Assignment : Guassian Naive Bayes

- Diabetes Dataset

1 . Importing Libraries

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
```

2 . Importing Dataset

In [ ]:

```
data = pd.read_csv('./Dataset/diabetes.csv')
```

In [ ]:

```
data.head()
```

Out[ ]:

	glucose	bloodpressure	diabetes
0	40	85	0
1	40	92	0
2	45	63	1
3	45	80	0
4	40	73	1

In [ ]:

```
X = data.iloc[:,:4].values
y = data['diabetes'].values
```

3 . Splitting Dataset into train and test

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

4 . Feature Scaling

In [ ]:

```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

5 . Training the Naive Bayes Classification model on the Training Set

In [ ]:

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

Out[ ]:

```
▾ GaussianNB
GaussianNB()
```

6 . Predicting the Test set results

```
In [ ]: y_pred = classifier.predict(X_test)
        y_pred
```

```
Out[ ]: array([0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1,
               0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0,
               0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1,
               1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
               0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0,
               1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,
               1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1,
               0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
               0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1,
               1], dtype=int64)
```

7 . Confusion Matrix and Accuracy

```
In [ ]: cm = confusion_matrix(y_test, y_pred)
        from sklearn.metrics import accuracy_score
        print ("Accuracy : ", accuracy_score(y_test, y_pred))
        cm
```

```
        Accuracy :  1.0

Out[ ]: array([[101,   0],
               [  0,  98]], dtype=int64)
```

8 . Comparing the Real Values with Predicted Values

```
In [ ]: df = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
        df.head(10)
```

Out[ ]:

	Real Values	Predicted Values
0	0	0
1	1	1
2	0	0
3	1	1
4	1	1
5	0	0
6	1	1
7	0	0
8	0	0
9	1	1