

TUGAS
MACHINE LEARNING



Dosen Pembimbing :

Dr. Eng Armin Lawi, S.Si.,M.Si

Disusun Oleh :

Eurico Devon Bura Pakilaran

H071191048

FAKULTAS MATEMATIKA & ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

2021/2022

SUMMARY

Association Rules adalah suatu prosedur yang mencari hubungan atau relasi antara 1 item dengan *item* lainnya dalam suatu dataset yang ditetapkan tersebut.. Association Rules ini terbagi atas 2 Tahapan yaitu : dalam mencari kombinasi sering terjadi dan untuk mendefinisikan kondisi yang telah diterapkan data tersebut. Association Rule kegunaan untuk menemukan hubungan penting antar item dalam setiap transaksi, sehingga hubungan tersebut dapat menandakan kuat dari tidaknya suatu aturan dalam asosiasi, Tujuan Association Rule secara simple untuk menemukan keterataan dalam data.

Association Rules Ini digunakan rumus “if” dan “then” jadi seperti IF-THEN yang menggabungkan beberapa items menjadi satu , contoh :

IF A and B THEN C

Association Rules Ini ada lhs dan rhs . Association Rules ini ada 3 bagian yaitu Support, Confidence, dan Lift.

1. Support (Analisis Pola Frekuensi Tinggi)

Support ini mencari kombinasi pada item list yang memenuhi syarat *minimum* dari nilai *support* dalam database, jadi seberapa besar dominasi suatu *itemset* atau *item* dari keseluruhan Transaksi item tersebut. Nilai Support dapat dirumuskan sebagai berikut :

$$support(A) = \frac{jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi}$$

Untuk Nilai 2 Variable item dari support sama seperti rumus diatas :

$$support(A,B) = \frac{\sum Transaksi\ mengandung\ A\ dan\ B}{\sum Total\ Transaksi}$$

Frequent itemset menunjukkan *itemset* yang memiliki frekuensi pada kemunculan lebih dari nilai *minimum* yang telah ditentukan (Φ). Contohnya $\Phi = 6$, maka semua dari *itemset* yang

frekuensi pada muncul lebih dari atau $= 6x$ disebut *frequent*. Himpunan pada *frequent k-itemset* pada dilambangkan dengan simbol " F_k ".

2. Pembentukan aturan Asosiasi (Confidence)

Confidence ini ukuran item yang menunjukkan hubungan antar item secara kondisional (contoh seberapa sering item B dibeli jika orang membeli item A) jadi contohnya kayak pergi ke supermarket, kita mau beli sesuatu misalnya telur, kemudian kita mau bayar dulu si B maka Si A sudah anggap dibeli maka transaksi itu harus memenuhi syarat tertentu untuk dibeli *item* tersebut. Bagaimana cara menemukan solusi dari *Confidence* yaitu harus menemukan dari nilai *Support* yang telah ditentukan nilai tersebut. *Support* yaitu jumlah itemset dpada transaksi tersebut.

3. Korelasi Lift

Lift itu Perbandingan antara *Confidence* dan persentase kasus pada berisi B yang telah diterapkan dalam paramater yang sangat penting selain *Support* dan *Confidence* dalam Association Rules. Pada Proses Mining Apriori ditandai dengan terbentuknya kekuatan hubungan kombinasi dengan *Itemset* dengan alur ukur asosiasi final. Meskipun demikian, untuk mengukur pada suatu valid atau tidaknya pada invalid asosiasi dapat menggunakan *lift ratio* [5]. *Lift ratio* adalah alat ukurnya pada *Support* dan *Confidence*

Lift dapa dirumuskan sebagai berikut :

$$Lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Keterangan:

- $Lift (A,B)$ = Korelasi Antara A dan B
- $P(A \cup B)$ = Jumlah Kemunculan Antara A dan B dibagi dengan Total Transaksi
- $P(A)P(B)$ = Jumlah Kemunculan A di kali dengan jumlah kemunculan B pada total Transaksi

Apabila Kemungkinan dari perhitungan tersebut maka menghasilkan nilai yang dibawah 1 maka dapat dinyatakan Korelasi Negatif (Minus). Sedangkan nilai yang diatas 1 maka

dapat dinyatakan Korelasi Positif. Jika Kemungkinan Nilai Sama dengan 1 Maka tidak ada Korelasi Antara A dan B Tersebut.

A. Apriori Algoritma

Apriori adalah salah satu Algoritma yang Bersifat Klasik Data Mining dimana sudah dikenal dalam melakukan pencarian *frequent itemset* dengan menggunakan Teknik Association Rules. Algoritma Apriori Dapat digunakan *Knowledge* mengenai *frequent itemset* yang sebelumnya, jadi untuk proses informasi selanjutnya. Biasanya Seorang mau konsumen membeli item A,B , Punya kemungkinan 50% dia akan membeli item C, sehingga pola ini sangat signifikan adanya dalam transaksi data tersebut. Apriori ini menggunakan nama barang pada prosesnya dalam item tersebut.

Contoh dalam Kehidupan Sehari hari : {Roti, Mentega} \rightarrow {Susu} {Support = 40, Confidence = 50% }

Contoh Code python Apriori dalam sederhana ada di Upload Github yang berisi file :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori, association_
rules
!pip install apyori
from apyori import apriori
```

Perlu Install Apyori

```
store_data = pd.read_excel('Buahan.xlsx', header=None)
```

```
store_data.head()
```

```
records = []
for i in range (0, num_records) :
    records.append([str(store_data.values[i,j]) for j in range(0,5)])
```

Perlu melihat rentang dari Dataset berapa Max yang telah ditentukan dari range tersebut.

```
association_rules = apriori(records, min_support=0.050, min_confiden
ce = 0.43, min_lift = 3, min_length = 2)
```

```

association_results = list(association_rules)

for item in association_results :
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + " --> " + items[1])

    print("Support : " + str(item[1]))

    print("Confidence : " + str(item[2][0][2]))
    print("Lift : " + str(item[2][0][3]))
    print("-----")
print("Banyaknya Hasil Association Rules : " , len(association_results))

```

B. FP Growth Algoritma

FP Growth ini dari pengembangan Apriori Algoritma . FP Growth Salah satu alternatif pada algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent Itemset*) dalam sebuah kumpulan data .

Disini FP Growth ini menggunakan kode barang pada prosesnya item dengan adanya di scan tersebut sehingga dia menemukan kode barang yang terdapat nama barang ditemukan kode barnag tersebut.

Contoh Code Untuk Sederhana :

```

import numpy as np
import pandas as pd
!pip install mlxtend
!pip install pyfpgrowth
import pyfpgrowth
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder

```

Perlu Install mlxtend dan pyfpgrowth

```
dataset = pd.read_csv('jakarta.csv', header=None)

dataset.head()

transactions = []
for sublist in dataset.values.tolist():
    clean_sublist = [item for item in sublist if item is not np.nan]
    transactions.append(clean_sublist)
patterns = pyfpgrowth.find_frequent_patterns(transactions, 2)

rules = pyfpgrowth.find_frequent_patterns(patterns, 6)

print(patterns)
print(rules)
```