Chapter 13 - Exercise 3: Adult transactions

- Cơ sở dữ liệu “Adult” được Ronny Kohavi và Barry Becker (Data Mining and Visualization, Silicon Graphics) trích xuất từ cơ sở dữ liệu của cục điều tra dân số tại http://www.census.gov/ (http://www.census.gov/) vào năm 1994. Ban đầu nó được sử dụng để dự đoán liệu income có vượt quá 50 nghìn USD/năm hay không dựa trên dữ liệu điều tra dân số. Sau đó, CSDL đã được thu thập thêm thuộc tính income với các level small và large (> 50K).
- Tiếp theo, bộ dữ liệu được dùng để tạo ra dữ liệu cho việc association mining (Xem thông tin chi tiết tại: https://rdrr.io/cran/arules/man/Adult.html (https://rdrr.io/cran/arules/man/Adult.html)). Và dữ liệu lúc này được lưu trong tập tin Adult_transactions.csv.

Yêu cầu: Áp dụng thuật toán Apriori để tính toán mức độ kết hợp giữa các item

- Chuẩn hóa dữ liệu
- Áp dụng Apriori, Tìm kết quả
- Tìm kiếm thông tin từ kết quả: trong thông tin kết quả có 'hours-per-week=Full-time' không? Nếu có thì 'hours-per-week=Full-time' kết hợp với item nào?"
- Trực quan hóa dữ liệu
- Cho biết 10 mục xuất hiện nhiều nhất. Vẽ biểu đồ.

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:
```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter13_Apriori/
```

In [3]:
```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
```

In [4]:
```
data = pd.read_csv("Adult_transactions.csv", index_col=0)
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48842 entries, 1 to 48842
Columns: 115 entries, age=Young to income=large
dtypes: bool(115)
memory usage: 5.7 MB
```

In [6]:
```
data.head()
```

Out[6]:

	age=Young	age=Middle-aged	age=Senior	age=Old	workclass=Federal-gov	workclass=Local-gov	workclass=
1	False	True	False	False	False	False	
2	False	False	True	False	False	False	
3	False	True	False	False	False	False	
4	False	False	True	False	False	False	
5	False	True	False	False	False	False	

5 rows × 115 columns

In [7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48842 entries, 1 to 48842
Columns: 115 entries, age=Young to income=large
dtypes: bool(115)
memory usage: 5.7 MB
```

In [8]:
```
# df.isnull().any()
```

In [9]:
```
frequent_itemsets = apriori(data, min_support=0.2, use_colnames=True)
frequent_itemsets.head(3)
```

Out[9]:

	support	itemsets
0	0.505119	(age=Middle-aged)
1	0.260862	(age=Senior)
2	0.694198	(workclass=Private)

In [10]:
```
frequent_itemsets.tail(3)
```

Out[10]:

	support	itemsets
614	0.212440	(native-country=United-States, hours-per-week=...
615	0.202592	(workclass=Private, native-country=United-Stat...
616	0.274456	(marital-status=Married-civ-spouse, native-cou...

In [11]:
```
from mlxtend.frequent_patterns import association_rules
association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

Out[11]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	
0	(age=Middle-aged)	(workclass=Private)	0.505119	0.694198	0.365669	0.723927	1.04
1	(age=Middle-aged)	(race=White)	0.505119	0.855043	0.425351	0.842082	0.98
2	(age=Middle-aged)	(capital-gain=None)	0.505119	0.917387	0.463208	0.917028	0.99
3	(age=Middle-aged)	(capital-loss=None)	0.505119	0.953278	0.480079	0.950428	0.99
4	(age=Middle-aged)	(native-country=United-States)	0.505119	0.897424	0.448876	0.888655	0.99
...	...	...	...	...	...	...	
3206	(race=White, relationship=Husband, capital-gai...	(marital-status=Married-civ-spouse, sex=Male, ...	0.321260	0.342103	0.274456	0.854311	2.49
3207	(native-country=United-States, relationship=Hu...	(marital-status=Married-civ-spouse, capital-lo...	0.363069	0.300295	0.274456	0.755935	2.51
3208	(relationship=Husband, capital-loss=None)	(marital-status=Married-civ-spouse, native-cou...	0.377892	0.298862	0.274456	0.726283	2.43
3209	(relationship=Husband, capital-gain=None)	(marital-status=Married-civ-spouse, native-cou...	0.355227	0.318230	0.274456	0.772622	2.42
3210	(race=White, relationship=Husband)	(marital-status=Married-civ-spouse, native-cou...	0.365628	0.297838	0.274456	0.750644	2.52

3211 rows × 9 columns

In [12]:
```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.4)
rules
```

Out[12]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	
0	(marital-status=Married-civ-spouse)	(relationship=Husband)	0.458192	0.403669	0.403423	0.880468	2
1	(relationship=Husband)	(marital-status=Married-civ-spouse)	0.403669	0.458192	0.403423	0.999391	2
2	(sex=Male)	(relationship=Husband)	0.668482	0.403669	0.403648	0.603828	1
3	(relationship=Husband)	(sex=Male)	0.403669	0.668482	0.403648	0.999949	1
4	(marital-status=Married-civ-spouse, age=Middle...	(relationship=Husband)	0.254637	0.403669	0.221244	0.868859	2
...	...	...	...	...	...	...	
1715	(relationship=Husband, capital-gain=None)	(marital-status=Married-civ-spouse, native-cou...	0.355227	0.318230	0.274456	0.772622	2
1716	(race=White, relationship=Husband)	(marital-status=Married-civ-spouse, native-cou...	0.365628	0.297838	0.274456	0.750644	2
1717	(marital-status=Married-civ-spouse)	(native-country=United-States, capital-loss=No...	0.458192	0.274641	0.274456	0.598999	2
1718	(sex=Male)	(marital-status=Married-civ-spouse, native-cou...	0.668482	0.274477	0.274456	0.410567	1
1719	(relationship=Husband)	(marital-status=Married-civ-spouse, native-cou...	0.403669	0.276524	0.274456	0.679905	2

1720 rows × 9 columns

In [13]:

```python
print(rules.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1720 entries, 0 to 1719
Data columns (total 9 columns):
antecedents           1720 non-null object
consequents           1720 non-null object
antecedent support    1720 non-null float64
consequent support    1720 non-null float64
support               1720 non-null float64
confidence            1720 non-null float64
lift                  1720 non-null float64
leverage              1720 non-null float64
conviction            1720 non-null float64
dtypes: float64(7), object(2)
memory usage: 121.1+ KB
None
```

In [14]:

```python
# "Có relationship=hours-per-week=Full-time không? nó kết hợp với item nào?"
for row in rules.iterrows():
    if "hours-per-week=Full-time" in row[1][0]:
        print(row)
```

```
(38, antecedents           (marital-status=Married-civ-spouse, hours-per-...
consequents                                        (relationship=Husband)
antecedent support                                               0.249908
consequent support                                               0.403669
support                                                          0.214774
confidence                                                       0.859413
lift                                                              2.12901
leverage                                                         0.113894
conviction                                                        4.24173
Name: 38, dtype: object)
(39, antecedents           (relationship=Husband, hours-per-week=Full-time)
consequents                              (marital-status=Married-civ-spouse)
antecedent support                                              0.214877
consequent support                                              0.458192
support                                                         0.214774
confidence                                                      0.999524
lift                                                             2.18145
leverage                                                         0.11632
conviction                                                       1137.26
Name: 39, dtype: object)
```

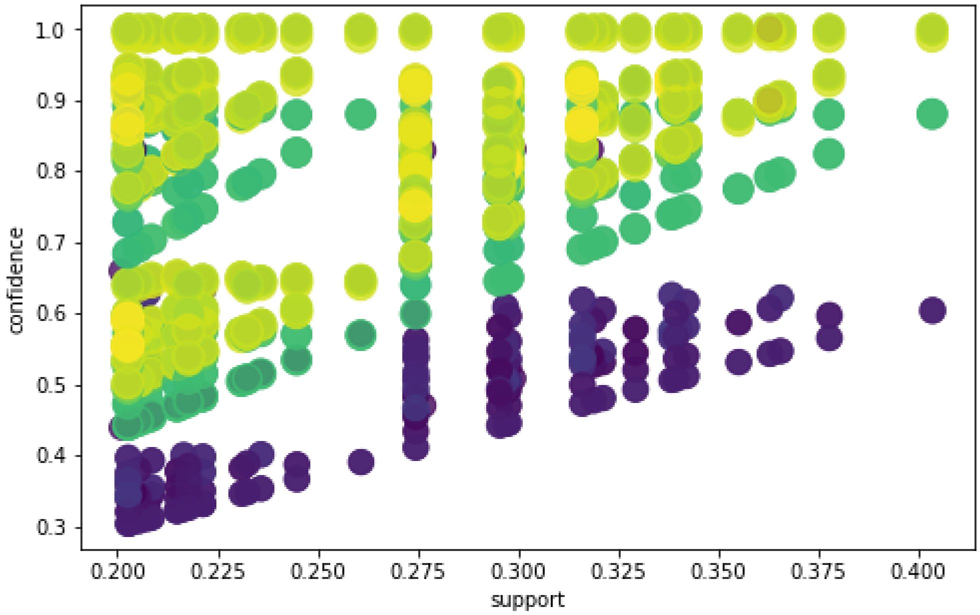
In [15]:

```python
support=rules['support'].values
confidence=rules['confidence'].values
lift = rules['lift'].values
```

In [16]:

```python
import matplotlib.pyplot as plt
```

In [17]:
```
plt.figure(figsize=(8,5))
plt.scatter(support, confidence, s= lift*100 ,alpha=0.8, c = lift)
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



In [18]:
```
result = data.apply(pd.value_counts).fillna(0)
result
```

Out[18]:

	age=Young	age=Middle-aged	age=Senior	age=Old	workclass=Federal-gov	workclass=Local-gov	workcl
False	39215	24171	36101	47039	47410	45706	
True	9627	24671	12741	1803	1432	3136	

2 rows × 115 columns

In [19]:
```
df_true = result.iloc[1,:]
df_true[:10]
```

Out[19]:
```
age=Young                        9627
age=Middle-aged                 24671
age=Senior                      12741
age=Old                          1803
workclass=Federal-gov            1432
workclass=Local-gov              3136
workclass=Never-worked             10
workclass=Private               33906
workclass=Self-emp-inc           1695
workclass=Self-emp-not-inc       3862
Name: True, dtype: int64
```

In [20]:
```
x = df_true.sort_values(ascending=False)
```

In [21]:
```
ten_ = x[:10]
ten_
```

Out[21]:
```
capital-loss=None                        46560
capital-gain=None                        44807
native-country=United-States             43832
race=White                               41762
workclass=Private                        33906
sex=Male                                 32650
hours-per-week=Full-time                 28577
income=small                             24720
age=Middle-aged                          24671
marital-status=Married-civ-spouse        22379
Name: True, dtype: int64
```

In [22]:
```
import numpy as np
pos = np.arange(len(ten_.values))
```

In [23]:
```
plt.figure(figsize=(8,5))
plt.bar(pos, ten_.values, align='center')
plt.xticks(pos, ten_.keys(), rotation='vertical')
plt.ylabel('Times')
plt.xlabel('Attributes')
plt.title('Ten Attributes')
plt.show()
```