

Лабораторна робота № 8. Дослідження алгоритмів асоціативних правил: Apriori, FP-Growth

Мета: сформулювати практичні навички у використанні алгоритмів асоціативних правил, таких як Apriori та FP-Growth, для аналізу великих даних і виявлення сильних асоціативних зв'язків між різними елементами у базі даних транзакцій. Ця лабораторна робота має на меті розвинути здатність здобувачів аналізувати реальні або синтетичні набори даних, виявляти значущі зв'язки між елементами та використовувати отримані знання для прийняття обґрунтованих рішень у сфері маркетингу, рекомендаційних систем, управління запасами та інших областях.

Теоретичні відомості

Асоціативні правила є одним з основних інструментів аналізу даних у машинному навчанні та інтелектуальному аналізі даних, які дозволяють виявляти важливі зв'язки, асоціації та кореляції між різними елементами в наборах даних. Правила можуть бути використані для виявлення закономірностей покупок у роздрібній торгівлі, для аналізу поведінки користувачів на веб-сайтах, у медицині для виявлення комбінацій симптомів та лікувань тощо.

Apriori – класичний алгоритм для пошуку асоціативних правил, що працює шляхом виявлення наборів елементів, які з'являються разом у транзакціях з частотою вище встановленого порога (порога мінімальної підтримки). **Apriori** використовує ітеративний підхід, де на кожному кроці генеруються кандидати з більшою кількістю елементів на основі наборів, що пройшли попередній крок, та перевіряються на мінімальну підтримку.

FP-Growth (Frequent Pattern Growth) – це ефективніший алгоритм для пошуку асоціативних правил, який не потребує «кандидатської генерації» (процесу створення можливих наборів елементів (кандидатів), які можуть задовольняти встановлені критеріїв), як **Apriori**. Замість цього, **FP-Growth** використовує структуру даних, відому як FP-дерево, для зберігання частотної інформації про елементи. Алгоритм працює в два кроки: спочатку будується FP-дерево, а потім, використовуючи це дерево, виявляються часті набори елементів.

Обидва алгоритми мають свої переваги та недоліки. **Apriori** простий у розумінні та

реалізації, але може бути неефективним на великих наборах даних через велику кількість сканувань бази даних і генерації кандидатів. **FP-Growth**, водночас, є значно швидшим за рахунок зменшення кількості сканувань бази даних та відсутності необхідності генерації кандидатських наборів, проте він вимагає більше пам'яті для зберігання FP-дерева.

Важливими поняттями при роботі з асоціативними правилами є підтримка (**support**), впевненість (**confidence**) і підйом (**lift**). Підтримка вказує, наскільки часто набір елементів з'являється в транзакціях; впевненість показує, наскільки часто правила застосовні, а підйом оцінює, наскільки асоціація між елементами сильніша, ніж випадково.

Приклад виконання роботи

Завдання

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Виконайте згрупування даних по транзакціям та перетворіть транзакції для подальшого аналізу.
3. Використайте алгоритми пошуку асоціативних правил: 1) Apriori; 2) FP-Growth. Виконайте оцінку та візуалізуйте результати у вигляді графу.
4. Сформулюйте рекомендації на основні виявлених асоціативних правил.
5. Проаналізуйте отримані результати.

Контрольне запитання №1



Вкажіть максимальне значення support для Apriori.

Відповідь: _____

Контрольне запитання №2



Вкажіть максимальне значення confidence для FP-Growth.

Відповідь: _____

Контрольне запитання №3



Вкажіть кількість виявлених правил при support = 0.10; confidence = 0.3.

Відповідь: _____

Хід роботи

1. Збережемо та імпортуємо набір даних за варіантом.

	A	B	C	D	E	F	G
1	Species	Weight	Length1	Length2	Length3	Height	Width
2	Bream	242	23.2	25.4	30	11.52	4.02
3	Bream	290	24	26.3	31.2	12.48	4.3056
4	Bream	340	23.9	26.5	31.1	12.3778	4.6961
5	Bream	363	26.3	29	33.5	12.73	4.4555
6	Bream	430	26.5	29	34	12.444	5.134
7	Bream	450	26.8	29.7	34.7	13.6024	4.9274

Рисунок 7.1. Датасет на Google Drive

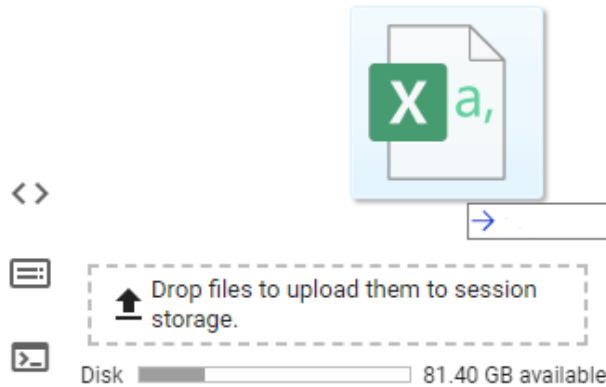


Рисунок 7.2. Завантаження датасету до Google Colab

2. Встановлення необхідних бібліотек та модулів

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules, fpgrowth
from mlxtend.preprocessing import TransactionEncoder
import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns
import ipywidgets as widgets
from IPython.display import display
```

3. Ініціалізація набору даних в Google Colab.

```
# Завдання №1. # Завантаження датасету
data = pd.read_csv("/content/33.csv", names=['items'], header=None)
data.head()
```

	items
0	MILK,BREAD,BISCUIT
1	BREAD,MILK,BISCUIT,CORNFLAKES
2	BREAD,TEA,BOURNVITA
3	JAM,MAGGI,BREAD,MILK
4	MAGGI,TEA,BISCUIT

Рисунок 7.3. Перші 5 рядків набору даних

4. Виконаємо згрупування даних по транзакціям та перетворення транзакцій

```
# Групування даних по транзакціям
transactions = data['items'].str.split(',')
```

```
# Перетворення транзакцій
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_transformed = pd.DataFrame(te_ary, columns=te.columns_)

transactions.head()
```

```
0          [MILK, BREAD, BISCUIT]
1  [BREAD, MILK, BISCUIT, CORNFLAKES]
2          [BREAD, TEA, BOURNVITA]
3          [JAM, MAGGI, BREAD, MILK]
4          [MAGGI, TEA, BISCUIT]
Name: items, dtype: object
```

Рисунок 7.4. Згруповані та перетворені транзакції

5. Підготуємо логіку для використання алгоритмів пошуку асоціативних правил Apriori та FP-Growth.

```
# Віджет для вибору алгоритму асоціативних правил
algorithm_widget = widgets.Dropdown(
    options=['Apriori', 'FP-Growth'],
    description='Algorithm:',
)

# Віджет для встановлення мінімальної підтримки
# % появи об'єкту у транзакціях
min_support_widget = widgets.FloatSlider(
    value=0.05,
    min=0.01,
    max=1,
    step=0.01,
    description='Min Support:',
)

# Віджет для вибору продукту
product_widget = widgets.Dropdown(options=te.columns_, description='Select Product:')
recommend_button = widgets.Button(description="Recommend")
recommendation_output = widgets.Output()

min_threshold_widget = widgets.FloatSlider(
    value=0.2,
    min=0.1,
    max=1,
    step=0.1,
    description='Min Threshold:',
)
```

```

build_button = widgets.Button(description="Build Rules")

output = widgets.Output()

def visualize_rules(rules):
    G = nx.DiGraph()
    labels_map = {}
    label_counter = 1

    for index, row in rules.iterrows():
        antecedents = tuple(row['antecedents'])
        consequents = tuple(row['consequents'])
        confidence = row['confidence']

        for antecedent in antecedents:
            if antecedent not in labels_map:
                labels_map[antecedent] = str(label_counter)
                label_counter += 1
            for consequent in consequents:
                if consequent not in labels_map:
                    labels_map[consequent] = str(label_counter)
                    label_counter += 1
            G.add_edge(labels_map[antecedent], labels_map[consequent],
confidence=f"{confidence:.2f}")

    pos = nx.spring_layout(G)
    edges = G.edges(data=True)
    nx.draw_networkx_nodes(G, pos, node_size=700)
    nx.draw_networkx_edges(G, pos, edgelist=edges, edge_color='black')
    nx.draw_networkx_labels(G, pos)
    edge_labels = dict(((u, v), d['confidence']) for u, v, d in edges)
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)
    plt.title('Association Rules')
    plt.axis('off')
    plt.show()

    print("Label mapping:")
    for item, label in labels_map.items():
        print(f"{label}: {item}")

def build_rules(button):
    global rules
    with output:
        output.clear_output()
        min_support = min_support_widget.value
        min_threshold = min_threshold_widget.value
        algorithm = algorithm_widget.value

```

```

        if algorithm == 'Apriori':
            frequent_itemsets = apriori(df_transformed,
min_support=min_support, use_colnames=True)
        elif algorithm == 'FP-Growth':
            frequent_itemsets = fpgrowth(df_transformed,
min_support=min_support, use_colnames=True)

        rules = association_rules(frequent_itemsets, metric="confidence",
min_threshold=min_threshold)

        if not rules.empty:
            print(f"Found {len(rules)} rules:")
            print(rules[['antecedents', 'consequents', 'support',
'confidence', 'lift']])

        else:
            print("No rules found with the given parameters.")

build_button.on_click(build_rules)

display(widgets.VBox([algorithm_widget, min_support_widget,
min_threshold_widget, build_button, output]))

```

Використаємо алгоритм пошуку асоціативних правил Apriori з параметрами Min Support = 0.12, Min threshold = 0.40.

Algorithm:

Apriori

Min Support: 0.12

Min Thresh... 0.40

Build Rules

Found 15 rules:

	antecedents	consequents	support	confidence	lift
0	(BISCUIT)	(BREAD)	0.20	0.571429	0.879121
1	(CORNFLAKES)	(BISCUIT)	0.15	0.500000	1.428571
2	(BISCUIT)	(CORNFLAKES)	0.15	0.428571	1.428571
3	(BOURNVITA)	(BREAD)	0.15	0.750000	1.153846
4	(MAGGI)	(BREAD)	0.15	0.600000	0.923077
5	(MILK)	(BREAD)	0.20	0.800000	1.230769
6	(SUGER)	(BREAD)	0.20	0.666667	1.025641
7	(TEA)	(BREAD)	0.20	0.571429	0.879121
8	(COCK)	(COFFEE)	0.15	1.000000	2.500000
9	(COFFEE)	(CORNFLAKES)	0.20	0.500000	1.666667
10	(CORNFLAKES)	(COFFEE)	0.20	0.666667	1.666667
11	(COFFEE)	(SUGER)	0.20	0.500000	1.666667
12	(SUGER)	(COFFEE)	0.20	0.666667	1.666667
13	(TEA)	(MAGGI)	0.20	0.571429	2.285714
14	(MAGGI)	(TEA)	0.20	0.800000	2.285714

Рисунок 7.5. Результат пошуку асоціативних правил за алгоритмом Apriori

Використаємо алгоритм пошуку асоціативних правил FP-Growth з параметрами
Min Support = 0.15, Min threshold = 0.60.

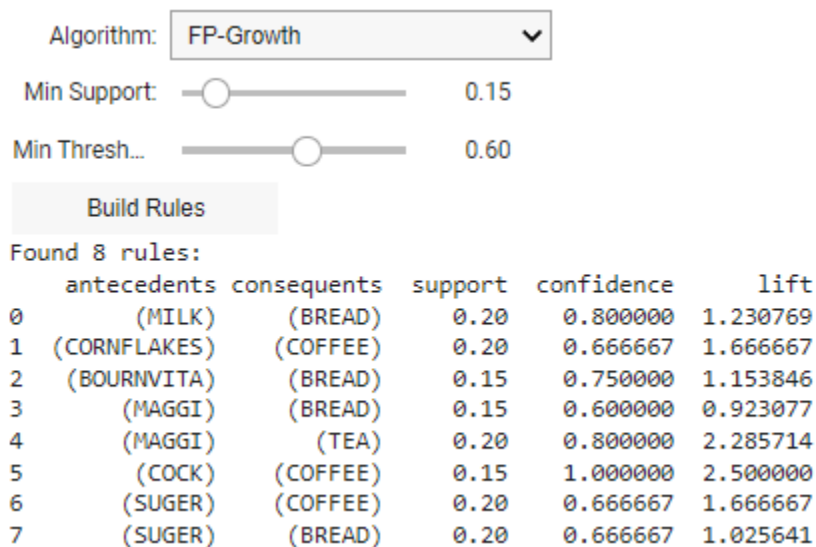


Рисунок 7.6. Результат пошуку асоціативних правил за алгоритмом FP-Growth

6. Виконаємо побудову графів для Apriori та FP-Growth

```
visualize_rules(rules)
```

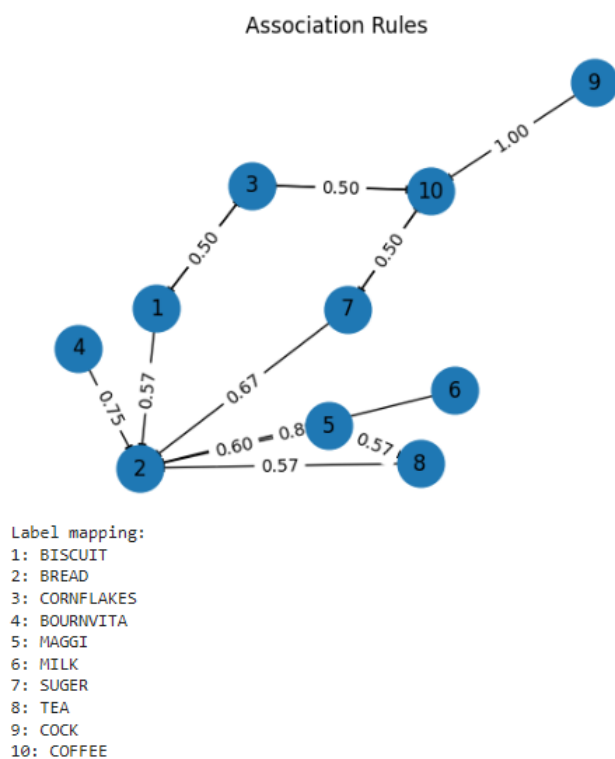


Рисунок 7.6. Граф асоціативних правил для Apriori

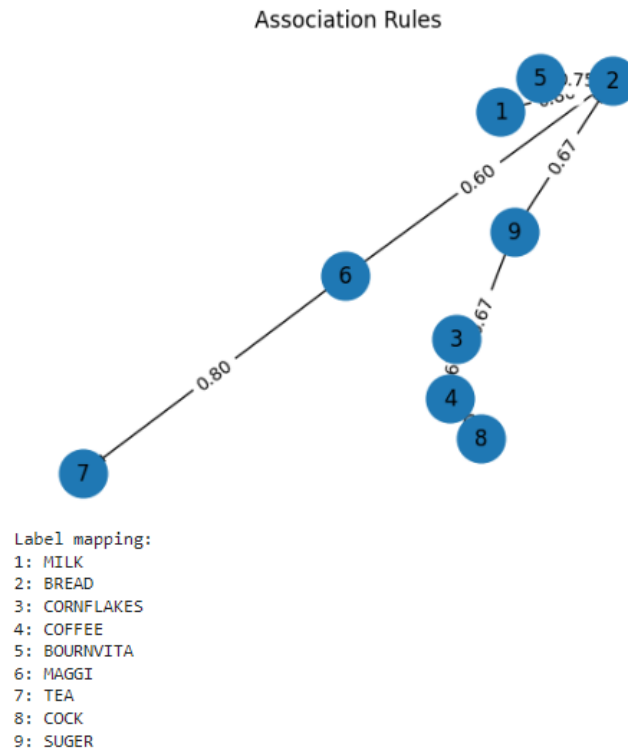


Рисунок 7.7. Граф асоціативних правил для FP-Growth

7. Виконаємо процес надання рекомендацій відповідно до моделі Apriori та FP-Growth

```
# Функція для отримання рекомендацій
def get_recommendations(button):
    with recommendation_output:
        recommendation_output.clear_output()
        selected_product = product_widget.value
        recommendations_set = set()

        recommended = rules[rules['antecedents'].apply(lambda x:
selected_product in x)]['consequents']

        if not recommended.empty:
            print(f"Recommendations for {selected_product}:")
            for index, item in recommended.iteritems():
                recommendations_set.update(item)

            for item in recommendations_set:
                print(f"- {item}")
        else:
            print(f"No recommendations found for {selected_product}")

recommend_button.on_click(get_recommendations)

display(widgets.VBox([product_widget, recommend_button,
recommendation_output]))
```

Select Prod... COCK

Recommend

Recommendations for COCK:

- COFFEE

Рисунок 7.8. Рекомендації моделі Apriori для «COCK»

Select Prod... SUGER

Recommend

Recommendations for SUGER:

- COFFEE
- BREAD

Рисунок 7.9. Рекомендації моделі FP-Growth для «COCK»

Таким чином, аналізуючи отримані дані, можна зазначити, що використання моделі SVM продемонструвало найкращі результати за показником Accuracy (0.9362). Моделі Decision Trees (0.7021) та Logistic Regression (0.7234) надали близькі один до одного результати за показником Accuracy, що є достатнім та дозволяє виконувати задачу класифікації для даного набору даних. Проте зі стандартним налаштуванням гіперпараметрів модель Decision Trees надала некоректний результат класифікації на довільному прикладі. При збільшенні значення гіперпараметрів, модель почала надавати більшу точність (0.8085) та правильно класифікувала приклад. Що стосується моделей K-NN (0.5106) та Naïve Bayes (0.5745), то їх показники точності є незадовільними та не дозволяють якісно виконувати класифікацію на заданому наборі даних, потребують доопрацювання.

Контрольне запитання №1



Вкажіть максимальне значення support для Apriori.

Відповідь: **0.20**

Контрольне запитання №2



Вкажіть максимальне значення confidence для FP-Growth.

Відповідь: **1.00**

Контрольне запитання №3



Вкажіть кількість виявлених правил при support = 0.10; confidence = 0.3.

Відповідь: **90**

Висновки: ...

Завдання для виконання лабораторної роботи

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.

Посилання на датасети					
1	6	11	16	21	26
2	7	12	17	22	27
3	8	13	18	23	28
4	9	14	19	24	29
5	10	15	20	25	30

2. Виконайте згрупування даних за транзакціями та перетворіть транзакції для подальшого аналізу.
3. Використайте алгоритми пошуку асоціативних правил: 1) Apriori; 2) FP-Growth. Виконайте оцінку та візуалізуйте результати у вигляді графу.
4. Сформууйте рекомендації на основні виявлених асоціативних правил.
5. Проаналізуйте отримані результати.

Контрольне запитання №1



Вкажіть максимальне значення support для Apriori.

Відповідь: _____

Контрольне запитання №2



Вкажіть максимальне значення confidence для FP-Growth.

Відповідь: _____

Контрольне запитання №3



Вкажіть кількість виявлених правил при support = 0.10; confidence = 0.3.

Відповідь: _____



Welcome to Google Colab

<https://colab.research.google.com/drive/1fK7wWaaifxKbJrOUftnDuYB88BfnqdIg?usp=sharing>