Apriorilliii english.pdf

by Raisa Turnitin

IMPLEMENTING APRIORI ALGORITHM IN SUPERMARKET SALES DATA

1st Jovinka Aphellia Salva
Computer Science Study Program, University of North Sumatra Medan, Indonesia
jovinkav@gmail.com

2nd Najwa Afifi Situmorang
Computer Science Study Program, University of North Sumatra Medan, Indonesia
najwaafifi121@gmail.com

3rd Dewi Sartika Br Ginting S.Kom., M.Kom
Computer Science Study Program, University of North Sumatra
Medan, Indonesia
dewidintingdg90@gmail.com

Abstract - Using association rule mining techniques, we aim to distribute groups of commonly occurring items and generate meaningful association rules that can provide valuable insights into customer purchasing behavior.

The dataset consists of further details such as invoice ID, product line, and payment model and has been pre-processed to simplify the main process. Our analysis reveals that the support value to Fashion accessories (0.178) which is the highest support value among others.

Keywords-association, a priori, sales, data

I. INTRODUCTION

A priori algorithm is the process of extracting information from a database and often generates elements or sets of elements and candidates to form association rule mining to obtain minimum support and minimum confidence values.

For a large enough database, the a priori algorithm will generate a large number of frequently used item/itemset patterns, as it needs to create candidates and keep track of recurring databases.

In this study, we applied the Aprior algorithm to a supermarket store dataset containing supermarket transaction sales data. The main objective of this research is to find common product groups that consumers often buy together. By identifying these buying behaviors, supermarket owners can gain insight into their marketing tactics and sales techniques to improve overall performance.

Before applying the Aprior algorithm, we perform preliminary data mining to understand the characteristics of the dataset and identify important information such as best-selling products, most popular payment methods, and sales by gender and month. The purpose of this step is to gain insights into customer buying patterns and prepare the data for further analysis.

II. THEORY

Apriori is an algorithm for mining target sets and learning association rules iteratively through a relational database. The process involves identifying specific objects that are found repeatedly in the database and then gradually increasing the number of such objects until they are available for analysis**.**

Mining frequently used products and their association rules is achieved through the use of the Apriori algorithm. In general, the apriori algorithm works on

A database that contains a large number of transactions. For example, consumer goods, but at the Grand Bazaar.

Support, confidence, and lift are the three main components in the association data mining process using the apriori algorithm.

The percentage form of the number of occurrences for a particular combination of items is called the Support(s) value.

$$Support, s(X \to Y) = \frac{(X \cup Y)}{N}$$

The importance of support values in association rules is emphasized, as low support levels indicate that associations are rare in the data set (all event data).

Calculating the percentage of the accuracy of the association rules that will be generated is called the Confident(c) value.

$$Confident, c(X \to Y) = \frac{(X \cup Y)}{X}$$

The magnitude of Y is defined as high confidence for events containing X.

Lift Ratio is a parameter that measures the strength of association rules created by support values and beliefs.

III. RESEARCH METHODOLOGY

This research applies the a priori algorithm from Data Mining to extract data related to supermarket sales.

The data used are supermarket sales records that contain information about all sales transactions, such as products purchased, payment methods and customer demographics.

1. *Data and Preprocessing*

1.1. Data Source:

The dataset used comes from a Kaggle CSV file titled "supermarket_sales.csv". The data in this dataset includes various attributes such as transaction ID, city, member, gender, product purchased, price, quantity, date, payment method, and others.

1.2. Data Reading:

The dataset is read using the `pandas` library, `numpy`, `matplotlib.pyplot`, `seaborn`, `itertools` and `warnings` with the following command:

```python
# Import pustaka yang diperlukan
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import warnings
warnings.filterwarnings("ignore")
from mlxtend.frequent_patterns import apriori, association_rules
from tabulate import tabulate
```

1.3. Data Preprocessing:

The data was processed to prepare it for further analysis.

- Converts a date column to datetime type.
- Added a column that separates the day, month, and year information from the date column.
- Delete columns that are not required for analysis.

```python
data['Date'] = pd.to_datetime(data['Date'])

months = ["january", "february", "march", "april", "may", "june", "july",
          "august", "septembre", "octobre", "novembre", "decembre"]

data["day"] = data["Date"].apply(lambda x : x.day)
data['Month_Name'] = data['Date'].dt.month.apply(lambda x: months[x-1])
data["year"] = data["Date"].apply(lambda x : x.year)
```

2. *Exploratory Data Analysis*

2.1. Visualization of Number of Products Purchased

```python
val_counts = dict(data["Product line"].value_counts()[:10])

plt.figure(figsize=(12,6))
sns.barplot(x=list(val_counts.keys()), y=list(val_counts.values()),
            palette="Blues_r")
```

2.2. Visualization of Payment Method Features

```python
payment = dict(data.groupby("Payment")["Product line"]
               .count().sort_values(ascending=False))

explode = [0] * len(payment)
explode[1] = 0.01 if len(payment) > 1 else 0
explode[2] = 0.2 if len(payment) > 2 else 0

plt.figure(figsize=(10, 6))
plt.pie(payment.values(), labels=payment.keys(), explode=explode,
        colors=sns.color_palette("Set2")[:len(payment)], autopct='%.2f%%')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```

2.3. Gender Feature Visualization

```python
gender = dict(data.groupby("Gender")["Product line"]
              .count().sort_values(ascending=False))

plt.figure(figsize=(10,6))
plt.pie(gender.values(), labels=gender.keys(), explode = [0, 0.01],
        colors = sns.color_palette("Set2")[5:7], autopct='%.2f%%')
plt.tight_layout()
plt.legend()
plt.show()
```

2.4. Moon Feature Visualization

```python
months = ["January", "February", "March", "April", "May", "June",
          "July", "August", "September", "October", "November", "December"]
month = dict(data.groupby("Month_Name")["Product line"].count())

ordered_months = ["january", "february", "march", "april", "may", "june",
                  "july", "august", "septembre", "octobre", "novembre", "decembre"]
sorted_month = {k: month[k] for k in ordered_months if k in month}

plt.figure(figsize=(14, 6))
sns.barplot(x=list(sorted_month.keys()), y=list(sorted_month.values()),
            palette="Purples_r")
plt.title('Jumlah Product Line per Bulan')
plt.xlabel('Month')
plt.ylabel('Jumlah Product Line')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

3. *Data Preprocess (for Pattern Data Mining)*

3.1. Summation of Each Purchased Product

```python
val_counts = data["Product line"].value_counts()
val_counts
```

3.2. Transaction ID Appearance

```python
invoices = []
for action in data["Invoice ID"].unique():
    if action not in excluded:
        invoice = data[data["Invoice ID"] == action]['Product line'].tolist()
        if len(invoice) > 0:
            invoices.append(invoice)
```

4. *Implementation of Apriori Algorithm*

```python
from itertools import combinations

def item_counter(data):
    counts = {}
    for action in data:
        for item in action:
            counts[item]=0
    for action in data:
        for item in action:
            counts[item] += 1

    return counts

def remove_non_sup(dic, min_sup):
    non_freq = []
    for k,v in dic.items():
        if v < min_sup:
            non_freq.append(k)
    [dic.pop(key) for key in non_freq]
    return dic

def check_valid_pairs(data, pairs):
    valid_pairs=[]
    for action in data:
        for pair in pairs:
            if all(x in action for x in pair):
                valid_pairs.append(pair)
    return list(set(valid_pairs))
```

5. *Retrieving the Final Result and Comparing it with the Original Algorithm/Facts*

5.1. Display TransactionEncorder

```
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
te_ary = te.fit(invoices).transform(invoices)
df = pd.DataFrame(te_ary, columns=te.columns_)
df
```

5.2. Formation of Frequent Itemsets

Frequent itemsets are formed using the `apriori` function from the `mlxtend.frequent_patterns` library with a predefined minimum support.

5.3. Formation of Largest Value in Frequent Itemsets

```
frequent_itemsets.nlargest(n = 15, columns = 'support')
```

5.4. Image Visualization of Frequent Itemsets

```
plt.figure(figsize=(12,6))
plt.xticks(rotation=90)
sns.barplot(x='itemsets', y='support', data=frequent_itemsets
            .nlargest(n = 15, columns = 'support'),
            palette="Purples_r")
```

6. Interpretation of Association Rules

```
confidence_association = association_rules(frequent_itemsets,
                         metric='lift', min_threshold=0.2)

confidence_association.head(10)
```

IV. RESULTS AND DISCUSSION

index ▲	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7

Figure 1: Learning Data has been Retrieved and Run

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	T
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	2
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	1
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	2
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	3

Figure 2: Data Preprocessing

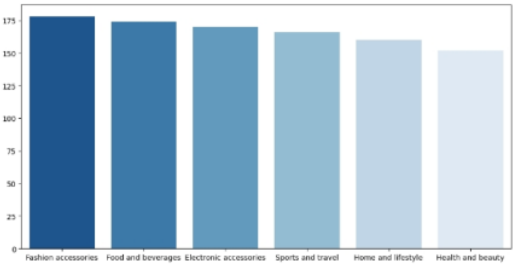


Figure 3: Visualization of Number of Products Purchased

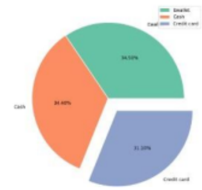
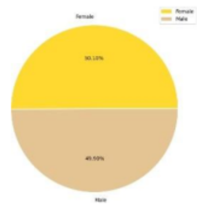


Figure 4: Payment Method Fitue Visualization



\\

Figure 5: Gender Fitue Visualization

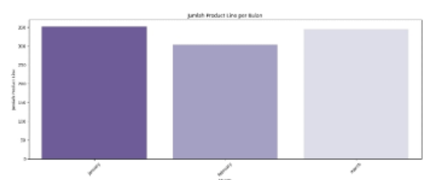


Figure 6: Moon Feature Visualization

```
Product line
Fashion accessories       178
Food and beverages        174
Electronic accessories    170
Sports and travel         166
Home and lifestyle        160
Health and beauty         152
Name: count, dtype: int64
```

Figure 7: Number of Each Product Purchased

	Electronic accessories	Fashion accessories	Food and beverages	Health and beauty	Home and lifestyle	Sports and travel
0	False	False	False	True	False	False
1	True	False	False	False	False	False
2	False	False	False	False	True	False
3	False	False	False	True	False	False
4	False	False	False	False	False	True
...	...	...	...	...	...	...
995	False	False	False	True	False	False
996	False	False	False	False	True	False
997	False	False	True	False	False	False
998	False	False	False	False	True	False
999	False	True	False	False	False	False

1000 rows × 6 columns

	support	itemsets	length
0	17.8	(Fashion accessories)	1
1	17.4	(Food and beverages)	1
2	17.0	(Electronic accessories)	1
3	16.6	(Sports and travel)	1
4	16.0	(Home and lifestyle)	1

Figure 9: 1 Itemset Support Table

	support	itemsets
1	0.178	(Fashion accessories)
2	0.174	(Food and beverages)
0	0.170	(Electronic accessories)
5	0.166	(Sports and travel)
4	0.160	(Home and lifestyle)
3	0.152	(Health and beauty)

Figure 10: Support Table

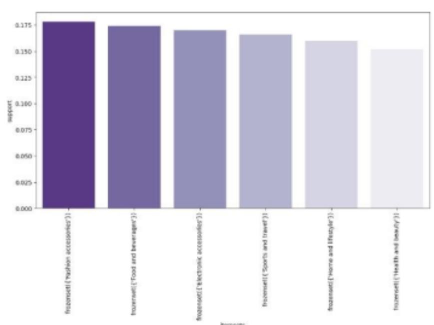


Figure 11: Figure drawing of Support

V. CONCLUSIONS

We use the Aprior algorithm to apply recurring pattern mining analysis to event sales supermarket data in this study. Functions such as "item counter", "*remove_non_sup*", "*check_valid_pairs*", "*pair_counter*", "*unique_elements*", and "apriori" created to implement the Apriori algorithm manually.

This analysis shows common clusters or groups of products that are frequently purchased by supermarket customers in large quantities in different time periods (e.g. 2, 3, 4, 5 items out of stock). Item sets with item lengths of 2 and 3 represent the most frequent purchase patterns observed in the transaction data. This research not only looks for the number of frequently viewed items, but also calculates the trust value between specific items or product groups. Trust is calculated using the equity_on_items and equity_on_sets functions. These indicate how often a product or product line is purchased when another product or product line is purchased in the same transaction.

The results of model mining and continuous trust analysis help supermarkets understand consumer purchasing behavior. This information can be used in marketing strategies, campaigns, product placement or new product development to meet customer preferences. Overall, this research shows that Aprior algorithm can find shopping patterns that are useful for business decision making through pattern mining analysis on supermarket transaction data.

VI. LITERATURE

https://en-m-wikipedia-org.translate.goog/wiki/Apriori_algorithm?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc

https://www-javatpoint-com.translate.goog/apriori-algorithm?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc

https://colab.research.google.com/drive/13bAvG56tL1kY113duD9pCNr1s1qXFFUB?authuser=0#scrollTo=tnSj_c8nI311

Aprioriiiii english.pdf

ORIGINALITY REPORT

6%	5%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	e-jurnal.lppmunsera.org Internet Source	3%
2	uhra.herts.ac.uk Internet Source	1%
3	Farshad Falahati, Daniel Ferreira, J-Sebastian Muehlboeck, Maria Eriksdotter, Andrew Simmons, Lars-Olof Wahlund, Eric Westman. "P2-270: Longitudinal Investigation of an MRI-Based Alzheimer's Disease Diagnostic Index in Adni", Alzheimer's & Dementia, 2016 Publication	1%
4	fasilkom-ti.usu.ac.id Internet Source	1%
5	patents.justia.com Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		