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

In [2]:
```python
adult = pd.read_csv(r"C:\Users\786\Downloads\archive (5)\adult.csv")
adult
```

Out[2]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Blac
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Whit
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Whit
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Blac
4	18	?	103497	Some-college	10	Never-married	?	Own-child	Whit
...	...	...	...	...	...	...	...	...	.
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	Whit
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	Whit
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Whit

48842 rows × 15 columns

In [3]:
```
adult.head()
```

Out[3]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	ge
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fe

In [4]:
```
adult.describe()
```

Out[4]:

	age	fnlwgt	educational-num	capital-gain	capital-loss	hours-per-week
count	48842.000000	4.884200e+04	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	1.896641e+05	10.078089	1079.067626	87.502314	40.422382
std	13.710510	1.056040e+05	2.570973	7452.019058	403.004552	12.391444
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.175505e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.781445e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.376420e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

In [5]:
```
adult["income"].describe()
```

Out[5]:
```
count      48842
unique         2
top         <=50K
freq       37155
Name: income, dtype: object
```

In [6]:
```
adult.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   age              48842 non-null  int64 
 1   workclass        48842 non-null  object
 2   fnlwgt           48842 non-null  int64 
 3   education        48842 non-null  object
 4   educational-num  48842 non-null  int64 
 5   marital-status   48842 non-null  object
 6   occupation       48842 non-null  object
 7   relationship     48842 non-null  object
 8   race             48842 non-null  object
 9   gender           48842 non-null  object
 10  capital-gain     48842 non-null  int64 
 11  capital-loss     48842 non-null  int64 
 12  hours-per-week   48842 non-null  int64 
 13  native-country   48842 non-null  object
 14  income           48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

In [7]:
```
adult.nunique()
```

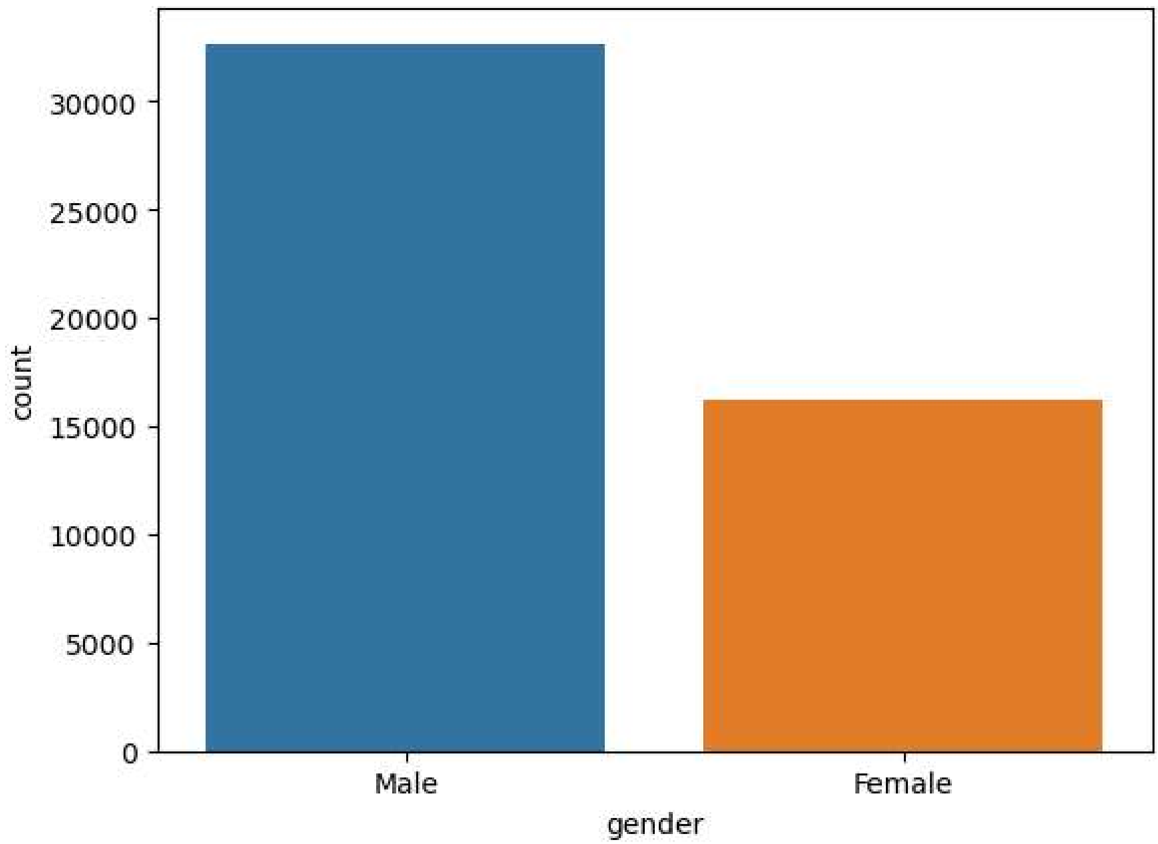
Out[7]:
```
age                   74
workclass              9
fnlwgt             28523
education             16
educational-num       16
marital-status         7
occupation            15
relationship           6
race                   5
gender                 2
capital-gain         123
capital-loss          99
hours-per-week        96
native-country        42
income                 2
dtype: int64
```

In [8]:
```
adult.index
```

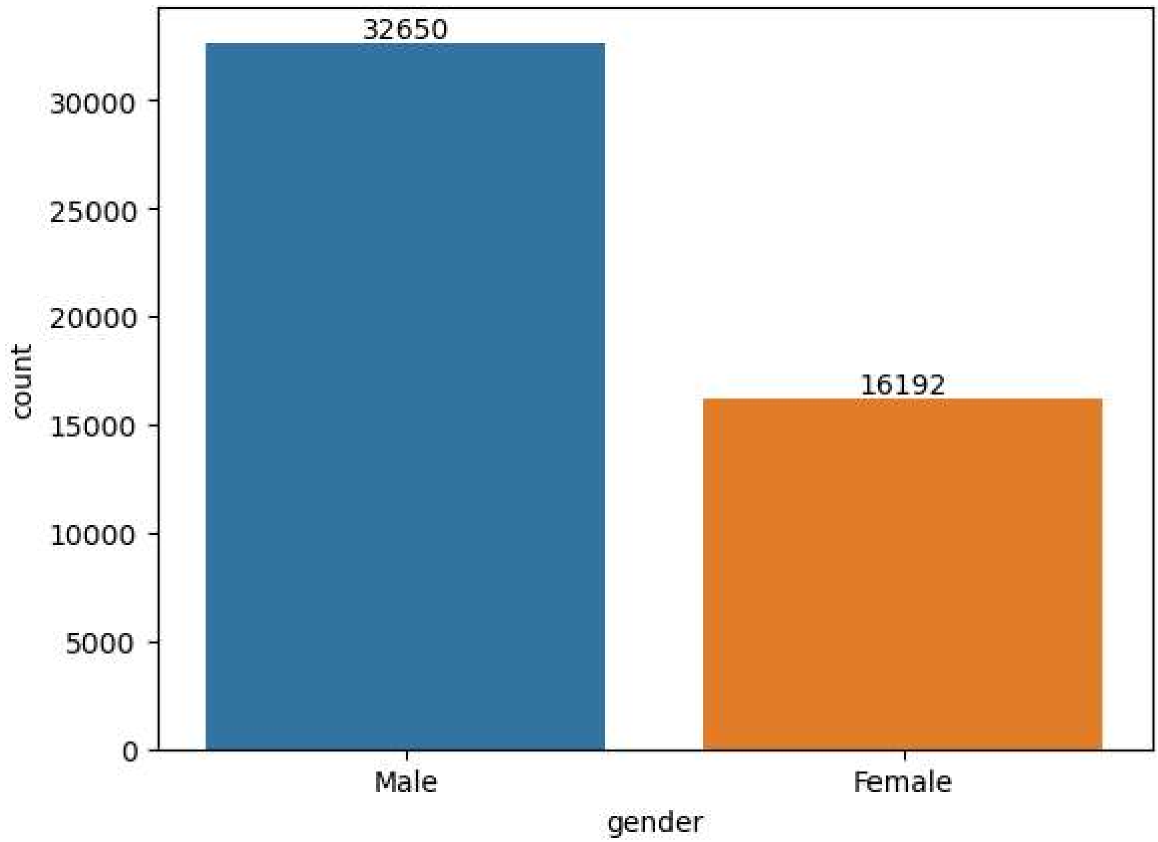
Out[8]:
```
RangeIndex(start=0, stop=48842, step=1)
```

In [9]:
```
adult.columns
```

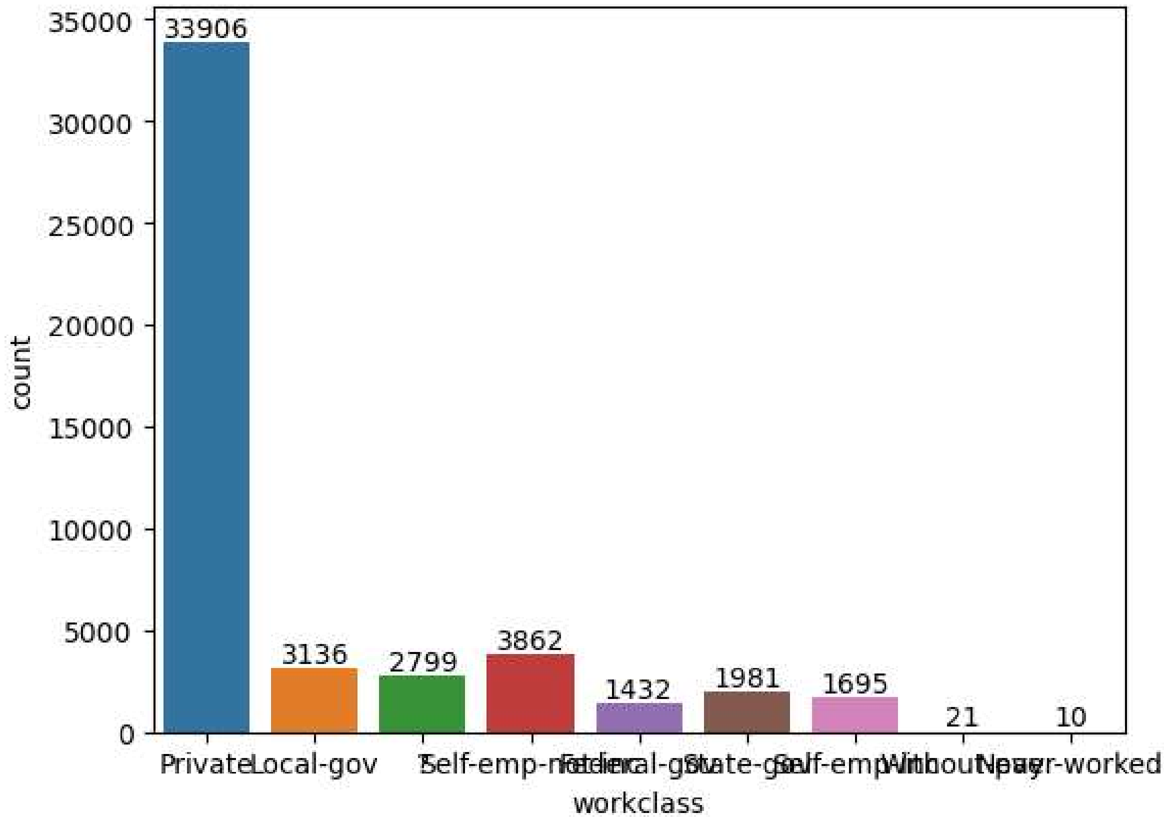
Out[9]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'educational-num',
       'marital-status', 'occupation', 'relationship', 'race', 'gender',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'income'],
      dtype='object')
```

In [10]:
```
sns.countplot(x="gender",data=adult)
```
Out[10]:
```
<AxesSubplot:xlabel='gender', ylabel='count'>
```


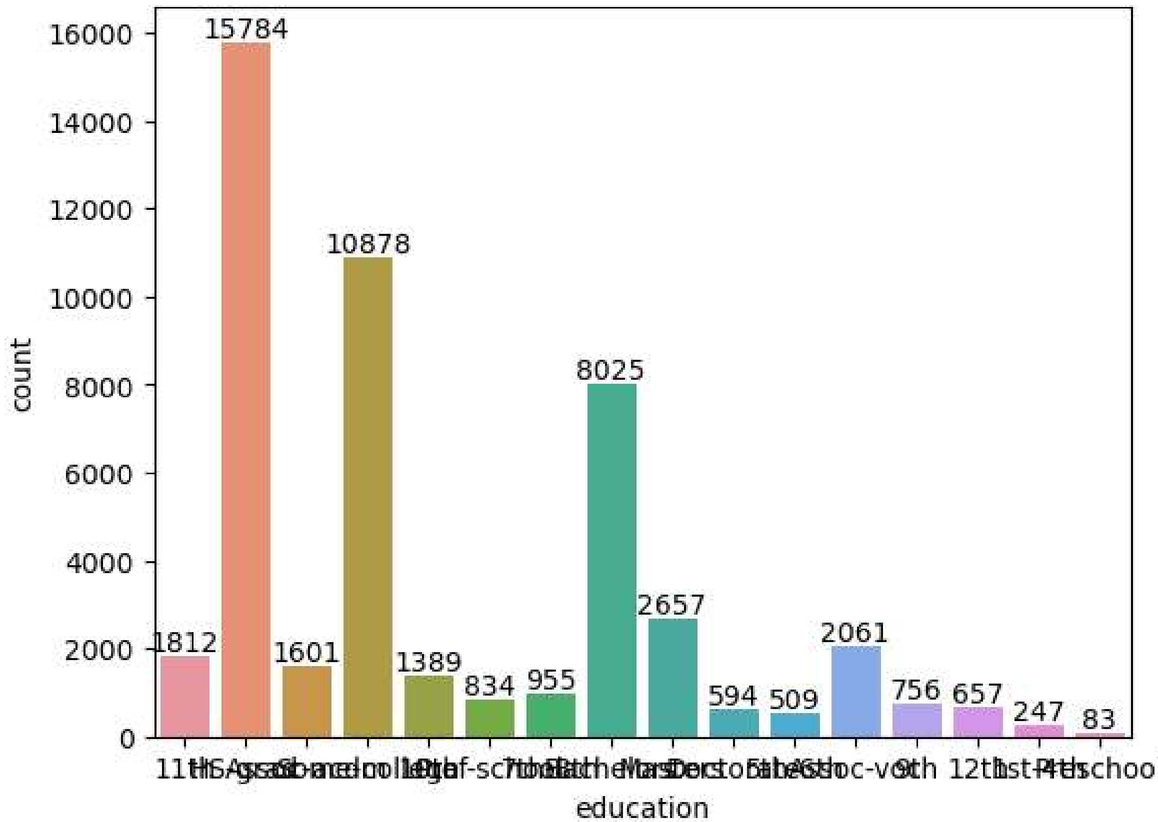
In [11]:
```
ax =sns.countplot(x="gender",data=adult)

for bars in ax.containers:
    ax.bar_label(bars)
```



In [12]:
```
ax =sns.countplot(x="workclass",data=adult)

for bars in ax.containers:
    ax.bar_label(bars)
```



In [13]:
```
ax =sns.countplot(x="education",data=adult)

for bars in ax.containers:
    ax.bar_label(bars)
```



In [14]:
```
adult
```

Out[14]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Blac
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Whit
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Whit
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Blac
4	18	?	103497	Some-college	10	Never-married	?	Own-child	Whit
...	...	...	...	...	...	...	...	...	.
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	Whit
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	Whit
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Whit

48842 rows × 15 columns

In [15]:
```
adult.isnull()
```

Out[15]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	..
48837	False	False	False	False	False	False	False	False	False
48838	False	False	False	False	False	False	False	False	False
48839	False	False	False	False	False	False	False	False	False
48840	False	False	False	False	False	False	False	False	False
48841	False	False	False	False	False	False	False	False	False

48842 rows × 15 columns

In [16]:
```
adult.isnull().sum()
```

Out[16]:
```
age                0
workclass          0
fnlwgt             0
education          0
educational-num    0
marital-status     0
occupation         0
relationship       0
race               0
gender             0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country     0
income             0
dtype: int64
```

In [17]:
```
len(adult)
```

Out[17]: 48842

In [18]:
```
adult["marital-status"].describe()
```

Out[18]:
```
count                   48842
unique                      7
top        Married-civ-spouse
freq                    22379
Name: marital-status, dtype: object
```

In [19]:
```
adult["marital-status"].unique()
```

Out[19]:
```
array(['Never-married', 'Married-civ-spouse', 'Widowed', 'Divorced',
       'Separated', 'Married-spouse-absent', 'Married-AF-spouse'],
      dtype=object)
```

In [20]:
```
adult.head()
```

Out[20]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	ge
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fe

In [21]:
```
adult
```

Out[21]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	rac
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Blac
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Whit
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Whit
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Blac
4	18	?	103497	Some-college	10	Never-married	?	Own-child	Whit
...	...	...	...	...	...	...	...	...	.
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	Whit
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Whit
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	Whit
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Whit

48842 rows × 15 columns

In [22]:
```
adult.head()
```

Out[22]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	ge
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Fe

In [23]:
```
adult=adult.drop(["education", "fnlwgt", "marital-status"],axis=1)
adult.head()
```

Out[23]:

	age	workclass	educational-num	occupation	relationship	race	gender	capital-gain	capital-loss	hours per weel
0	25	Private	7	Machine-op-inspct	Own-child	Black	Male	0	0	4(
1	38	Private	9	Farming-fishing	Husband	White	Male	0	0	5(
2	28	Local-gov	12	Protective-serv	Husband	White	Male	0	0	4(
3	44	Private	10	Machine-op-inspct	Husband	Black	Male	7688	0	4(
4	18	?	10	?	Own-child	White	Female	0	0	3(

In [24]:
```python
adult.replace('?', np.NaN,inplace = True)
adult.head()
```

Out[24]:

	age	workclass	educational-num	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-weel
0	25	Private	7	Machine-op-inspct	Own-child	Black	Male	0	0	4(
1	38	Private	9	Farming-fishing	Husband	White	Male	0	0	5(
2	28	Local-gov	12	Protective-serv	Husband	White	Male	0	0	4(
3	44	Private	10	Machine-op-inspct	Husband	Black	Male	7688	0	4(
4	18	NaN	10	NaN	Own-child	White	Female	0	0	3(

In [25]:
```python
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
```

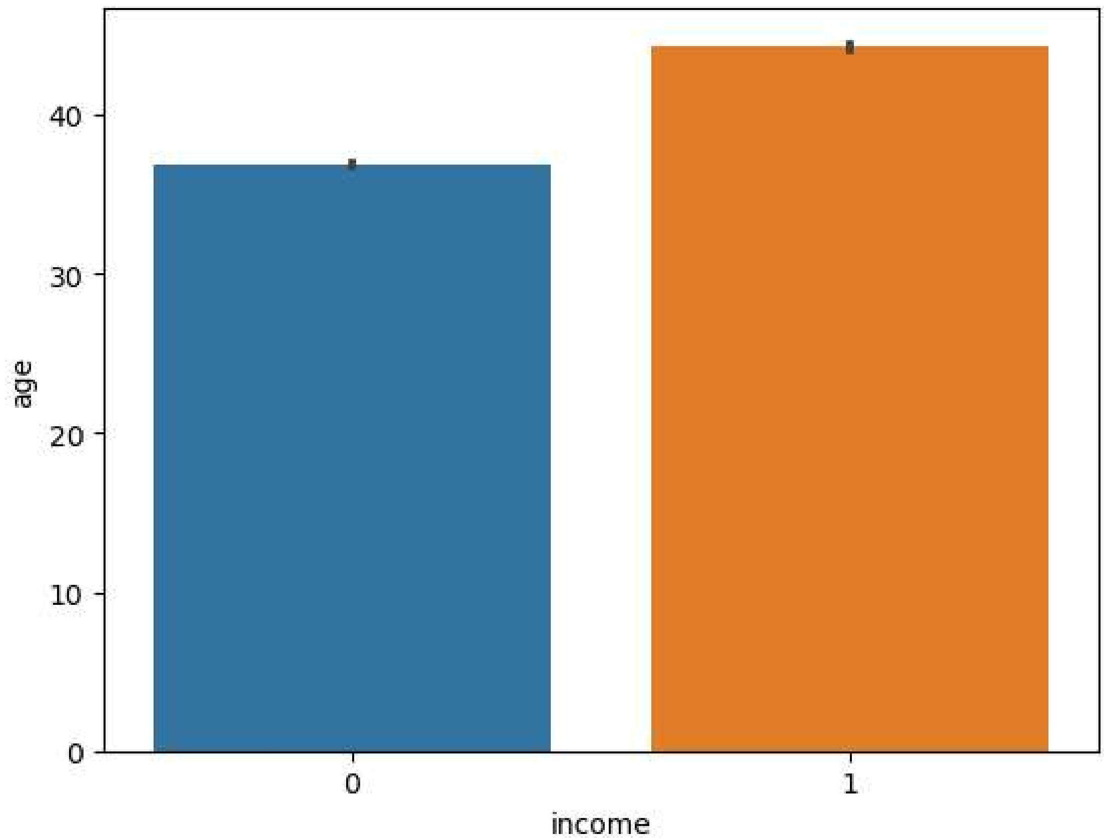
In [26]:
```python
adult["workclass"] = le.fit_transform(adult["workclass"])
adult["occupation"] = le.fit_transform(adult["occupation"])
adult["relationship"] = le.fit_transform(adult["relationship"])
adult["race"] = le.fit_transform(adult["race"])
adult["gender"] = le.fit_transform(adult["gender"])
adult["native-country"] = le.fit_transform(adult["native-country"])
adult["income"] = le.fit_transform(adult["income"])

adult.head()
```

Out[26]:

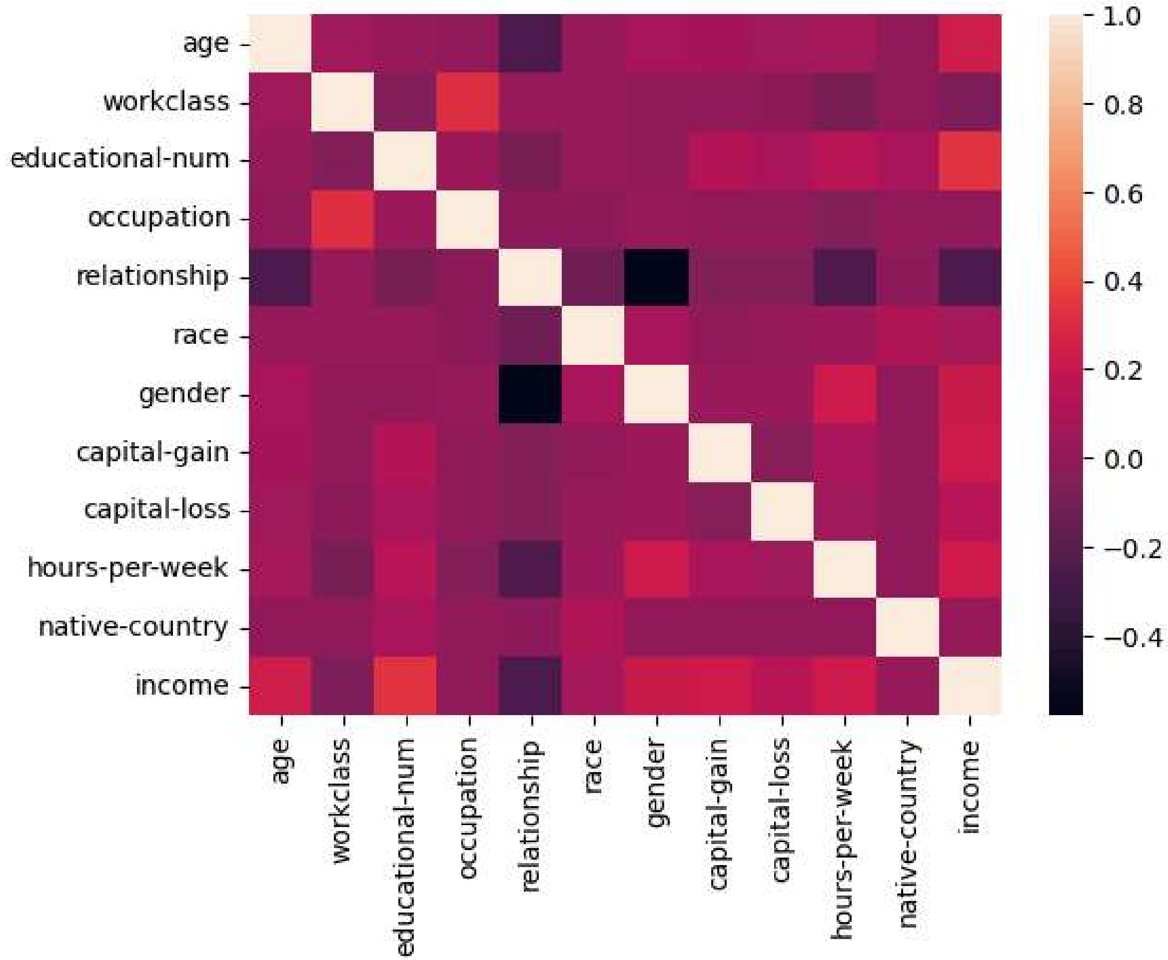
	age	workclass	educational-num	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
0	25	3	7	6	3	2	1	0	0	40
1	38	3	9	4	0	4	1	0	0	50
2	28	1	12	10	0	4	1	0	0	40
3	44	3	10	6	0	2	1	7688	0	40
4	18	8	10	14	3	4	0	0	0	30

In [27]:
```
sns.barplot(x = 'income', y='age', data=adult)
```

Out[27]:
```
<AxesSubplot:xlabel='income', ylabel='age'>
```



In [28]:
```
sns.heatmap(adult.corr())
```
Out[28]: <AxesSubplot:>



In [29]:
```
x = adult.drop(['income'],axis=1)
y = adult['income']
```

In [30]:
```
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)
```

In [31]:
```
from sklearn.ensemble import RandomForestClassifier
Clf = RandomForestClassifier()
```

In [32]:
```
Clf.fit(x_train,y_train)
```
Out[32]: RandomForestClassifier()

In [33]:
```
y_pred = Clf.predict(x_test)
y_pred
```
Out[33]: array([0, 0, 1, ..., 0, 0, 0])

In [34]:
```
!pip install scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: scikit-learn in c:\programdata\anaconda3\li
b\site-packages (1.0.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\programdata\anac
onda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: scipy>=1.1.0 in c:\programdata\anaconda3\li
b\site-packages (from scikit-learn) (1.9.1)
Requirement already satisfied: joblib>=0.11 in c:\programdata\anaconda3\li
b\site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: numpy>=1.14.6 in c:\programdata\anaconda3\l
ib\site-packages (from scikit-learn) (1.21.5)
```

In [35]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.92      0.91     11236
           1       0.71      0.63      0.67      3417

    accuracy                           0.85     14653
   macro avg       0.80      0.78      0.79     14653
weighted avg       0.85      0.85      0.85     14653
```

In [36]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[36]:
```
array([[10360,   876],
       [ 1269,  2148]], dtype=int64)
```

In [37]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[37]:
```
0.8536135944857708
```

In [38]:
```
x = adult.drop(['income'],axis=1)
y = adult['income']
```

In [39]:
```
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)
```

In [40]:
```
from sklearn.naive_bayes import GaussianNB
gb = GaussianNB()
```

In [41]:
```
gb.fit(x_train,y_train)
```

Out[41]:
```
GaussianNB()
```

In [42]:
```
y_pred = gb.predict(x_test)
y_pred
```

Out[42]:
```
array([0, 1, 1, ..., 0, 0, 0])
```

In [43]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,y_pred)
```

Out[43]:
```
array([[10613,   557],
       [ 2405,  1078]], dtype=int64)
```

In [44]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[44]:
```
0.797857094110421
```

In [45]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.95      0.88     11170
           1       0.66      0.31      0.42      3483

    accuracy                           0.80     14653
   macro avg       0.74      0.63      0.65     14653
weighted avg       0.78      0.80      0.77     14653
```

In [ ]:
```

```