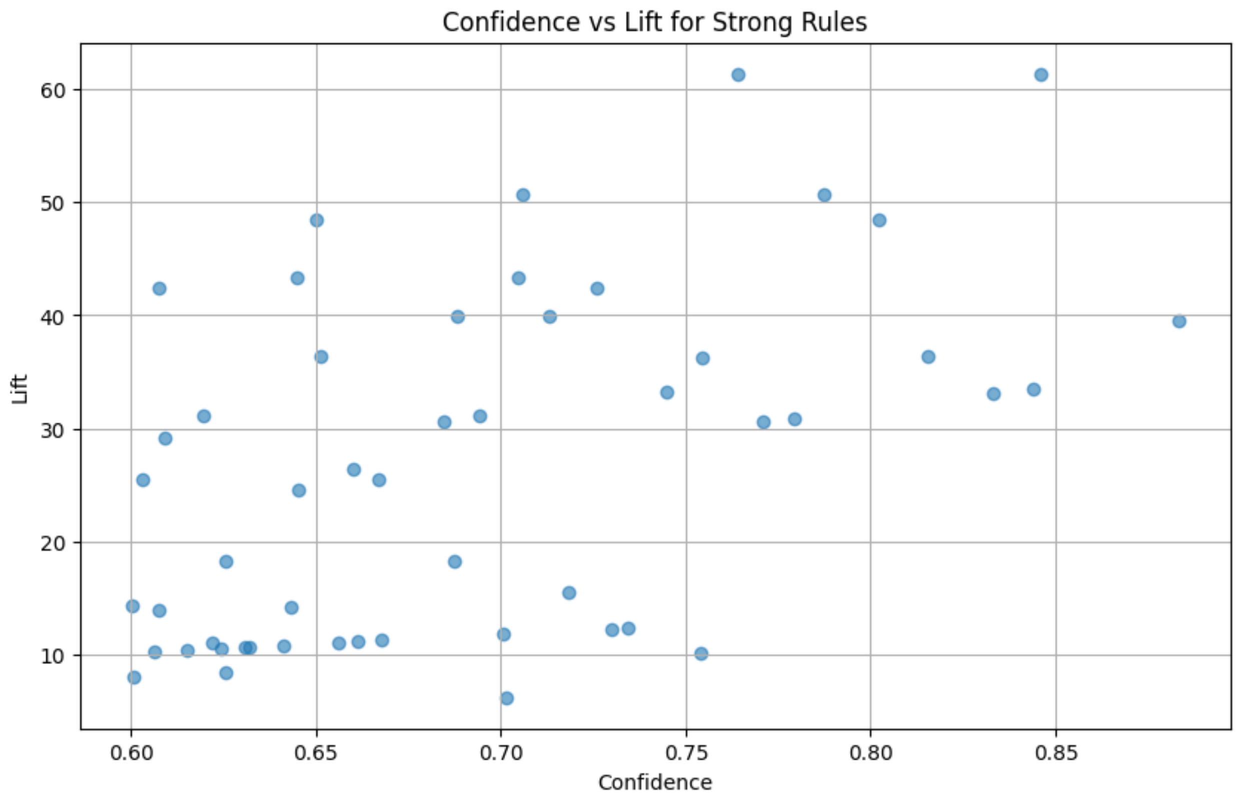
Association Guidelines and Suggestions

Code

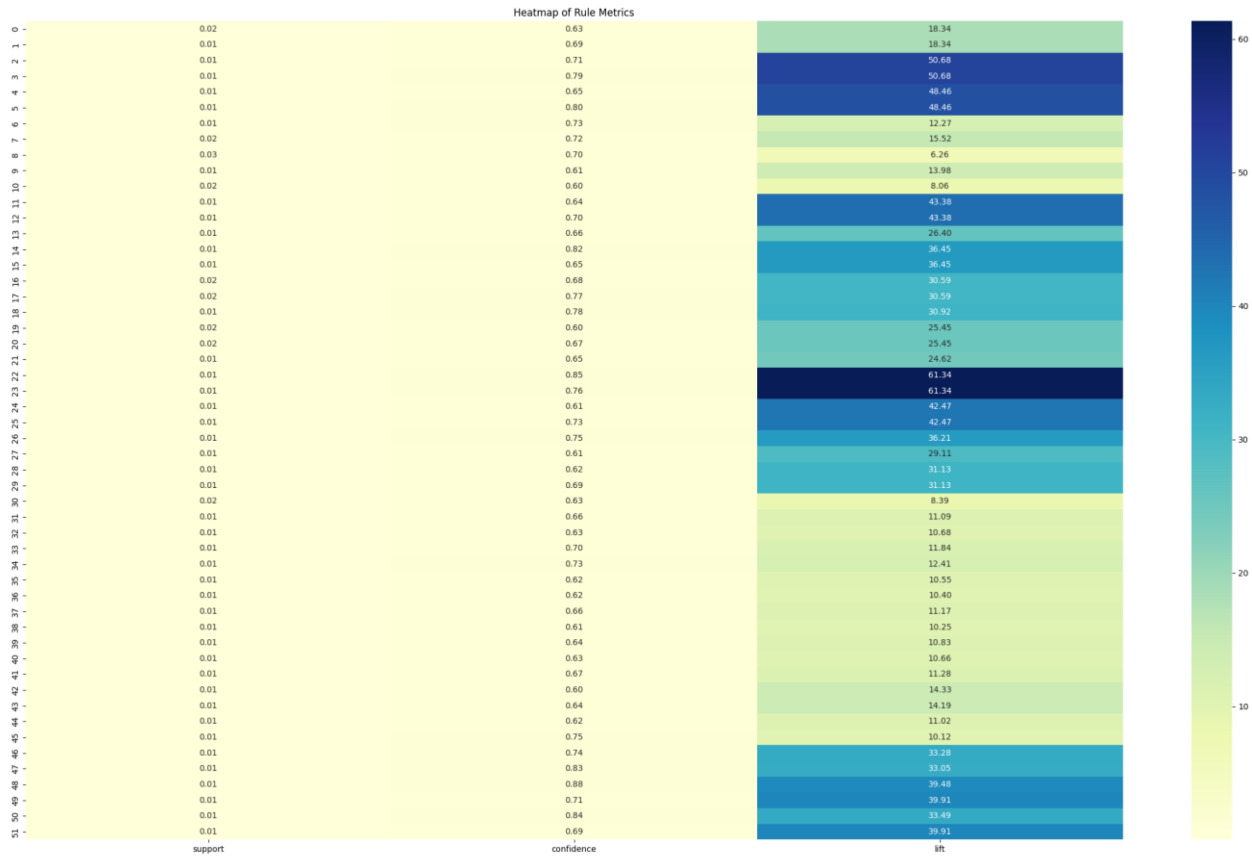
```
# association rule mining for items purchased together
transactions = df.groupby('Invoice')['Description'].apply(list).reset_index()
te = TransactionEncoder()
te_data =
te.fit(transactions['Description']).transform(transactions['Description'])
df_trans = pd.DataFrame(te_data, columns=te.columns_)

frequent_itemsets = apriori(df_trans, min_support=0.01, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="confidence",
min_threshold=0.6, num_itemsets=len(frequent_itemsets))
strong_rules = rules[rules['lift'] > 1]
```

```
# Plot Confidence vs Lift
plt.figure(figsize=(10, 6))
plt.scatter(strong_rules['confidence'], strong_rules['lift'], alpha=0.6)
plt.title('Confidence vs Lift for Strong Rules')
plt.xlabel('Confidence')
plt.ylabel('Lift')
plt.grid(True)
plt.show()
```




```
# heatmap of rule metrics
rule_metrics = strong_rules[['antecedents', 'consequents', 'support',
'confidence', 'lift']]
plt.figure(figsize=(30, 18))
sns.heatmap(rule_metrics[['support', 'confidence', 'lift']], annot=True,
cmap="YlGnBu", fmt=".2f")
plt.title('Heatmap of Rule Metrics')
plt.show()
```

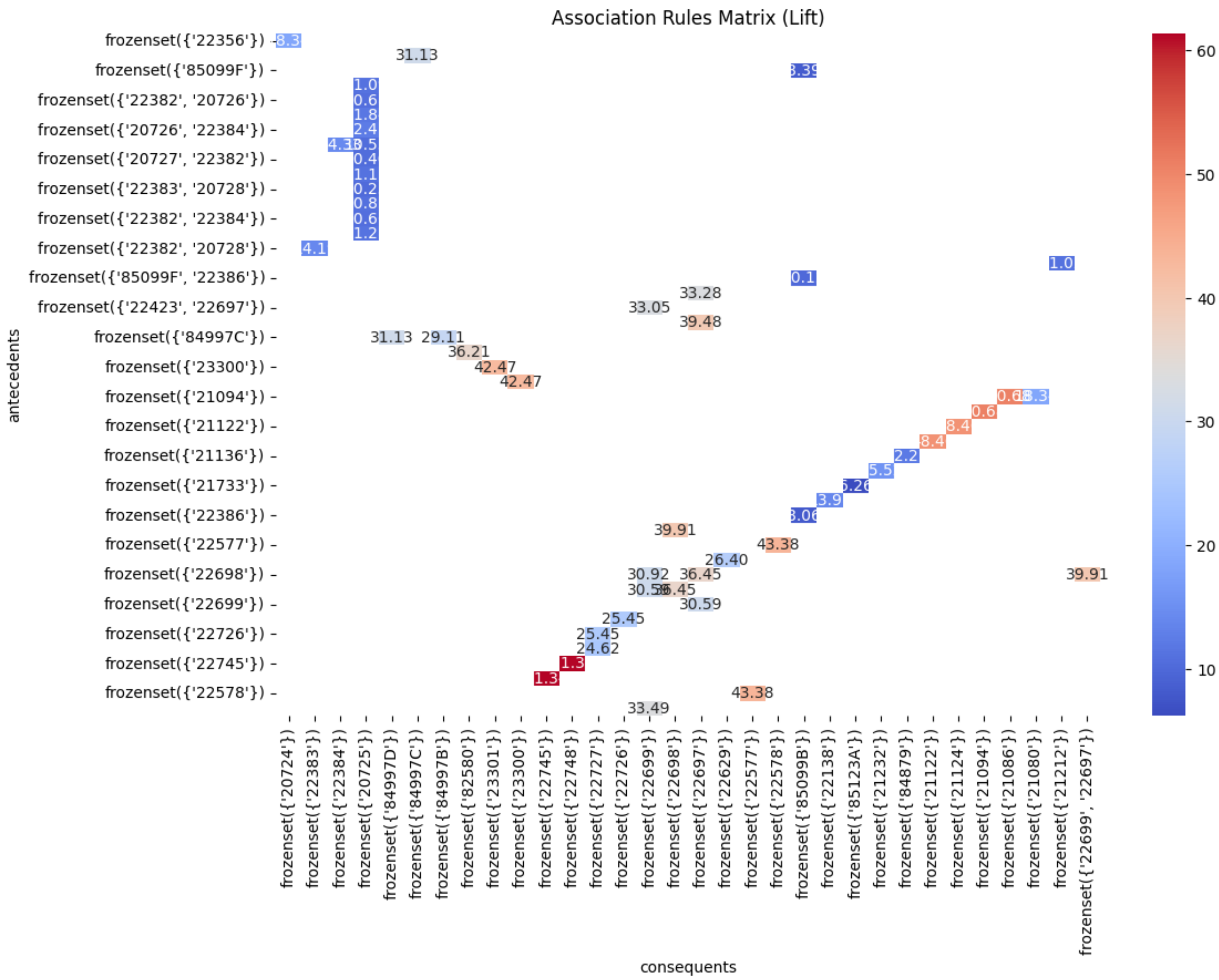



```
# sort rules by lift and show top 10
top_rules = strong_rules.sort_values(by='lift', ascending=False).head(10)
print(top_rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

Antecedent	Consequent
POPPY'S PLAYHOUSE KITCHEN	POPPY'S PLAYHOUSE BEDROOM
POPPY'S PLAYHOUSE BEDROOM	POPPY'S PLAYHOUSE KITCHEN
SET/6 RED SPOTTY PAPER PLATES	SET/6 RED SPOTTY PAPER CUPS
SET/6 RED SPOTTY PAPER CUPS	SET/6 RED SPOTTY PAPER PLATES
WOODEN HEART CHRISTMAS SCANDINAVIAN	WOODEN STAR CHRISTMAS SCANDINAVIAN
WOODEN STAR CHRISTMAS SCANDINAVIAN	WOODEN HEART CHRISTMAS SCANDINAVIAN
GARDENERS KNEELING PAD KEEP CALM	GARDENERS KNEELING PAD CUP OF TEA
GARDENERS KNEELING PAD CUP OF TEA	GARDENERS KNEELING PAD KEEP CALM
PINK REGENCY TEACUP AND SAUCER	ROSES REGENCY TEACUP AND SAUCER , GREEN REGENCY TEACUP AND SAUCER
GREEN REGENCY TEACUP AND SAUCER	PINK REGENCY TEACUP AND SAUCER

```
# plotting association rules as matrix plot
```

```
rules_matrix = strong_rules.pivot_table(index='antecedents',
columns='consequents', values='lift', aggfunc='mean')
plt.figure(figsize=(12, 8))
sns.heatmap(rules_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title('Association Rules Matrix (Lift)')
plt.show()
```




```
# bar plots for support, confidence, and lift of top rules
top_rules = strong_rules.sort_values(by='lift', ascending=False).head(10)

fig, axes = plt.subplots(1, 3, figsize=(15, 5))

axes[0].bar(top_rules['antecedents'].astype(str), top_rules['support'])
axes[0].set_title('Support of Top Rules')
axes[0].set_xlabel('Rule')
axes[0].set_ylabel('Support')

axes[1].bar(top_rules['antecedents'].astype(str), top_rules['confidence'])
```

```python
axes[1].set_title('Confidence of Top Rules')
axes[1].set_xlabel('Rule')
axes[1].set_ylabel('Confidence')

axes[2].bar(top_rules['antecedents'].astype(str), top_rules['lift'])
axes[2].set_title('Lift of Top Rules')
axes[2].set_xlabel('Rule')
axes[2].set_ylabel('Lift')

plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

Analysis and Recommendations

The algorithm implemented performs association rule mining to identify relationships between items frequently purchased together. First, the dataset is grouped by invoices to create a list of items purchased in each transaction. These transactions are then encoded into a binary matrix, where rows represent transactions, columns represent unique items, and cells indicate whether an item was present in a transaction. The Apriori algorithm is applied to this matrix to generate frequent itemsets, which are groups of items appearing together in transactions with a support value exceeding 1%. Frequent itemsets are then used to generate association rules, which describe relationships between antecedents and consequents. These rules are filtered using confidence, a measure of the likelihood of purchasing consequent items given the antecedents, and lift, which evaluates the strength of the association compared to random chance. The result is a set of strong association rules that provide insights into purchasing patterns. According to the strongest association rules, there are several pairs of items that are often bought together. For example, the ‘Poppy’s Playhouse Kitchen’ and ‘Poppy’s Playhouse Bath’ are associated with each other. Positioning these items together on the website so consumers looking for one learn about the other would take advantage of this correlation. Selling these items as bundles would be another recommendation, as consumers may be willing to buy related items altogether at a slightly lower price when they would otherwise skip out on some.