```
#To implement Naïve bayes algorithm using any inbuild and external data set.
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
##Importing of dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [1, 2, 3]].values
y = dataset.iloc[:, -1].values
Χ
    array([['Male', 19, 19000],
['Male', 35, 20000],
          ['Female', 26, 43000],
          ['Female', 50, 20000],
          ['Male', 36, 33000],
['Female', 49, 36000]], dtype=object)
    array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
          1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
          0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1,
          1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1,
          1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1,
          0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1,
          1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
          0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0,
          1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1,
          0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1,
          1, 1, 0, 1])
dataset
```

		User ID	Gender	Age	EstimatedSalary	Purchased
	0	15624510	Male	19	19000	0
	1	15810944	Male	35	20000	0
	2	15668575	Female	26	43000	0
	3	15603246	Female	27	57000	0
	4	15804002	Male	19	76000	0
;	395	15691863	Female	46	41000	1
;	396	15706071	Male	51	23000	1
;	397	15654296	Female	50	20000	1
;	398	15755018	Male	36	33000	0
;	399	15594041	Female	49	36000	1
400 rows × 5 columns						

```
##LabelEncoder
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X[:,0] = le.fit_transform(X[:,0])
```

Χ

array([[1, 19, 19000],

```
[1, 35, 20000],
            [0, 26, 43000],
            [0, 50, 20000],
            [1, 36, 33000],
            [0, 49, 36000]], dtype=object)
##Train test Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
##Feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X train = sc.fit transform(X train)
X_test = sc.transform(X_test)
##Training the Naive Bayes model on the training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
     GaussianNB()
y_pred = classifier.predict(X_test)
y_pred
     array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
            0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,
            1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
            0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1])
y_test
     array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
            0,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,
            1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
            0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1])
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test,y_pred)
ac
     0.925
cm
     array([[56, 2],
            [ 4, 18]])
# Single code
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

```
# Feature Scaling
from \ sklearn.preprocessing \ import \ StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
# Predicting the Test set results
y_pred = classifier.predict(X_test)
# Making the Confusion Matrix
\begin{tabular}{ll} \hline from $sklearn.metrics import confusion\_matrix, accuracy\_score \\ \hline \end{tabular}
ac = accuracy_score(y_test,y_pred)
cm = confusion_matrix(y_test, y_pred)
ac
     0.9125
cm
     array([[55, 3],
[ 4, 18]])
```

• ×