t2_예상3

```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("data/adult/adult.csv")
```

```
df.head(2)
```

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capit
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0

```
df.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
```

```
df.isnull().sum()
```

```
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
```

```python
# 시험환경 세팅 (코드 변경 X)
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

def exam_data_load(df, target, id_name="", null_name=""):
    if id_name == "":
        df = df.reset_index().rename(columns={"index": "id"})
        id_name = 'id'
    else:
        id_name = id_name

    if null_name != "":
        df[df == null_name] = np.nan

    X_train, X_test = train_test_split(df, test_size=0.2, random_state=2021)

    y_train = X_train[[id_name, target]]
    X_train = X_train.drop(columns=[target])


    y_test = X_test[[id_name, target]]
    X_test = X_test.drop(columns=[target])
    return X_train, X_test, y_train, y_test
```

```python
df=pd.read_csv("data/adult/adult.csv")
X_train, X_test, y_train, y_test = exam_data_load(df, target='income', null_name='?')

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((26048, 15), (6513, 15), (26048, 2), (6513, 2))
```

```python
X_train.isnull().sum()
```

```
id                   0
age                  0
workclass         1456
fnlwgt               0
education            0
education.num        0
marital.status       0
occupation        1463
relationship         0
race                 0
sex                  0
capital.gain         0
capital.loss        0
hours.per.week      0
native.country    461
dtype: int64
```

```
X_test.isnull().sum()
```

```
id                  0
age                 0
workclass         380
fnlwgt              0
education           0
education.num       0
marital.status      0
occupation        380
relationship        0
race                0
sex                 0
capital.gain        0
capital.loss        0
hours.per.week      0
native.country    122
dtype: int64
```

```
X_train.head(2)
```

	id	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex
21851	21851	36	Private	241998	Bachelors	13	Married-civ-spouse	Craft-repair	Husband	White	Male
7632	7632	53	Private	103950	Masters	14	Divorced	Prof-specialty	Not-in-family	White	Fema

```
y_train.head()
```

	id	income
21851	21851	>50K
7632	7632	<=50K
27878	27878	<=50K
14121	14121	<=50K
32345	32345	<=50K

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

```
native.country
United-States                 23381
Mexico                          516
Philippines                     158
Germany                         108
Canada                           88
Puerto-Rico                      87
El-Salvador                      76
```

```
India                            73
Cuba                             73
England                          69
Italy                            63
South                            62
Jamaica                          59
Vietnam                          57
China                            57
Guatemala                        54
Dominican-Republic               51
Japan                            49
Poland                           47
Columbia                         44
Taiwan                           37
Haiti                            37
Iran                             34
Portugal                         32
Peru                             29
Nicaragua                        27
Ecuador                          25
Greece                           24
France                           23
Ireland                          19
Cambodia                         18
Hong                             17
Trinadad&Tobago                  17
Thailand                         16
Laos                             13
Outlying-US(Guam-USVI-etc)       11
Yugoslavia                       11
Honduras                          9
Hungary                           8
Scotland                          7
Holand-Netherlands                1
Name: count, dtype: int64
```

```
X_train['native.country'].isnull().sum()
```

```
461
```

```
# 결측치 카테고리형 빈도수 높은걸로 채우기
cols= ['native.country','workclass','occupation']
for col in cols:
    X_train[col]=X_train[col].fillna("UnKnown")
    X_test[col]=X_test[col].fillna("UnKnown")
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
```

```
X_train.pop('id')
X_test.pop('id')
```

```
20901    20901
14170    14170
1776      1776
30428    30428
8602      8602
         ...
31222    31222
10861    10861
8929      8929
2066      2066
25782    25782
Name: id, Length: 6513, dtype: int64
```

```
# X_train.info()
```

```
X_train= pd.get_dummies(X_train)
X_test=pd.get_dummies(X_test)
```

```
X_train.columns.equals(X_test.columns)
```

```
False
```

```
X_train, X_test= X_train.align(X_test,join='left',axis=1,fill_value=0)
```

```
X_train.columns.equals(X_test.columns)
```

```
True
```

```
X_train.shape,X_test.shape
```

```
((26048, 108), (6513, 108))
```

```
rf=RandomForestClassifier(random_state=42,n_estimators=500)
```

```
rf.fit(X_train,y_train['income'])
```

```
RandomForestClassifier(n_estimators=500, random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
pred=rf.predict(X_test)
```

```
score= accuracy_score(y_test['income'],pred)
```

```
score
```

```
0.8545984953170582
```

```
pred
```

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