Customer Purchasing Pattern Analysis

CSIT 558

VINAYAK VEMULA

DESCRIPTIVE DATA MINING

Dataset Description

- The *Online Retail II* dataset contains 1.06 million retail transactions from a UK-based store (2009–2011).

- After cleaning missing descriptions and invalid quantities, 1.04 million records remained.

- It includes 8 attributes, such as Invoice, StockCode, Description, Quantity, Price, Customer ID, and Country.

- Top frequent products were one-hot encoded for Apriori.

- Customer purchases were time-sorted for PrefixSpan sequence mining.

```
Dataset loaded successfully!
  Invoice StockCode                          Description  Quantity  \
0  489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS        12
1  489434    79323P                   PINK CHERRY LIGHTS        12
2  489434    79323W                  WHITE CHERRY LIGHTS        12
3  489434     22041         RECORD FRAME 7" SINGLE SIZE         48
4  489434     21232       STRAWBERRY CERAMIC TRINKET BOX        24

           InvoiceDate  Price  Customer ID         Country
0  2009-12-01 07:45:00   6.95      13085.0  United Kingdom
1  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
2  2009-12-01 07:45:00   6.75      13085.0  United Kingdom
3  2009-12-01 07:45:00   2.10      13085.0  United Kingdom
4  2009-12-01 07:45:00   1.25      13085.0  United Kingdom
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067371 entries, 0 to 1067370
Data columns (total 8 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   Invoice      1067371 non-null  object
 1   StockCode    1067371 non-null  object
 2   Description  1062989 non-null  object
 3   Quantity     1067371 non-null  int64
 4   InvoiceDate  1067371 non-null  object
 5   Price        1067371 non-null  float64
 6   Customer ID  824364 non-null   float64
 7   Country      1067371 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 65.1+ MB
None
Columns in dataset:
['Invoice', 'StockCode', 'Description', 'Quantity', 'InvoiceDate', 'Price', 'Customer ID', 'Country']

 Data cleaned successfully!
Number of rows after cleaning: 1042728
```

Objective and Project Goal

Goal: Analyze customer purchasing behavior using descriptive data mining.

- Identify frequently co-purchased items
- Discover strong association rules
- Identify sequential patterns in customer transactions
- Use Apriori/FPGrowth for association rules and PrefixSpan for sequence mining
- Results help in product bundling, inventory planning, and recommendation strategies

Data Preprocessing

- Removing cancelled invoices, missing customer IDs, and negative quantities.
- Selecting the most frequent product descriptions to reduce dimensionality.
- Converting transactions into a basket format and applying one-hot encoding for association rule mining.
- Sorting customer transactions by date to create sequences for PrefixSpan.

```
df = df.dropna(subset=['Invoice', 'StockCode', 'Description'])
df = df[df['Quantity'] > 0]
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'], errors='coerce')

print("Data cleaned successfully!")
print(f"Number of rows after cleaning: {len(df)}")

print("\nSample data:")
print(df.head())
```

```
Data cleaned successfully!
Number of rows after cleaning: 1042728

Sample data:
  Invoice StockCode                          Description  Quantity  \
0  489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS        12
1  489434    79323P                   PINK CHERRY LIGHTS        12
2  489434    79323W                  WHITE CHERRY LIGHTS        12
3  489434     22041         RECORD FRAME 7" SINGLE SIZE         48
4  489434     21232       STRAWBERRY CERAMIC TRINKET BOX        24

          InvoiceDate  Price  Customer ID         Country
0 2009-12-01 07:45:00   6.95      13085.0  United Kingdom
1 2009-12-01 07:45:00   6.75      13085.0  United Kingdom
2 2009-12-01 07:45:00   6.75      13085.0  United Kingdom
3 2009-12-01 07:45:00   2.10      13085.0  United Kingdom
4 2009-12-01 07:45:00   1.25      13085.0  United Kingdom
```

Apriori Experiment 1

Parameters: Support = 0.05 (5%), Confidence = 0.60 (60%)

Purpose: Identify only the strongest and most frequent item associations.

Observations:

- Frequent Itemsets: 27
- Association Rules: 0

```
# Parameter Set A (Strict)
support, confidence = 0.05, 0.6
fis, rules = run_apriori(support, confidence)

print(f"Parameter Set A: Support={support}, Confidence={confidence} ")
print(f"Frequent Itemsets found: {len(fis)}")
print(f"Association Rules found: {len(rules)}\n")
```

Apriori Experiment 1

Effectiveness: This approach is useful when looking for only the strongest and most frequently occurring patterns. However, the thresholds are overly restrictive, resulting in almost no actionable rules and limiting insights for exploratory analysis.

```
***  Parameter Set A: Support=0.05, Confidence=0.6
     Frequent Itemsets found: 27
     Association Rules found: 0

     No rules found for this parameter set.
```

Apriori Experiment 2

Parameters: Support = 0.02 (2%), Confidence = 0.50 (50%)

Purpose: Balance between strictness and discovery of useful rules.

Observations:

- Frequent Itemsets: 317
- Rules: 19
- Avg Lift ≈ 14.69

```
# Parameter Set B (Balanced)
support, confidence = 0.02, 0.5
fis, rules = run_apriori(support, confidence)

print(f"Parameter Set B: Support={support}, Confidence={confidence} ")
print(f"Frequent Itemsets found: {len(fis)}")
print(f"Association Rules found: {len(rules)}\n")

if not rules.empty:
    print("Top 5 Rules (sorted by lift):")
    print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(5))
else:
    print("No rules found for this parameter set.")
```

Apriori Experiment 2

Effectiveness: This setting is well-suited for uncovering actionable co-purchase patterns. It provides strong, interpretable rules that can support bundling strategies and targeted recommendations without overwhelming the analysis with low-frequency noise.

```
Parameter Set B: Support=0.02, Confidence=0.5
Frequent Itemsets found: 317
Association Rules found: 19

Top 5 Rules (sorted by lift):
                                antecedents                                consequents
4      (PINK REGENCY TEACUP AND SAUCER)    (GREEN REGENCY TEACUP AND SAUCER)
5    (GREEN REGENCY TEACUP AND SAUCER)      (PINK REGENCY TEACUP AND SAUCER)
13  (ROSES REGENCY TEACUP AND SAUCER )      (PINK REGENCY TEACUP AND SAUCER)
12     (PINK REGENCY TEACUP AND SAUCER)  (ROSES REGENCY TEACUP AND SAUCER )
7    (ROSES REGENCY TEACUP AND SAUCER )    (GREEN REGENCY TEACUP AND SAUCER)

      support  confidence       lift
4   0.023193    0.834146  22.896479
5   0.023193    0.636634  22.896479
13  0.021837    0.570517  20.518587
12  0.021837    0.785366  20.518587
7   0.027778    0.725726  19.920461
```

Apriori Experiment 3

Parameters: Support = 0.01 (1%), Confidence = 0.30 (30%)

Purpose: Discover large variety of item associations, including less frequent ones.

Observations:

- Frequent Itemsets: 1006
- Rules: 667
- Avg Lift ≈ 13.30

```
# Parameter Set C (Lenient)
support, confidence = 0.01, 0.3
fis, rules = run_apriori(support, confidence)

print(f"Parameter Set C: Support={support}, Confidence={confidence} ")
print(f"Frequent Itemsets found: {len(fis)}")
print(f"Association Rules found: {len(rules)}\n")

if not rules.empty:
    print("Top 5 Rules (sorted by lift):")
    print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(5))
else:
    print("No rules found for this parameter set.")
```

Apriori Experiment 3

Effectiveness: This approach is ideal for comprehensive exploratory analysis. It captures a large number of associations that provide deeper insights into purchasing behavior, though some rules may be weaker and require further filtering for practical use.

```
Parameter Set C: Support=0.01, Confidence=0.3
Frequent Itemsets found: 1006
Association Rules found: 667

Top 5 Rules (sorted by lift):
                                          antecedents  \
645                    (POPPY'S PLAYHOUSE LIVINGROOM )
644  (POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...
642  (POPPY'S PLAYHOUSE LIVINGROOM , POPPY'S PLAYHO...
647                      (POPPY'S PLAYHOUSE BEDROOM )
643  (POPPY'S PLAYHOUSE LIVINGROOM , POPPY'S PLAYHO...

                                          consequents   support  confidence
645  (POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...  0.011095    0.725177
644                    (POPPY'S PLAYHOUSE LIVINGROOM )  0.011095    0.734291
642                      (POPPY'S PLAYHOUSE BEDROOM )  0.011095    0.862869
647  (POPPY'S PLAYHOUSE LIVINGROOM , POPPY'S PLAYHO...  0.011095    0.581792
643                       (POPPY'S PLAYHOUSE KITCHEN)  0.011095    0.887202

         lift
645  47.994499
644  47.994499
642  45.247241
647  45.247241
643  44.077904
```

Analysis of Experiments

Experiment 1: Very strict thresholds resulted in few itemsets and no rules, making the findings too limited for meaningful insights.

Experiment 2: Moderate thresholds produced a balanced number of itemsets and strong rules, offering the best combination of quality and coverage.

Experiment 3: Low thresholds generated many itemsets and rules, providing broad insights but including some weaker associations.

Conclusion: Experiment 2 provides the highest quality rules; Experiment 3 gives widest coverage.

Robustness Analysis

Different random seeds tested: 0, 7, 42, 123

Observations:

- Rules ranged: 19–25
- Mean lift stable: 13.0–14.4

Robustness results across seeds:

	seed	itemsets	rules	mean_lift
0	0	312	19	14.361603
1	7	319	20	14.416897
2	42	333	23	13.431698
3	123	317	25	13.000786

Interpretation: Results are consistent → strong robustness.

Confirms that discovered patterns are not dependent on sampling randomness.

Sequence Mining (PrefixSpan)

Support thresholds tested: 500, 200, 100

Findings:

- Higher support → fewer but stronger patterns
- Lower support → longer sequences detected
- Interpretation: Customers purchase themed items in sequential groups.

…

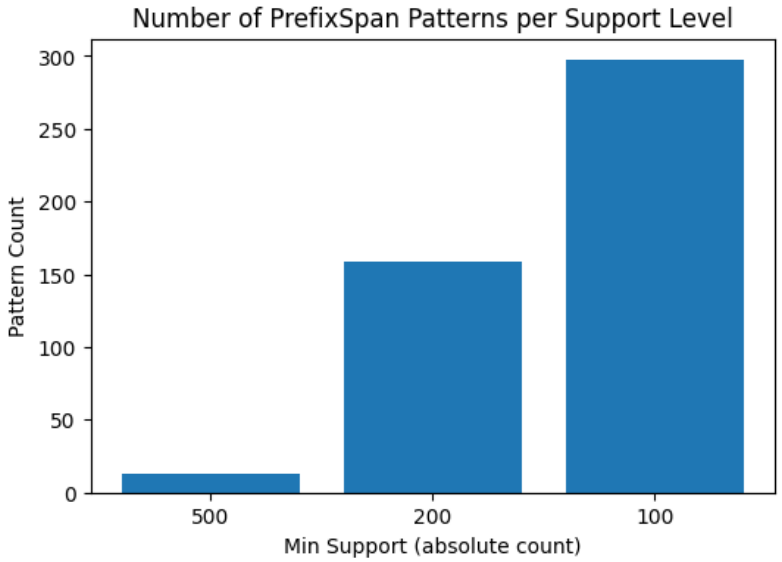
```
 FAST PrefixSpan: min_support = 500
Patterns found: 13
Top 10 patterns:
(1249, ['WHITE HANGING HEART T-LIGHT HOLDER'])
(910, ['REGENCY CAKESTAND 3 TIER'])
(867, ['BAKING SET 9 PIECE RETROSPOT '])
(735, ['ASSORTED COLOUR BIRD ORNAMENT'])
(599, ['REX CASH+CARRY JUMBO SHOPPER'])
(561, ['60 TEATIME FAIRY CAKE CASES'])
(539, ['PARTY BUNTING'])
(536, ['VINTAGE SNAP CARDS'])
(533, ['NATURAL SLATE HEART CHALKBOARD '])
(529, ['PACK OF 72 RETRO SPOT CAKE CASES'])

 FAST PrefixSpan: min_support = 200
Patterns found: 159
Top 10 patterns:
(1249, ['WHITE HANGING HEART T-LIGHT HOLDER'])
(910, ['REGENCY CAKESTAND 3 TIER'])
(867, ['BAKING SET 9 PIECE RETROSPOT '])
(735, ['ASSORTED COLOUR BIRD ORNAMENT'])
(599, ['REX CASH+CARRY JUMBO SHOPPER'])
(561, ['60 TEATIME FAIRY CAKE CASES'])
(539, ['PARTY BUNTING'])
(536, ['VINTAGE SNAP CARDS'])
(533, ['NATURAL SLATE HEART CHALKBOARD '])
(529, ['PACK OF 72 RETRO SPOT CAKE CASES'])

 FAST PrefixSpan: min_support = 100
Patterns found: 297
Top 10 patterns:
(1249, ['WHITE HANGING HEART T-LIGHT HOLDER'])
(910, ['REGENCY CAKESTAND 3 TIER'])
(867, ['BAKING SET 9 PIECE RETROSPOT '])
(735, ['ASSORTED COLOUR BIRD ORNAMENT'])
(599, ['REX CASH+CARRY JUMBO SHOPPER'])
(561, ['60 TEATIME FAIRY CAKE CASES'])
(539, ['PARTY BUNTING'])
(536, ['VINTAGE SNAP CARDS'])
(533, ['NATURAL SLATE HEART CHALKBOARD '])
(529, ['PACK OF 72 RETRO SPOT CAKE CASES'])
```

Insights from Sequential Patterns

- Customers frequently buy themed or related items in sequence.
- Strong sequential trends in Regency-style cups, saucers, and décor items.
- Useful for creating product bundles and improving recommendation flows.



Conclusion

- **Experiment 1** reveals only the most common and reliable behaviors due to strict thresholds, but the insights are limited and lack variety.
- **Experiment 2** provides the best overall balance, producing strong, meaningful, and actionable multi-item rules with solid support and confidence.
- **Experiment 3** uncovers the broadest range of patterns, offering valuable exploratory insights, though some associations are weaker and require careful interpretation.

References

- UCI Machine Learning Repository. (2019). *Online Retail II Dataset*.
- Borgelt, C. (2012). *Frequent Itemset Mining Implementations: Apriori & FPGrowth*.
- Agrawal, R., & Srikant, R. (1995). “Mining Sequential Patterns.”
- *Proceedings of the International Conference on Data Engineering (ICDE).*