Chairul Imam
1608107010040
Association Rule Mining

Deskripsi Dataset

Saya menggunakan dataset **BreadBasket_DMS.csv**. Dataset terdiri dari **4 feature** dan **21293 instance**. Nama Feature yaitu **Date**, **Time**, **Transactions** dan **Item**.

Untuk mengunduh dataset, silakan akses URL berikut: http://bit.ly/3h8abzF

Dataset Preprocessing

Langkah dan screenshot yang dimuat pada dokumen ini adalah snippet utama dari notebook yang lebih lengkap. Source code yang lebih lengkap dapat diakses pada bagian **Source Code**.

Drop Feature Date dan Time

Feature **Date** dan **Time** dihapus karena tidak digunakan dalam analisis *association rule*.

```
# drop feature date and time
df.drop(columns=['Date', 'Time'],
        inplace=True)
```

Drop Transaksi dengan Item NONE

Dari eksplorasi item terlaris, ditemukan item **NONE** didalam dataset. Item **NONE** ini mungkin terjadi karena kesalahan penginputan.

Transaksi dengan item NONE dihapus dengan mencari index transaksi tersebut. Dengan informasi index transaksi, fungsi drop akan menghapus transaksi tersebut.

```
# save index where Item == NONE
index_of_none = df[ df.Item == 'NONE' ].index

# dimension of  instance where item == NONE
index_of_none.shape
```

```
(786,)
```

```
# drop instance according to index_of_none
df.drop(index=index_of_none,
        inplace=True)
```

Penyesuaian Struktur Dataset

Library **mlxtend** mengharuskan dataset transaksi dalam bentuk list-of-list. Item didalam list transaksi tidak boleh duplikat.

```
list_transaksi = []

for id in transaksi_unik:

    # set agar item di transaksi tidak duplikat
    transaksi = set( df[ df.Transaction == id]['Item'] )

    # konversi ke list
    listed_transaksi = list(transaksi)

    # append list yang telah disort
    list_transaksi.append( sorted(listed_transaksi) )
```

```
# view list transaksi
list_transaksi[:15]
```

```
[['Bread'],
 ['Scandinavian'],
 ['Cookies', 'Hot chocolate', 'Jam'],
 ['Muffin'],
 ['Bread', 'Coffee', 'Pastry'],
 ['Medialuna', 'Muffin', 'Pastry'],
 ['Coffee', 'Medialuna', 'Pastry', 'Tea'],
 ['Bread', 'Pastry'],
 ['Bread', 'Muffin'],
 ['Medialuna', 'Scandinavian'],
 ['Bread', 'Medialuna'],
 ['Coffee', 'Jam', 'Pastry', 'Tartine', 'Tea'],
 ['Basket', 'Bread', 'Coffee'],
 ['Bread', 'Medialuna', 'Pastry'],
 ['Mineral water', 'Scandinavian']]
```

Implementasi Apriori

Encode Item Transaksi

```
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
te_array = te.fit_transform(list_transaksi)

te_array
```

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

```
# konversi ke dataframe
encoded_transactions = pd.DataFrame(data=te_array,
                                    columns=te.columns_)
# simpan transaksi setelah diencode
encoded_transactions.to_csv('../output/encoded_transactions.csv',
                            index=False)
```

```
encoded_transactions.head()
```

	Adjustment	Afternoon with the baker	Alfajores	Argentina Night	Art Tray	Bacon	Baguette	Bakewell	Bare Popcorn	Basket	...	The BART
0	False	False	False	False	False	False	False	False	False	False	...	False
1	False	False	False	False	False	False	False	False	False	False	...	False
2	False	False	False	False	False	False	False	False	False	False	...	False
3	False	False	False	False	False	False	False	False	False	False	...	False
4	False	False	False	False	False	False	False	False	False	False	...	False

5 rows × 94 columns

Cari Frequent Itemsets

```
from mlxtend.frequent_patterns import apriori

# frequent itemset dengan parameter min_support
frequent_itemsets = apriori(df=encoded_transactions,
                            min_support=0.005,
                            use_colnames=True)

# sort itemsets dengan support tertinggi
frequent_itemsets.sort_values(by=['support'],
                              ascending=False,
                              inplace=True)

# save frequent itemsets
frequent_itemsets.to_csv('../output/frequent_itemsets.csv',
                         index=False)
```

Cari Association Rule

Dari frequent itemset yang memenuhi kriteria **min_support=0.005**, dicari association rule.

```
from mlxtend.frequent_patterns import association_rules

# cari rule dengan lift
assoc_rules = association_rules(df=frequent_itemsets,
                                metric='lift',
                                min_threshold=2.5)

# sort association rules
assoc_rules.sort_values(by=['lift'],
                        ascending=False,
                        inplace=True)
```

Kesimpulan

Dari kriteria awal yang diberikan, tidak ditemukan rule yang memenuhi kriteria tersebut. Saya berinisiatif untuk menurunkan parameter **min_lift** menjadi 2.5 dan mengabaikan parameter **min_confidence**. Jadi parameter baru yang dibuat adalah **min_lift=2.5** dan **min_support=0.005**.

Nilai **confidence** diabaikan karena dapat menimbulkan misleading. Nilai **Confidence** rule yang memiliki **consequent_support** yang tinggi akan selalu tinggi. Untuk mencegah misleading dari nilai confidence yang tinggi, saya memprioritaskan nilai lift.

Berikut adalah rule yang dihasilkan dengan parameter **min_support=0.05** dan **min_lift=2.5**

```
assoc_rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(Coke)	(Sandwich)	0.019440	0.071844	0.005177	0.266304	3.706722	0.00378	1.265043
2	(Sandwich)	(Coke)	0.071844	0.019440	0.005177	0.072059	3.706722	0.00378	1.056705
0	(Juice)	(Cookies)	0.038563	0.054411	0.006128	0.158904	2.920442	0.00403	1.124234
1	(Cookies)	(Juice)	0.054411	0.038563	0.006128	0.112621	2.920442	0.00403	1.083457

Source Code

Untuk mengakses dan mencoba source code, silakan mengakses pada URL berikut:

https://github.com/chairul-imam/Data-Mining-and-Machine-Learning/tree/main/Association-Rules-Mining

Referensi

Fast Algorithms for Mining Association Rules

https://towardsdatascience.com/association-rules-2-aa9a77241654

https://www.kaggle.com/aboliveira/bakery-market-basket-analysis

http://rasbt.github.io/mlxtend/api_subpackages/mlxtend.frequent_patterns/

http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/

http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/