```
# Sumber File => https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/
# Sumber Dataset => http://archive.ics.uci.edu/ml/datasets/Online+Retail
```

```
# Langkah 1: Mengimpor library yang diperlukan
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

```
# Langkah 2: Meload dan Mengeksplor data
# Memuat Data
data = pd.read_excel("Online_Retail.xlsx")
data.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Coun
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	Ur Kingo
1	536365	71053	WHITE METAL	6	2010-12-01 08:26:00	3.39	17850.0	Ur Kingo

```
# Mengeksplor kolom data
data.columns
```

```
Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
       'UnitPrice', 'CustomerID', 'Country'],
      dtype='object')
```

```
# Mengeksplor berbagai wilayah transaksi
data.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', 'United Arab Emirates', 'Saudi Arabia',
       'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',
       'European Community', 'Malta', 'RSA'], dtype=object)
```

```
# Langkah 3: Membersihkan Data
# Menghapus ruang ekstra dalam deskripsi
data['Description'] = data['Description'].str.strip()

# Menghapus baris tanpa nomor faktur (InvoiceNo)
data.dropna(axis = 0, subset =['InvoiceNo'], inplace = True)
data['InvoiceNo'] = data['InvoiceNo'].astype('str')

# Menghapus semua transaksi yang dilakukan secara kredit
data = data[~data['InvoiceNo'].str.contains('C')]
```

```
# Langkah 4: Memisahkan data sesuai dengan wilayah transaksi
# Transaksi dilakukan di Prancis
basket_France = (data[data['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))

# Transaksi dilakukan di United Kingdom
basket_UK = (data[data['Country'] =="United Kingdom"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))

# Transaksi dilakukan di Portugal
basket_Por = (data[data['Country'] =="Portugal"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

```
# Transaksi dilakukan di Sweden
basket_Sweden = (data[data['Country'] =="Sweden"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
basket_Sweden
```

```
# Langkah 5: Pengkodean Data dengan "Hot Encoding"
# Mendefinisikan fungsi "Hot Encoding" untuk membuat data sesuai untuk library yang bersangkutan
def hot_encode(x):
    if(x<= 0):
        return 0
    if(x>= 1):
        return 1
```

```
# Mengkodekan dataset
basket_encoded = basket_France.applymap(hot_encode)
basket_France = basket_encoded

basket_encoded = basket_UK.applymap(hot_encode)
basket_UK = basket_encoded
```

```
basket_encoded = basket_Por.applymap(hot_encode)
basket_Por = basket_encoded

basket_encoded = basket_Sweden.applymap(hot_encode)
basket_Sweden = basket_encoded
basket_Sweden
```

```
    574690        24.0        0.0      0.0        0.0        0.0         0.0         0.0    0.0
```

```
# Langkah 6: Membangun model dan menganalisis hasilnya
# 1) France
# Membangun model
frq_items = apriori(basket_France, min_support = 0.05, use_colnames = True)

# Mengumpulkan aturan yang disimpulkan dalam dataframe
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
#print(rules.head())
rules.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	convict
44	(JUMBO BAG WOODLAND ANIMALS)	(POSTAGE)	0.076531	0.765306	0.076531	1.000	1.306667	0.017961	
258	(PLASTERS IN TIN CIRCUS PARADE	(POSTAGE)	0.051020	0.765306	0.051020	1.000	1.306667	0.011974	

```
542911          0          0          0          0          1          0          0          0
```

```
# 2) United Kingdom
frq_items = apriori(basket_UK, min_support = 0.01, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
#print(rules.head())
rules.head(10)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
116	(BEADED CRYSTAL HEART PINK ON STICK)	(DOTCOM POSTAGE)	0.011036	0.037928	0.010768	0.975728	25.725872	0.010349
2018	(JAM MAKING SET PRINTED, SUKI SHOULDER BAG)	(DOTCOM POSTAGE)	0.011625	0.037928	0.011196	0.963134	25.393807	0.010755
2295	(HERB MARKER THYME, HERB MARKER MINT)	(HERB MARKER ROSEMARY)	0.010714	0.012375	0.010232	0.955000	77.173095	0.010099
2300	(HERB MARKER PARSLEY, HERB MARKER ROSEMARY)	(HERB MARKER THYME)	0.011089	0.012321	0.010553	0.951691	77.240055	0.010417
2301	(HERB MARKER PARSLEY,	(HERB MARKER	0.011089	0.012375	0.010553	0.951691	76.905682	0.010416

```
# Portugal
frq_items = apriori(basket_Por, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
#print(rules.head())
rules.head(6)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	convi
1170	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET 12 COLOUR PENCILS SPACEBOY)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	
1171	(SET 12 COLOUR PENCILS SPACEBOY)	(SET 12 COLOUR PENCILS DOLLY	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	

```
# Sweden
frq_items = apriori(basket_Sweden, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, metric ="lift", min_threshold = 1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
#print(rules.head())
rules.head(3)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(12 PENCILS SMALL TUBE SKULL)	(PACK OF 72 SKULL CAKE CASES)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	inf
	(PACK OF 72	(12 PENCILS							