DWM

Experiment 8

Aim: Implementation of Association Rule Mining algorithm

i) Apriori algorithm

ii) FP Tree algorithm

1)Apriori algorithm

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder

# Define the dataset (TIDs 100-500)
transactions = [
    {'f', 'a', 'c', 'd', 'g', 'i', 'm', 'p'},
    {'a', 'b', 'c', 'f', 'l', 'm', 'o'},
    {'b', 'f', 'h', 'j', 'o', 'w'},
    {'b', 'c', 'k', 's', 'p'},
    {'a', 'f', 'c', 'e', 'l', 'p', 'm', 'n'}
]
N_transactions = len(transactions)

# Convert to Boolean DataFrame
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_transactions = pd.DataFrame(te_ary, columns=te.columns_)

# Set the threshold: min_support = 0.6 (3 out of 5 transactions)
MIN_SUPPORT = 0.6

print(f"Total Transactions (N): {N_transactions}")
print(f"Minimum Support Count required: {N_transactions * MIN_SUPPORT}")
print("-" * 40)
```

```
Total Transactions (N): 5
Minimum Support Count required: 3.0
----------------------------------------
```

```python
# Assuming df_transactions and MIN_SUPPORT are defined correctly

# Run Apriori to generate ALL frequent itemsets.
# The 'min_len' and 'max_len' arguments are REMOVED.
frequent_itemsets_all = apriori(
    df_transactions,
    min_support=MIN_SUPPORT,
    use_colnames=True
)

frequent_itemsets_all['length'] = frequent_itemsets_all['itemsets'].apply(lambda x: len(x))
frequent_itemsets_all['support_count'] = frequent_itemsets_all['support'] * len(df_transactions)

print("Total Frequent Itemsets Found:\n", frequent_itemsets_all[['itemsets', 'length', 'support_count']])
print("-" * 50)
```

```
Total Frequent Itemsets Found:
          itemsets  length  support_count
0              (a)       1            3.0
1              (b)       1            3.0
2              (c)       1            4.0
3              (f)       1            4.0
4              (m)       1            3.0
5              (p)       1            3.0
6           (a, c)       2            3.0
7           (a, f)       2            3.0
8           (a, m)       2            3.0
9           (c, f)       2            3.0
10          (c, m)       2            3.0
11          (c, p)       2            3.0
12          (m, f)       2            3.0
13       (a, c, f)       3            3.0
14       (a, c, m)       3            3.0
15       (a, m, f)       3            3.0
16       (c, m, f)       3            3.0
17    (a, c, m, f)       4            3.0
```

```python
import pandas as pd
from mlxtend.frequent_patterns import apriori
# Assuming df_transactions and MIN_SUPPORT (0.6) are correctly defined

# Run Apriori once to generate ALL frequent itemsets (L1, L2, L3, etc.)
# Note: min_len and max_len parameters are REMOVED.
frequent_itemsets_all = apriori(
    df_transactions,
    min_support=MIN_SUPPORT,
    use_colnames=True
)

# Calculate the size of each itemset and add it as a new column
frequent_itemsets_all['length'] = frequent_itemsets_all['itemsets'].apply(lambda x: len(x))
frequent_itemsets_all['support_count'] = frequent_itemsets_all['support'] * len(df_transactions)

print("Total Frequent Itemsets Found:")
print(frequent_itemsets_all[['itemsets', 'length', 'support_count']].sort_values('length').head(10))
print("-" * 50)
```

```
Total Frequent Itemsets Found:
  itemsets  length  support_count
0      (a)       1            3.0
1      (b)       1            3.0
2      (c)       1            4.0
3      (f)       1            4.0
4      (m)       1            3.0
5      (p)       1            3.0
6   (a, c)       2            3.0
7   (a, f)       2            3.0
8   (a, m)       2            3.0
9   (c, f)       2            3.0
```

2)FPTree

```
...
✅ Part B: FP-Growth Frequent Patterns (Support >= 60%):
    support      itemsets
0       0.8           (c)
1       0.8           (f)
2       0.6           (m)
3       0.6           (p)
4       0.6           (a)
5       0.6           (b)
6       0.6        (c, f)
7       0.6        (m, f)
8       0.6        (c, m)
9       0.6     (c, m, f)
10      0.6        (c, p)
11      0.6        (a, m)
12      0.6        (a, f)
13      0.6        (a, c)
14      0.6     (a, m, f)
15      0.6     (a, c, m)
16      0.6     (a, c, f)
17      0.6  (a, c, m, f)
```