library installation

```
[1]: !pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\raksh\anaconda3\lib\site-packages (0.24.0)
Requirement already satisfied: scipy>=1.16.3 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (1.17.0)
Requirement already satisfied: numpy>=2.3.5 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (2.4.1)
Requirement already satisfied: pandas>=2.3.3 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (2.3.3)
Requirement already satisfied: scikit-learn>=1.8.0 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (1.8.0)
Requirement already satisfied: matplotlib>=3.10.8 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (3.10.8)
Requirement already satisfied: joblib>=1.5.2 in c:\users\raksh\anaconda3\lib\site-packages (from mlxtend) (1.5.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (11.1.0)
Requirement already satisfied: pyparsing>=3 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\raksh\anaconda3\lib\site-packages (from matplotlib>=3.10.8->mlxtend) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\raksh\anaconda3\lib\site-packages (from pandas>=2.3.3->mlxtend) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\raksh\anaconda3\lib\site-packages (from pandas>=2.3.3->mlxtend) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\raksh\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.10.8->mlxtend) (1.17.0)
Requirement already satisfied: threadpoolctl>=3.2.0 in c:\users\raksh\anaconda3\lib\site-packages (from scikit-learn>=1.8.0->mlxtend) (3.5.0)
```

```
[2]: from mlxtend.frequent_patterns import apriori, association_rules
```

Imports ¶

```
[3]: import pandas as pd
     from mlxtend.frequent_patterns import apriori, association_rules
```

```
[4]:  print("Python is running fine")

      Python is running fine

[6]:  import os
      os.getcwd()

[6]:  'C:\\Users\\raksh'
```

Imports CSV File

```
[7]:  import pandas as pd

      df = pd.read_csv("purchase_pattern_analyses3.csv")
      df.head()
```

[7]:

	BillNo	Itemname	Quantity	Price	CustomerID	Country	Present_Date
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2.55	17850	United Kingdom	2010-12-01 08:26:00
1	536365	WHITE METAL LANTERN	6	3.39	17850	United Kingdom	2010-12-01 08:26:00
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2.75	17850	United Kingdom	2010-12-01 08:26:00
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	3.39	17850	United Kingdom	2010-12-01 08:26:00
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	3.39	17850	United Kingdom	2010-12-01 08:26:00

Load & Clean

```
[8]:  data = df.loc[:, ['BillNo', 'Itemname']].dropna()
```

Apply Apriori Algorithm

```python
frequent_itemsets = apriori(
    basket,
    min_support=0.05,     # HIGH support
    max_len=2,            # 🔥 LIMIT ITEMSET SIZE
    use_colnames=True
)

frequent_itemsets.shape
```

```
(171, 2)
```

Basket

```python
basket = (
    data
    .groupby(['BillNo', 'Itemname'])
    .size()
    .unstack(fill_value=0)
)

basket = basket.gt(0).astype(bool)
```

```
# Select required columns (safe copy)
data = df.loc[:, ['BillNo', 'Itemname']].dropna()

data.head()
```

	BillNo	Itemname
0	536365	WHITE HANGING HEART T-LIGHT HOLDER
1	536365	WHITE METAL LANTERN
2	536365	CREAM CUPID HEARTS COAT HANGER
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE
4	536365	RED WOOLLY HOTTIE WHITE HEART.

```
basket = (
    data
    .groupby(['BillNo', 'Itemname'])
    .size()
    .unstack(fill_value=0)
)

basket.head()
```

Itemname	10 COLOUR SPACEBOY PEN	12 DAISY PEGS IN WOOD BOX	12 MESSAGE CARDS WITH ENVELOPES	12 PENCILS TALL TUBE SKULLS	3 PIECE SPACEBOY COOKIE CUTTER SET	3 STRIPEY MICE FELTCRAFT	3 TIER CAKE TIN GREEN AND CREAM	3 TIER CAKE TIN RED AND CREAM	36 FOIL HEART CAKE CASES	36 FOIL STAR CAKE CASES	...	WOODEN FRAME ANTIQUE WHITE	WOODEN HEART CHRISTMAS SCANDINAVIAN	WOODEN OWLS LIGHT GARLAND	WOODEN PICTURE FRAME WHITE FINISH	
BillNo																
536365	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
536366	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
536367	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
536368	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
536369	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	

5 rows × 607 columns

```
[15]:  basket = basket.astype(bool)

       basket.head()
```

[15]:

Itemname	10 COLOUR SPACEBOY PEN	12 DAISY PEGS IN WOOD BOX	12 MESSAGE CARDS WITH ENVELOPES	12 PENCILS TALL TUBE SKULLS	3 PIECE SPACEBOY COOKIE CUTTER SET	3 STRIPEY MICE FELTCRAFT	3 TIER CAKE TIN GREEN AND CREAM	3 TIER CAKE TIN RED AND CREAM	36 FOIL HEART CAKE CASES	36 FOIL STAR CAKE CASES	...	WOODEN FRAME ANTIQUE WHITE	WOODEN HEART CHRISTMAS SCANDINAVIAN	WOODEN OWLS LIGHT GARLAND	WOODEN PICTURE FRAME WHITE FINISH
BillNo															
536365	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
536366	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
536367	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
536368	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
536369	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False

5 rows × 607 columns

```
[16]:  frequent_itemsets = apriori(
           basket,
           min_support=0.02,   # increase if dataset is large
           use_colnames=True
       )

       frequent_itemsets.head()
```

[16]:

	support	itemsets
0	0.031746	(3 PIECE SPACEBOY COOKIE CUTTER SET)
1	0.031746	(3 STRIPEY MICE FELTCRAFT)
2	0.031746	(36 FOIL HEART CAKE CASES)
3	0.031746	(4 TRADITIONAL SPINNING TOPS)
4	0.031746	(5 HOOK HANGER MAGIC TOADSTOOL)

```python
[14]: basket_small = (
          data_small.groupby(['BillNo', 'Itemname'])['Itemname']
          .count()
          .unstack()
          .fillna(0)
      )

      basket_small = basket_small.astype(bool)
      print(basket_small.shape)
```

(54, 100)

Generate Association Rules

```python
[12]: rules = association_rules(
          frequent_itemsets,
          metric="confidence",
          min_threshold=0.7
      )

      rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head()
```

	antecedents	consequents	support	confidence	lift
0	(CREAM CUPID HEARTS COAT HANGER)	(EDWARDIAN PARASOL RED)	0.063492	0.800000	12.6
1	(EDWARDIAN PARASOL RED)	(CREAM CUPID HEARTS COAT HANGER)	0.063492	1.000000	12.6
2	(GLASS STAR FROSTED T-LIGHT HOLDER)	(CREAM CUPID HEARTS COAT HANGER)	0.079365	1.000000	12.6
3	(CREAM CUPID HEARTS COAT HANGER)	(GLASS STAR FROSTED T-LIGHT HOLDER)	0.079365	1.000000	12.6
4	(KNITTED UNION FLAG HOT WATER BOTTLE)	(CREAM CUPID HEARTS COAT HANGER)	0.079365	0.833333	10.5

```python
[22]:  # Simple Python visualization
       import matplotlib.pyplot as plt

       top_items = (
           frequent_itemsets
           .assign(item_count=frequent_itemsets['itemsets'].astype(str))
           .sort_values('support', ascending=False)
           .head(10)
       )

       plt.barh(top_items['itemsets'].astype(str), top_items['support'])
       plt.xlabel("Support")
       plt.title("Top 10 Frequent Itemsets")
       plt.show()
```

Top 10 Frequent Itemsets
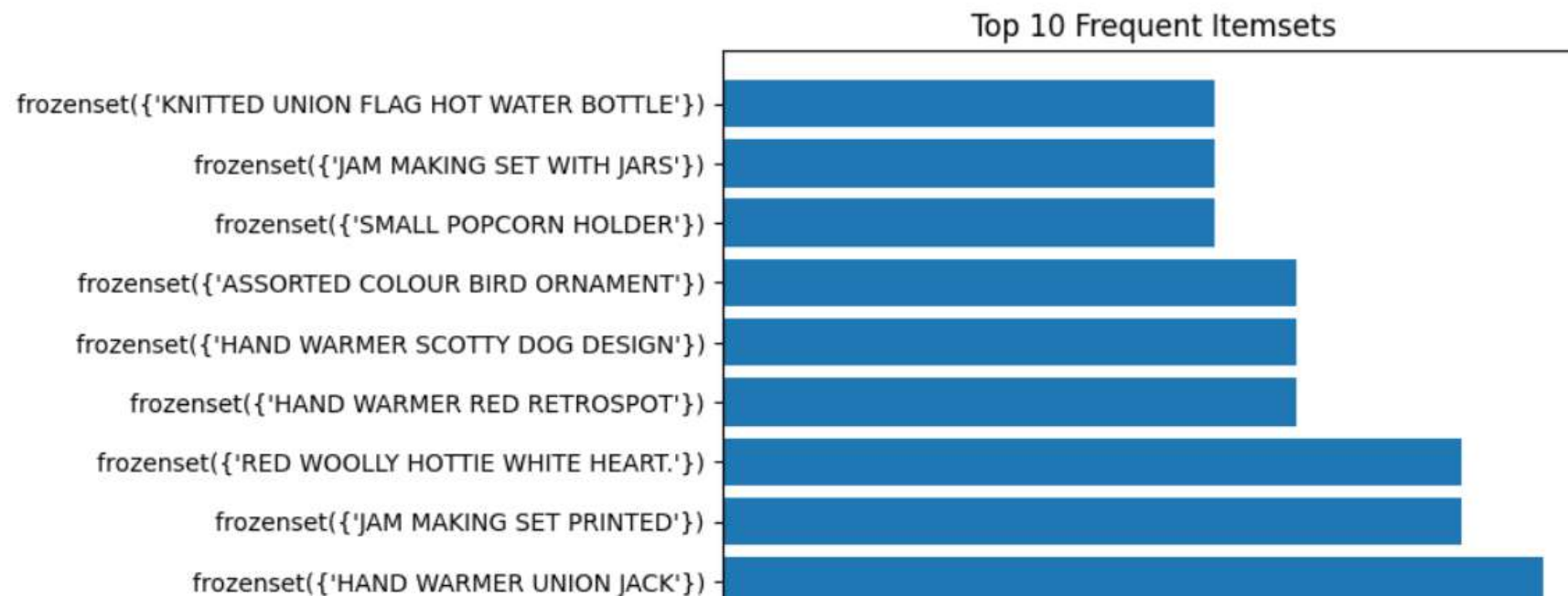
```
[23]:   # Rule strength visualization
        plt.scatter(rules['confidence'], rules['lift'])
        plt.xlabel("Confidence")
        plt.ylabel("Lift")
        plt.title("Association Rules Strength")
        plt.show()
```


Association Rules Strength