In [1]:

```
import pandas as pd
import os
import joblib as jb
import sklearn
import pydotplus
```

In [3]:

```
from sklearn.preprocessing import LabelEncoder
```

In [4]:

```
pd.read_excel('Combined_Updated.xlsx')
```

Out[4]:

	AgeCategory	Workclass	Education	EducationNum	MaritalStatus	Occupation	Relati
0	3	6	9	12	4	0	
1	0	5	9	12	2	3	
2	3	3	11	8	0	5	
3	0	3	1	6	2	5	
4	3	3	9	12	2	9	
...	...	...	...	...	...	...	
48837	3	3	9	12	0	9	
48838	2	3	11	8	6	9	
48839	3	3	9	12	2	9	
48840	0	3	9	12	0	0	
48841	3	4	9	12	2	3	

48842 rows × 15 columns

In [5]:

```
data=pd.read_excel('Combined_Updated.xlsx')
```

In [6]:

```
xc=['AgeCategory','Workclass','Education','EducationNum','MaritalStatus','Occupation','R
y=['Yes','No']
all_input=data[xc]
all_class=data['Class']
```

In [8]:

```
from sklearn.model_selection import train_test_split
```

In [9]:

```
(X_train,X_test,Y_train,Y_test)=train_test_split(all_input,all_class,train_size=0.67,ran
```

In [11]:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

In [12]:

```
clf=GaussianNB()
clf.fit(X_train,Y_train)
```

Out[12]:

GaussianNB

GaussianNB()

In [14]:

```
Y_train_pred=clf.predict(X_train)

Y_test_pred=clf.predict(X_test)


print(classification_report(Y_train, Y_train_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.87      0.87     24905
           1       0.59      0.59      0.59      7819

    accuracy                           0.81     32724
   macro avg       0.73      0.73      0.73     32724
weighted avg       0.81      0.81      0.81     32724
```

In [15]:

```
from sklearn import metrics,model_selection,preprocessing
wrong_train_pred=(Y_train !=Y_train_pred).sum()
print("Total wrong detected on training data= {}".format(wrong_train_pred))

accuracy_train=metrics.accuracy_score(Y_train,Y_train_pred)
print("Accuracy of this model on training data= {:.3f}".format(accuracy_train))
```

```
Total wrong detected on training data= 6372
Accuracy of this model on training data= 0.805
```

In [16]:

```
wrong_test_pred=(Y_test !=Y_test_pred).sum()
print("Total wrong detected on test data = {}".format(wrong_test_pred))

accuracy_test=metrics.accuracy_score(Y_test,Y_test_pred)
print("Accuracy of this model on test data = {:.3f}".format(accuracy_test))
```

```
Total wrong detected on test data = 3228
Accuracy of this model on test data = 0.800
```

In [ ]: