NAANMUDHALVAN

AI_PHASE4

Performing Association Rules and Generating Insights

Mohammed kani A
711021106003

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

# Load the dataset
data = pd.read_csv('https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis', header=No
ne)

# Preprocess the data and convert it into a transaction format (list of lists)
transactions = []
for i in range(data.shape[0]):
    transaction = [str(data.values[i, j]) for j in range(data.shape[1]) if not pd.isnull(data.values[i, j])]
    transactions.append(transaction)

# Apply the Apriori algorithm to find frequent item sets
min_support = 0.1  # Adjust this support threshold as needed
frequent_item_sets = apriori(transactions, min_support=min_support, use_colnames=True)

# Generate association rules
min_confidence = 0.7  # Adjust this confidence threshold as needed
rules = association_rules(frequent_item_sets, metric="confidence", min_threshold=min_confidence)

# Display the frequent item sets and association rules
print("Frequent Item Sets:")
print(frequent_item_sets)

print("\nAssociation Rules:")
print(rules)

# Generate insights from the association rules
print("\nInsights:")
for index, row in rules.iterrows():
    antecedents = set(row['antecedents'])
    consequents = set(row['consequents'])
    support = row['support']
    confidence = row['confidence']
    lift = row['lift']

    # Example insights: You can customize these based on your data and domain
```

```
    if len(antecedents) == 1 and len(consequents) == 1:
        print(f"If '{', '.join(antecedents)}' is bought, it's highly likely that '{', '.join(consequents)}' is also bought.
")
        print(f"Support: {support:.2f}, Confidence: {confidence:.2f}, Lift: {lift:.2f}")
    elif len(antecedents) == 1:
        print(f"Customers who buy '{', '.join(antecedents)}' often buy '{', '.join(consequents)}' as well.")
        print(f"Support: {support:.2f}, Confidence: {confidence:.2f}, Lift: {lift:.2f}")
    elif len(consequents) == 1:
        print(f"Customers who buy '{', '.join(consequents)}' often start with '{', '.join(antecedents)}'.")
        print(f"Support: {support:.2f}, Confidence: {confidence:.2f}, Lift: {lift:.2f}")
    else:
        print(f"A pattern found: '{', '.join(antecedents)}' leads to '{', '.join(consequents)}'.")
        print(f"Support: {support:.2f}, Confidence: {confidence:.2f}, Lift: {lift:.2f}")
    print()
```