

Nama: Deannys Anugerah F

NIM: G.231.22.0006

Praktikum Data Mining Basket

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx')
df.head()
```

Kode ini mengimpor pustaka pandas untuk pengolahan data, serta apriori dan association_rules dari mlxtend.frequent_patterns untuk menemukan pola asosiasi dalam data.

Di sini, data dibaca dari file Excel yang terdapat di URL yang diberikan dan ditampilkan 5 baris pertama untuk melihat sekilas data.

```
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
```

Membersihkan data dengan menghilangkan spasi di sekitar deskripsi produk, menghapus baris yang memiliki nilai InvoiceNo kosong, mengubah tipe data InvoiceNo menjadi string, dan menghapus faktur yang memiliki 'C' yang menunjukkan pengembalian.

```
basket = (df[df['Country'] == "France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

Mengelompokkan data berdasarkan InvoiceNo dan Description, menjumlahkan Quantity untuk setiap kelompok, dan merestrukturisasi data sehingga setiap faktur memiliki kolom untuk setiap produk.

```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)
```

Fungsi ini mengonversi nilai kuantitas menjadi biner: 0 jika kuantitas kurang dari atau sama dengan nol, dan 1 jika kuantitas lebih besar dari atau sama dengan satu. Menerapkan fungsi `encode_units` ke semua elemen dalam `basket`, dan menghapus kolom 'POSTAGE'.

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

Menggunakan algoritma apriori untuk menemukan itemset yang sering muncul dengan dukungan minimal 7%.

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

Menggunakan itemset yang sering muncul untuk menghitung aturan asosiasi berdasarkan metrik `lift` dengan ambang batas minimum 1 dan menampilkan 5 aturan pertama.

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

Menyaring aturan asosiasi dengan nilai `lift` minimal 6 dan `confidence` minimal 0.8.

```
basket['ALARM CLOCK BAKELIKE GREEN'].sum()

340.0

basket['ALARM CLOCK BAKELIKE RED'].sum()

316.0
```

Menghitung total kuantitas produk 'ALARM CLOCK BAKELIKE GREEN' dan 'ALARM CLOCK BAKELIKE RED' yang terjual.

```
basket2 = (df[df['Country'] == "Germany"]
           .groupby(['InvoiceNo', 'Description'])['Quantity']
           .sum().unstack().reset_index().fillna(0)
           .set_index('InvoiceNo'))

basket_sets2 = basket2.applymap(encode_units)
basket_sets2.drop('POSTAGE', inplace=True, axis=1)
frequent_itemsets2 = apriori(basket_sets2, min_support=0.05, use_colnames=True)
rules2 = association_rules(frequent_itemsets2, metric="lift", min_threshold=1)

rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5)]
```

Langkah yang sama seperti sebelumnya, tetapi kali ini untuk negara Jerman.

Mengonversi data menjadi biner dan menghapus kolom 'POSTAGE'.

Menerapkan algoritma apriori dengan dukungan minimal 5% untuk negara Jerman.

Menghitung aturan asosiasi dengan metrik `lift` dan menyaring aturan dengan nilai `lift` minimal 4 dan `confidence` minimal 0.5 untuk negara Jerman.

Link github:

https://github.com/P1nkP1sY/Deannys_G.231.22.0006_MarketAnalysis/blob/main/Basket_G_231_22_0006.ipynb