

Nama : Moh. Harysakti Rahanyamtel

Nim : G.231.22.0045

Matkul: Data Mining

```
1 import pandas as pd
2 from mlxtend.frequent_patterns import apriori
3 from mlxtend.frequent_patterns import association_rules
4
5 df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online20Retail1.xlsx')
6 df.head()
```

- Mengimpor pustaka pandas untuk manipulasi data dan pustaka MLxtend untuk analisis asosiasi.
- Membaca data dari file Excel dengan url.
- Menampilkan data yang telah dibaca untuk melihat contoh data awal

```
1 df.rename(columns={'InvoiceNo': 'NOFaktur',
2                    'Description': 'Deskripsi',
3                    'StockCode': 'KodeStok',
4                    'Quantity': 'Jumlah',
5                    'InvoiceDate': 'FakturTanggal',
6                    'UnitPrice': 'HargaSatuan',
7                    'CustomerID': 'IDPelanggan',
8                    'Country': 'Negara'}, inplace=True)
9 df['Deskripsi'] = df['Deskripsi'].str.strip()
10 df.dropna(axis=0, subset=['NOFaktur'], inplace=True)
11 df['NOFaktur'] = df['NOFaktur'].astype('str')
12 df = df[~df['NOFaktur'].str.contains('C')]
```

- Mengubah nama kolom seperti mengubah 'InvoiceNo' menjadi 'NOFaktur' dan 'Description' menjadi 'Deskripsi' dan lainnya
- Menghapus spasi ekstra dari nilai dalam kolom 'Deskripsi'.
- Menghapus baris yang memiliki nilai kosong pada kolom 'NOFaktur'.
- Mengonversi nilai dalam kolom 'NOFaktur' menjadi string.
- Menghapus baris yang memiliki 'NOFaktur' yang mengandung huruf 'C'.

```
1 basket = (df[df['Negara'] == "France"]
2           .groupby(['NOFaktur', 'Deskripsi'])['Jumlah']
3           .sum().unstack().reset_index().fillna(0)
4           .set_index('NOFaktur'))
```

- Memilih baris-baris di mana nilai kolom 'Negara' sama dengan "France".
- Mengelompokkan data berdasarkan kolom 'NOFaktur' dan 'Deskripsi'.
- Menghitung jumlah item ('Jumlah') yang dibeli dalam setiap grup.
- Mengubah bentuk data hasil pengelompokan agar lebih mudah dibaca.
- Mengisi nilai-nilai kosong dengan 0.
- Mengatur 'NOFaktur' sebagai indeks DataFrame yang baru

```
1 def encode_units(x):
2     if x <= 0:
3         return 0
4     if x >= 1:
5         return 1
6
7 basket_sets = basket.applymap(encode_units)
8 basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

- definisikan sebuah fungsi encode_units yang akan mengubah nilai-nilai dalam DataFrame basket menjadi 0 jika nilainya kurang dari atau sama dengan 0, dan menjadi 1 jika nilainya lebih besar atau sama dengan 1.

- fungsi encode_units diterapkan pada setiap elemen DataFrame basket menggunakan metode .applymap(), dan hasilnya disimpan dalam DataFrame baru yang disebut basket_sets.

- kolom 'POSTAGE' dihapus dari DataFrame basket_sets menggunakan metode .drop().

```
1 frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
```

- menggunakan algoritma Apriori untuk mengidentifikasi kelompok item yang sering muncul bersama dalam DataFrame basket_sets. Pengaturan min_support=0.07 berarti hanya kelompok item yang muncul dalam setidaknya 7% dari transaksi yang dianggap penting. use_colnames=True menandakan bahwa nama item yang digunakan dalam DataFrame akan dipertahankan dalam hasil kelompok item yang sering muncul bersama.

```
1 rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
2 rules.head()
```

- hasil frequent itemsets yang dihasilkan sebelumnya untuk membuat aturan asosiasi menggunakan metrik "lift" dengan ambang minimum 1. Hasil aturan asosiasi disimpan dalam variabel rules, dan lima baris pertama dari aturan tersebut ditampilkan menggunakan metode .head().

```
1 rules[ (rules['lift'] >= 6) &
2        (rules['confidence'] >= 0.8) ]
```

- mengambil aturan asosiasi dari variabel rules dimana nilai "lift" lebih besar atau sama dengan 6 dan nilai "confidence" lebih besar atau sama dengan 0.8.

```
1 basket['ALARM CLOCK BAKELIKE GREEN'].sum()
2
3 340.0
4
5 basket['ALARM CLOCK BAKELIKE RED'].sum()
6
7 316.0
```

- menghitung jumlah total pembelian untuk produk "ALARM CLOCK BAKELIKE GREEN" dan "ALARM CLOCK BAKELIKE RED" dari DataFrame basket

```
1 basket2 = (df[df['Negara'] == "Germany"]
2            .groupby(['NOFaktur', 'Deskripsi'])['Jumlah']
3            .sum().unstack().reset_index().fillna(0)
4            .set_index('NOFaktur'))
5
6 basket_sets2 = basket2.applymap(encode_units)
7 basket_sets2.drop('POSTAGE', inplace=True, axis=1)
8 frequent_itemsets2 = apriori(basket_sets2, min_support=0.05, use_colnames=True)
9 rules2 = association_rules(frequent_itemsets2, metric="lift", min_threshold=1)
10
11 rules2[ (rules2['lift'] >= 4) &
12         (rules2['confidence'] >= 0.5)]
```

- Memilih baris-baris di mana nilai kolom 'Negara' sama dengan "Germany".
- Mengelompokkan data berdasarkan kolom 'NOFaktur' dan 'Deskripsi'.
- Menghitung jumlah item ('Jumlah') yang dibeli dalam setiap grup.
- Mengubah bentuk data hasil pengelompokan agar lebih mudah dibaca.
- Mengisi nilai-nilai kosong dengan 0.
- Mengatur 'NOFaktur' sebagai indeks DataFrame yang baru.
- Menggunakan fungsi encode_units untuk mengonversi data.
- Menghapus kolom 'POSTAGE' dari DataFrame.
- Menggunakan algoritma Apriori untuk menghasilkan frequent itemsets dengan support minimum 0.05.
- Menghasilkan aturan asosiasi dengan menggunakan metrik "lift" dan ambang minimum 1.
- Mengambil aturan asosiasi yang memiliki nilai "lift" minimal 4 dan nilai "confidence" minimal 0.5.