

In [1]:

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

In [2]:

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

In [3]:

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

Out[3]:

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

48842 rows × 15 columns

In [4]:

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

In [5]:

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

In [6]:

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

In [7]:

```
_train,X_test,Y_train,Y_test)=train_test_split(all_input,all_class,train_size=0.67,random_state=1)
```

In [8]:

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

In [10]:

```
from sklearn.preprocessing import StandardScaler
```

In [11]:

```
clf = StandardScaler()
```

In [13]:

```
X_train_scaled = clf.fit_transform(X_train)
X_test_scaled = clf.fit_transform(X_test)
```

In [16]:

```
from sklearn.linear_model import LogisticRegression
```

In [17]:

```
clf = LogisticRegression(random_state=0).fit(X_train_scaled,Y_train)
```

In [18]:

```
Y_train_pred=clf.predict(X_train_scaled)
Y_test_pred=clf.predict(X_test_scaled)
```

In [19]:

```
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= 5879
Accuracy of this model on training data= 0.820

In [20]:

```
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 = 2886
Accuracy of this model on test data = 0.821

In []: