ai-phase4

October 31, 2023

[10]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

[11]:
```
!pip install openpyxl
```

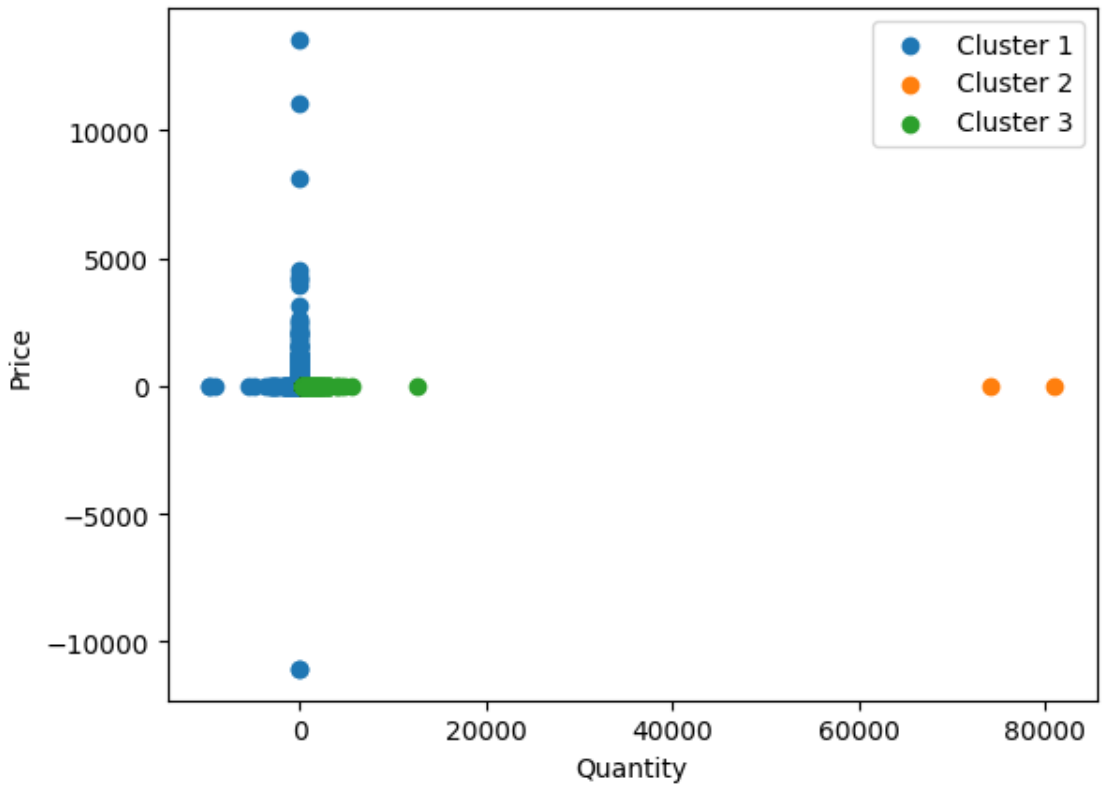
```
Requirement already satisfied: openpyxl in c:\users\d e l
l\appdata\local\programs\python\python310\lib\site-packages (3.1.2)
Requirement already satisfied: et-xmlfile in c:\users\d e l
l\appdata\local\programs\python\python310\lib\site-packages (from openpyxl)
(1.1.0)


[notice] A new release of pip available: 22.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[12]:
```
data = pd.read_excel('Assignment-1_Data.xlsx')

selected_features = data[['Quantity', 'Price']]
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters)
data['Cluster'] = kmeans.fit_predict(selected_features)
```

```
C:\Users\D E L L\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

[13]:
```
for cluster in range(num_clusters):
    cluster_data = data[data['Cluster'] == cluster]
    plt.scatter(cluster_data['Quantity'], cluster_data['Price'],␣
 ↪label=f'Cluster {cluster + 1}')

plt.xlabel('Quantity')
plt.ylabel('Price')
plt.legend()
```

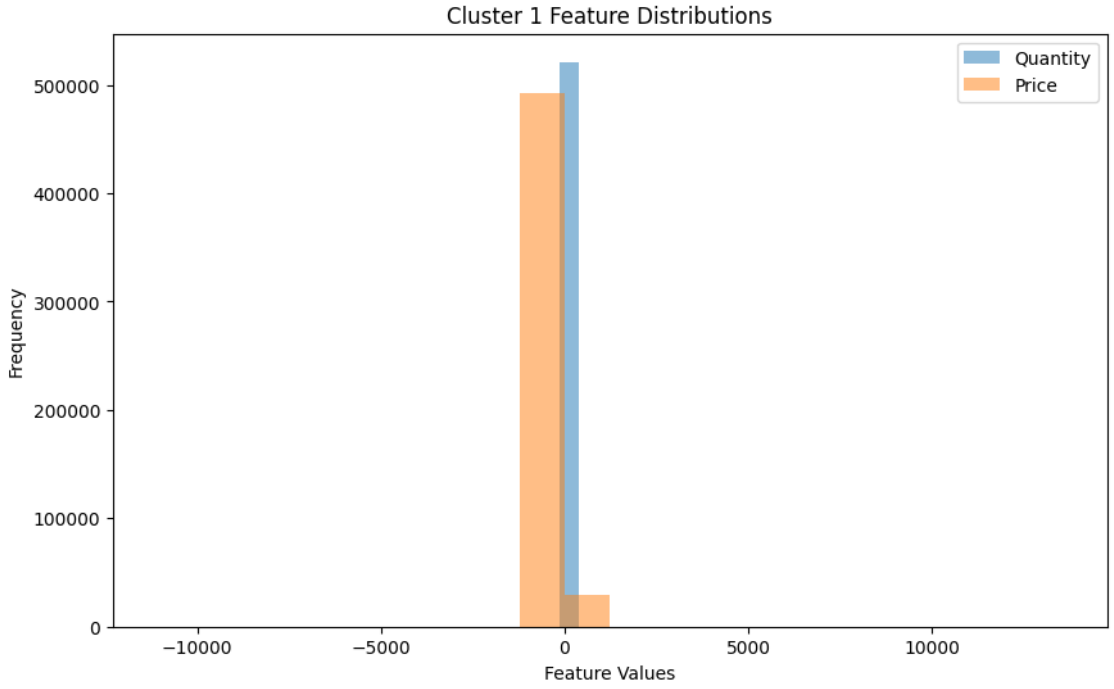
```
plt.show()
```



[14]:
```
for cluster in range(num_clusters):
    cluster_data = data[data['Cluster'] == cluster]
    print(f'Cluster {cluster + 1}:')
    print(cluster_data.describe())
    centroid = kmeans.cluster_centers_[cluster]
    print(f'Centroid for Cluster: {cluster + 1}')
    print(f'Quantity: {centroid[0]}')
    print(f'Price: {centroid[1]}')
    plt.figure(figsize=(10, 6))
    plt.hist(cluster_data['Quantity'], bins=20, alpha=0.5, label='Quantity')
    plt.hist(cluster_data['Price'], bins=20, alpha=0.5, label='Price')
    plt.xlabel('Feature Values')
    plt.ylabel('Frequency')
    plt.title(f'Cluster {cluster + 1} Feature Distributions')
    plt.legend()
    plt.show()
```

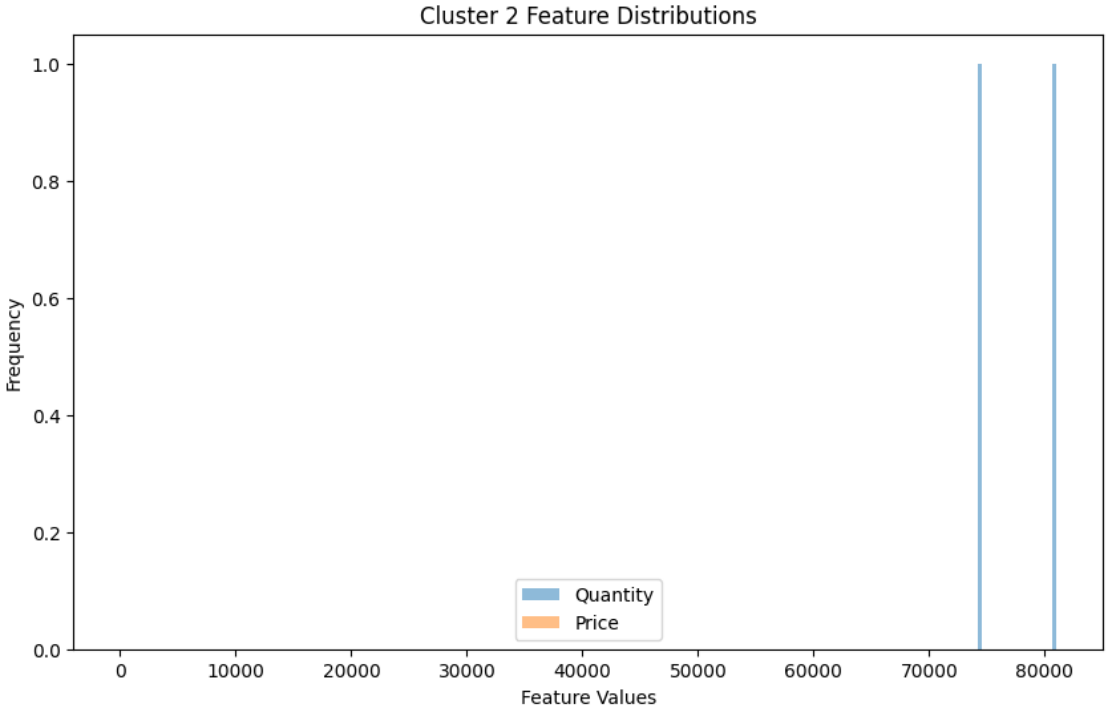
```
Cluster 1:
            Quantity                           Date          Price  \
count  521308.000000                         521308  521308.000000
```

```
mean        8.723591  2011-07-04 12:57:38.102695680       3.830624
min     -9600.000000            2010-12-01 08:26:00  -11062.060000
25%         1.000000            2011-03-28 10:15:00       1.250000
50%         3.000000            2011-07-20 08:59:00       2.080000
75%        10.000000            2011-10-19 14:22:00       4.130000
max       378.000000            2011-12-09 12:50:00   13541.330000
std        36.890530                            NaN      41.930819

          CustomerID   Cluster
count  387303.000000  521308.0
mean    15315.979801       0.0
min     12347.000000       0.0
25%     13950.000000       0.0
50%     15261.000000       0.0
75%     16837.000000       0.0
max     18287.000000       0.0
std      1721.397926       0.0
Centroid for Cluster: 1
Quantity: 8.72359142771426
Price: 3.830624436993143
```



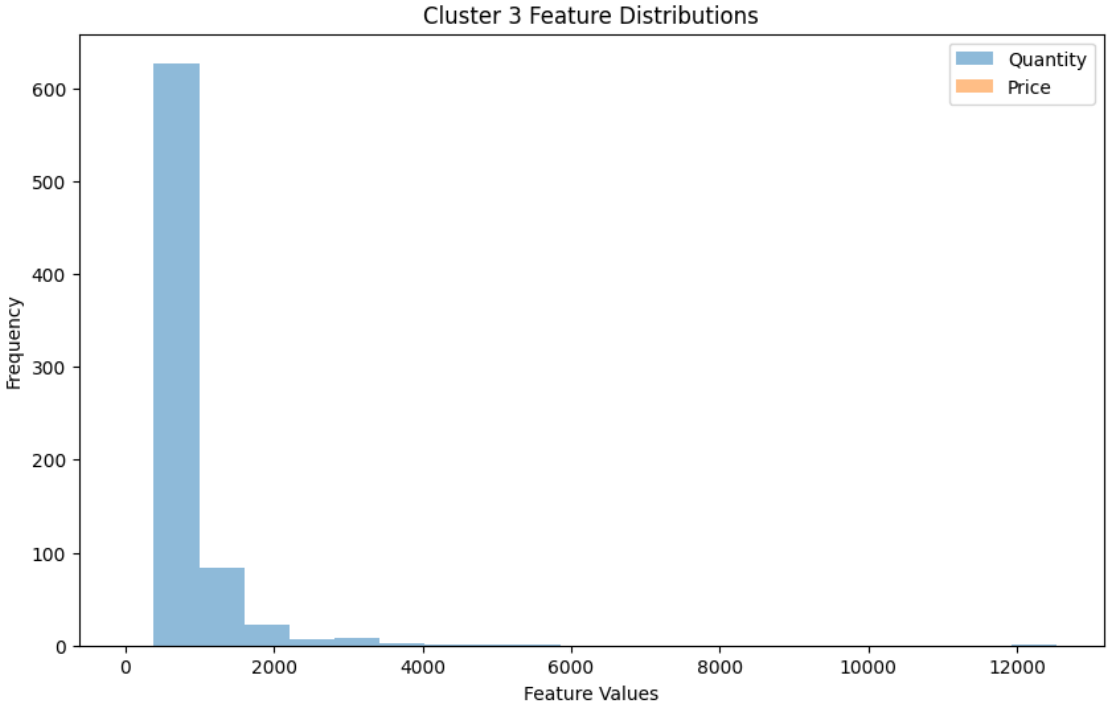

```
Cluster 2:
          Quantity                 Date     Price    CustomerID  Cluster
count      2.000000                    2  2.000000      2.000000      2.0
mean   77605.000000  2011-06-29 21:38:00  1.560000  14396.000000      1.0
```

```
min    74215.000000  2011-01-18 10:01:00  1.040000  12346.000000      1.0
25%    75910.000000  2011-04-09 15:49:30  1.300000  13371.000000      1.0
50%    77605.000000  2011-06-29 21:38:00  1.560000  14396.000000      1.0
75%    79300.000000  2011-09-19 03:26:30  1.820000  15421.000000      1.0
max    80995.000000  2011-12-09 09:15:00  2.080000  16446.000000      1.0
std     4794.183976                  NaN  0.735391   2899.137803      0.0
Centroid for Cluster: 2
Quantity: 77605.0
Price: 1.56
```




```
Cluster 3:
          Quantity                           Date       Price    CustomerID  \
count    754.000000                            754  754.000000    718.000000
mean     749.290451  2011-07-01 12:41:03.740052992    1.189589  15832.974930
min      384.000000            2010-12-01 09:58:00    0.000000  12415.000000
25%      432.000000            2011-04-01 12:25:30    0.360000  14101.000000
50%      576.000000            2011-07-24 12:58:30    0.720000  16333.000000
75%      748.500000            2011-10-05 10:06:00    1.650000  17450.000000
max    12540.000000            2011-12-08 18:46:00    8.150000  18251.000000
std      694.918323                            NaN    1.327377   1879.120714

       Cluster
count    754.0
mean       2.0
min        2.0
```

```
25%         2.0
50%         2.0
75%         2.0
max         2.0
std         0.0
Centroid for Cluster: 3
Quantity: 749.2904509283826
Price: 1.1895888594164465
```



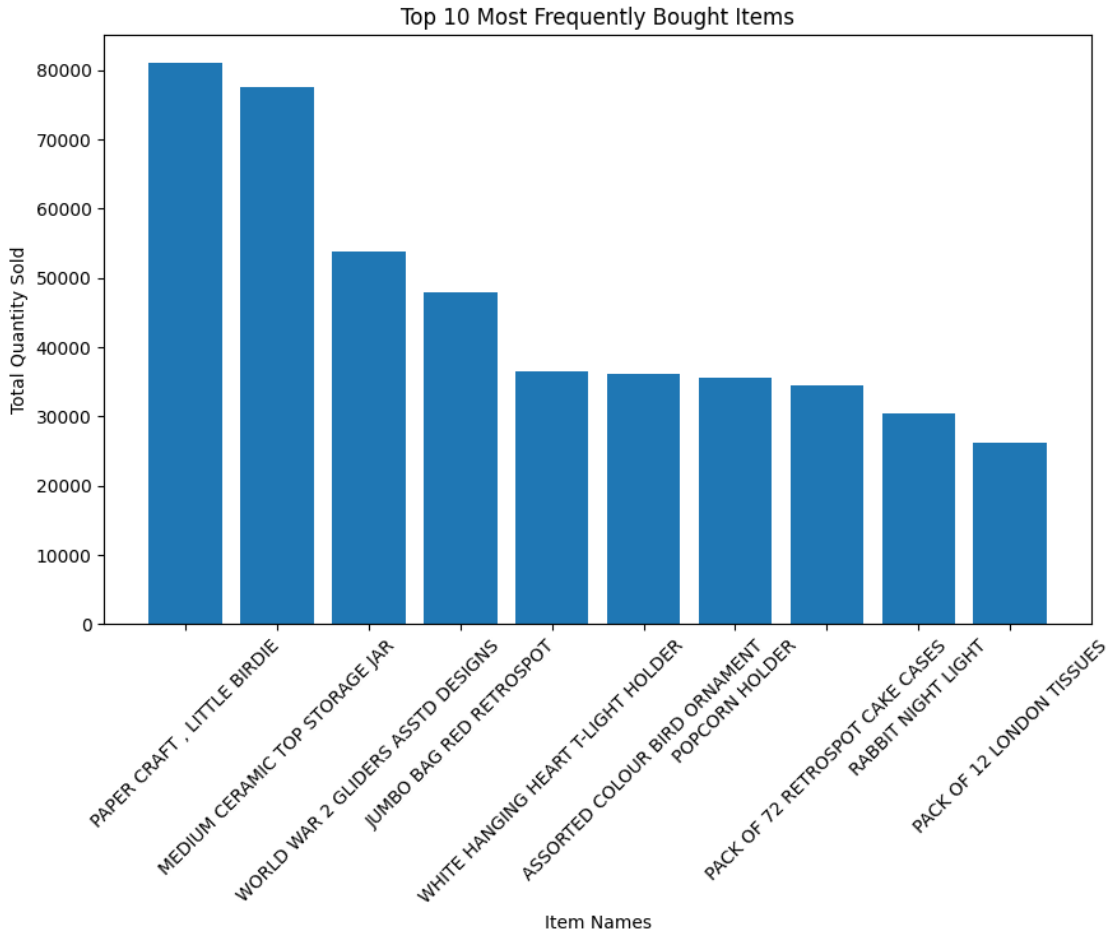
[15]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules


basket = (data.groupby(['BillNo', 'Itemname'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('BillNo'))
basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1))
frequent_itemsets = apriori(basket_sets, min_support=0.01, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)
print("Association Rules:")
print(rules)
```

```
C:\Users\D E L L\AppData\Local\Temp\ipykernel_12816\3169484420.py:10:
FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
instead.
```

```
  basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1))

Association Rules:
                         antecedents  \
0          (JAM MAKING SET PRINTED)
1          (6 RIBBONS RUSTIC CHARM)
2          (6 RIBBONS RUSTIC CHARM)
3        (JAM MAKING SET WITH JARS)
4          (6 RIBBONS RUSTIC CHARM)
…                                …
2999     (STRAWBERRY CHARLOTTE BAG)
3000    (CHARLOTTE BAG SUKI DESIGN)
3001  (RED RETROSPOT CHARLOTTE BAG)
3002  (CHARLOTTE BAG PINK POLKADOT)
3003       (WOODLAND CHARLOTTE BAG)

                                           consequents  antecedent support  \
0                             (6 RIBBONS RUSTIC CHARM)            0.055226
1                             (JAM MAKING SET PRINTED)            0.046615
2                           (JAM MAKING SET WITH JARS)            0.046615
3                             (6 RIBBONS RUSTIC CHARM)            0.053890
4                            (JUMBO BAG RED RETROSPOT)            0.046615
…                                                  …                   …
2999  (RED RETROSPOT CHARLOTTE BAG, CHARLOTTE BAG PI…            0.035432
3000  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.043300
3001  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.050871
3002  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.036520
3003  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.040924

      consequent support   support  confidence       lift  leverage  \
0               0.046615  0.011530    0.208781   4.478826  0.008956
1               0.055226  0.011530    0.247346   4.478826  0.008956
2               0.053890  0.010095    0.216561   4.018599  0.007583
3               0.046615  0.010095    0.187328   4.018599  0.007583
4               0.102138  0.010689    0.229299   2.245001  0.005928
…                  …       …          …         …        …
2999            0.012371  0.010046    0.283520  22.917453  0.009607
3000            0.011926  0.010046    0.232000  19.453344  0.009529
3001            0.010936  0.010046    0.197471  18.056517  0.009489
3002            0.012767  0.010046    0.275068  21.544841  0.009579
3003            0.012074  0.010046    0.245466  20.329375  0.009551

      conviction  zhangs_metric
0       1.204957       0.822130
1       1.255257       0.814705
2       1.207637       0.787884
3       1.173148       0.793942
4       1.164995       0.581681
```

```
...          ...          ...
2999    1.378445     0.991495
3000    1.286555     0.991528
3001    1.232433     0.995248
3002    1.361828     0.989730
3003    1.309318     0.991382

[3004 rows x 10 columns]
```

[16]:
```
item_sales = data.groupby('Itemname')['Quantity'].sum().
  ↪sort_values(ascending=False)

top_n = 10;
plt.figure(figsize=(10, 6))
plt.bar(item_sales.index[:top_n], item_sales.values[:top_n])
plt.xlabel('Item Names')
plt.ylabel('Total Quantity Sold')
plt.title(f'Top {top_n} Most Frequently Bought Items')
plt.xticks(rotation=45)
plt.show()
```

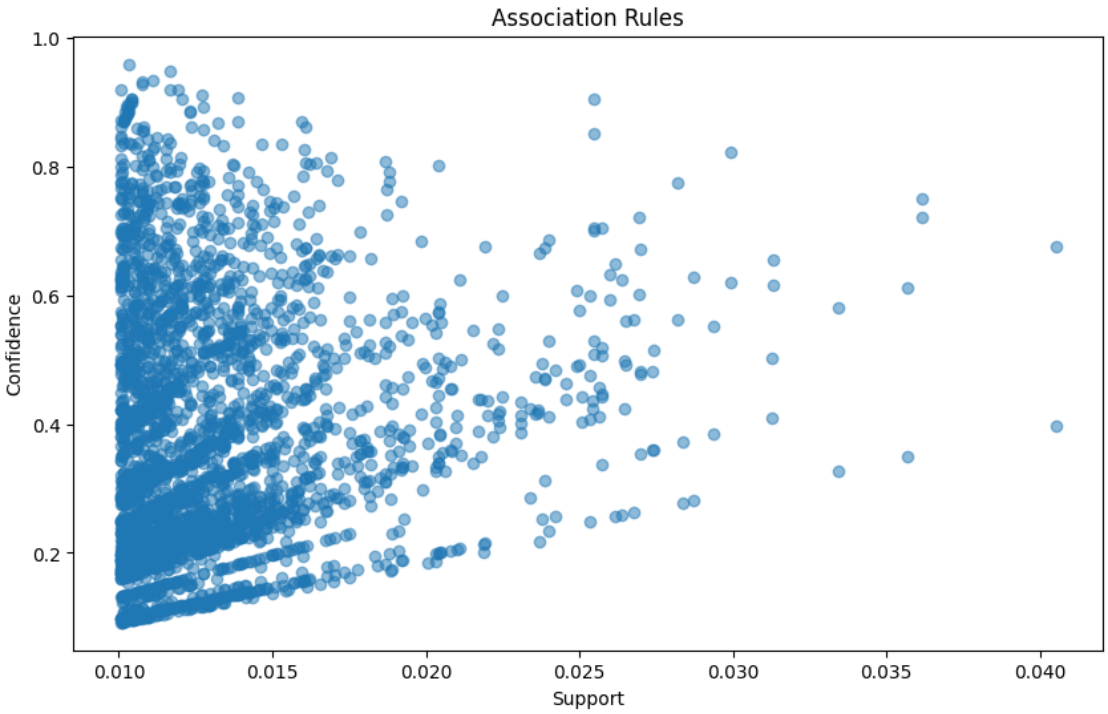


[17]:
```
basket = (data.groupby(['BillNo', 'Itemname'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('BillNo'));
basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1))
frequent_itemsets = apriori(basket_sets, min_support=0.01, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)
print("Association Rules:")
print(rules)
plt.figure(figsize=(10, 6))
plt.scatter(rules['support'], rules['confidence'], alpha=0.5)
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.title('Association Rules')
plt.show()
```

```
C:\Users\D E L L\AppData\Local\Temp\ipykernel_12816\303242403.py:11:
FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
instead.
  basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1))
```

```
Association Rules:
                         antecedents  \
0          (JAM MAKING SET PRINTED)
1          (6 RIBBONS RUSTIC CHARM)
2          (6 RIBBONS RUSTIC CHARM)
3        (JAM MAKING SET WITH JARS)
4          (6 RIBBONS RUSTIC CHARM)
…                               …
2999     (STRAWBERRY CHARLOTTE BAG)
3000    (CHARLOTTE BAG SUKI DESIGN)
3001  (RED RETROSPOT CHARLOTTE BAG)
3002  (CHARLOTTE BAG PINK POLKADOT)
3003       (WOODLAND CHARLOTTE BAG)

                                              consequents  antecedent support  \
0                              (6 RIBBONS RUSTIC CHARM)            0.055226
1                              (JAM MAKING SET PRINTED)            0.046615
2                            (JAM MAKING SET WITH JARS)            0.046615
3                              (6 RIBBONS RUSTIC CHARM)            0.053890
4                             (JUMBO BAG RED RETROSPOT)            0.046615
…                                                   …                   …
2999  (RED RETROSPOT CHARLOTTE BAG, CHARLOTTE BAG PI…            0.035432
3000  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.043300
3001  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.050871
3002  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.036520
```

```
3003  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.040924

      consequent support   support  confidence       lift  leverage  \
0               0.046615  0.011530    0.208781   4.478826  0.008956
1               0.055226  0.011530    0.247346   4.478826  0.008956
2               0.053890  0.010095    0.216561   4.018599  0.007583
3               0.046615  0.010095    0.187328   4.018599  0.007583
4               0.102138  0.010689    0.229299   2.245001  0.005928
…                  …       …          …         …        …
2999            0.012371  0.010046    0.283520  22.917453  0.009607
3000            0.011926  0.010046    0.232000  19.453344  0.009529
3001            0.010936  0.010046    0.197471  18.056517  0.009489
3002            0.012767  0.010046    0.275068  21.544841  0.009579
3003            0.012074  0.010046    0.245466  20.329375  0.009551

      conviction  zhangs_metric
0       1.204957       0.822130
1       1.255257       0.814705
2       1.207637       0.787884
3       1.173148       0.793942
4       1.164995       0.581681
…          …            …
2999    1.378445       0.991495
3000    1.286555       0.991528
3001    1.232433       0.995248
3002    1.361828       0.989730
3003    1.309318       0.991382

[3004 rows x 10 columns]
```



[18]:
```
basket = (data.groupby(['BillNo', 'Itemname'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('BillNo'));
basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1));
frequent_itemsets = apriori(basket_sets, min_support=0.01, use_colnames=True);
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0);
print("Association Rules:")
print(rules)
sorted_rules = rules.sort_values(by=['lift'], ascending=False)
print("Sorted Association Rules:")
print(sorted_rules)
```

```
C:\Users\D E L L\AppData\Local\Temp\ipykernel_12816\3757603289.py:9:
FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
instead.
  basket_sets = basket.applymap(lambda quantity: bool(quantity >= 1));

Association Rules:
                         antecedents  \
0          (JAM MAKING SET PRINTED)
1          (6 RIBBONS RUSTIC CHARM)
2          (6 RIBBONS RUSTIC CHARM)
3        (JAM MAKING SET WITH JARS)
4          (6 RIBBONS RUSTIC CHARM)
…                               …
```

```
2999     (STRAWBERRY CHARLOTTE BAG)
3000    (CHARLOTTE BAG SUKI DESIGN)
3001  (RED RETROSPOT CHARLOTTE BAG)
3002  (CHARLOTTE BAG PINK POLKADOT)
3003       (WOODLAND CHARLOTTE BAG)

                                          consequents  antecedent support  \
0                            (6 RIBBONS RUSTIC CHARM)            0.055226
1                            (JAM MAKING SET PRINTED)            0.046615
2                          (JAM MAKING SET WITH JARS)            0.046615
3                            (6 RIBBONS RUSTIC CHARM)            0.053890
4                           (JUMBO BAG RED RETROSPOT)            0.046615
…                                                  …                   …
2999  (RED RETROSPOT CHARLOTTE BAG, CHARLOTTE BAG PI…            0.035432
3000  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.043300
3001  (CHARLOTTE BAG PINK POLKADOT, STRAWBERRY CHARL…            0.050871
3002  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.036520
3003  (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL…            0.040924

      consequent support   support  confidence       lift  leverage  \
0               0.046615  0.011530    0.208781   4.478826  0.008956
1               0.055226  0.011530    0.247346   4.478826  0.008956
2               0.053890  0.010095    0.216561   4.018599  0.007583
3               0.046615  0.010095    0.187328   4.018599  0.007583
4               0.102138  0.010689    0.229299   2.245001  0.005928
…                  …       …          …         …        …
2999            0.012371  0.010046    0.283520  22.917453  0.009607
3000            0.011926  0.010046    0.232000  19.453344  0.009529
3001            0.010936  0.010046    0.197471  18.056517  0.009489
3002            0.012767  0.010046    0.275068  21.544841  0.009579
3003            0.012074  0.010046    0.245466  20.329375  0.009551

      conviction  zhangs_metric
0       1.204957       0.822130
1       1.255257       0.814705
2       1.207637       0.787884
3       1.173148       0.793942
4       1.164995       0.581681
…          …             …
2999    1.378445       0.991495
3000    1.286555       0.991528
3001    1.232433       0.995248
3002    1.361828       0.989730
3003    1.309318       0.991382

[3004 rows x 10 columns]
Sorted Association Rules:
                              antecedents  \
```

```
482                 (HERB MARKER ROSEMARY)
483                    (HERB MARKER THYME)
481                    (HERB MARKER THYME)
480                  (HERB MARKER PARSLEY)
479                  (HERB MARKER PARSLEY)
…                                      …
1429            (REGENCY CAKESTAND 3 TIER)
1308  (WHITE HANGING HEART T-LIGHT HOLDER)
1309      (PAPER CHAIN KIT 50'S CHRISTMAS)
827               (JUMBO BAG RED RETROSPOT)
826             (REGENCY CAKESTAND 3 TIER)

                              consequents  antecedent support  \
482                    (HERB MARKER THYME)             0.011580
483                 (HERB MARKER ROSEMARY)             0.011530
481                  (HERB MARKER PARSLEY)             0.011530
480                    (HERB MARKER THYME)             0.011481
479                 (HERB MARKER ROSEMARY)             0.011481
…                                      …                    …
1429  (WHITE HANGING HEART T-LIGHT HOLDER)             0.094220
1308      (PAPER CHAIN KIT 50'S CHRISTMAS)             0.108967
1309  (WHITE HANGING HEART T-LIGHT HOLDER)             0.056562
827             (REGENCY CAKESTAND 3 TIER)             0.102138
826               (JUMBO BAG RED RETROSPOT)             0.094220

      consequent support   support  confidence       lift  leverage  \
482             0.011530  0.010738    0.927350  80.428744  0.010605
483             0.011580  0.010738    0.931330  80.428744  0.010605
481             0.011481  0.010392    0.901288  78.505254  0.010260
480             0.011530  0.010392    0.905172  78.505254  0.010260
479             0.011580  0.010392    0.905172  78.169761  0.010259
…                  …        …          …         …        …
1429            0.108967  0.016973    0.180147   1.653230  0.006707
1308            0.056562  0.010095    0.092643   1.637910  0.003932
1309            0.108967  0.010095    0.178478   1.637910  0.003932
827             0.094220  0.013757    0.134690   1.429524  0.004133
826             0.102138  0.013757    0.146008   1.429524  0.004133

      conviction  zhangs_metric
482    13.605998       0.999136
483    14.393872       0.999086
481    10.014131       0.998778
480    10.423865       0.998728
479    10.423343       0.998673
…          …             …
1429    1.086821       0.436224
1308    1.039765       0.437094
1309    1.084612       0.412815
```

```
827     1.046769       0.334647
826     1.051371       0.331721

[3004 rows x 10 columns]
```