

Ujian Akhir Semester

Analisis Frequent Patterns dan Aturan Asosiasi
dari Dataset Groceries.csv

21 Januari 2022

6181801078/Adira Islah

Program Data Science
Jurusan Teknik Informatika
Universitas Katolik Parahyangan

I. Deskripsi Dataset

Nama File : Groceries.csv

Ukuran File : 1.078 KB

Jumlah Records : 38765 baris

Jumlah Atribut : 3

Nama Atribut	Tipe Data
Member_number	int
Date	object
itemDescription	object

II. Tujuan

Melakukan analisis Frequent items dan mencari aturan asosiasi dari dataset Groceries.csv

III. Metodologi Analisis

- Melakukan penyiapan data dengan membuat data set baru agar bisa mengimplement algoritma apriori dan fp-growtg
- proses data dengan algoritma apriori & fp-growth
- menganalisis hasil frequent items dan association rules
-

IV. Hasil Eksperimen

4.1. Penyiapan Data

Penyiapan data dilakukan dengan membuat dataset baru, dengan kolom Member_number sebagai index, Date dan semua jenis item lalu dibaris rekord jenis item terdapat berapa quantity pembelian item tersebut.

basket - DataFrame

	Member_number	Date	t food pro	UHT-milk	asive clea	if. sweete	ay cosme	bags	king pow
0	1000	15-03-2015	0	0	0	0	0	0	0
1	1000	24-06-2014	0	0	0	0	0	0	0
2	1000	24-07-2015	0	0	0	0	0	0	0
3	1000	25-11-2015	0	0	0	0	0	0	0
4	1000	27-05-2015	0	0	0	0	0	0	0
5	1001	02-05-2015	0	0	0	0	0	0	0
6	1001	07-02-2014	0	0	0	0	0	0	0
7	1001	12-12-2014	0	0	0	0	0	0	0
8	1001	14-04-2015	0	0	0	0	0	0	0
9	1001	20-01-2015	0	0	0	0	0	0	0

berikut adalah potongan screen shot data frame baru dengan format baru, ini dilakukan agar dataframe baru bisa diumpankan ke kode apriori dan FP-Growth

```
#data processing
df['itemDescription'] = df['itemDescription'].str.strip()
df['Member_number'] = df['Member_number'].astype('str')

df = df.reset_index()
basket = df.pivot_table(index=('Member_number','Date'),columns='itemDescription',

def encode_units3(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units3)
```

berikut adalah potongan code saat melakukan penyiapan data baru dengan membuat data frame baru, saya melakukan transformasi kolom dan baris dari semua atribut (Member_number, Date dan itemDescription) dari file csv lalu diencode.

4.3. Analisis Frequent Patterns dan Aturan Asosiasi

4.3.1. Analisis Frequent Patterns

Frequent Item dengan Algoritma Apriori

Minimum Support : 1% / 0.001

Minimum Lift : 0.7

Terdapat sebanyak 750 record untuk frequent itemsets yang ditemukan dengan minimum support 0.001

berikut beberapa record untuk hasil frequent item dengan min support = 0.001

Support	Item
0.00400989	Instant food products
0.0213861	UHT-Milk
0.00147029	abrasive cleaner
0.00193811	artif. sweetener
0.00808661	baking powder
0.00113614	bathroom cleaner
0.0339504	beef

Frequent Item dengan Algoritma Apriori

Minimum Support : 1% / 0.001

Minimum Lift : 0.7

Terdapat sebanyak 750 record untuk frequent itemsets yang ditemukan dengan minimum support 0.001

berikut beberapa record untuk hasil frequent item dengan min support = 0.001

Support	Item
0.157923	whole milk
0.0858785	yogurt
0.0603489	sausage
0.00949008	semi finished-bread
0.0517276	pastry
0.0187797	salty snack
0.0469157	canned beer

4.3.1. Analisis Aturan Asosiasi

Aturan Asosiasi dengan Algoritma Apriori

Minimum Support : 1% / 0.001

Minimum Lift : 0.7

Running time : 0:00:04.049395

Terdapat sebanyak 1010 records untuk hasil Aturan Asosiasi

Berikut 6 Aturan Asosiasi Terbaik (lift \geq 1)

Antecedants	Consequents	Antecedants support	Consequents support	Support	Confidence	Lift
sausage, yogurt	whole milk	0.00574751	0.157923	0.00147029	0.255814	1.61987
sausage, rolls/bun	whole milk	0.00534652	0.157923	0.00113614	0.2125	1.34559
susage, soda	whole milk	0.00594801	0.157923	0.0010693	0.179775	1.13837
semi-finished bread	whole milk	0.00949008	0.157923	0.00167079	0.176056	1.11482
yogurt, rolls/bun	whole milk	0.00781929	0.157923	0.00133663	0.17094	1.08243
detergent	whole milk	0.00862127	0.157923	0.00140346	0.162791	1.03082

dari beberapa record masih ada hasil lift yang kurang dari 1, maka record tersebut tidak dimasukkan ke aturan asosiasi terbaik karena lift kurang dari 1 adalah aturan asosiasi yang kurang baik/menarik hasil dari 6 record ini dapat dimanfaatkan untuk meningkatkan penjualan karena pada faktanya hasil dari aturan asosiasi support > min support dan lift > min lift/ lift > 1.

dari tabel diatas maka :

- setiap pembelian sausage, yogurt maka orang tersebut membeli whole milk
- setiap pembelian sausage, rolls/bun maka orang tersebut membeli whole milk
- setiap pembelian sausage, soda maka orang tersebut membeli whole milk
- setiap pembelian semi-finished bread maka orang tersebut membeli whole milk
- setiap pembelian detergent maka orang tersebut membeli whole milk

Aturan Asosiasi dengan Algoritma FP-Growth

Minimum Support : 1% / 0.001

Minimum Lift : 0.7

Running time :0:00:00.017991

Terdapat sebanyak 1010 records untuk hasil Aturan Asosiasi

Berikut 6 Aturan Asosiasi Terbaik (lift \geq 1)

Antecedants	Consequents	Antecedant s support	Consequents support	Support	Confide nce	Lift
whole milk, yogurt	sausage	0.0111609	0.0603489	0.00147 029	0.13173 7	2.18292
sausage, yogurt	whole milk	0.00574751	0.157923	0.00147 029	0.25581 4	1.61987
sausage, rolls/bun	whole milk	0.00534652	0.157923	0.00113 614	0.2125	1.34559
susage, soda	whole milk	0.00594801	0.157923	0.00106 93	0.17977 5	1.13837
semi-finished bread	whole milk	0.00949008	0.157923	0.00167 079	0.17605 6	1.11482
yogurt, rolls/bun	whole milk	0.00781929	0.157923	0.00133 663	0.17094	1.08243

dari beberapa record masih ada hasil lift yang kurang dari 1, maka record tersebut tidak dimasukan ke aturan asosiasi terbaik karena lift kurang dari 1 adalah aturan asosiasi yang kurang baik/menarik hasil dari 6 record ini dapat dimanfaatkan untuk meningkatkan penjualan karena pada faktanya hasil dari aturan asosiasi support > min support dan lift > min lift/ lift > 1.

dari tabel diatas maka :

- setiap pembelian whole milk, yogurt maka orang tersebut membeli sausage
- setiap pembelian sausage, rolls/bun maka orang tersebut membeli whole milk
- setiap pembelian sausage, soda maka orang tersebut membeli whole milk
- setiap pembelian semi-finished bread maka orang tersebut membeli whole milk
- setiap pembelian sausage, yogurt maka orang tersebut membeli whole milk

V. Kesimpulan

Dengan dibuatnya analisis ini, saya mendapat berbagai insights untuk market basket analysis dari data set Groceries.csv, diantaranya setiap pembelian kebutuhan pangan sehari-hari seperti sausage, semi-finished bread, yogurt maka orang-orang akan membeli whole milk juga. fakta tersebut benar adanya karena dilihat dari hasil lift, support dan confidence setelah dibandingkan dengan minimal thresholdnya masing-masing menghasilkan nilai yang lebih besar.

Dengan adanya hasil analisis ini, groceries dapat melakukan promo bundling dilihat dari hasil aturan asosiasi terbaik.

Lampiran

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 21 09:12:21 2022

@author: Adira Islah (6181801078)
"""

import time
import pandas as pd
from apyori import apriori
from datetime import datetime
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import fpgrowth
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import association_rules

# load data groceries
df = pd.read_csv("Groceries.csv")

df.dtypes

#data processing
df['itemDescription'] = df['itemDescription'].str.strip()
df['Member_number'] = df['Member_number'].astype('str')

df = df.reset_index()
basket =
df.pivot_table(index=('Member_number', 'Date'), columns='itemDescription', values='index', aggfunc='count', fill_value=0)

def encode_units3(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units3)

#Applying Apriori
#Frequent item
apriori_start = datetime.now()
frequent_itemset = apriori(basket_sets, min_support=0.001,
use_colnames=True)
frequent_itemset_sort =
frequent_itemset.sort_values(['support'], ascending=[False, False])
apriori_end = datetime.now()
```



```

running_time = apriori_end-apriori_start
print("Apriori Running Time: ", str(running_time))

#Association Rules
apriori_start = datetime.now()
rules_apriori = association_rules(frequent_itemset, metric="lift",
min_threshold=0.7)
rules_apriori[ (rules_apriori['lift'] >= 0.7) &
(rules_apriori['support'] >= 0.001)]
rules_apriori = rules_apriori.sort_values(['confidence', 'lift'],
ascending = [False, False])
apriori_end = datetime.now()
running_time = apriori_end-apriori_start
print("Apriori Running Time: ", str(running_time))

#Applying FP-Growth
#Frequent Item
fpgrowth_start = datetime.now()
fpgrowth_frequent_itemsets = fpgrowth(basket_sets,
min_support=0.001, use_colnames=True)
fpgrowth_end = datetime.now()
running_time = fpgrowth_end-fpgrowth_start
print("FP-Growth Running Time: ", str(running_time))

#asc_rules_2 = association_rules(fpgrowth_frequent_itemsets,
metric="confidence", min_threshold=0.2)
fpgrowth_start = datetime.now()
fprules = association_rules(fpgrowth_frequent_itemsets,
metric="lift", min_threshold=0.7)
fprules = fprules[ (fprules['support'] >= 0.001) &
(fprules['lift'] > 1) ]
fprules = fprules.sort_values(['support', 'lift'], ascending =
[False, False])
apriori_end = datetime.now()
running_time = fpgrowth_end-fpgrowth_start
print("Apriori Running Time: ", str(running_time))
~~~~00o~~~

```