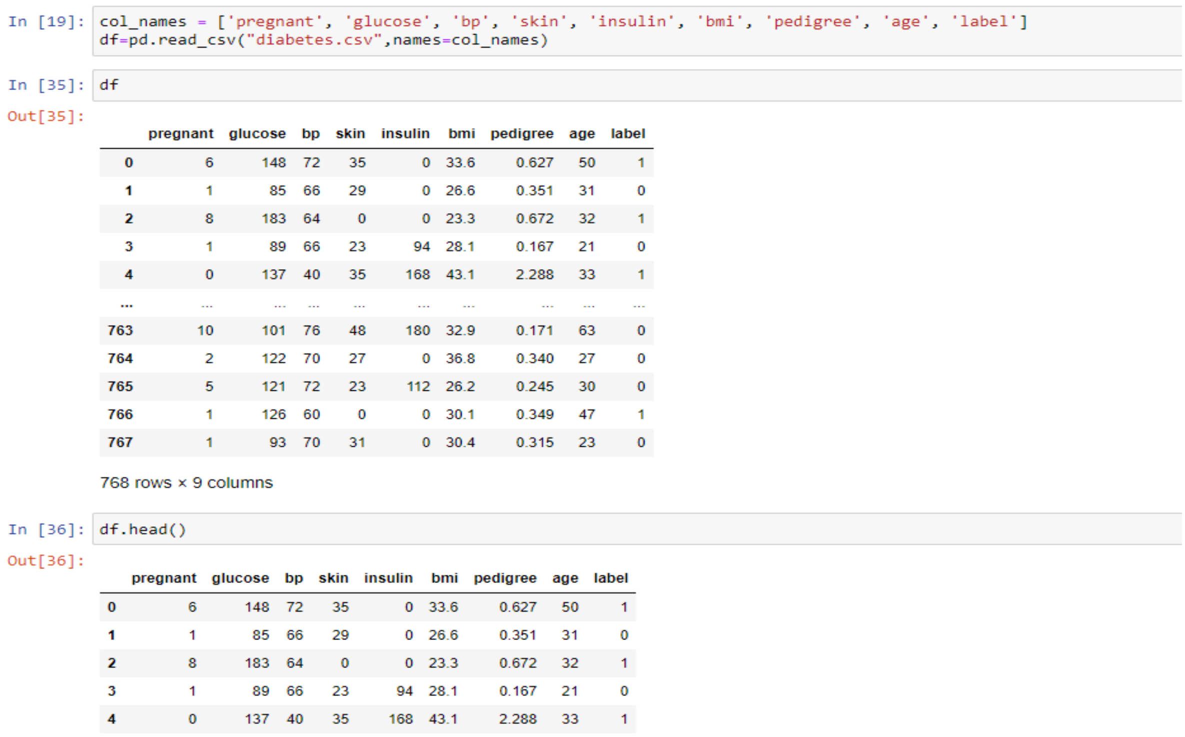
```
In [16]: df=pd.read_csv("diabetes.csv")
```

```
In [17]: df
```

Out[17]:

	6	148	72	35	0	33.6	0.627	50	1
0	1	85	66	29	0	26.6	0.351	31	0
1	8	183	64	0	0	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.167	21	0
3	0	137	40	35	168	43.1	2.288	33	1
4	5	116	74	0	0	25.6	0.201	30	0
...	...	...	...	...	...	...	...	...	...
762	10	101	76	48	180	32.9	0.171	63	0
763	2	122	70	27	0	36.8	0.340	27	0
764	5	121	72	23	112	26.2	0.245	30	0
765	1	126	60	0	0	30.1	0.349	47	1
766	1	93	70	31	0	30.4	0.315	23	0

767 rows × 9 columns



In [19]:
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
df=pd.read_csv("diabetes.csv",names=col_names)
```

In [35]:
```
df
```

Out[35]:

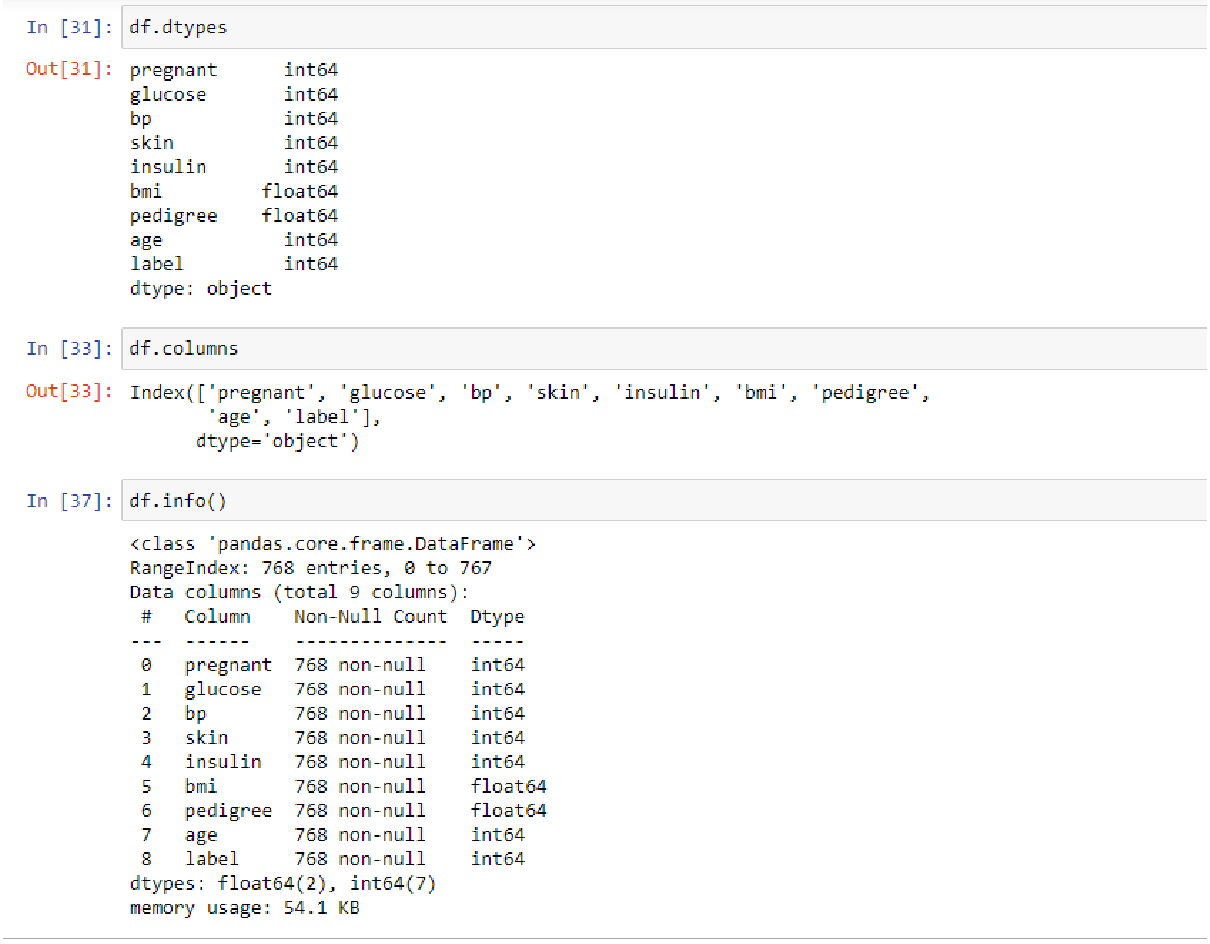
	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [36]:
```
df.head()
```

Out[36]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1



In [31]:
```
df.dtypes
```

Out[31]:
```
pregnant      int64
glucose       int64
bp            int64
skin          int64
insulin       int64
bmi         float64
pedigree    float64
age           int64
label         int64
dtype: object
```

In [33]:
```
df.columns
```

Out[33]:
```
Index(['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree',
       'age', 'label'],
      dtype='object')
```

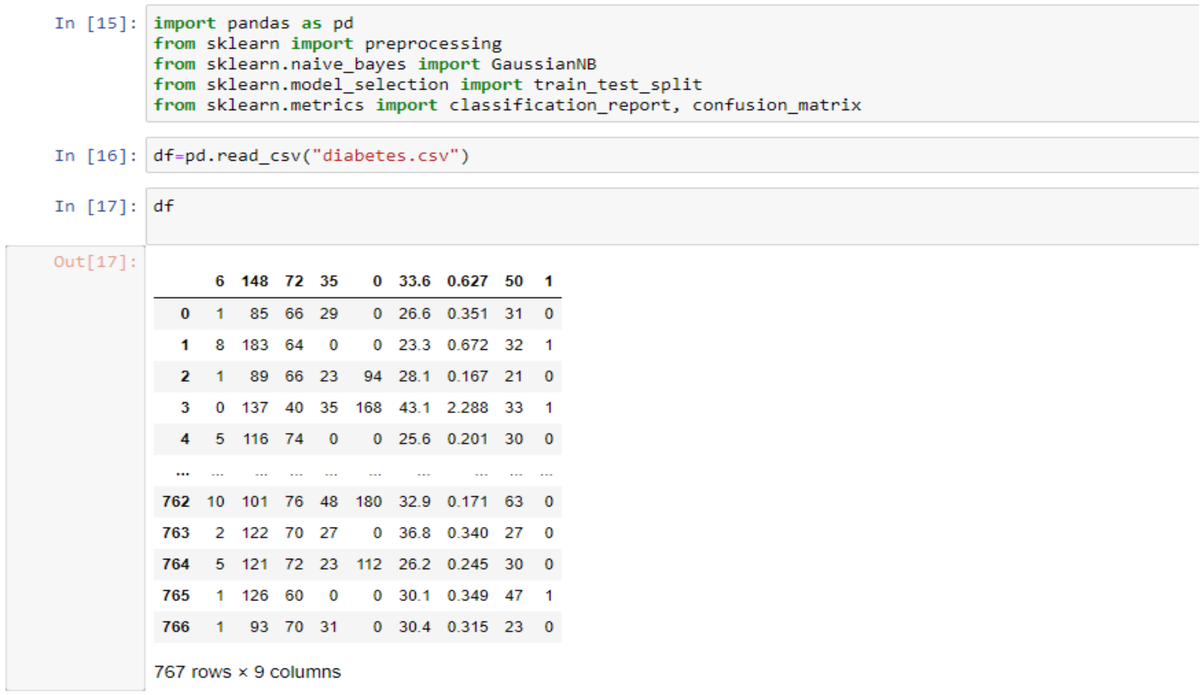
In [37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   pregnant  768 non-null    int64  
 1   glucose   768 non-null    int64  
 2   bp        768 non-null    int64  
 3   skin      768 non-null    int64  
 4   insulin   768 non-null    int64  
 5   bmi       768 non-null    float64
 6   pedigree  768 non-null    float64
 7   age       768 non-null    int64  
 8   label     768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Attributes and their types:

Pregnant	-	Numerical, Continuous
Glucose	-	Numerical, Continuous
Bp	-	Numerical, Continuous
Skin	-	Numerical, Continuous
Insulin	-	Numerical, Continuous
BMI	-	Categorical, Numerical, Continuous
Pedigree	-	Numerical, Continuous
Age	-	Numerical, Continuous
Label	-	Categorical, Asymmetric binary

2. Load the dataset and implement Naïve Bayes Classification algorithm using python. Divide the dataset to Training set and testing set. Calculate probabilities and build prediction model. Print the Prediction for Test set, confusion Matrix and accuracy.



In [15]:
```
import pandas as pd
from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

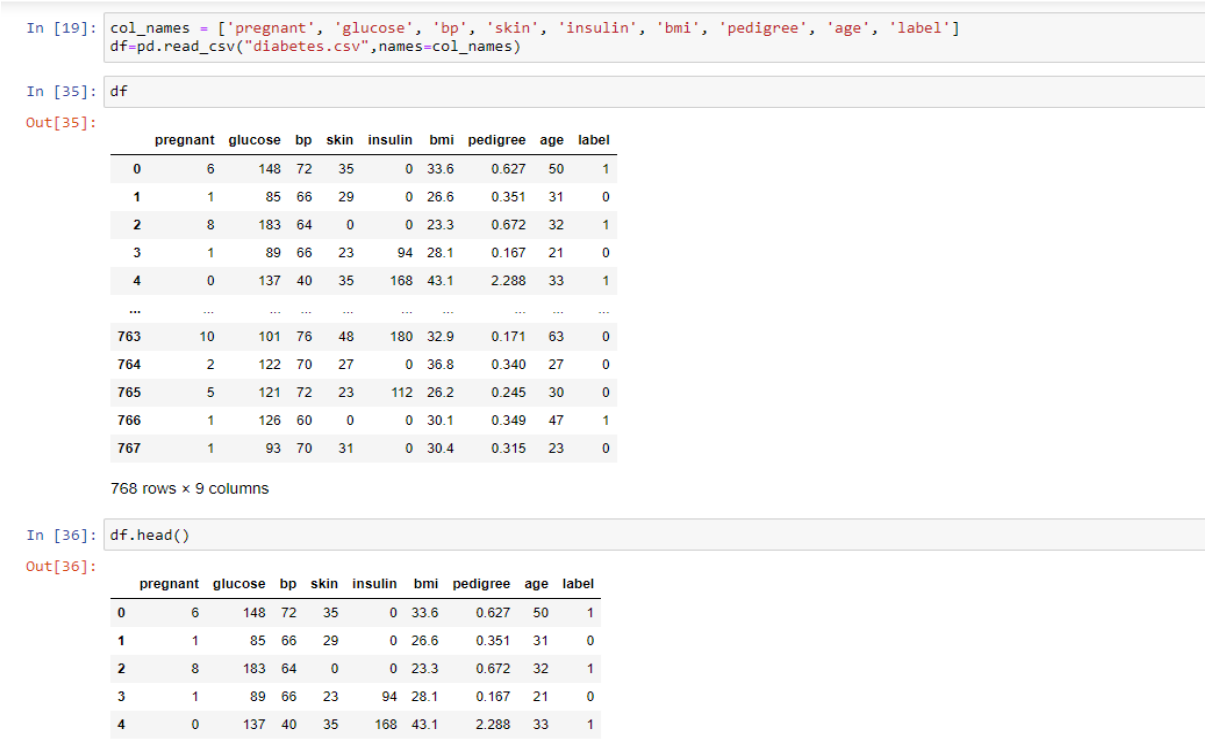
In [16]:
```
df=pd.read_csv("diabetes.csv")
```

In [17]:
```
df
```

Out[17]:

	6	148	72	35	0	33.6	0.627	50	1
0	1	85	66	29	0	26.6	0.351	31	0
1	8	183	64	0	0	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.167	21	0
3	0	137	40	35	168	43.1	2.288	33	1
4	5	116	74	0	0	25.6	0.201	30	0
...	...	...	...	...	...	...	...	...	...
762	10	101	76	48	180	32.9	0.171	63	0
763	2	122	70	27	0	36.8	0.340	27	0
764	5	121	72	23	112	26.2	0.245	30	0
765	1	126	60	0	0	30.1	0.349	47	1
766	1	93	70	31	0	30.4	0.315	23	0

767 rows × 9 columns



In [19]:
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
df=pd.read_csv("diabetes.csv",names=col_names)
```

In [35]:
```
df
```

Out[35]:

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

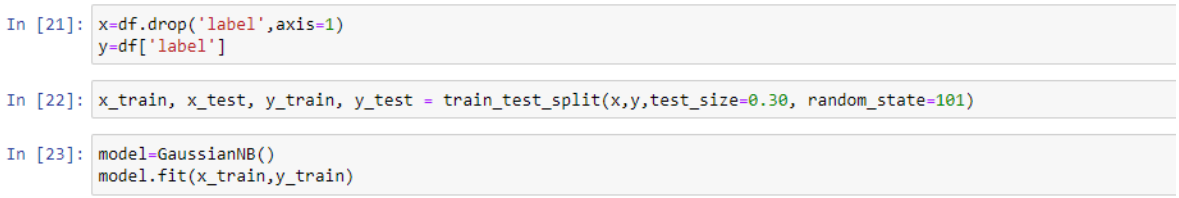
768 rows × 9 columns

In [36]:
```
df.head()
```

Out[36]:

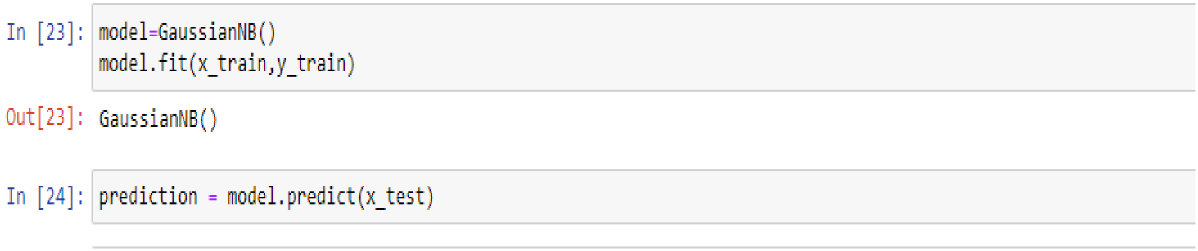
	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Divide dataset to training and test dataset:



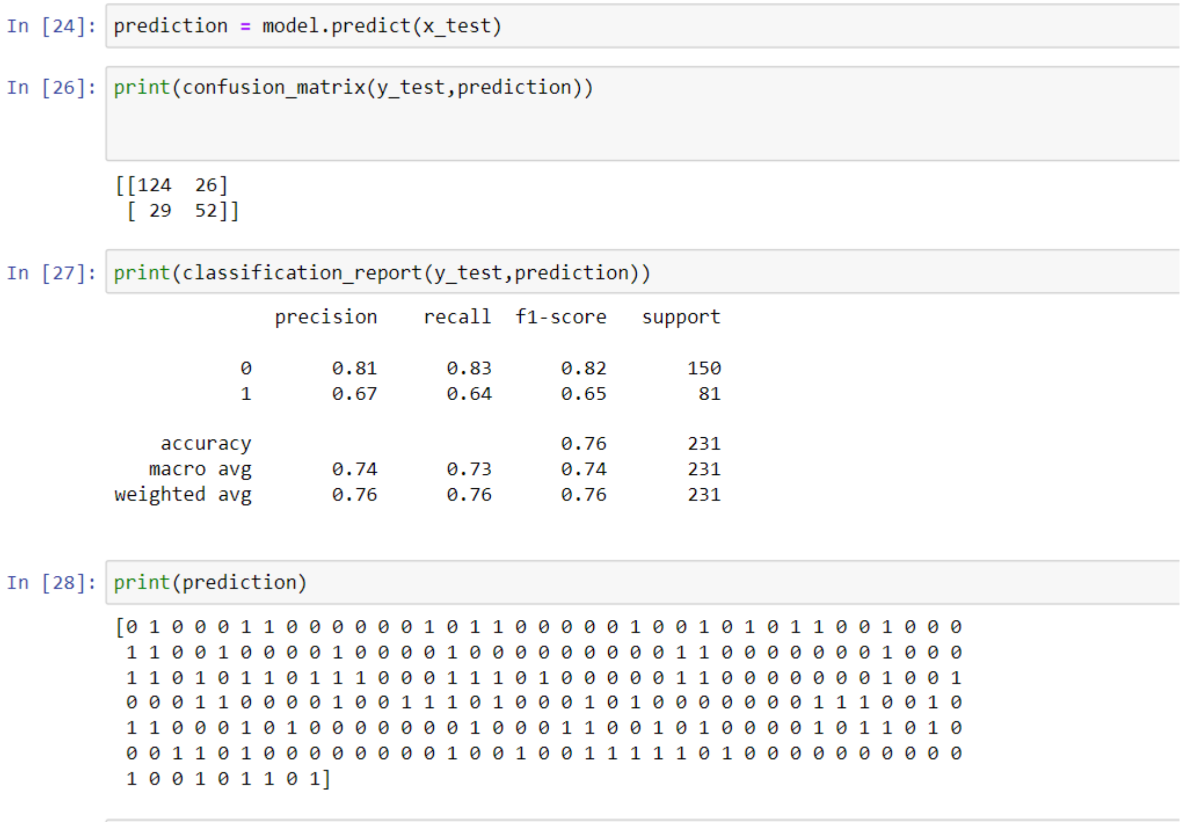
```
In [21]: x=df.drop('label',axis=1)
         y=df['label']

In [22]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.30, random_state=101)

In [23]: model=GaussianNB()
         model.fit(x_train,y_train)
```

prediction model:



```
In [23]: model=GaussianNB()
         model.fit(x_train,y_train)

Out[23]: GaussianNB()

In [24]: prediction = model.predict(x_test)
```

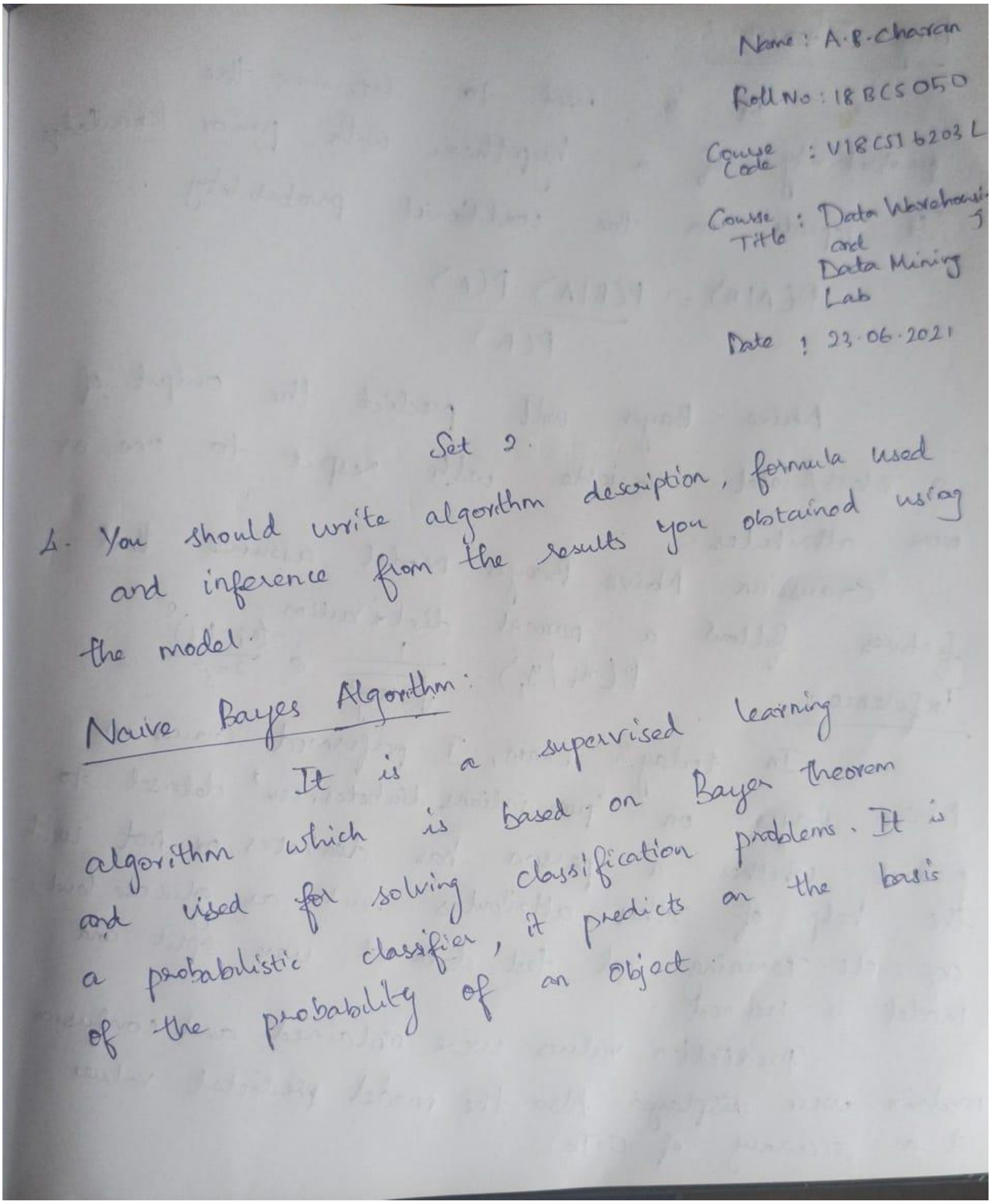
Prediction results and confusion matrix and Accuracy:



```
In [24]: prediction = model.predict(x_test)

In [26]: print(confusion_matrix(y_test,prediction))

         [[124  26]
          [ 29  52]]

In [27]: print(classification_report(y_test,prediction))

                       precision    recall  f1-score   support

                    0       0.81      0.83      0.82       150
                    1       0.67      0.64      0.65        81

             accuracy                           0.76       231
            macro avg       0.74      0.73      0.74       231
         weighted avg       0.76      0.76      0.76       231


In [28]: print(prediction)

         [0 1 0 0 0 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0
          1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0
          1 1 0 1 0 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1
          0 0 0 1 1 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 1 0 0 1 0
          1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 0 1 0 1 1 0 1 0
          0 0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0
          1 0 0 1 0 1 1 0 1]
```

3. Upload in your GITHUB account. Provide the link for access.

4. you should write algorithm description, formulas used and inference from the results you obtained using the model. You should write, scan and upload as pdf.



Name : A.B.Charan

Roll No : 18BCS050

Course Code : V18CS16203L

Course Title : Data Warehousing and Data Mining Lab

Date : 23.06.2021

Set 2.

4. You should write algorithm description, formula used and inference from the results you obtained using the model.

Naive Bayes Algorithm:

It is a supervised learning algorithm which is based on Bayes theorem and used for solving classification problems. It is a probablistic classifier, it predicts on the basis of the probability of an object.

Bayes Theorem :

It is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Naive Bayes will predict the output of a class/ label variable with respect to one or more attributes.

Gaussian Naive Bayes Model assumes that features follow a normal distribution

$$P(x_i / y_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}}\, e^{-\frac{(x_i-\mu_j)^2}{2\sigma_j^2}}$$

Inference:

In today's exam, I performed Gaussian Naive Bayes on 'pima_indians_diabetes.csv' dataset to predict whether a person has diabetes or not with the help of other attributes such as glucose level, age, etc. Training and test data set were split and model is trained.

Prediction values were obtained and confusion matrix were displayed. Also the model predicted values at a accuracy of 0.76.