PROG 5 NAIVE BAYES CLASSIFICATION

AIM:

To upload Iris.csv dataset with sepal length, Sepal width, petal length, petal width.

Employee Naive Bayes Classification methodology and predict the type of species for a given set of attributes.

Note : Do not use Predefined Function

SOURCE CODE:

```
import pandas as pd

import numpy as np

from sklearn.metrics import confusion_matrix, f1_score

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_squared_error



data = pd.read_csv("./iris.csv")

data = data.drop(data[data["variety"] == "Setosa"].index)

uni = data["variety"].unique()

data["variety"] = data["variety"].replace(uni, [0, 1])



def calculate_prior(df, Y):

    classes = sorted(list(df[Y].unique()))

    prior = []

    z = len(df)

    for i in classes:

        prior.append(len(df[df[Y] == i])/z)

    return prior



def calculate_likelihood_gaussian(df, feat_name, feat_val, Y, label):

    feat = list(df.columns)
    df = df[df[Y]==label]
    mean, std = df[feat_name].mean(), df[feat_name].std()
    p_x_given_y = (1 / (np.sqrt(2 * np.pi) * std)) *  np.exp(-((feat_val-mean)*2 / (2 * std*2 )))
    return p_x_given_y


def naive_bayes_gaussian(df, X, Y):
    # get feature names
    features = list(df.columns)[:-1]

    # calculate prior
    prior = calculate_prior(df, Y)

    Y_pred = []
    # loop over every data sample
    for x in X:
        # calculate likelihood
        labels = sorted(list(df[Y].unique()))
        likelihood = [1]*len(labels)
        for j in range(len(labels)):
            for i in range(len(features)):
                likelihood[j] *= calculate_likelihood_gaussian(df, features[i], x[i], Y, labels[j])

        # calculate posterior probability (numerator only)
        post_prob = [1]*len(labels)
        for j in range(len(labels)):
            post_prob[j] = likelihood[j] * prior[j]

        Y_pred.append(np.argmax(post_prob))

    return np.array(Y_pred)
```

```
train, test = train_test_split(data, test_size=.2, random_state=41)


X_test = test.iloc[:,:-1].values
Y_test = test.iloc[:,-1].values
Y_pred = naive_bayes_gaussian(train, X=X_test, Y="variety")



print("The accuracy of the model is",f1_score(Y_test, Y_pred))
```

OUTPUT:

```
The accuracy of the model is 0.8181818181818181
```

RESULT:

Thus the program is executed and the output is verified successfully.