In [20]:

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

warnings.filterwarnings('ignore')
```

In [48]:

```
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 [49]:

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

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

In [50]:

```
y = train['income']
test['income'] = "blank"
```

In [51]:

```
all_dat = pd.concat([train, test], axis=0)
print(all_dat.shape)
```

```
(32561, 16)
```

In [52]:

```
all_dat.income.value_counts()
```

Out[52]:

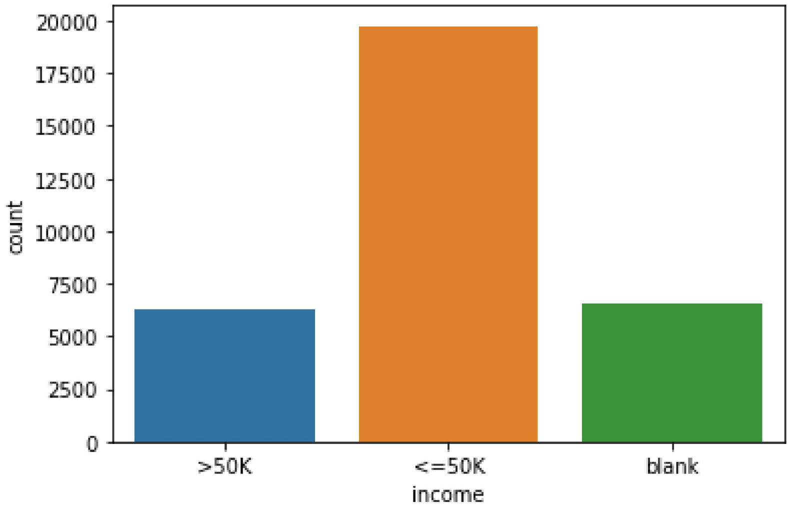
```
<=50K     19744
blank      6512
>50K       6305
Name: income, dtype: int64
```

In [53]:

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

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b3caaa8070>
```



In [54]:

```
all_dat.loc[ all_dat['income']=='>50K' , 'target'] = 1
all_dat.loc[ all_dat['income']=='<=50K' , 'target'] = 0
all_dat.loc[ all_dat['income']=='blank' , 'target'] = 999
all_dat['target'] = all_dat.target.astype("int")
```

In [55]:

```
all_dat.head()
```

Out[55]:

	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 [56]:

```
all_dat.columns
```

Out[56]:

```
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 [57]:

```
sel_cat = ['workclass', 'education', 'marital_status',
           'occupation', 'relationship', 'race',
           'sex', 'native_country' ]

X_cat = all_dat[sel_cat]
y = all_dat['target']
```

In [58]:

```
X_dummy = pd.get_dummies(X_cat)
X_dummy
```

Out[58]:

	workclass_?	workclass_Federal-gov	workclass_Local-gov	workclass_Never-worked	workclass_Private
0	0	0	0	0	1
1	0	0	0	0	1
2	0	0	0	0	1
3	0	0	0	0	1
4	0	0	0	0	1
...	...	...	...	...	...
6507	0	0	0	0	1
6508	0	0	0	0	0
6509	0	0	0	0	1
6510	0	0	0	0	1
6511	0	0	0	0	1

32561 rows × 102 columns

In [59]:

```
all_dat_n = pd.concat([all_dat, X_dummy], axis=1)
all_dat_n
```

Out[59]:

	id	age	workclass	fnlwgt	education	education_num	marital_status	occupation	rel
0	0	40	Private	168538	HS-grad	9	Married-civ-spouse	Sales	
1	1	17	Private	101626	9th	5	Never-married	Machine-op-inspct	
2	2	18	Private	353358	Some-college	10	Never-married	Other-service	
3	3	21	Private	151158	Some-college	10	Never-married	Prof-specialty	
4	4	24	Private	122234	Some-college	10	Never-married	Adm-clerical	Nc
...	...	...	...	...	...	...	...	...	
6507	6507	35	Private	61343	Bachelors	13	Married-civ-spouse	Sales	
6508	6508	41	Self-emp-inc	32185	Bachelors	13	Married-civ-spouse	Tech-support	
6509	6509	39	Private	409189	5th-6th	3	Married-civ-spouse	Other-service	
6510	6510	35	Private	180342	HS-grad	9	Married-civ-spouse	Craft-repair	
6511	6511	28	Private	156819	HS-grad	9	Divorced	Handlers-cleaners	U

32561 rows × 119 columns

In [60]:

```
train_n = all_dat_n.loc[ (all_dat_n['target']==0) | (all_dat_n['target']==1)  , : ]
test_n = all_dat_n.loc[ all_dat_n['target']==999  ,  : ]
```

In [61]:

```
print(train_n.shape, test_n.shape)
```

```
(26049, 119) (6512, 119)
```

In [62]:

```
sel_cat = ['workclass', 'education', 'marital_status',
           'occupation', 'relationship', 'race',
           'sex', 'native_country', 'income']

train_n = train_n.drop(sel_cat, axis=1)
test_n = test_n.drop(sel_cat, axis=1)

print(train_n.shape, test_n.shape)
```

```
(26049, 110) (6512, 110)
```

In [65]:

```
X = train_n.drop(['target'], axis=1)
y = train_n['target']

test_X = test_n.drop(['target'], axis=1)
```

In [66]:

```
print(X.shape, y.shape, test_X.shape)
```

```
(26049, 109) (26049,) (6512, 109)
```

In [67]:

```
# 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 [68]:

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

로지스틱 모델

In [15]:

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

In [69]:

```
model = LogisticRegression()
model.fit(X, y)
pred = model.predict(test_X)
```

In [70]:

```
sub.columns
```

Out[70]:

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

In [71]:

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

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

In [72]:

```
sub['prediction'] = pred
sub.to_csv("secondSub4th.csv", index=False)
```

In [ ]: