In [5]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import warnings

warnings.filterwarnings('ignore')
```

In [4]:

```
train = pd.read_csv('data/4th_kaggle/train.csv')
test = pd.read_csv('data/4th_kaggle/test.csv')
sub = pd.read_csv('data/4th_kaggle/sample_submission.csv')
```

데이터 탐색

- 컬럼명 : [].columns
- 행열 : [].shape
- 정보 : [].info()
- 수치 데이터 요약정보 : [].describe()
- 결측치 : [].isnull().sum()

데이터 정보

```
    age : 나이
    workclass : 고용 형태
    fnlwgt : 사람 대표성을 나타내는 가중치 (final weight의 약자)
    education : 교육 수준 (최종 학력)
    education_num : 교육 수준 수치
    marital_status: 결혼 상태
    occupation : 업종
    relationship : 가족 관계
    race : 인종
    sex : 성별
    capital_gain : 양도 소득
    capital_loss : 양도 손실
    hours_per_week : 주당 근무 시간
    native_country : 국적
    income : 수익 (예측해야 하는 값, target variable)
```

In [6]:

```
train.columns
```

Out[6]:

```
Index(['id', '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 [7]:

```
test.columns
```

Out[7]:

```
Index(['id', 'age', 'workclass', 'fnlwgt', 'education', 'education_num',
       'marital_status', 'occupation', 'relationship', 'race', 'sex',
       'capital_gain', 'capital_loss', 'hours_per_week', 'native_country'],
      dtype='object')
```

In [8]:

```
sub.columns
```

Out[8]:

```
Index(['id', 'prediction'], dtype='object')
```

In [11]:

```
print("학습용 데이터 : ", train.shape)
print("테스트용 데이터 : ", test.shape)
```

```
학습용 데이터 :  (26049, 16)
테스트용 데이터 :  (6512, 15)
```

In [12]:

```
train.isnull().sum()
```

Out[12]:

```
id                0
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
income            0
dtype: int64
```

In [13]:

```
test.isnull().sum()
```

Out[13]:

```
id                0
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 [14]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26049 entries, 0 to 26048
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   id              26049 non-null  int64 
 1   age             26049 non-null  int64 
 2   workclass       26049 non-null  object
 3   fnlwgt          26049 non-null  int64 
 4   education       26049 non-null  object
 5   education_num   26049 non-null  int64 
 6   marital_status  26049 non-null  object
 7   occupation      26049 non-null  object
 8   relationship    26049 non-null  object
 9   race            26049 non-null  object
 10  sex             26049 non-null  object
 11  capital_gain    26049 non-null  int64 
 12  capital_loss    26049 non-null  int64 
 13  hours_per_week  26049 non-null  int64 
 14  native_country  26049 non-null  object
 15  income          26049 non-null  object
dtypes: int64(7), object(9)
memory usage: 3.2+ MB
```

In [15]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6512 entries, 0 to 6511
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   id              6512 non-null   int64 
 1   age             6512 non-null   int64 
 2   workclass       6512 non-null   object
 3   fnlwgt          6512 non-null   int64 
 4   education       6512 non-null   object
 5   education_num   6512 non-null   int64 
 6   marital_status  6512 non-null   object
 7   occupation      6512 non-null   object
 8   relationship    6512 non-null   object
 9   race            6512 non-null   object
 10  sex             6512 non-null   object
 11  capital_gain    6512 non-null   int64 
 12  capital_loss    6512 non-null   int64 
 13  hours_per_week  6512 non-null   int64 
 14  native_country  6512 non-null   object
dtypes: int64(7), object(8)
memory usage: 763.2+ KB
```

In [16]:

```
train.income.unique()
```

Out[16]:

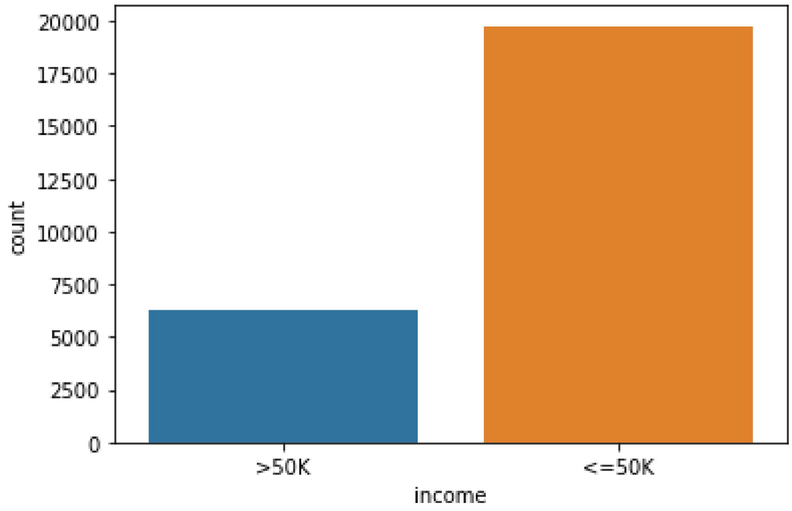
```
array(['>50K', '<=50K'], dtype=object)
```

In [19]:

```
sns.countplot(x="income", data=train)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21a3b20bb20>
```



In [31]:

```
train.loc[ train['income']=='>50K' , 'target'] = 1
train.loc[ train['income']=='<=50K' , 'target'] = 0
train['target'] = train.target.astype("int")
```

In [32]:

```
train.head()
```

Out[32]:

	id	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relations
0	0	40	Private	168538	HS-grad	9	Married-civ-spouse	Sales	Husba
1	1	17	Private	101626	9th	5	Never-married	Machine-op-inspct	Own-cl
2	2	18	Private	353358	Some-college	10	Never-married	Other-service	Own-cl
3	3	21	Private	151158	Some-college	10	Never-married	Prof-specialty	Own-cl
4	4	24	Private	122234	Some-college	10	Never-married	Adm-clerical	Not-in-far

In [23]:

```
test.head()
```

Out[23]:

on_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_
10	Never-married	Adm-clerical	Other-relative	White	Female	0	0	
9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	
10	Never-married	Handlers-cleaners	Own-child	White	Male	0	0	
11	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	
16	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	

In [24]:

```
train.columns
```

Out[24]:

```
Index(['id', 'age', 'workclass', 'fnlwgt', 'education', 'education_num',
       'marital_status', 'occupation', 'relationship', 'race', 'sex',
       'capital_gain', 'capital_loss', 'hours_per_week', 'native_country',
       'income', 'target'],
      dtype='object')
```

In [40]:

```
sel = ['id', 'age', 'fnlwgt', 'education_num', 'capital_gain', 'capital_loss', 'hours_per_week']

X = train[sel]
y = train['target']

test_X = test[sel]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,
                                                    stratify=train.target,
                                                    random_state=42)
```

In [26]:

```
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(19536, 7) (6513, 7) (19536,) (6513,)
```

로지스틱 모델

In [34]:

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

In [41]:

```
model = LogisticRegression()
model.fit(X_train, y_train)
pred = model.predict(test_X)
```

In [42]:

```
sub.columns
```

Out[42]:

```
Index(['id', 'prediction'], dtype='object')
```

In [43]:

```
print( sub.shape )
print( pred.shape )
```

```
(6512, 2)
(6512,)
```

In [45]:

```
sub['prediction'] = pred
sub.to_csv("firstSub4th.csv", index=False)
```

0.78545

In [ ]: