Name: Harsh Patel

Roll NO: A42

Experiment No:7

Aim: Aim: Write a python program to evaluate an Applying gaussian Naïve Bayes learning on iris Dataset

Code :

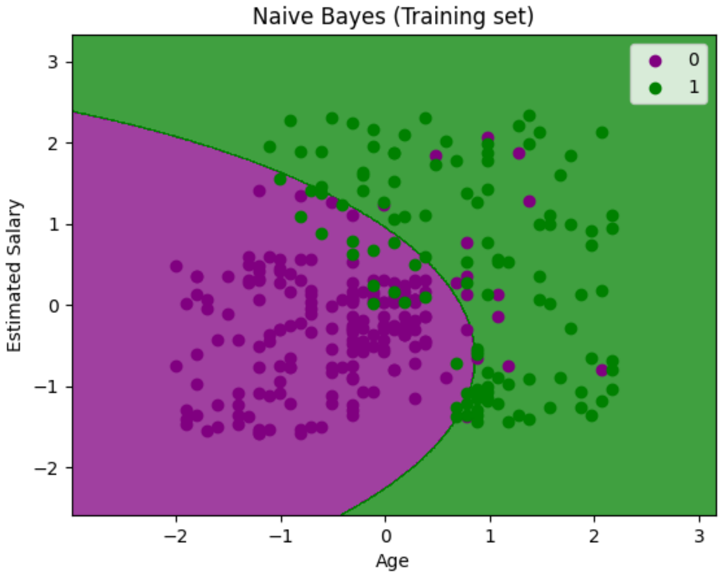
```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

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

from matplotlib.colors import ListedColormap


# Importing the dataset

dataset = pd.read_csv('user_data.csv')

x = dataset.iloc[:, [2, 3]].values

y = dataset.iloc[:, 4].values


# Splitting the dataset into the Training set and Test set

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)


# Feature Scaling

sc = StandardScaler()

x_train = sc.fit_transform(x_train)

x_test = sc.transform(x_test)


# Fitting Naive Bayes to the Training set

classifier = GaussianNB()
```

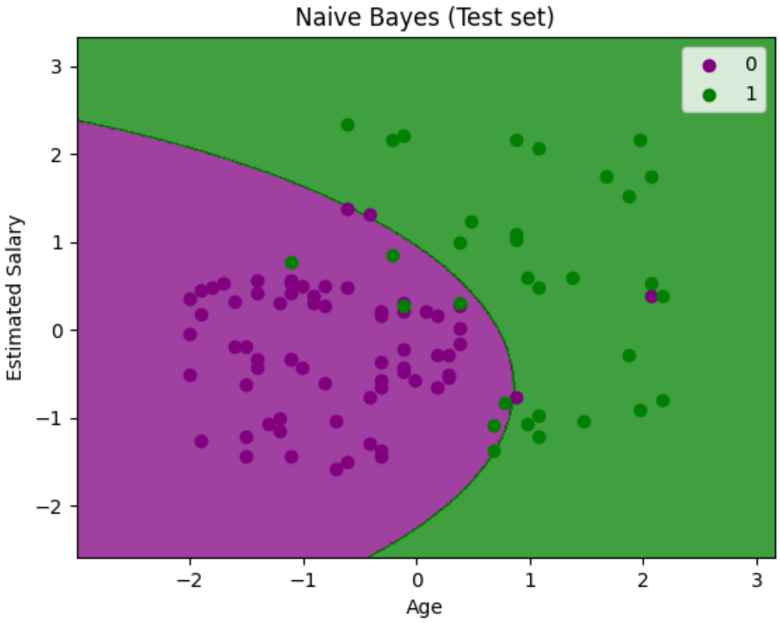
```
classifier.fit(x_train, y_train)


# Predicting the Test set results
y_pred = classifier.predict(x_test)


# Making the Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)


# Visualising the Training set results
def visualize_results(x_set, y_set, title):
    X1, X2 = np.meshgrid(np.arange(start=x_set[:, 0].min() - 1, stop=x_set[:, 0].max() + 1, step=0.01),
                         np.arange(start=x_set[:, 1].min() - 1, stop=x_set[:, 1].max() + 1, step=0.01))
    plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
                 alpha=0.75, cmap=ListedColormap(('purple', 'green')))
    plt.xlim(X1.min(), X1.max())
    plt.ylim(X2.min(), X2.max())
    for i, j in enumerate(np.unique(y_set)):
        plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
                    c=ListedColormap(('purple', 'green'))(i), label=j)
    plt.title(title)
    plt.xlabel('Age')
    plt.ylabel('Estimated Salary')
    plt.legend()
    plt.show()
visualize_results(x_train, y_train, 'Naive Bayes (Training set)')


# Visualising the Test set results
```

```
visualize_results(x_test, y_test, 'Naive Bayes (Test set)')
```

OUTPUT:

```
Confusion Matrix:
[[65  3]
 [ 7 25]]
```


Naive Bayes (Training set)
Estimated Salary
Age


Naive Bayes (Test set)
Estimated Salary
Age