Chapter 8 - Ex2: Adult Dataset

- Adult Dataset được cung cấp bởi UCI (University of California, Irvine) được sử dụng để phát triển mô hình dự đoán Predictive Model Development.
- Bộ dữ liệu adult.data và adult.test chứa 48.842 mẫu và có 14 attributes/features. Dữ liệu này được dùng để xây dựng model dự đoán và kiểm tra một mẫu có thu nhập >50K USD hay không.

Attribute Information:

- age: continuous.
- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt: continuous.
- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- education-num: continuous.
- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex: Female, Male.
- capital-gain: continuous.
- capital-loss: continuous.
- hours-per-week: continuous.
- native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- Class: >50K, <=50K.

Yêu cầu:

- Đọc dữ liệu adult.data, tiền xử lý dữ liệu.
- Xem xét tính cân bằng giữa hai loại mẫu. Trực quan hóa. Nhận xét.
- Nếu 2 loại mẫu này không cân bằng, hãy chọn một phương pháp cân bằng dữ liệu và thực hiện. Trực quan hóa kết quả.

- Tham khảo: link (https://towardsdatascience.com/under-sampling-a-performance-booster-on-imbalanced-data-a79ff1559fab)

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

In [2]:
```
# Đọc dữ liệu, kiểm tra sơ bộ bau đầu, trực quan hóa, tiền xử lý dữ liệu
adult_train = pd.read_csv("adult/adult.data", header=None)
```

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

Out[3]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40

In [4]:
```
adult_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
0     32561 non-null int64
1     32561 non-null object
2     32561 non-null int64
3     32561 non-null object
4     32561 non-null int64
5     32561 non-null object
6     32561 non-null object
7     32561 non-null object
8     32561 non-null object
9     32561 non-null object
10    32561 non-null int64
11    32561 non-null int64
12    32561 non-null int64
13    32561 non-null object
14    32561 non-null object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [5]:
```
adult_train.to_csv("aldult_data.csv")
```

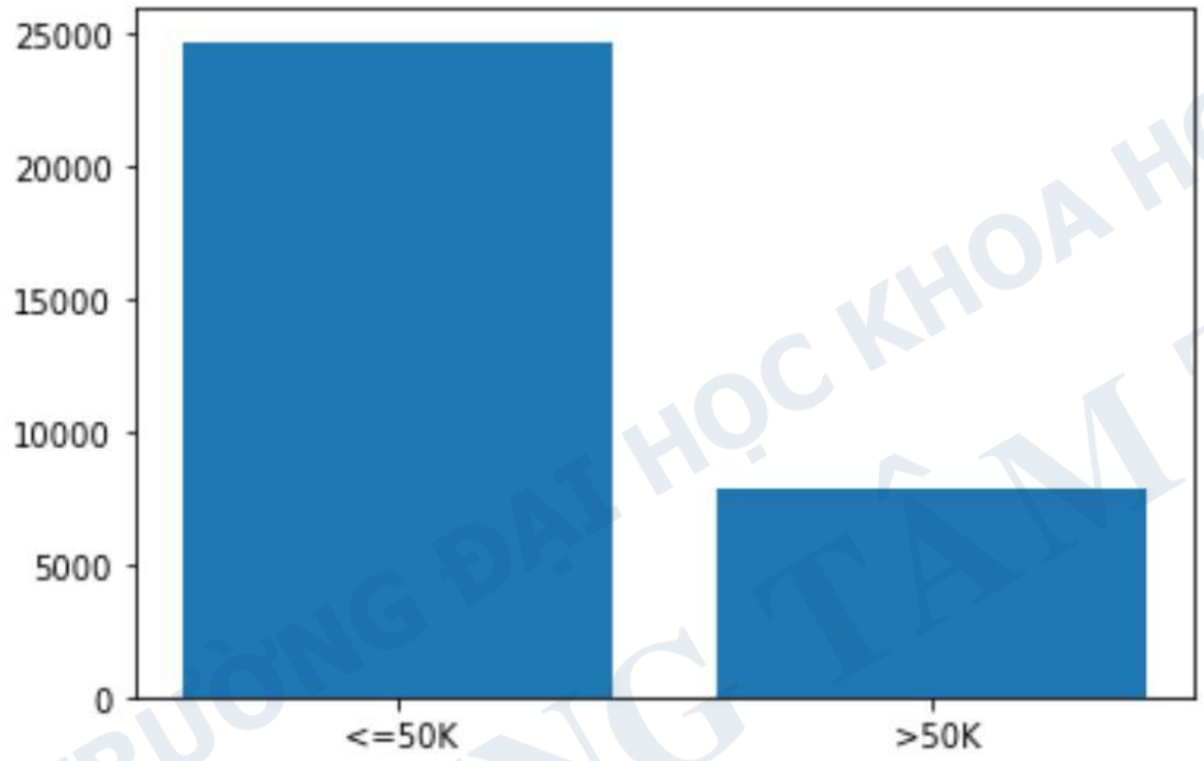
In [6]:
```
# Không có dữ liệu null
```

In [7]:
```
# Đếm theo loại: hiếm, phổ biến
occ = adult_train[14].value_counts()
occ
```

Out[7]:
```
 <=50K    24720
 >50K      7841
Name: 14, dtype: int64
```

In [8]:
```
plt.bar(occ.index.values, occ.values)
```

Out[8]:
```
<BarContainer object of 2 artists>
```



- Chuyển dữ liệu phân loại thành dạng numeric dùng Label encoder và dummy encoder

In [9]:
```
y_train = adult_train[14]
X_train = adult_train.drop([14], axis=1)
```

In [10]:
```
X_train.head(2)
```

Out[10]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	Unite Sta
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	Unite Sta

In [11]:
```
y_train[:2]
```

Out[11]:
```
0     <=50K
1     <=50K
Name: 14, dtype: object
```

In [12]:
```
from sklearn.preprocessing import LabelEncoder
```

In [13]:
```
label_encoder = LabelEncoder()
y_train_l = label_encoder.fit_transform(y_train)
```

In [14]:
```
y_train_l[:2]
```

Out[14]:
```
array([0, 0])
```

In [15]:
```
# Categorical boolean mask
categorical_feature_mask = X_train.dtypes==object
# filter categorical columns using mask and turn it into a list
categorical_cols = X_train.columns[categorical_feature_mask].tolist()
categorical_cols
```

Out[15]:
```
[1, 3, 5, 6, 7, 8, 9, 13]
```

In [16]:
```
X_train_d = pd.get_dummies(data=X_train,
                           columns=categorical_cols,
                           drop_first=True)
```

In [17]:
```
X_train_d.head(2)
```

Out[17]:

	0	2	4	10	11	12	1_ Federal-gov	1_ Local-gov	1_ Never-worked	1_ Private	...	13_ Portugal	13_ Puerto-Rico	13_ Scotland
0	39	77516	13	2174	0	40	0	0	0	0	...	0	0	0
1	50	83311	13	0	0	13	0	0	0	0	...	0	0	0

2 rows × 100 columns

In [18]:
```
from collections import Counter
sorted(Counter(y_train_l).items())
```

Out[18]:
```
[(0, 24720), (1, 7841)]
```

In [19]:
```
# Vì lượng dữ liệu class 1 tương đối nhiều => do đó ta sẽ áp dụng Udersampling
# để giảm số mẫu của nhóm <=50k bằng với nhóm >50k
```

In [20]:
```
from sklearn.utils import resample
```

In [21]:
```
# có thể dùng cách resample
```

In [22]:
```
data_train = X_train_d
data_train[14] = y_train_l
```

In [23]:
```
data_0 = data_train[data_train[14]==0]
data_1 = data_train[data_train[14]==1]
```

In [24]:
```
display(data_0.shape, data_1.shape)
```

```
(24720, 101)

(7841, 101)
```

In [25]:
```
from sklearn.utils import resample
```

In [26]:
```
data_0_resample = resample(data_0,
                           replace = False, # sample without replacement
                           n_samples = data_1.shape[0], # match minority n
                           random_state = 27) # reproducible results
```

In [27]:
```
downsampled = pd.concat([data_0_resample, data_1])
downsampled.head()
```

Out[27]:

	0	2	4	10	11	12	1_ Federal-gov	1_ Local-gov	1_ Never-worked	1_ Private	...	13_ Puerto-Rico	13_ Scotland	13_ South
31749	22	199426	10	0	0	17	0	0	0	1	...	0	0	0
24093	31	91964	13	0	0	40	0	0	0	1	...	0	0	0
21539	37	60313	9	0	0	40	0	0	0	1	...	0	0	0
24582	30	85708	9	0	0	40	0	0	0	1	...	0	0	0
622	65	109351	5	0	0	24	0	0	0	1	...	0	0	0

5 rows × 101 columns

In [28]:
```
display(data_0_resample.shape, data_1.shape)
```

```
(7841, 101)

(7841, 101)
```