

LAPORAN EKSPERIMEN

Pengantar Penambangan Data dengan Python

Apriori dan FP-Growth
Dataset: OnlineRetail.csv

Dibuat oleh:

6181801078/Adira Islah
6181801057/Catherine Elizabeth Josephine

Tanggal:
18 Desember 2021

Universitas Katolik Parahyangan

I. Tujuan Eksperimen

Tujuan dari eksperimen ini adalah untuk pencarian frequent pattern dan aturan asosiasi yang menarik untuk Benua Eropa, Benua Asia dan Benua Australia dengan algoritma Apriori dan FP-Growth.

II. Deskripsi Dataset dan Penyiapan Data

Nama Data : OnlineRetail.csv

Ukuran File: 44 KB

Jumlah Record: 541909 baris

Jumlah Atribut: 8 atribut

Tabel berisi nama dan tipe data atribut pada dataset:

Nama Atribut	Tipe Data
InvoiceNo	object
StockCode	object
Description	object
Quantity	int64
InvoiceDate	object
UnitPrice	float64
CustomerID	float64
Country	object

Keadaan data: Terdapat nilai Null dan value yang bernilai minus, sehingga harus dibersihkan terlebih dahulu.

Penyiapan Data

- Mengganti nilai pada atribut Country yang sebelumnya EIRE menjadi Ireland, Channel Island menjadi Guernsey.
- Mengecek apakah terdapat null dalam data, jika ada drop baris yang memiliki missing value
- Menghilangkan baris yang memiliki nilai minus di atribut Quantity
- Menghapus baris yang memiliki nilai Unspecified di atribut Country

III. Metode Eksperimen

- Melakukan penyiapan data dengan melakukan pembersihan data di beberapa atribut yang memiliki nilai null dan nilai minus
- Melakukan pencarian aturan asosiasi per benua (Eropa, Australia dan Asia) dengan algoritma Apriori dan FP-Growth

IV. Hasil Eksperimen

Australia dengan mengambil 5 rule terbaik

minsup = 0.07

min threshold = 1

lift = 3

confidence = 0.5

Antecedents	Consequents	Antecedents Support	Consequents Support	Confidence	Lift
36 PENCILS TUBE RED RETROSPOT	RED RETROSPOT CAKE STAND	0.07	0.07 0.07	1	14
SET OF 4 PANTRY JELLY MOULDS	FELTCRAFT DOLL MOLLY	0.07	0.07 0.07	1	14
SET OF 12 FAIRY CAKE BAKING CASES	SET OF 12 MINI LOAF BAKING CASES	0.07	0.07 0.07	1	14
SET OF 12 FAIRY CAKE BAKING CASES	SET OF 6 SNACK LOAF BAKING CASES	0.07	0.07 0.07 0.07 0.07	1	14
SET OF 12 FAIRY CAKE BAKING CASES	SET OF 6 TEA TIME BAKING CASES	0.07		1	14

Dilihat dari tabel peraturan asosiasi diatas, maka di benua Australia :

- jika seorang membeli Pencils Tube Red Retrospot maka akan membeli Red Retrospot cake stand
- jika seorang membeli SET OF 4 PANTRY JELLY MOULDS maka akan membeli FELTCRAFT DOLL MOLLY
- jika seorang membeli SET OF 12 FAIRY CAKE BAKING CASES maka akan membeli SET OF 12 MINI LOAF BAKING CASES
- jika seorang membeli SET OF 12 FAIRY CAKE BAKING CASES maka akan membeli SET OF 6 SNACK LOAF BAKING CASES
- jika seorang membeli SET OF 12 FAIRY CAKE BAKING CASES maka akan membeli SET OF 6 TEA TIME BAKING CASES

Kode Program

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 20 16:11:50 2021

@author: Adira Islah (6181801078)
         Catherine Elizabeth Josephine (6181801057)
"""

import pandas as pd
import pycountry_convert as pc
from apyori import apriori
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

df = pd.read_csv('D:\Catherine\Semester 7\PPD Python\Tubes part
2\OnlineRetail.csv')
df.head()

df['Country'].value_counts() # melihat ada country apa saja

# untuk cek apakah kota tersebut terdapat kode benuanya tidak
(continent_name)
country_code = pc.country_name_to_country_alpha2("Australia", cn_name_format
= "default")
print(country_code)
continent_name = pc.country_alpha2_to_continent_code(country_code)
print(continent_name)

df= df[~df['Country'].str.contains('Unspecified')] # hapus country bernama
Unspecified

df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df.dropna() # Membersihkan missing value
df = df[(df.Quantity>0)]
```

```

df.describe() #untuk cek min, max, dll

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

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)

frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules = rules.sort_values(['confidence', 'lift'], ascending = [False, False])
# mengurutkan confidence dan lift yang paling tinggi
rules.head()

rules[ (rules['lift'] >= 3) & (rules['confidence'] >= 0.5) ]
rules.head()

```