Naive Bayes Classifier in Python

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
dataset=pd.read_csv(r"D:\Data Science with AI\8th-jan-2023\project\adult.csv")
```

In [4]:
```
dataset
```

Out[4]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Fe
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Fe
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Fe
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Fe
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Fe
...	...	...	...	...	...	...	...	...	...	
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	White	
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Fe
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Fe
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	

32561 rows × 15 columns

In [5]:
```
dataset.shape
```

Out[5]:
```
(32561, 15)
```

In [6]:
```
dataset.head()
```

Out[6]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female

In [7]:
```
#rename columns names

col_names=['age' ,'workclass','fnlwgt', 'education',    'education.num',    'marital.st
dataset.columns=col_names
dataset.columns
```

Out[7]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education.num',
       'marital.status', 'occupation', 'relationship', 'race', 'sex',
       'capital.gain', 'capital.loss', 'hours.per.week', 'native.country',
       'income'],
      dtype='object')
```

In [8]:
```
# let's again preview the dataset

dataset.head()
```

Out[8]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female

In [9]:
```
#summary
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64 
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64 
 11  capital.loss    32561 non-null  int64 
 12  hours.per.week  32561 non-null  int64 
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [10]:
```
#explore categorical variable

categorical=[var for var in dataset.columns if dataset[var].dtype=='O']

print('there are {} categorical variables\n'.format(len(categorical)))

print('the categorical variables are :\n\n', categorical)
```

```
there are 9 categorical variables

the categorical variables are :

 ['workclass', 'education', 'marital.status', 'occupation', 'relationship', 'race', 's
ex', 'native.country', 'income']
```

In [11]:
```
# view the categorical variables

dataset[categorical].head()
```

Out[11]:

	workclass	education	marital.status	occupation	relationship	race	sex	native.country	income
0	?	HS-grad	Widowed	?	Not-in-family	White	Female	United-States	<=50K
1	Private	HS-grad	Widowed	Exec-managerial	Not-in-family	White	Female	United-States	<=50K
2	?	Some-college	Widowed	?	Unmarried	Black	Female	United-States	<=50K
3	Private	7th-8th	Divorced	Machine-op-inspct	Unmarried	White	Female	United-States	<=50K
4	Private	Some-college	Separated	Prof-specialty	Own-child	White	Female	United-States	<=50K

In [12]:
```
#missing value in categorical values

dataset[categorical].isnull().sum()
```

Out[12]:
```
workclass         0
education         0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
native.country    0
income            0
dtype: int64
```

In [13]:
```
# frequency count of values in categorical variables
```

In [14]:
```
for var in categorical:
    print(dataset[var].value_counts())
```

```
workclass
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
?                    1836
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: count, dtype: int64
education
HS-grad         10501
Some-college     7291
Bachelors        5355
Masters          1723
Assoc-voc        1382
11th             1175
Assoc-acdm       1067
```

In [15]:
```
#view frequecy distribution of categorical variables

for var in categorical:
    print(dataset[var].value_counts()/np.float16(len(dataset)))
```

```
workclass
Private             0.697052
Self-emp-not-inc    0.078041
Local-gov           0.064281
?                   0.056388
State-gov           0.039865
Self-emp-inc        0.034275
Federal-gov         0.029484
Without-pay         0.000430
Never-worked        0.000215
Name: count, dtype: float64
education
HS-grad         0.322512
Some-college    0.223925
Bachelors       0.164466
Masters         0.052918
Assoc-voc       0.042445
11th            0.036087
Assoc-acdm      0.032770
```

In [ ]:
```

```

In [16]:
```
#exploare workclass variable

dataset.workclass.unique()
```

Out[16]:
```
array(['?', 'Private', 'State-gov', 'Federal-gov', 'Self-emp-not-inc',
       'Self-emp-inc', 'Local-gov', 'Without-pay', 'Never-worked'],
      dtype=object)
```

In [17]:
```
# replace '?' values in workclass variable with 'NAN'

dataset['workclass'].replace('?',np.NAN, inplace=True)
```

In [18]:
```
# again check the frequency distribution of values in workclass variable
dataset.workclass.value_counts()
```

Out[18]:
```
workclass
Private             22696
Self-emp-not-inc     2541
Local-gov            2093
State-gov            1298
Self-emp-inc         1116
Federal-gov           960
Without-pay            14
Never-worked            7
Name: count, dtype: int64
```

In [19]:
```
# explore occupation variable
dataset.occupation.unique()
```

Out[19]:
```
array(['?', 'Exec-managerial', 'Machine-op-inspct', 'Prof-specialty',
       'Other-service', 'Adm-clerical', 'Craft-repair',
       'Transport-moving', 'Handlers-cleaners', 'Sales',
       'Farming-fishing', 'Tech-support', 'Protective-serv',
       'Armed-Forces', 'Priv-house-serv'], dtype=object)
```

In [20]:
```
# check frequency distribution of values in occupation variable
dataset.occupation.value_counts()
```

Out[20]:
```
occupation
Prof-specialty       4140
Craft-repair         4099
Exec-managerial      4066
Adm-clerical         3770
Sales                3650
Other-service        3295
Machine-op-inspct    2002
?                    1843
Transport-moving     1597
Handlers-cleaners    1370
Farming-fishing       994
Tech-support          928
Protective-serv       649
Priv-house-serv       149
Armed-Forces            9
Name: count, dtype: int64
```

In [21]:
```
# replace'?' values in occupation variable with nan
dataset['occupation'].replace('?',np.NAN, inplace=True)
```

In [22]:
```
# check the frequency distribution values in occupation variable

dataset.occupation.value_counts()
```

Out[22]:
```
occupation
Prof-specialty       4140
Craft-repair         4099
Exec-managerial      4066
Adm-clerical         3770
Sales                3650
Other-service        3295
Machine-op-inspct    2002
Transport-moving     1597
Handlers-cleaners    1370
Farming-fishing       994
Tech-support          928
Protective-serv       649
Priv-house-serv       149
Armed-Forces            9
Name: count, dtype: int64
```

explore native_country variable

In [23]:
```
#check labels in native country variable

dataset['native.country'].unique()
```

Out[23]:
```
array(['United-States', '?', 'Mexico', 'Greece', 'Vietnam', 'China',
       'Taiwan', 'India', 'Philippines', 'Trinadad&Tobago', 'Canada',
       'South', 'Holand-Netherlands', 'Puerto-Rico', 'Poland', 'Iran',
       'England', 'Germany', 'Italy', 'Japan', 'Hong', 'Honduras', 'Cuba',
       'Ireland', 'Cambodia', 'Peru', 'Nicaragua', 'Dominican-Republic',
       'Haiti', 'El-Salvador', 'Hungary', 'Columbia', 'Guatemala',
       'Jamaica', 'Ecuador', 'France', 'Yugoslavia', 'Scotland',
       'Portugal', 'Laos', 'Thailand', 'Outlying-US(Guam-USVI-etc)'],
      dtype=object)
```

In [24]:
```
dataset
```

Out[24]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
0	90	NaN	77053	HS-grad	9	Widowed	NaN	Not-in-family	White	Fe
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Fe
2	66	NaN	186061	Some-college	10	Widowed	NaN	Unmarried	Black	Fe
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Fe
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Fe
...	...	...	...	...	...	...	...	...	...	
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	White	
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Fe
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Fe
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	

32561 rows × 15 columns

In [25]:
```
# check frequency distribution of values in native_country variable
dataset['native.country'].value_counts()
```

Out[25]:
```
native.country
United-States                 29170
Mexico                          643
?                               583
Philippines                     198
Germany                         137
Canada                          121
Puerto-Rico                     114
El-Salvador                     106
India                           100
Cuba                             95
England                          90
Jamaica                          81
South                            80
China                            75
Italy                            73
Dominican-Republic               70
Vietnam                          67
Guatemala                        64
Japan                            62
Poland                           60
Columbia                         59
Taiwan                           51
Haiti                            44
Iran                             43
Portugal                         37
Nicaragua                        34
Peru                             31
Greece                           29
France                           29
Ecuador                          28
Ireland                          24
Hong                             20
Cambodia                         19
Trinadad&Tobago                  19
Laos                             18
Thailand                         18
Yugoslavia                       16
Outlying-US(Guam-USVI-etc)       14
Hungary                          13
Honduras                         13
Scotland                         12
Holand-Netherlands                1
Name: count, dtype: int64
```

In [26]:
```
# replace '?' values in native_country variable with 'NAN'

dataset['native.country'].replace('?',np.NAN,inplace=True)
```

In [27]:
```
#check frequency distribution values

dataset['native.country'].value_counts()
```

Out[27]:
```
native.country
United-States                 29170
Mexico                          643
Philippines                     198
Germany                         137
Canada                          121
Puerto-Rico                     114
El-Salvador                     106
India                           100
Cuba                             95
England                          90
Jamaica                          81
South                            80
China                            75
Italy                            73
Dominican-Republic               70
Vietnam                          67
Guatemala                        64
Japan                            62
Poland                           60
Columbia                         59
Taiwan                           51
Haiti                            44
Iran                             43
Portugal                         37
Nicaragua                        34
Peru                             31
Greece                           29
France                           29
Ecuador                          28
Ireland                          24
Hong                             20
Trinadad&Tobago                  19
Cambodia                         19
Thailand                         18
Laos                             18
Yugoslavia                       16
Outlying-US(Guam-USVI-etc)       14
Hungary                          13
Honduras                         13
Scotland                         12
Holand-Netherlands                1
Name: count, dtype: int64
```

check missing values in categorical variables again

In [28]:
```
dataset[categorical].isnull().sum()
```

Out[28]:
```
workclass         1836
education            0
marital.status       0
occupation        1843
relationship         0
race                 0
sex                  0
native.country     583
income               0
dtype: int64
```

number of labels: cardinality

In [29]:
```
# check for cardinaliity in categorical variable
for var in categorical:
    print(var,'contains',len(dataset[var].unique()),'labels')
```

```
workclass contains 9 labels
education contains 16 labels
marital.status contains 7 labels
occupation contains 15 labels
relationship contains 6 labels
race contains 5 labels
sex contains 2 labels
native.country contains 42 labels
income contains 2 labels
```

explore numerical variables

In [30]:
```
# find numerical variable

numerical=[var for var in dataset.columns if dataset[var].dtype!='o']
print('there are {} numerical variables\n'.format(len(numerical)))
print('the numeric variables are :',numerical)
```

```
there are 15 numerical variables

the numeric variables are : ['age', 'workclass', 'fnlwgt', 'education', 'education.nu
m', 'marital.status', 'occupation', 'relationship', 'race', 'sex', 'capital.gain', 'ca
pital.loss', 'hours.per.week', 'native.country', 'income']
```

In [31]:
```
# view numerical variables

dataset[numerical].head()
```

Out[31]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
0	90	NaN	77053	HS-grad	9	Widowed	NaN	Not-in-family	White	Female
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female
2	66	NaN	186061	Some-college	10	Widowed	NaN	Unmarried	Black	Female
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female

missing values in numerical variable

In [32]:
```
# check missing values in numerical variables

dataset[numerical].isnull().sum()
```

Out[32]:
```
age                 0
workclass        1836
fnlwgt              0
education           0
education.num       0
marital.status      0
occupation       1843
relationship        0
race                0
sex                 0
capital.gain        0
capital.loss        0
hours.per.week      0
native.country    583
income              0
dtype: int64
```

declare feature vector and target variable

In [33]:
```
x=dataset.drop(['income'],axis=1)
y=dataset['income']
```

split the data into training and testing set

In [34]:
```
# split x and y

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [35]:
```
# check the shape of x_train and x_test

x_train.shape,x_test.shape
```

Out[35]:
```
((22792, 14), (9769, 14))
```

featuere engineering

In [36]:
```
# check datatype in x_train
x_train.dtypes
```

Out[36]:
```
age               int64
workclass        object
fnlwgt            int64
education        object
education.num     int64
marital.status   object
occupation       object
relationship     object
race             object
sex              object
capital.gain      int64
capital.loss      int64
hours.per.week    int64
native.country   object
dtype: object
```

In [37]:
```
# display categorical variables
categorical=[col for col in x_train.columns if x_train[col].dtypes=='O']
categorical
```

Out[37]:
```
['workclass',
 'education',
 'marital.status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native.country']
```

In [38]:
```
# display numerical variables

numerical=[col for col in x_train.columns if x_train[col].dtypes!='O']
numerical
```

Out[38]:
```
['age',
 'fnlwgt',
 'education.num',
 'capital.gain',
 'capital.loss',
 'hours.per.week']
```

engineering missing values in categorical variables

In [39]:
```
# print percentage of missing values in categorical variables

x_train[categorical].isnull().mean()
```

Out[39]:
```
workclass         0.056774
education         0.000000
marital.status    0.000000
occupation        0.057038
relationship      0.000000
race              0.000000
sex               0.000000
native.country    0.018208
dtype: float64
```

In [40]:
```
#print categorical variables with missing data

for col in categorical:
    if x_train[col].isnull().mean()>0:
        print(col, (x_train[col].isnull().mean()))
```

```
workclass 0.056774306774306775
occupation 0.057037557037557036
native.country 0.018208143208143207
```

In [41]:
```
#impute missing categorical variables with most frequent value

for df2 in [x_train,x_test]:
    df2['workclass'].fillna(x_train['workclass'].mode()[0],inplace=True)
    df2['occupation'].fillna(x_train['occupation'].mode()[0],inplace=True)
    df2['native.country'].fillna(x_train['native.country'].mode()[0],inplace=True)
```

In [42]:
```
#check missing values in categorical variables in x_train

x_train[categorical].isnull().sum()
```

Out[42]:
```
workclass         0
education         0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
native.country    0
dtype: int64
```

In [43]:
```
x_test[categorical].isnull().sum()
```

Out[43]:
```
workclass         0
education         0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
native.country    0
dtype: int64
```

In [44]:
```
#checking missing values in x_train
x_train.isnull().sum()
```

Out[44]:
```
age               0
workclass         0
fnlwgt            0
education         0
education.num     0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
capital.gain      0
capital.loss      0
hours.per.week    0
native.country    0
dtype: int64
```

In [45]:
```
x_test.isnull().sum()
```

Out[45]:
```
age               0
workclass         0
fnlwgt            0
education         0
education.num     0
marital.status    0
occupation        0
relationship      0
race              0
sex               0
capital.gain      0
capital.loss      0
hours.per.week    0
native.country    0
dtype: int64
```

encode categorical variable

In [46]:
```
#print categorical variable
categorical
```

Out[46]:
```
['workclass',
 'education',
 'marital.status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native.country']
```

In [47]:
```
x_train[categorical].head()
```

Out[47]:

	workclass	education	marital.status	occupation	relationship	race	sex	native.country
32098	State-gov	Bachelors	Married-civ-spouse	Exec-managerial	Wife	White	Female	United-States
25206	Local-gov	HS-grad	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	United-States
23491	Private	Some-college	Never-married	Exec-managerial	Not-in-family	White	Female	United-States
12367	Local-gov	HS-grad	Never-married	Farming-fishing	Own-child	White	Male	United-States
7054	Federal-gov	Masters	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States

In [48]:
```
#!pip install --upgrade category_encoders
```

In [49]:
```
# import categorical encoder

import category_encoders as ce
```

In [50]:
```
x_train.head()
```

Out[50]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	State-gov	31627	Bachelors	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	Local-gov	236391	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	Private	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	Local-gov	273929	HS-grad	9	Never-married	Farming-fishing	Own-child	White	
7054	38	Federal-gov	99527	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	

In [51]:
```
x_train.shape
```

Out[51]:
```
(22792, 14)
```

In [52]:
```
x_test.head()
```

Out[52]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	Private	274475	9th	5	Widowed	Sales	Unmarried	White
8950	19	Private	237455	HS-grad	9	Never-married	Handlers-cleaners	Own-child	White
7838	23	Private	125491	Some-college	10	Never-married	Other-service	Not-in-family	Asian-Pac-Islander
16505	37	Federal-gov	48779	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
19140	49	Private	423222	Masters	14	Married-civ-spouse	Sales	Husband	White

In [53]:
```
x_train.shape
```

Out[53]:
```
(22792, 14)
```

In [54]:
```
x_train.head()
```

Out[54]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	State-gov	31627	Bachelors	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	Local-gov	236391	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	Private	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	Local-gov	273929	HS-grad	9	Never-married	Farming-fishing	Own-child	White	
7054	38	Federal-gov	99527	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	

feature scaling

In [55]:
```
cols=x_train.columns
```

In [56]:
```
cols
```

Out[56]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education.num',
       'marital.status', 'occupation', 'relationship', 'race', 'sex',
       'capital.gain', 'capital.loss', 'hours.per.week', 'native.country'],
      dtype='object')
```

In [57]:
```
x_train
```

Out[57]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	State-gov	31627	Bachelors	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	Local-gov	236391	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	Private	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	Local-gov	273929	HS-grad	9	Never-married	Farming-fishing	Own-child	White	
7054	38	Federal-gov	99527	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
...	...	...	...	...	...	...	...	...	...	...
13123	90	Self-emp-not-inc	282095	Some-college	10	Married-civ-spouse	Farming-fishing	Husband	White	
19648	36	Private	279721	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White	
9845	26	Private	51961	12th	8	Never-married	Sales	Other-relative	Black	
10799	44	Private	115323	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
2732	39	Private	224531	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	

22792 rows × 14 columns

In [58]:
```
#converting categorical to numeriacl data
```

In [59]:
```
from sklearn.preprocessing import RobustScaler
scaler=RobustScaler()
x_train1=scaler.fit_transform(x_train[['age']])
x_train1
```

Out[59]:
```
array([[ 0.15],
       [ 0.1 ],
       [ 0.25],
       ...,
       [-0.55],
       [ 0.35],
       [ 0.1 ]])
```

In [60]:
```
x_train['workclass']
```

Out[60]:
```
32098          State-gov
25206          Local-gov
23491            Private
12367          Local-gov
7054         Federal-gov
               ...
13123    Self-emp-not-inc
19648            Private
9845             Private
10799            Private
2732             Private
Name: workclass, Length: 22792, dtype: object
```

In [61]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['workclass']=lb.fit_transform(x_train[['workclass']])
```

In [62]:
```
x_train
```

Out[62]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	6	31627	Bachelors	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	1	236391	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	3	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	1	273929	HS-grad	9	Never-married	Farming-fishing	Own-child	White	
7054	38	0	99527	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	Some-college	10	Married-civ-spouse	Farming-fishing	Husband	White	
19648	36	3	279721	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White	
9845	26	3	51961	12th	8	Never-married	Sales	Other-relative	Black	
10799	44	3	115323	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
2732	39	3	224531	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	

22792 rows × 14 columns

In [63]:
```
cols=x_train.columns
```

In [64]:
```
x_train
```

Out[64]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	6	31627	Bachelors	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	1	236391	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	3	194710	Some-college	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	1	273929	HS-grad	9	Never-married	Farming-fishing	Own-child	White	
7054	38	0	99527	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	Some-college	10	Married-civ-spouse	Farming-fishing	Husband	White	
19648	36	3	279721	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White	
9845	26	3	51961	12th	8	Never-married	Sales	Other-relative	Black	
10799	44	3	115323	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	
2732	39	3	224531	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	

22792 rows × 14 columns

In [65]:
```
['workclass']
```

Out[65]:
```
['workclass']
```

In [66]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['education']=lb.fit_transform(x_train[['education']])
```

In [67]:
```
x_train
```

Out[67]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	6	31627	9	13	Married-civ-spouse	Exec-managerial	Wife	White	Fe
25206	39	1	236391	11	9	Married-civ-spouse	Machine-op-inspct	Husband	White	
23491	42	3	194710	15	10	Never-married	Exec-managerial	Not-in-family	White	Fe
12367	27	1	273929	11	9	Never-married	Farming-fishing	Own-child	White	
7054	38	0	99527	12	14	Married-civ-spouse	Exec-managerial	Husband	White	
...	...	...	...	...	...	...	...	...	...	
13123	90	5	282095	15	10	Married-civ-spouse	Farming-fishing	Husband	White	
19648	36	3	279721	11	9	Married-civ-spouse	Transport-moving	Husband	White	
9845	26	3	51961	2	8	Never-married	Sales	Other-relative	Black	
10799	44	3	115323	12	14	Married-civ-spouse	Exec-managerial	Husband	White	
2732	39	3	224531	11	9	Married-civ-spouse	Craft-repair	Husband	White	

22792 rows × 14 columns

In [68]:
```
x_train['marital.status']
```

Out[68]:
```
32098    Married-civ-spouse
25206    Married-civ-spouse
23491         Never-married
12367         Never-married
7054     Married-civ-spouse
                ...
13123    Married-civ-spouse
19648    Married-civ-spouse
9845          Never-married
10799    Married-civ-spouse
2732     Married-civ-spouse
Name: marital.status, Length: 22792, dtype: object
```

In [69]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['marital.status']=lb.fit_transform(x_train[['marital.status']])
```

In [70]:
```
x_train['occupation']
```

Out[70]:
```
32098      Exec-managerial
25206    Machine-op-inspct
23491      Exec-managerial
12367      Farming-fishing
7054       Exec-managerial
               ...
13123      Farming-fishing
19648     Transport-moving
9845                 Sales
10799      Exec-managerial
2732          Craft-repair
Name: occupation, Length: 22792, dtype: object
```

In [71]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['occupation']=lb.fit_transform(x_train[['occupation']])
```

In [72]:
```
x_train
```

Out[72]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	6	31627	9	13	2	3	Wife	White	Fe
25206	39	1	236391	11	9	2	6	Husband	White	
23491	42	3	194710	15	10	4	3	Not-in-family	White	Fe
12367	27	1	273929	11	9	4	4	Own-child	White	
7054	38	0	99527	12	14	2	3	Husband	White	
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	15	10	2	4	Husband	White	
19648	36	3	279721	11	9	2	13	Husband	White	
9845	26	3	51961	2	8	4	11	Other-relative	Black	
10799	44	3	115323	12	14	2	3	Husband	White	
2732	39	3	224531	11	9	2	2	Husband	White	

22792 rows × 14 columns

In [73]:
```
x_train['relationship']
```

Out[73]:
```
32098             Wife
25206          Husband
23491    Not-in-family
12367        Own-child
7054           Husband
               ...
13123          Husband
19648          Husband
9845     Other-relative
10799          Husband
2732           Husband
Name: relationship, Length: 22792, dtype: object
```

In [74]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['relationship']=lb.fit_transform(x_train[['relationship']])
```

In [75]:
```
x_train
```

Out[75]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	se
32098	40	6	31627	9	13	2	3	5	White	Fe
25206	39	1	236391	11	9	2	6	0	White	
23491	42	3	194710	15	10	4	3	1	White	Fe
12367	27	1	273929	11	9	4	4	3	White	
7054	38	0	99527	12	14	2	3	0	White	
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	15	10	2	4	0	White	
19648	36	3	279721	11	9	2	13	0	White	
9845	26	3	51961	2	8	4	11	2	Black	
10799	44	3	115323	12	14	2	3	0	White	
2732	39	3	224531	11	9	2	2	0	White	

22792 rows × 14 columns

In [76]:
```
x_train['race']
```

Out[76]:
```
32098    White
25206    White
23491    White
12367    White
7054     White
         ...
13123    White
19648    White
9845     Black
10799    White
2732     White
Name: race, Length: 22792, dtype: object
```

In [77]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['race']=lb.fit_transform(x_train[['race']])
```

In [78]:
```
x_train
```

Out[78]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
32098	40	6	31627	9	13	2	3	5	4	Fer
25206	39	1	236391	11	9	2	6	0	4	N
23491	42	3	194710	15	10	4	3	1	4	Fer
12367	27	1	273929	11	9	4	4	3	4	N
7054	38	0	99527	12	14	2	3	0	4	N
...	...	...	...	...	...	...	...	...	...	
13123	90	5	282095	15	10	2	4	0	4	N
19648	36	3	279721	11	9	2	13	0	4	N
9845	26	3	51961	2	8	4	11	2	2	N
10799	44	3	115323	12	14	2	3	0	4	N
2732	39	3	224531	11	9	2	2	0	4	N

22792 rows × 14 columns

In [79]:
```
x_train['sex']
```

Out[79]:
```
32098    Female
25206      Male
23491    Female
12367      Male
7054       Male
           ...
13123      Male
19648      Male
9845       Male
10799      Male
2732       Male
Name: sex, Length: 22792, dtype: object
```

In [80]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['sex']=lb.fit_transform(x_train['sex'])
```

In [81]:
```
x_train
```

Out[81]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
32098	40	6	31627	9	13	2	3	5	4	0
25206	39	1	236391	11	9	2	6	0	4	1
23491	42	3	194710	15	10	4	3	1	4	0
12367	27	1	273929	11	9	4	4	3	4	1
7054	38	0	99527	12	14	2	3	0	4	1
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	15	10	2	4	0	4	1
19648	36	3	279721	11	9	2	13	0	4	1
9845	26	3	51961	2	8	4	11	2	2	1
10799	44	3	115323	12	14	2	3	0	4	1
2732	39	3	224531	11	9	2	2	0	4	1

22792 rows × 14 columns

In [82]:
```
x_train['native.country']
```

Out[82]:
```
32098    United-States
25206    United-States
23491    United-States
12367    United-States
7054     United-States
             ...
13123    United-States
19648    United-States
9845     United-States
10799    United-States
2732     United-States
Name: native.country, Length: 22792, dtype: object
```

In [83]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_train['native.country']=lb.fit_transform(x_train[['native.country']])
```

In [84]:
```
x_train
```

Out[84]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
32098	40	6	31627	9	13	2	3	5	4	0
25206	39	1	236391	11	9	2	6	0	4	1
23491	42	3	194710	15	10	4	3	1	4	0
12367	27	1	273929	11	9	4	4	3	4	1
7054	38	0	99527	12	14	2	3	0	4	1
...	...	...	...	...	...	...	...	...	...	...
13123	90	5	282095	15	10	2	4	0	4	1
19648	36	3	279721	11	9	2	13	0	4	1
9845	26	3	51961	2	8	4	11	2	2	1
10799	44	3	115323	12	14	2	3	0	4	1
2732	39	3	224531	11	9	2	2	0	4	1

22792 rows × 14 columns

In [85]:
```
from sklearn.preprocessing import RobustScaler
scaler=RobustScaler()
x_train2=scaler.fit_transform(x_train)
x_train2
```

Out[85]:
```
array([[ 0.15      ,  3.        , -1.22924847, ...,  0.        ,
        -4.        ,  0.        ],
       [ 0.1       , -2.        ,  0.4831759 , ...,  0.        ,
        -0.4       ,  0.        ],
       [ 0.25      ,  0.        ,  0.13460115, ...,  0.        ,
         0.        ,  0.        ],
       ...,
       [-0.55      ,  0.        , -1.05919691, ...,  0.        ,
         2.2       ,  0.        ],
       [ 0.35      ,  0.        , -0.52930575, ...,  0.        ,
         0.        ,  0.        ],
       [ 0.1       ,  0.        ,  0.3839917 , ...,  0.        ,
         0.        ,  0.        ]])
```

In [86]:
```
x_train = pd.DataFrame(x_train2,columns=[cols])
```

In [87]:
```
x_train
```

Out[87]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
0	0.15	3.0	-1.229248	-0.666667	1.000000	0.0	-0.500000	1.333333	0.0
1	0.10	-2.0	0.483176	0.000000	-0.333333	0.0	0.000000	-0.333333	0.0
2	0.25	0.0	0.134601	1.333333	0.000000	1.0	-0.500000	0.000000	0.0
3	-0.50	-2.0	0.797103	0.000000	-0.333333	1.0	-0.333333	0.666667	0.0
4	0.05	-3.0	-0.661406	0.333333	1.333333	0.0	-0.500000	-0.333333	0.0
...	...	...	...	...	...	...	...	...	...
22787	2.65	2.0	0.865395	1.333333	0.000000	0.0	-0.333333	-0.333333	0.0
22788	-0.05	0.0	0.845541	0.000000	-0.333333	0.0	1.166667	-0.333333	0.0
22789	-0.55	0.0	-1.059197	-3.000000	-0.666667	1.0	0.833333	0.333333	-2.0
22790	0.35	0.0	-0.529306	0.333333	1.333333	0.0	-0.500000	-0.333333	0.0
22791	0.10	0.0	0.383992	0.000000	-0.333333	0.0	-0.666667	-0.333333	0.0

22792 rows × 14 columns

In [ ]:
```
```

In [89]:
```
x_train.head()
```

Out[89]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
0	0.15	3.0	-1.229248	-0.666667	1.000000	0.0	-0.500000	1.333333	0.0	-1.0
1	0.10	-2.0	0.483176	0.000000	-0.333333	0.0	0.000000	-0.333333	0.0	0.0
2	0.25	0.0	0.134601	1.333333	0.000000	1.0	-0.500000	0.000000	0.0	-1.0
3	-0.50	-2.0	0.797103	0.000000	-0.333333	1.0	-0.333333	0.666667	0.0	0.0
4	0.05	-3.0	-0.661406	0.333333	1.333333	0.0	-0.500000	-0.333333	0.0	0.0

model training

In [90]:
```
#train a gaussian naive bayes classifier on the training set
from sklearn.naive_bayes import GaussianNB

#instantiate the model
gnb=GaussianNB()

#fit the model
gnb.fit(x_train,y_train)
```

Out[90]:
```
GaussianNB
GaussianNB()
```

In [91]:
```
x_test
```

Out[91]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	Private	274475	9th	5	Widowed	Sales	Unmarried	White
8950	19	Private	237455	HS-grad	9	Never-married	Handlers-cleaners	Own-child	White
7838	23	Private	125491	Some-college	10	Never-married	Other-service	Not-in-family	Asian-Pac-Islander
16505	37	Federal-gov	48779	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
19140	49	Private	423222	Masters	14	Married-civ-spouse	Sales	Husband	White
...	...	...	...	...	...	...	...	...	...
21949	37	Self-emp-not-inc	29054	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
26405	26	Private	165673	HS-grad	9	Married-civ-spouse	Sales	Husband	White
23236	25	Private	156848	Some-college	10	Never-married	Prof-specialty	Own-child	White
26823	34	Private	148226	HS-grad	9	Never-married	Machine-op-inspct	Unmarried	Black
20721	50	Private	94391	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	White

9769 rows × 14 columns

In [93]:
```
x_test['workclass']
```

Out[93]:
```
22278             Private
8950              Private
7838              Private
16505         Federal-gov
19140             Private
                 ...
21949    Self-emp-not-inc
26405             Private
23236             Private
26823             Private
20721             Private
Name: workclass, Length: 9769, dtype: object
```

In [94]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['workclass']=lb.fit_transform(x_test[['workclass']])
```

In [95]:
```
x_test
```

Out[95]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	3	274475	9th	5	Widowed	Sales	Unmarried	White
8950	19	3	237455	HS-grad	9	Never-married	Handlers-cleaners	Own-child	White
7838	23	3	125491	Some-college	10	Never-married	Other-service	Not-in-family	Asian-Pac-Islander
16505	37	0	48779	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
19140	49	3	423222	Masters	14	Married-civ-spouse	Sales	Husband	White
...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
26405	26	3	165673	HS-grad	9	Married-civ-spouse	Sales	Husband	White
23236	25	3	156848	Some-college	10	Never-married	Prof-specialty	Own-child	White
26823	34	3	148226	HS-grad	9	Never-married	Machine-op-inspct	Unmarried	Black
20721	50	3	94391	Assoc-voc	11	Married-civ-spouse	Craft-repair	Husband	White

9769 rows × 14 columns

In [96]:
```
x_test['education']
```

Out[96]:
```
22278            9th
8950         HS-grad
7838    Some-college
16505       Bachelors
19140         Masters
              ...
21949         HS-grad
26405         HS-grad
23236    Some-college
26823         HS-grad
20721       Assoc-voc
Name: education, Length: 9769, dtype: object
```

In [97]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['education']=lb.fit_transform(x_test[['education']])
```

In [98]:
```
x_test
```

Out[98]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	3	274475	6	5	Widowed	Sales	Unmarried	White
8950	19	3	237455	11	9	Never-married	Handlers-cleaners	Own-child	White
7838	23	3	125491	15	10	Never-married	Other-service	Not-in-family	Asian-Pac-Islander
16505	37	0	48779	9	13	Married-civ-spouse	Prof-specialty	Husband	White
19140	49	3	423222	12	14	Married-civ-spouse	Sales	Husband	White
...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	Married-civ-spouse	Farming-fishing	Husband	White
26405	26	3	165673	11	9	Married-civ-spouse	Sales	Husband	White
23236	25	3	156848	15	10	Never-married	Prof-specialty	Own-child	White
26823	34	3	148226	11	9	Never-married	Machine-op-inspct	Unmarried	Black
20721	50	3	94391	8	11	Married-civ-spouse	Craft-repair	Husband	White

9769 rows × 14 columns

In [99]:
```
x_test['marital.status']
```

Out[99]:
```
22278              Widowed
8950         Never-married
7838         Never-married
16505    Married-civ-spouse
19140    Married-civ-spouse
                ...
21949    Married-civ-spouse
26405    Married-civ-spouse
23236         Never-married
26823         Never-married
20721    Married-civ-spouse
Name: marital.status, Length: 9769, dtype: object
```

In [100]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['marital.status']=lb.fit_transform(x_test[['marital.status']])
```

In [101]:
```
x_test
```

Out[101]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	3	274475	6	5	6	Sales	Unmarried	White
8950	19	3	237455	11	9	4	Handlers-cleaners	Own-child	White
7838	23	3	125491	15	10	4	Other-service	Not-in-family	Asian-Pac-Islander
16505	37	0	48779	9	13	2	Prof-specialty	Husband	White
19140	49	3	423222	12	14	2	Sales	Husband	White
...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	2	Farming-fishing	Husband	White
26405	26	3	165673	11	9	2	Sales	Husband	White
23236	25	3	156848	15	10	4	Prof-specialty	Own-child	White
26823	34	3	148226	11	9	4	Machine-op-inspct	Unmarried	Black
20721	50	3	94391	8	11	2	Craft-repair	Husband	White

9769 rows × 14 columns

In [102]:
```
x_test['occupation']
```

Out[102]:
```
22278               Sales
8950      Handlers-cleaners
7838          Other-service
16505        Prof-specialty
19140                 Sales
                 ...
21949       Farming-fishing
26405                 Sales
23236        Prof-specialty
26823     Machine-op-inspct
20721          Craft-repair
Name: occupation, Length: 9769, dtype: object
```

In [103]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['occupation']=lb.fit_transform(x_test[['occupation']])
```

In [104]:
```
x_test
```

Out[104]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	3	274475	6	5	6	11	Unmarried	White
8950	19	3	237455	11	9	4	5	Own-child	White
7838	23	3	125491	15	10	4	7	Not-in-family	Asian-Pac-Islander
16505	37	0	48779	9	13	2	9	Husband	White
19140	49	3	423222	12	14	2	11	Husband	White
...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	2	4	Husband	White
26405	26	3	165673	11	9	2	11	Husband	White
23236	25	3	156848	15	10	4	9	Own-child	White
26823	34	3	148226	11	9	4	6	Unmarried	Black
20721	50	3	94391	8	11	2	2	Husband	White

9769 rows × 14 columns

In [105]:
```
x_test['relationship']
```

Out[105]:
```
22278        Unmarried
8950         Own-child
7838     Not-in-family
16505          Husband
19140          Husband
              ...
21949          Husband
26405          Husband
23236        Own-child
26823        Unmarried
20721          Husband
Name: relationship, Length: 9769, dtype: object
```

In [106]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['relationship']=lb.fit_transform(x_test[['relationship']])
```

In [107]:
```
x_test
```

Out[107]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
22278	56	3	274475	6	5	6	11	4	White
8950	19	3	237455	11	9	4	5	3	White
7838	23	3	125491	15	10	4	7	1	Asian-Pac-Islander
16505	37	0	48779	9	13	2	9	0	White
19140	49	3	423222	12	14	2	11	0	White
...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	2	4	0	White
26405	26	3	165673	11	9	2	11	0	White
23236	25	3	156848	15	10	4	9	3	White
26823	34	3	148226	11	9	4	6	4	Black
20721	50	3	94391	8	11	2	2	0	White

9769 rows × 14 columns

In [108]:
```
x_test['race']
```

Out[108]:
```
22278                 White
8950                  White
7838     Asian-Pac-Islander
16505                 White
19140                 White
                 ...
21949                 White
26405                 White
23236                 White
26823                 Black
20721                 White
Name: race, Length: 9769, dtype: object
```

In [109]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['race']=lb.fit_transform(x_test[['race']])
```

In [110]:
```
x_test
```

Out[110]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
22278	56	3	274475	6	5	6	11	4	4	Fer
8950	19	3	237455	11	9	4	5	3	4	Fer
7838	23	3	125491	15	10	4	7	1	1	Fer
16505	37	0	48779	9	13	2	9	0	4	M
19140	49	3	423222	12	14	2	11	0	4	M
...	...	...	...	...	...	...	...	...	...	
21949	37	5	29054	11	9	2	4	0	4	M
26405	26	3	165673	11	9	2	11	0	4	M
23236	25	3	156848	15	10	4	9	3	4	M
26823	34	3	148226	11	9	4	6	4	2	Fer
20721	50	3	94391	8	11	2	2	0	4	M

9769 rows × 14 columns

In [111]:
```
x_test['sex']
```

Out[111]:
```
22278    Female
8950     Female
7838     Female
16505      Male
19140      Male
          ...
21949      Male
26405      Male
23236      Male
26823    Female
20721      Male
Name: sex, Length: 9769, dtype: object
```

In [112]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['sex']=lb.fit_transform(x_test[['sex']])
```

In [113]:
```
x_test
```

Out[113]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
22278	56	3	274475	6	5	6	11	4	4	0
8950	19	3	237455	11	9	4	5	3	4	0
7838	23	3	125491	15	10	4	7	1	1	0
16505	37	0	48779	9	13	2	9	0	4	1
19140	49	3	423222	12	14	2	11	0	4	1
...	...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	2	4	0	4	1
26405	26	3	165673	11	9	2	11	0	4	1
23236	25	3	156848	15	10	4	9	3	4	1
26823	34	3	148226	11	9	4	6	4	2	0
20721	50	3	94391	8	11	2	2	0	4	1

9769 rows × 14 columns

In [114]:
```
x_test['native.country']
```

Out[114]:
```
22278    United-States
8950     United-States
7838           Vietnam
16505    United-States
19140    United-States
             ...
21949    United-States
26405    United-States
23236    United-States
26823    United-States
20721    United-States
Name: native.country, Length: 9769, dtype: object
```

In [115]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
x_test['native.country']=lb.fit_transform(x_test[['native.country']])
```

In [116]:
```
x_test
```

Out[116]:

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
22278	56	3	274475	6	5	6	11	4	4	0
8950	19	3	237455	11	9	4	5	3	4	0
7838	23	3	125491	15	10	4	7	1	1	0
16505	37	0	48779	9	13	2	9	0	4	1
19140	49	3	423222	12	14	2	11	0	4	1
...	...	...	...	...	...	...	...	...	...	...
21949	37	5	29054	11	9	2	4	0	4	1
26405	26	3	165673	11	9	2	11	0	4	1
23236	25	3	156848	15	10	4	9	3	4	1
26823	34	3	148226	11	9	4	6	4	2	0
20721	50	3	94391	8	11	2	2	0	4	1

9769 rows × 14 columns

predict the result

In [117]:
```
y_pred=gnb.predict(x_test)
y_pred
```

Out[117]:
```
array(['<=50K', '<=50K', '<=50K', ..., '<=50K', '<=50K', '<=50K'],
      dtype='<U5')
```

check accuracy score

In [118]:
```
from sklearn.metrics import accuracy_score
print('model accuracy score:{0:0.4f}'.format(accuracy_score(y_test,y_pred)))
```

```
model accuracy score:0.7585
```

here y_test is the true class labels and y_pred is the predicted class labes in the test set

compare the train set and test set accuracy

now we will compare the train set and test set accuracy to check for overfitting

In [119]:
```
from sklearn.preprocessing import RobustScaler
scaler=RobustScaler()
x_test2=scaler.fit_transform(x_test)
x_test2
```

Out[119]:
```
array([[ 1.        ,  0.        ,  0.81616505, ...,  0.        ,
         0.        ,  0.        ],
       [-0.94736842,  0.        ,  0.50353837, ...,  0.        ,
        -3.        ,  0.        ],
       [-0.73684211,  0.        , -0.44197575, ...,  0.        ,
        -1.        ,  1.        ],
       ...,
       [-0.63157895,  0.        , -0.177172  , ...,  0.        ,
         0.        ,  0.        ],
       [-0.15789474,  0.        , -0.24998311, ...,  0.        ,
         1.6       ,  0.        ],
       [ 0.68421053,  0.        , -0.70460917, ...,  0.        ,
         2.        ,  0.        ]])
```

In [120]:
```
y_pred_train=gnb.predict(x_test2)

y_pred_train.shape
```

Out[120]:
```
(9769,)
```

In [121]:
```
y_train = y_train.reset_index(drop = True)
y_train.head(9769)
```

Out[121]:
```
0       <=50K
1        >50K
2       <=50K
3       <=50K
4        >50K
        ...
9764    <=50K
9765    <=50K
9766     >50K
9767    <=50K
9768    <=50K
Name: income, Length: 9769, dtype: object
```

In [122]:
```
print('Training-set  accuracy score:{0:0.4f}'.format(accuracy_score(y_train.head(9769),
```

```
Training-set  accuracy score:0.6971
```

check for overfitting and underfitting

In [123]:
```
#print the scores on training and test set

print('training set score :{:.4f}'.format(gnb.score(x_train2,y_train)))
print('test set score :{:.4f}'.format(gnb.score(x_test2,y_test)))
```

```
training set score :0.7996
test set score :0.7995
```

compare model accuracy with null accuracy

In [124]:
```
#check class distribution in test set
y_test.value_counts()
```

Out[124]:
```
income
<=50K    7410
>50K     2359
Name: count, dtype: int64
```

In [125]:
```
#check null accuracy score
null_accuracy =(7407/(7407+2362))
print('Null_accuracy score{0:0.4f}'.format(null_accuracy))
```

```
Null_accuracy score0.7582
```

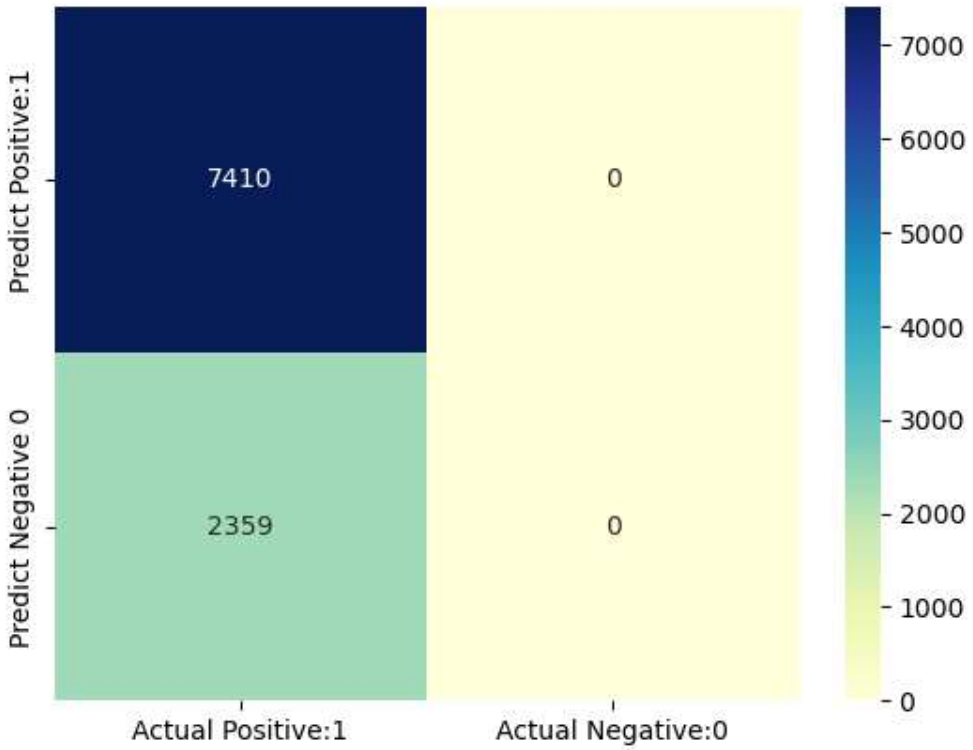
confusion matrix

In [126]:
```
#print the confusion matrix and slice into four pices
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)

print('confusion matrix\n\n',cm)
print('\nTrue Positive(TP)=',cm[0,0])
print('\nTrue Negative(TN)=',cm[1,1])
print('\nFalse Positive(FP)=',cm[0,1])
print('\nFalse Negative(FN)=',cm[1,0])
```

```
confusion matrix

 [[7410    0]
 [2359    0]]

True Positive(TP)= 7410

True Negative(TN)= 0

False Positive(FP)= 0

False Negative(FN)= 2359
```

In [130]:
```
#visulize confusion matrix with seaborn heatmap
cm_matrix = pd.DataFrame(data=cm,columns=['Actual Positive:1','Actual Negative:0'] ,
                                        index= ['Predict Positive:1','Predict Negative 0']

sns.heatmap(cm_matrix, annot=True ,fmt='d',cmap='YlGnBu')
```

Out[130]: <Axes: >



In [ ]:

In [ ]: