[Thực hành] Thực hành trên bộ dữ liệu Online Retail

Mục tiêu: Thực hành xử lý dữ liệu trên bộ dữ liệu Online Retail bằng các kỹ thuật đã học.

Thông tin bộ dữ liệu:

- InvoiceNo: Số hóa đơn
- StockCode: mã hàng
- Description: Mô tả hàng
- Quantity: Số lượng
- InvoiceDate: Ngày bán
- UnitPrice: Đơn giá
- CustomerID: Mã khách
- Country: Nước sản xuất

Khai báo thư viện cần dùng

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, RobustScaler, StandardScaler
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, OrdinalEncoder
```

```
# đọc dữ liệu
df = pd.read_csv("OnlineRetail.csv", encoding = "ISO-8859-1")

# in ra kich thuoc du lieu
df.shape
```

```
(250981, 8)
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cour
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6.0	12/1/2010 8:26	2.55	17850.0	Ur King
1	536365	71053	WHITE METAL LANTERN	6.0	12/1/2010 8:26	3.39	17850.0	Ur King
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8.0	12/1/2010 8:26	2.75	17850.0	Ur King
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6.0	12/1/2010 8:26	3.39	17850.0	Ur King
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6.0	12/1/2010 8:26	3.39	17850.0	Ur King

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12462 entries, 0 to 12461
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   InvoiceNo    12462 non-null  object
```

```
 1   StockCode    12462 non-null  object
 2   Description  12417 non-null  object
 3   Quantity     12461 non-null  float64
 4   InvoiceDate  12461 non-null  object
 5   UnitPrice    12461 non-null  float64
 6   CustomerID   8956 non-null   float64
 7   Country      12461 non-null  object
dtypes: float64(3), object(5)
memory usage: 779.0+ KB
```

Kiểm tra dữ liệu bị khuyết

```
# kiểm tra dữ liệu bị khuyết
df.isna()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
75101	False	False	False	False	False	False	True	False
75102	False	False	False	False	False	False	True	False
75103	False	False	False	False	False	False	True	False
75104	False	False	False	False	False	False	True	False
75105	False	False	False	True	True	True	True	True

75106 rows × 8 columns

```
# kiểm tra dữ liệu không bị khuyết
df['CustomerID'].notna()
```

```
0         True
1         True
2         True
3         True
4         True
         ...
75101    False
75102    False
75103    False
75104    False
75105    False
Name: CustomerID, Length: 75106, dtype: bool
```

```
# in những dòng ngoại lai Quantity < 0
df[df['Quantity'] < 0]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cour
141	C536379	D	Discount	-1.0	12/1/2010 9:41	27.50	14527.0	Ur King
154	C536383	35004C	SET OF 3 COLOURED FLYING DUCKS	-1.0	12/1/2010 9:49	4.65	15311.0	Ur King
235	C536391	22556	PLASTERS IN TIN CIRCUS PARADE	-12.0	12/1/2010 10:24	1.65	17548.0	Ur King
236	C536391	21984	PACK OF 12 PINK PAISLEY TISSUES	-24.0	12/1/2010 10:24	0.29	17548.0	Ur King
237	C536391	21983	PACK OF 12 BLUE PAISLEY TISSUES	-24.0	12/1/2010 10:24	0.29	17548.0	Ur King
...	...	...	...	...	...	...	...	
250430	C559030	21794	CLASSIC FRENCH STYLE BASKET NATURAL	-2.0	7/5/2011 14:51	3.95	16571.0	Ur King
250431	C559030	20984	12 PENCILS TALL TUBE POSY	-24.0	7/5/2011 14:51	0.29	16571.0	Ur King
250468	C559031	82580	BATHROOM METAL SIGN	-250.0	7/5/2011 14:53	0.42	15985.0	Ur King
250469	C559033	22760	TRAY, BREAKFAST IN BED	-1.0	7/5/2011 14:54	10.95	17389.0	Ur King

```
#Xóa bỏ dòng ngoại lai của Quantity
df = df[df['Quantity'] >= 0]
```

```
5347 rows × 8 columns
```

```
# xóa những dòng chứa giá trị bị khuyết
df1 = df.dropna()
```

```
df1.shape
```

```
(46828, 8)
```

```
# xóa những dòng chứa toàn giá trị khuyết
df2 = df.dropna(how='all')
```

```
df2.shape
```

```
(75106, 8)
```

```
# giữ những dòng có ít nhất 7 giá trị không bị khuyết
df3 = df.dropna(thresh=7)
```

```
df3.shape
```

```
(74902, 8)
```

```
# xóa những hàng mà có giá trị bị khuyết trên cột CustomerID
df4 = df.dropna(subset=["CustomerID"])
```

```
df4.shape
```

```
(46828, 8)
```

Thay thế dữ liệu bị khuyết

```
# thay thế những giá trị bị khuyết trên cột CustomerID bằng giá trị -1
df5 = df
df5['CustomerID'] = df['CustomerID'].fillna(-1)
```

```
# hiển thị những dòng có CustomerID = -1 vừa được thay thế
df5[df5['CustomerID'] == -1]
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cour
622	536414	22139	NaN	56.0	12/1/2010 11:52	0.00	-1.0	Ur King
1443	536544	21773	DECORATIVE ROSE BATHROOM BOTTLE	1.0	12/1/2010 14:32	2.51	-1.0	Ur King
1444	536544	21774	DECORATIVE CATS BATHROOM BOTTLE	2.0	12/1/2010 14:32	2.51	-1.0	Ur King
1445	536544	21786	POLKADOT RAIN HAT	4.0	12/1/2010 14:32	0.85	-1.0	Ur King
1446	536544	21787	RAIN PONCHO RETROSPOT	2.0	12/1/2010 14:32	1.66	-1.0	Ur King
...	...	...	...	...	...	...	...	
75101	542541	22124	SET OF 2 TEA TOWELS PING MICROWAVE	1.0	1/28/2011 14:25	2.46	-1.0	Ur King
75102	542541	22131	FOOD CONTAINER SET 3 LOVE HEART	1.0	1/28/2011 14:25	4.13	-1.0	Ur King
75103	542541	22135	MINI LADLE LOVE HEART PINK	1.0	1/28/2011 14:25	0.83	-1.0	Ur King
75104	542541	22148	EASTER CRAFT 4 CHICKS	3.0	1/28/2011 14:25	4.13	-1.0	Ur King
75105	542541	22149	FELTCRAFT 6 FLOWER FRIENDS	NaN	NaN	NaN	-1.0	

28278 rows × 8 columns

```
# thay thế các giá trị bị khuyết ở cột Country bằng giá trị trước nó
df5['Country'] = df['Country'].fillna(method='ffill')
```

```
df5
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Cour
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6.0	12/1/2010 8:26	2.55	17850.0	Ur King
1	536365	71053	WHITE METAL LANTERN	6.0	12/1/2010 8:26	3.39	17850.0	Ur King
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8.0	12/1/2010 8:26	2.75	17850.0	Ur King
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6.0	12/1/2010 8:26	3.39	17850.0	Ur King
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6.0	12/1/2010 8:26	3.39	17850.0	Ur King
...	...	...	...	...	...	...	...	
75101	542541	22124	SET OF 2 TEA TOWELS PING MICROWAVE	1.0	1/28/2011 14:25	2.46	-1.0	Ur King

Xử lý dữ liệu ngoại lai

```
sns.boxplot(x=df1['Quantity'])  # vẽ box plot cho dữ liệu ở cột Quantity
```

Xóa dữ liệu ngoại lai bằng IQR score

```
Q1 = df1['Quantity'].quantile(0.25)
Q3 = df1['Quantity'].quantile(0.75)
IQR = Q3 - Q1


# xác định phần tử không phải ngoại lai
df6 = df1
df6['outlier'] = ~((df1['Quantity'] < (Q1 - 1.5*IQR)) | (df1['Quantity'] > (Q3 + 1.5*IQR)))
```

```
# xóa phần tử ngoại lai
df6 = df6[df6['outlier'] == True]


sns.boxplot(x=df6['Quantity'])  # vẽ box plot cho dữ liệu ở cột Quantity
```

Chuẩn hóa dữ liệu bằng

```
# vẽ biểu đồ hộp cho cột Quantity
sns.boxplot(x=df1['Quantity'])



# mô tả dữ liệu
df1['Quantity'].describe()
```

```
count    74902.000000
mean         0.768581
std         55.283898
min     -10602.571429
25%         -0.285714
50%          0.000000
75%          0.714286
max      10601.714286
Name: Quantity, dtype: float64
```

```
# chuẩn hóa dữ liệu với minmax scaling
scaler = MinMaxScaler()

# Chuẩn hóa dữ liệu trong df với MinMaxScaler ở 2 cột Quantity và UnitPrice
df_s = scaler.fit_transform(df1[['Quantity']])


# mô tả dữ liệu sau chuẩn hóa
```

```
pd.DataFrame(df_s).describe()
```

	0
count	74902.000000
mean	0.500056
std	0.002607
min	0.000000
25%	0.500007
50%	0.500020
75%	0.500054
max	1.000000

```
# vẽ lại biểu đồ hộp
sns.boxplot(x=df_s)
```

```
# chuẩn hóa dữ liệu với robust scaling
scaler = RobustScaler()

# Chuẩn hóa dữ liệu trong df với RobustScaler ở 2 cột Quantity và UnitPrice
df_s = scaler.fit_transform(df1[['Quantity']])


# mô tả dữ liệu sau chuẩn hóa
pd.DataFrame(df_s).describe()
```

	0
count	74902.000000
mean	0.768581
std	55.283898
min	-10602.571429
25%	-0.285714

```
# vẽ lại biểu đồ hộp
sns.boxplot(x=df_s)
```

```
# chuẩn hóa dữ liệu với z-score scaling
scaler = StandardScaler()

# Chuẩn hóa dữ liệu trong df với StandardScaler ở 2 cột Quantity và UnitPrice
df_s = scaler.fit_transform(df1[['Quantity']])


# mô tả dữ liệu sau chuẩn hóa
pd.DataFrame(df_s).describe()
```

	0
count	7.490200e+04

```
sns.boxplot(x=df_s)
```

```
sns.kdeplot(data=df_s)
```

50%	1.390254e-02

Mã hóa dữ liệu

max	1.917560e+02

```
# các giá trị ở cột Country
df1['Country'].unique()
```

```
array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',
       'Norway', 'EIRE', 'Switzerland', 'Spain', 'Poland', 'Portugal',
       'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',
       'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',
       'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',
       'Lebanon'], dtype=object)
```

```
# mã hóa cột Country với One-hot encoder sử dụng scikit learn
encoder = OneHotEncoder()

encoded_data = encoder.fit_transform(np.asarray(df1['Country']).reshape(-1,1))
encoded_data.todense()
```

```
matrix([[0., 0., 0., ..., 0., 0., 1.],
        [0., 0., 0., ..., 0., 0., 1.],
        [0., 0., 0., ..., 0., 0., 1.],
        ...,
        [0., 0., 0., ..., 0., 0., 1.],
        [0., 0., 0., ..., 0., 0., 1.],
        [0., 0., 0., ..., 0., 0., 1.]])
```

```
# mã hóa cột Country với One-hot encoder sử dụng pandas
pd.get_dummies(df1['Country'])
```

	Australia	Austria	Bahrain	Belgium	Channel Islands	Cyprus	Denmark	EIRE	Finland	France	Germany	Greece	Hong Kong	Ice
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
75100	0	0	0	0	0	0	0	0	0	0	0	0	0	
75101	0	0	0	0	0	0	0	0	0	0	0	0	0	
75102	0	0	0	0	0	0	0	0	0	0	0	0	0	
75103	0	0	0	0	0	0	0	0	0	0	0	0	0	
75104	0	0	0	0	0	0	0	0	0	0	0	0	0	

74902 rows × 28 columns

```
# mã hóa cột Country với Label encoder sử dụng scikit learn
encoder = LabelEncoder()

encoded_data = encoder.fit_transform(np.asarray(df1['Country']))
encoded_data
```

```
array([27, 27, 27, ..., 27, 27, 27])
```

```
# mã hóa cột Country với Label encoder sử dụng pandas
df1['Country'].astype('category').cat.codes
```

```
0        27
1        27
2        27
3        27
4        27
         ..
75100    27
75101    27
75102    27
75103    27
75104    27
Length: 74902, dtype: int8
```

Rời rạc hóa dữ liệu

```
df1.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	outl
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	0.428571	12/1/2010 8:26	2.55	17850.0	United Kingdom	.
1	536365	71053	WHITE METAL LANTERN	0.428571	12/1/2010 8:26	3.39	17850.0	United Kingdom	.
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	0.714286	12/1/2010 8:26	2.75	17850.0	United Kingdom	.

```
# Rời rạc hóa dữ liệu ở cột UnitPrice

# chia thành 4 khoảng giá trị có độ dài bằng nhau
cats = pd.cut(df1['UnitPrice'], 4)
cats
```

```
0        (-16.888, 4222.005]
1        (-16.888, 4222.005]
2        (-16.888, 4222.005]
3        (-16.888, 4222.005]
4        (-16.888, 4222.005]
                ...
75100    (-16.888, 4222.005]
75101    (-16.888, 4222.005]
75102    (-16.888, 4222.005]
75103    (-16.888, 4222.005]
75104    (-16.888, 4222.005]
Name: UnitPrice, Length: 74902, dtype: category
Categories (4, interval[float64]): [(-16.888, 4222.005] < (4222.005, 8444.01] < (8444.01, 12666.015] <
                                    (12666.015, 16888.02]]
```

```
# số lượng phần từ ở mỗi phần
pd.value_counts(cats)
```

```
(-16.888, 4222.005]      74894
(12666.015, 16888.02]        6
(4222.005, 8444.01]          2
(8444.01, 12666.015]         0
Name: UnitPrice, dtype: int64
```

```
# chia thành 4 phần có số lượng phần tử tương đương nhau
cats = pd.qcut(df1['UnitPrice'], 4)
cats
```

```
0           (2.51, 4.24]
1           (2.51, 4.24]
2           (2.51, 4.24]
3           (2.51, 4.24]
4           (2.51, 4.24]
               ...
75100    (4.24, 16888.02]
75101       (1.25, 2.51]
75102       (2.51, 4.24]
75103     (-0.001, 1.25]
75104       (2.51, 4.24]
Name: UnitPrice, Length: 74902, dtype: category
Categories (4, interval[float64]): [(-0.001, 1.25] < (1.25, 2.51] < (2.51, 4.24] < (4.24, 16888.02]]
```

```
# số lượng phần từ ở mỗi phần
pd.value_counts(cats)
```

```
(-0.001, 1.25]      19906
(1.25, 2.51]        19459
(4.24, 16888.02]    18643
(2.51, 4.24]        16894
Name: UnitPrice, dtype: int64
```

Tổng kết

Qua bài thực hành này ta đã ôn tập lại các kiến thức xử lý dữ liệu