

TUGAS KONSEP APLIKASI DATA MAINING
ASSOCIATION RULE (MENGHITUNG SUPPORT DAN CONFIDANCE)
MENGGUNAKAN JUPYTER NETBOOK



Aminurachma Aisyah Nilatika
17.52.001

KEMENTRIAN RISET DAN TEKNOLOGI PENDIDIKAN TINGGI
SEKOLAH TINGGI MANAJEMEN INFORMATIKA KOMPUTER
PRADNYA PARAMITA
MALANG
2020

Secara sederhana beberapa istilah yang akan sering di gunakan pada *Association Rule* ini adalah:

- **Support** adalah indikasi seberapa sering itemset muncul di dataset.
- **Confidence** adalah indikasi seberapa sering aturan itu terbukti benar.
- **Lift Ratio** adalah rasio dari nilai pengamatan *Support* , diharapkan jika dua aturan itu independen

1. Yang pertama adalah melakukan import library, yang pertama adalah pandas nanti di sebut sebagai pd, matplotlib.pyplot nanti di sebut sebagai plt, Kemudian mlxtend melakukan import fungsi apriori dan association_rules

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from apyori import apriori
```

2. Selanjutnya adalah membaca file excel menggunakan library pandas. Dimana nama file excel yang di gunakan adalah ('D:/Aminurachma Aisyah Nilatika/STIMATA/Semester 6/Data Mining/UAS/datamining/Uas/dataset_soal No. 3.xls',header = None) lalu, header = None artinya adalah menghilangkan Header atasnya, jika sudah berhasil maka akan muncul head tabel dari data tersebut.

```
In [4]: data = pd.read_excel('D:/Aminurachma Aisyah Nilatika/STIMATA/Semester 6/Data Mining/UAS/datamining/Uas/dataset_soal No. 3.xls',header = None)
In [5]: data.head()
```

Dengan output :

	0	1	2	3	4	5	6	7	8	9
0	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
1	burgers	meatballs	eggs	low fat yogurt	NaN	mineral water	salmon	low fat yogurt	NaN	mineral water
2	chutney	low fat yogurt	NaN	whole wheat pasta	french fries	mineral water	salmon	whole wheat pasta	french fries	mineral water
3	turkey	whole wheat pasta	french fries	soup	light cream	shallot	NaN	soup	light cream	shallot
4	mineral water	soup	light cream	frozen vegetables	spaghetti	green tea	NaN	frozen vegetables	spaghetti	green tea

3. Hitung jumlah dataset yang kita miliki dengan menggunakan syntax berikut ini :

```
In [6]: data.shape
```

Dengan output :

```
Out[6]: (2055, 10)
```

4. Selanjutnya, Library Apriori yang akan kita gunakan membutuhkan dataset dalam bentuk list. Saat ini kami memiliki dataset dalam bentuk dataset pandas. Untuk mengonversi dataset dalam pandas kami ke List di dalam list, jalankan skrip berikut:

```
In [8]: records = []
        for i in range(0, 2055):
            records.append([str(data.values[i,j]) for j in range(0, 10)])
```

Penjelasan syntax : records merupakan nama head yang akan berisi hasil dari perulangan. Append merupakan syntax untuk menambah baris.

5. Langkah selanjutnya adalah menerapkan algoritma Apriori pada dataset. Untuk melakukannya, kita bisa menggunakan kelas apriori yang kita impor dari perpustakaan apyori. Jalankan skrip berikut :

```
In [9]: association_rules = apriori(records, min_support=0.2, min_confidence=0.2, min_lift=0.2, min_length=2)
        association_results = list(association_rules)
```

6. Pertama mari kita temukan jumlah total aturan yang diexecute oleh Library apriori. Jalankan skrip berikut:

```
In [10]: print(len(association_results))

61
```

7. Selanjut mari menampilkan item pertama di association_rules untuk melihat apa aturan pertama. Jalankan skrip berikut:

```
In [11]: association_results[0]
```

Dengan output:

```
Out[11]: RelationRecord(items=frozenset({'avocado'}), support=0.3143552311435523, ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'avocado'}), confidence=0.3143552311435523, lift=1.0)])
```

8. Script untuk menampilkan support, confidence, dan lift, maka jalankan script dibawah ini:

```
In [14]: results = []
        for item in association_results:
            pair = item[0]
            items = [x for x in pair]

            value0 = str(items[0])
            value2 = str(item[1])[:10]
            value3 = str(item[2][0][2])[:10]
            value4 = str(item[2][0][3])[:10]

            rows = (value0, value2, value3, value4)

            results.append(rows)

        label = ['Title1'], ['Support'], ['Confidence'], ['Lift']
        print("=====")

        store_suggestion = pd.DataFrame.from_records(results, columns=label)

        print(store_suggestion)
```

Dengan output :

9. Selanjutnya jalankan script berikut ini untuk menampilkan kesimpulan

```
store_suggestion.describe()
```

Dengan Output :

```
In [35]: store_suggestion.describe()
```

Out[35]:

	(Title1,)	(Support,)	(Confidence,)	(Lift,)
count	61	61	61	61
unique	15	53	53	1
top	nan	0.24294060	0.24294060	1.0
freq	14	4	4	61

10. Selanjutnya meneksport hasilnya ke bentuk excel dengan menggunakan script berikut ini :

```
In [17]: store_suggestion.to_excel('C:/Users/Aminurachma Aisyah/Outputs_AR_Nomer3.xls')
```

Kesimpulan

Algoritma Association rule mining seperti Apriori sangat berguna untuk menemukan asosiasi sederhana. Mereka mudah diimplementasikan dan memiliki kemampuan menjelaskan yang tinggi. Pada tugas ini saya mencoba (min_support=0.2, min_confidence=0.2, min_lift=0.2, min_length=2) dan mendapatkan 61 list.