

Praktikum 8
Algoritma Apriori: Association Rule



Disusun Oleh:
Muhammad Risalah Naufal (21051214008)
S1 Sistem Informasi B

Mata Kuliah: Data Mining
Dosen Pengampu: Wiyli Yustanti, S.Si., M.Kom.

PROGRAM STUDI SISTEM INFORMASI
RUMPUN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SURABAYA 2023

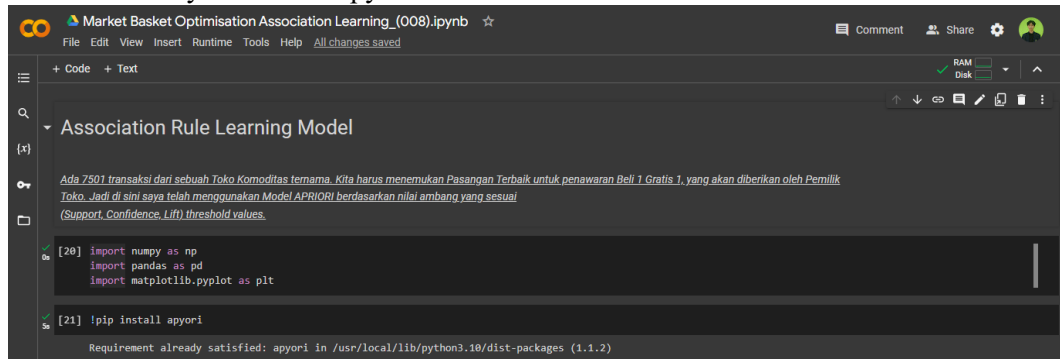
A. Dataset Penjualan

Dataset ini merupakan dataset yang berisi tentang hasil penjualan barang pada suatu toko. Data yang ada berisi tentang pembeli membeli barang apa saja dan nantinya akan dicari pembelian apa yang saling berasosiasi atau pembelian barang apa saja yang saling berhubungan.

Pada pengolahan data kali ini diharapkan dapat memberikan informasi terkait item apa saja yang memiliki asosiasi satu dengan lainnya, nantinya informasi yang ada akan digunakan untuk mengambil keputusan.

B. Metode Penelitian

1. Masukan library dan install apyori



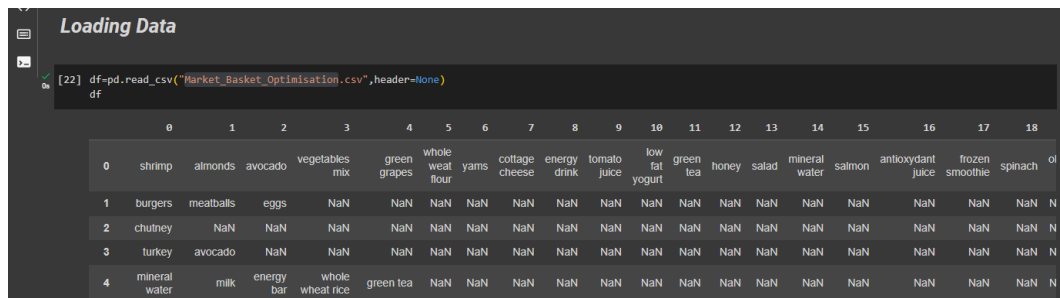
```
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Association Rule Learning Model
Ada 7501 transaksi dari sebuah Toko Komoditas ternama. Kita harus menemukan Pasangan Terbaik untuk penawaran Beli 1 Gratis 1, yang akan diberikan oleh Pemilik Toko. Jadi di sini saya telah menggunakan Model APRIORI berdasarkan nilai ambang yang sesuai (Support, Confidence, Lift) threshold values.
[20] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
[21] !pip install apyori
Requirement already satisfied: apyori in /usr/local/lib/python3.10/dist-packages (1.1.2)
```

Kode di atas adalah implementasi Python menggunakan beberapa pustaka, seperti NumPy, Pandas, dan Matplotlib. Berikut adalah penjelasan singkatnya:

- `import numpy as np`: Mengimpor pustaka NumPy dan memberikan alias "np" untuk digunakan dalam kode. NumPy digunakan untuk operasi numerik efisien.
- `import pandas as pd`: Mengimpor pustaka Pandas dan memberikan alias "pd". Pandas digunakan untuk manipulasi dan analisis data, terutama menggunakan struktur data seperti DataFrame.
- `import matplotlib.pyplot as plt`: Mengimpor modul pyplot dari pustaka Matplotlib dan memberikan alias "plt". Matplotlib digunakan untuk membuat visualisasi data, dan pyplot adalah modul yang menyediakan antarmuka mirip MATLAB untuk membuat plot.

Perintah tersebut, `!pip install apyori`, secara singkat digunakan untuk menginstal pustaka "apyori" menggunakan pip, yang diperlukan untuk implementasi Algoritma APRIORI dalam analisis asosiasi data.

2. Load data



```
Loading Data
[22] df=pd.read_csv("Market_Basket_Optimisation.csv",header=None)
df
0 shrimp almonds avocado vegetables mix green grapes whole wheat flour yams cottage cheese energy drink tomato juice low fat yogurt green tea honey salad mineral water salmon antioxidant juice frozen smoothie spinach ol
1 burgers meatballs eggs NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
2 chutney NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
3 turkey avocado NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
4 mineral water milk energy bar whole wheat rice green tea NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
```

Kode tersebut membaca file CSV ("Market_Basket_Optimisation.csv") menggunakan Pandas dan menyimpannya dalam DataFrame yang disebut df, tanpa baris judul. Selanjutnya, DataFrame tersebut ditampilkan.

3. Pre-Process

```
Data Preprocessing

In [23]: x=df.iloc[:,].values
        lis=[]
        count=0
        for i in x:
            lis.append([str(y) for y in i])

In [24]: from apyori import apriori
        association=apriori(transactions=lis,min_support=0.003,min_confidence=0.2,min_lift=3,min_length=2,max_length=2)
        confidence=[]
        lift=[]
        support=[]
        base_item=[]
        dependent_item=[]
        for sub in association:
            lift.append(sub[2][0][3])
            confidence.append(sub[2][0][2])
            support.append(sub[1])
            base_item.append(list(sub[2][0][0])[0])
            dependent_item.append(list(sub[2][0][1])[0])
        data=zip(base_item,dependent_item,support,confidence,lift)
```

Kode tersebut mengambil nilai dari seluruh baris dan kolom dari DataFrame df, kemudian mengonversi setiap elemen dalam setiap baris menjadi string, dan menyimpannya dalam bentuk sublist di dalam list lis. Tujuan dari langkah ini adalah untuk mempersiapkan data transaksi dalam format yang sesuai untuk analisis menggunakan algoritma asosiasi.

Kode kedua menggunakan pustaka apyori untuk menerapkan Algoritma APRIORI pada data transaksi yang telah dipersiapkan sebelumnya. Hasil analisis disimpan dalam variabel association. Selanjutnya, kode melakukan ekstraksi informasi terkait aturan asosiasi, seperti tingkat support, confidence, dan lift, untuk setiap aturan yang dihasilkan.

- min_support, min_confidence, min_lift, min_length, dan max_length adalah parameter yang mengatur ambang batas untuk mengekstraksi aturan asosiasi.
- Kode kemudian melakukan iterasi melalui hasil asosiasi dan mengekstrak informasi seperti lift, confidence, support, serta item dasar (base item) dan item tergantung (dependent item) untuk setiap aturan.
- Data hasil ekstraksi disusun dalam bentuk zip dari beberapa list, termasuk base_item, dependent_item, support, confidence, dan lift.

4. Membuat data frame

```
Making a Data Frame

In [25]: df2=pd.DataFrame(columns=["Base_Item","Dependent_Item","Support","Confidence","Lift"])

In [26]: list2=[list(i) for i in data]

In [27]: j=0
        for z in list2:
            df2.loc[j]=z
            j=j+1
        df2
```

	Base_Item	Dependent_Item	Support	Confidence	Lift
0	light cream	chicken	0.004533	0.290598	4.843951
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
2	pasta	escalope	0.005866	0.372881	4.700812
3	fromage blanc	honey	0.003333	0.245098	5.164271
4	herb & pepper	ground beef	0.015998	0.323450	3.291994
5	tomato sauce	ground beef	0.005333	0.377358	3.840659

Kode di atas membuat DataFrame baru yang disebut df2 dengan kolom-kolom yang ditentukan, yaitu "Base_Item", "Dependent_Item", "Support", "Confidence", dan "Lift". DataFrame ini akan digunakan untuk menyimpan hasil ekstraksi informasi dari aturan asosiasi yang dihasilkan oleh algoritma APRIORI.

Kode kedua membuat list baru yang disebut list2, di mana setiap elemen dalam list tersebut adalah sublist yang dihasilkan dari mengonversi setiap tuple dalam variabel data menjadi list. Dengan kata lain, mengubah format data dari tuple ke dalam bentuk list.

Kode ketiga melakukan iterasi melalui setiap sublist dalam list2 dan memasukkan nilainya ke dalam baris baru dalam DataFrame df2. Setiap iterasi menambahkan satu baris

ke DataFrame dengan nilai dari sublist saat itu. Akhirnya, DataFrame df2 akan berisi informasi dari aturan asosiasi yang diekstrak sebelumnya.

5. Urutan Terurut dari Pasangan Barang Terbaik (Untuk Beli 1 Gratis 1)

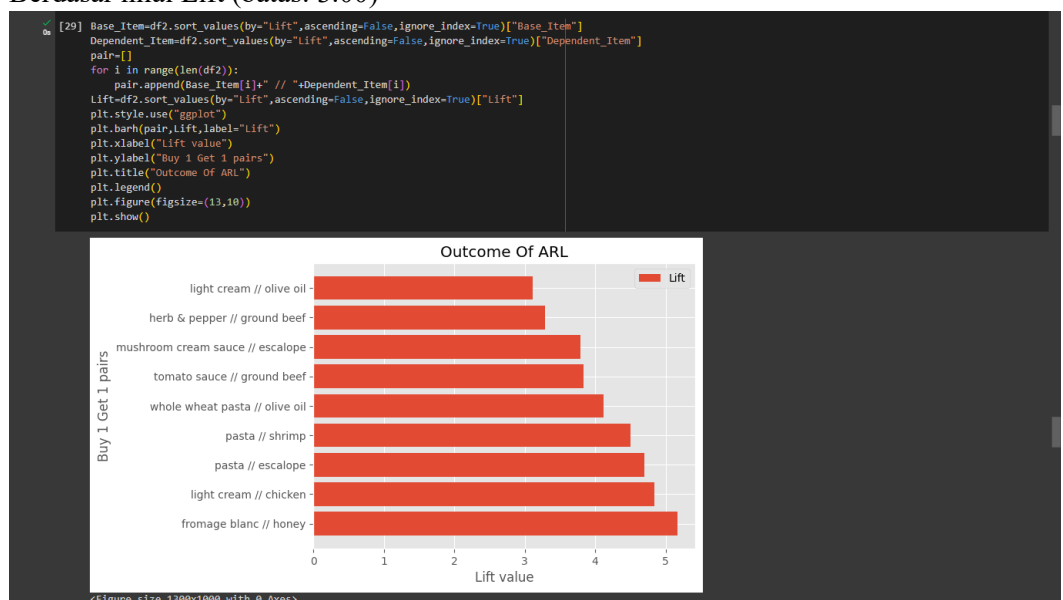
Following is the Sorted Order of Best Paired Items (For Buy 1 Get 1), based on their Lift (Threshold=3.00)

```
[28] df2.sort_values(by="Lift",ascending=False,ignore_index=True)
```

	Base_Item	Dependent_Item	Support	Confidence	Lift
0	fromage blanc	honey	0.003333	0.245098	5.164271
1	light cream	chicken	0.004533	0.290598	4.843951
2	pasta	escalope	0.005866	0.372881	4.700812
3	pasta	shrimp	0.005066	0.322034	4.506672
4	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
6	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
7	herb & pepper	ground beef	0.015998	0.323450	3.291994
8	light cream	olive oil	0.003200	0.205128	3.114710

Kode di atas mengurutkan DataFrame df2 berdasarkan kolom "Lift" secara menurun (descending) dan mengabaikan indeks aslinya, kemudian hasilnya disimpan kembali ke dalam DataFrame df2. Dengan demikian, DataFrame tersebut akan berisi aturan asosiasi yang diurutkan berdasarkan nilai "Lift" dari yang tertinggi ke yang terendah.

- Berdasar nilai Lift (batas: 3.00)



Kode di atas membuat diagram batang horizontal (bar chart) untuk menampilkan pasangan item (Buy 1 Get 1 pairs) berdasarkan nilai "Lift" dari DataFrame df2. Berikut adalah penjelasan singkatnya:

- Sort dan Ekstraksi Data:
 - DataFrame df2 diurutkan berdasarkan kolom "Lift" secara menurun.
 - Ekstraksi kolom-kolom yang diperlukan, yaitu "Base_Item", "Dependent_Item", dan "Lift".
- Pembuatan Pasangan:
 - Menggabungkan item dasar dan item tergantung untuk membentuk pasangan, disimpan dalam list pair.
- Visualisasi:
 - Membuat diagram batang horizontal (bar chart) dengan menggunakan Matplotlib.
 - Label sumbu x diberi nama "Lift value".

- Label sumbu y diberi nama "Buy 1 Get 1 pairs".
- Menambahkan judul "Outcome Of ARL".
- Menampilkan legenda untuk menunjukkan bahwa yang ditampilkan adalah nilai "Lift".
- Menampilkan diagram batang horizontal.

Dengan cara ini, kode tersebut memberikan visualisasi yang memperlihatkan pasangan item berdasarkan nilai "Lift" untuk analisis asosiasi.

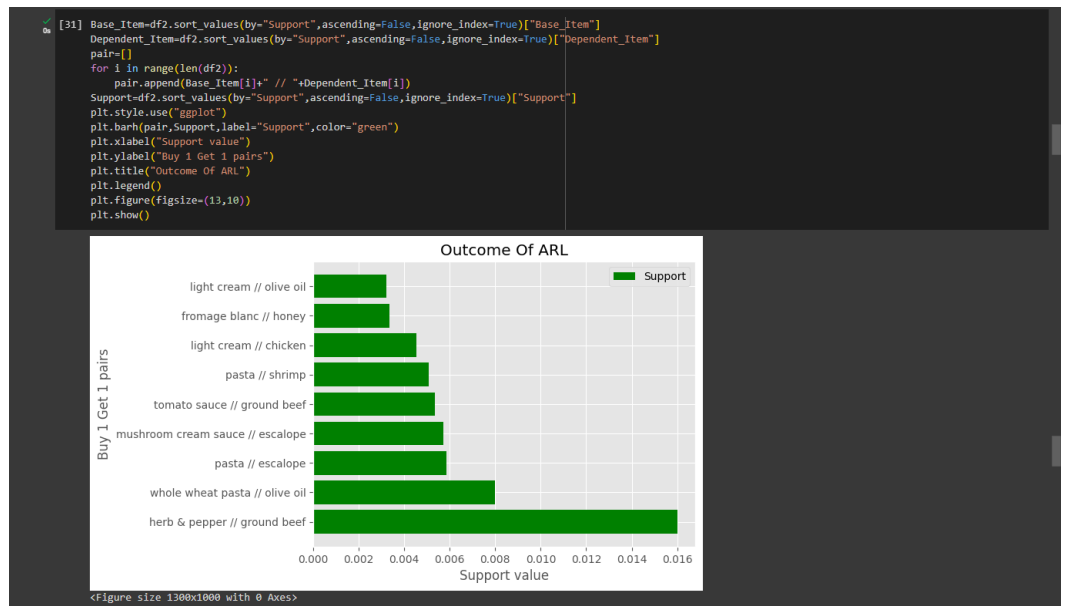
- Confidence (batas: 0.2)



Membuat diagram batang horizontal (bar chart) untuk menampilkan pasangan item (Buy 1 Get 1 pairs) berdasarkan nilai "Confidence" dari DataFrame df2. Berikut adalah penjelasan singkatnya:

- Sort dan Ekstraksi Data:
 - DataFrame df2 diurutkan berdasarkan kolom "Confidence" secara menurun.
 - Ekstraksi kolom-kolom yang diperlukan, yaitu "Base_Item", "Dependent_Item", dan "Confidence".
- Pembuatan Pasangan:
 - Menggabungkan item dasar dan item tergantung untuk membentuk pasangan, disimpan dalam list pair.
- Visualisasi:
 - Membuat diagram batang horizontal (bar chart) dengan menggunakan Matplotlib.
 - Label sumbu x diberi nama "Confidence value".
 - Label sumbu y diberi nama "Buy 1 Get 1 pairs".
 - Menambahkan judul "Outcome Of ARL".
 - Menampilkan legenda untuk menunjukkan bahwa yang ditampilkan adalah nilai "Confidence".
 - Memberikan warna hijau pada batang diagram menggunakan parameter color="green".
 - Menampilkan diagram batang horizontal.

- Support (batas: 0.003)



membuat diagram batang horizontal (bar chart) untuk menampilkan pasangan item (Buy 1 Get 1 pairs) berdasarkan nilai "Support" dari DataFrame df2. Berikut adalah penjelasan singkatnya:

- Sort dan Ekstraksi Data:
 - DataFrame df2 diurutkan berdasarkan kolom "Support" secara menurun.
 - Ekstraksi kolom-kolom yang diperlukan, yaitu "Base_Item", "Dependent_Item", dan "Support".
- Pembuatan Pasangan:
 - Menggabungkan item dasar dan item tergantung untuk membentuk pasangan, disimpan dalam list pair.
- Visualisasi:
 - Membuat diagram batang horizontal (bar chart) dengan menggunakan Matplotlib.
 - Label sumbu x diberi nama "Support value".
 - Label sumbu y diberi nama "Buy 1 Get 1 pairs".
 - Menambahkan judul "Outcome Of ARL".
 - Menampilkan legenda untuk menunjukkan bahwa yang ditampilkan adalah nilai "Support".
 - Memberikan warna hijau pada batang diagram menggunakan parameter color="green".
 - Menampilkan diagram batang horizontal.

6. Hasil running bila batas nilai Support dan Confidence: 0.7

```

[33] x=df.iloc[:,:].values
     lis=[]
     count=0
     for i in x:
         lis.append([str(y) for y in i])

[34] from apyori import apriori
     assosiation=apriori(transactions=lis,min_support=0.7,min_confidence=0.7,min_lift=3,min_length=2,max_length=2)
     confidence=[]
     lift=[]
     support=[]
     base_item=[]
     dependent_item=[]
     for sub in assosiation:
         lift.append(sub[2][0][3])
         confidence.append(sub[2][0][2])
         support.append(sub[1])
         base_item.append(list(sub[2][0][0])[0])
         dependent_item.append(list(sub[2][0][1])[0])
     data=zip(base_item,dependent_item,support,confidence,lift)

[35] df2=pd.DataFrame(columns=["Base_Item","Dependent_Item","Support","Confidence","Lift"])

[36] list2=[list(i) for i in data]

j=0
for z in list2:
    df2.loc[j]=z
    j=j+1
df2

```

Base_Item	Dependent_Item	Support	Confidence	Lift
-----------	----------------	---------	------------	------

Hasil running yang keluar bila batas nilai Support dan Confidence = 0,7. Tidak ada hasil yang dikeluarkan.

C. Kesimpulan

- Pre-Processing data menerapkan nilai batas yang berlaku pada Confidence dan Support adalah 0.003 untuk Support, dan 0,2 untuk Confidence.
- Lift: Berguna untuk menilai apakah aturan asosiasi memiliki nilai prediktif yang signifikan. Pemilihan berdasarkan lift dilakukan jika berfokus pada meningkatkan relevansi atau kebermanfaatan aturan tersebut.
- Support: Berguna untuk menilai seberapa umum atau jarang aturan asosiasi terjadi dalam dataset. Pemilihan berdasarkan support berguna jika ingin mengetahui popularitas atau distribusi aturan tersebut di antara transaksi.
- Confidence: Berguna untuk menilai seberapa kuat korelasi antara item A dan item B dalam aturan asosiasi. Pemilihan berdasarkan confidence berguna jika ingin menetapkan tingkat kepercayaan pada kebenaran aturan tersebut.
- Pada nilai Lift dapat dilihat bahwa asosiasi barang yang paling baik adalah fromage blanc dengan honey. Artinya ke dua barang tersebut memiliki relevansi paling bagus dalam hubungannya dengan nilai 5.164271.
- Pada nilai Support dapat dilihat bahwa asosiasi barang yang paling baik adalah herb & pepper dengan ground beef. Artinya ke dua barang tersebut adalah barang yang paling sering muncul dan populer dalam tiap transaksi, dan memiliki nilai 0.015998.
- Pada nilai Confidence dapat dilihat bahwa asosiasi barang yang paling baik adalah tomato sauce dengan ground beef. Artinya ke dua barang tersebut memiliki korelasi paling kuat dan sering muncul secara bersamaan, dan memiliki nilai 0.015998.
- Ketika nilai batas Confidence dan Support diubah menjadi 0,7 hasil yang dikeluarkan tidak ada dikarenakan tidak ada asosiasi antar barang yang memiliki nilai setinggi 0,7.