Heatmaps

MARKET BASKET ANALYSIS IN PYTHON

Isaiah Hull
Economist

MovieLens dataset

```python
import pandas as pd

# Load ratings data.
ratings = pd.read_csv('datasets/movie_ratings.csv')
print(ratings.head())
```

```
   userId  movieId  title
0    3149    54286  Bourne Ultimatum, The (2007)
1    3149     1220  Blues Brothers, The (1980)
2    3149     4007  Wall Street (1987)
3    3149     7156  Fog of War: Eleven...
4    3149    97304  Argo (2012)
```

Creating "transactions" from ratings

```python
# Recover unique user IDs.
user_id = movies['userId'].unique()

# Create library of highly rated movies for each user.
libraries = [list(ratings[ratings['userId'] == u].title) for u in user_id]

# Print example library.
print(library[0])
```

```
['Battlestar Galactica (2003)',
 'Gorgon, The (1964)',
 'Under the Skin (2013)',
 'Upstream Color (2013)',
 'Destry Rides Again (1939)',
 'Dr. Phibes Rises Again (1972)']
```

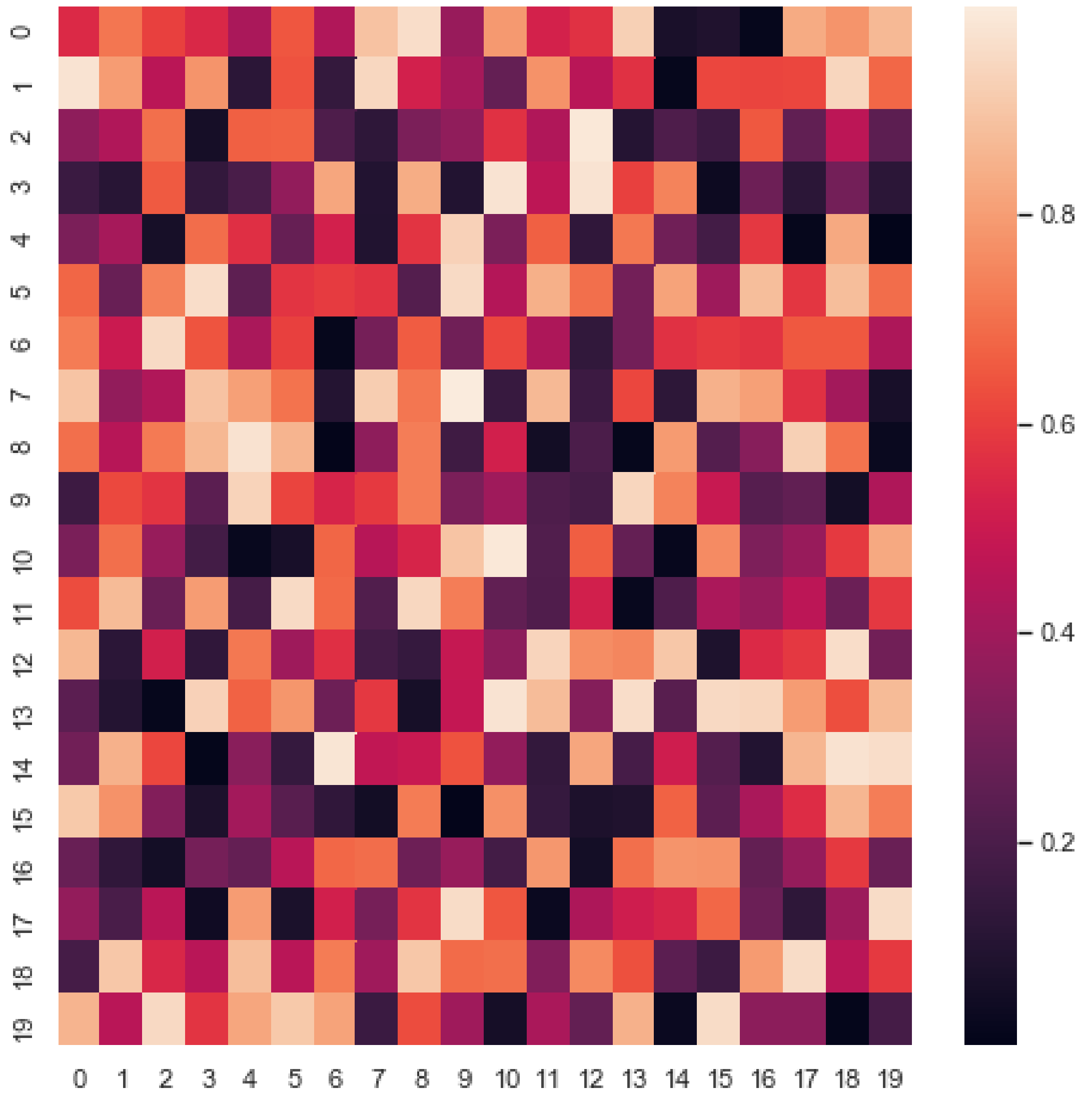
One-hot encoding transactions

```python
from mlxtend.preprocessing import TransactionEncoder

# Instantiate transaction encoder.
encoder = TransactionEncoder()

# One-hot encode libraries.
onehot = encoder.fit(libraries).transform(libraries)

# Use movie titles as column headers.
onehot = pd.DataFrame(onehot, columns = encoder.columns_)

# Print onehot header.
print(onehot.head())
```

One-hot encoding transactions

```
   (500) Days of Summer (2009)  .45 (2006)  10 Things I Hate About You (1999)
0                        False       False                              False
1                        False       False                              False
2                        False       False                              False
3                        False       False                              False
4                        False       False                              False
```

What is a heatmap?



Preparing the data

1. **Generate the rules.**
   - Use Apriori algorithm and association rules.
2. **Convert antecedents and consequents into strings.**
   - Stored as frozen sets by default in mlxtend.
3. **Convert rules into matrix format.**
   - Suitable for use in heatmaps.

Preparing the data

```
from mlxtend.frequent_patterns import association_rules, apriori
import seaborn as sns
```

```
# Apply the apriori algorithm
frequent_itemsets = apriori(onehot, min_support=0.10,
                            use_colnames=True, max_len=2)

# Recover the association rules
rules = association_rules(frequent_itemsets)
```

Generating a heatmap

```
# Convert antecedents and consequents into strings
rules['antecedents'] = rules['antecedents'].apply(lambda a: ','.join(list(a)))
rules['consequents'] = rules['consequents'].apply(lambda a: ','.join(list(a)))
```

```
# Print example.
print(rules[['antecedents','consequents']])
```

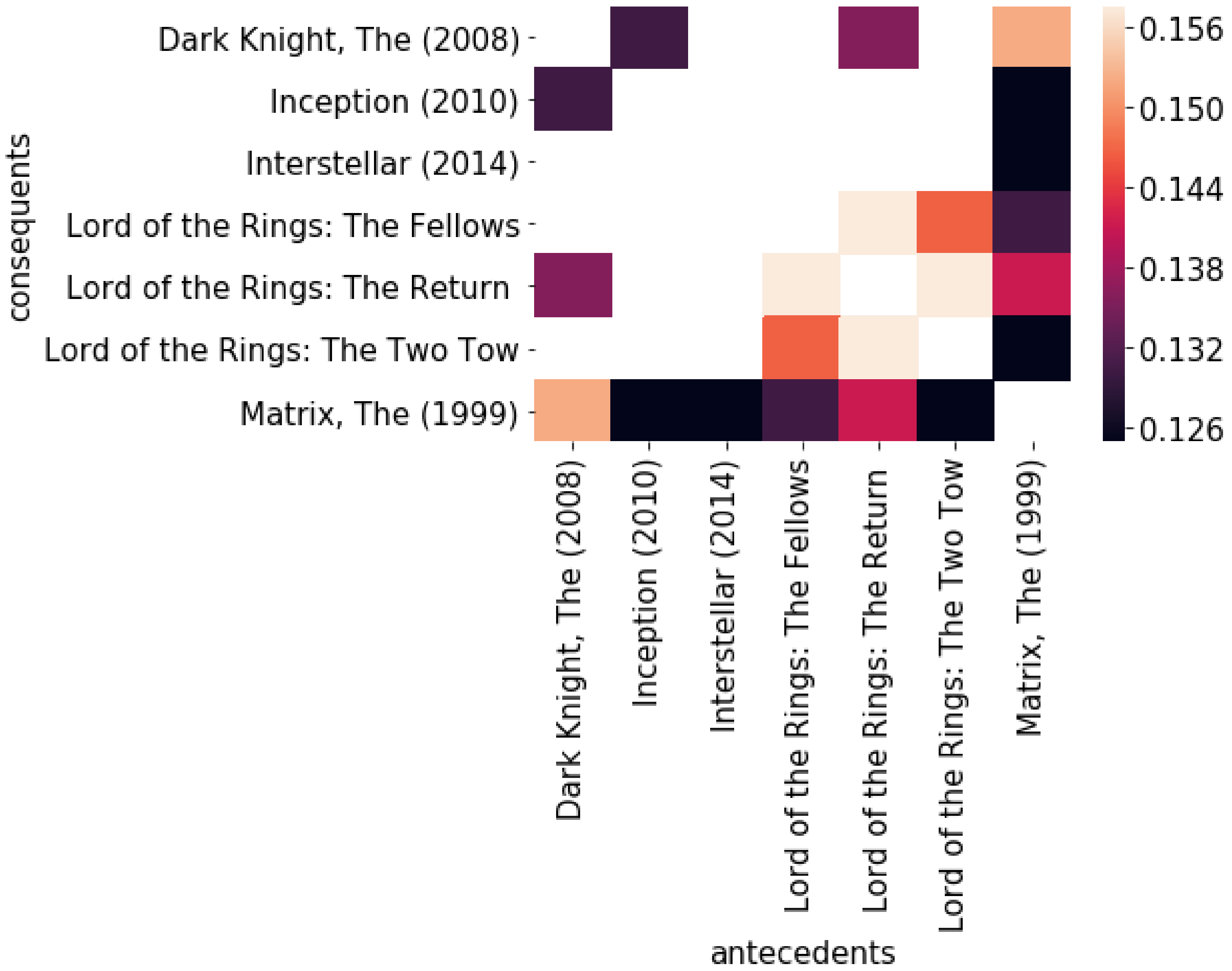
```
            antecedents                    consequents
0  Batman Begins (2005)  Dark Knight Rises, The (2012)
```

Generating a heatmap

```
# Transform antecedent, consequent, and support columns into matrix
support_table = rules.pivot(index='consequents', columns='antecedents',
                                values='support')
```

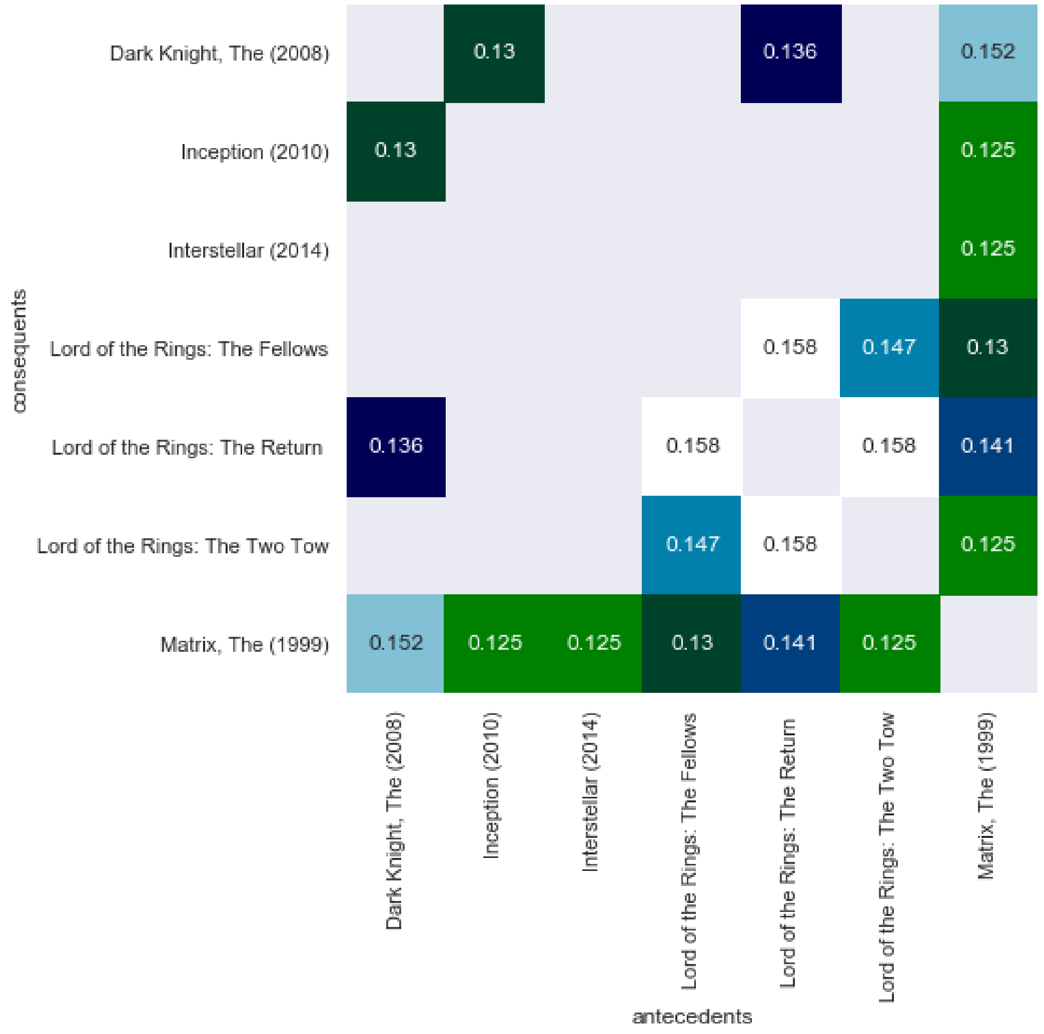
```
# Generate heatmap
sns.heatmap(support_table)
```

Generating a heatmap



Customizing heatmaps

```
sns.heatmap(pivot, annot=True, cbar=False, cmap='ocean')
```



Let's practice!

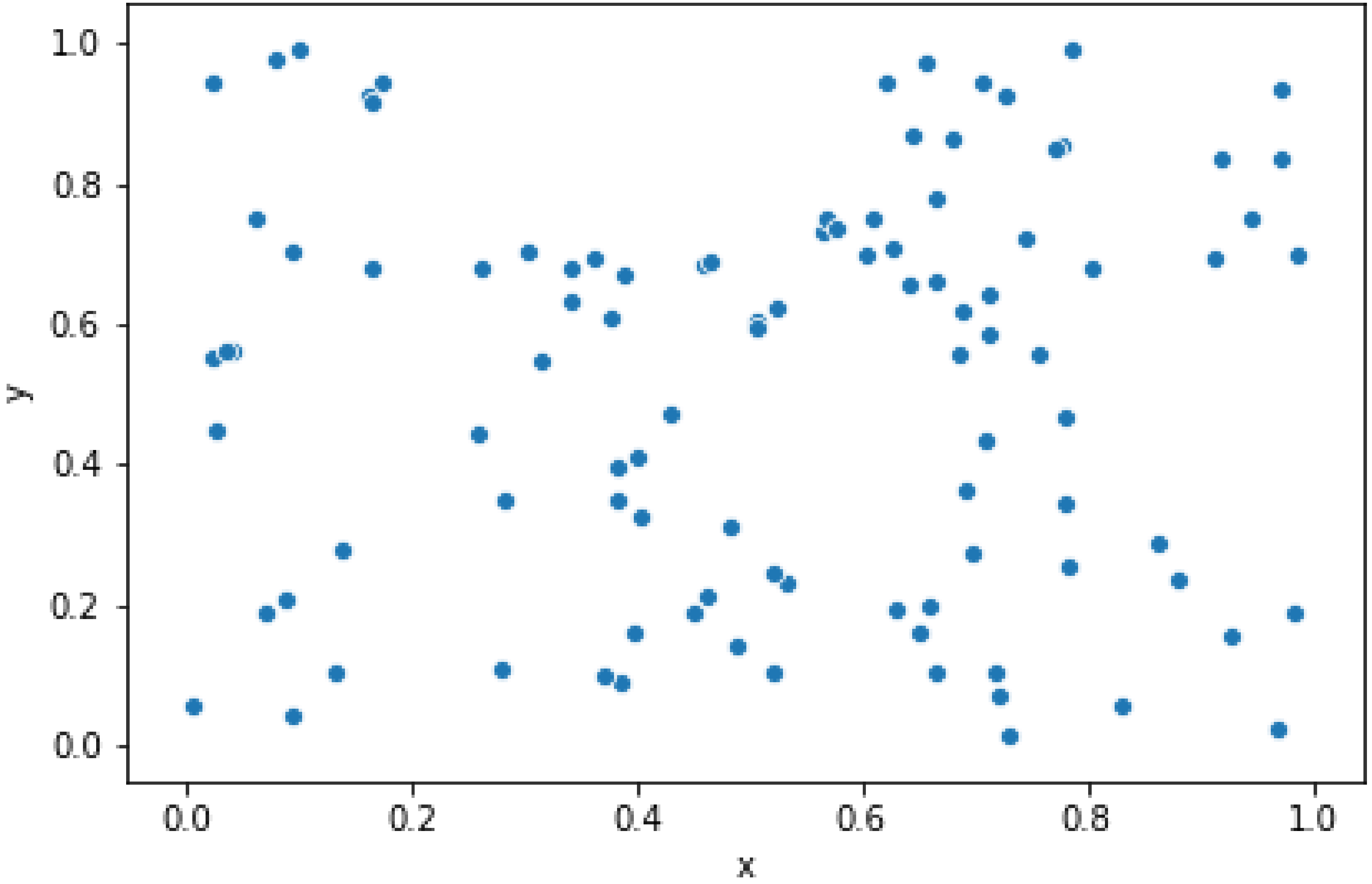
MARKET BASKET ANALYSIS IN PYTHON

Scatterplots

MARKET BASKET ANALYSIS IN PYTHON

Isaiah Hull
Economist

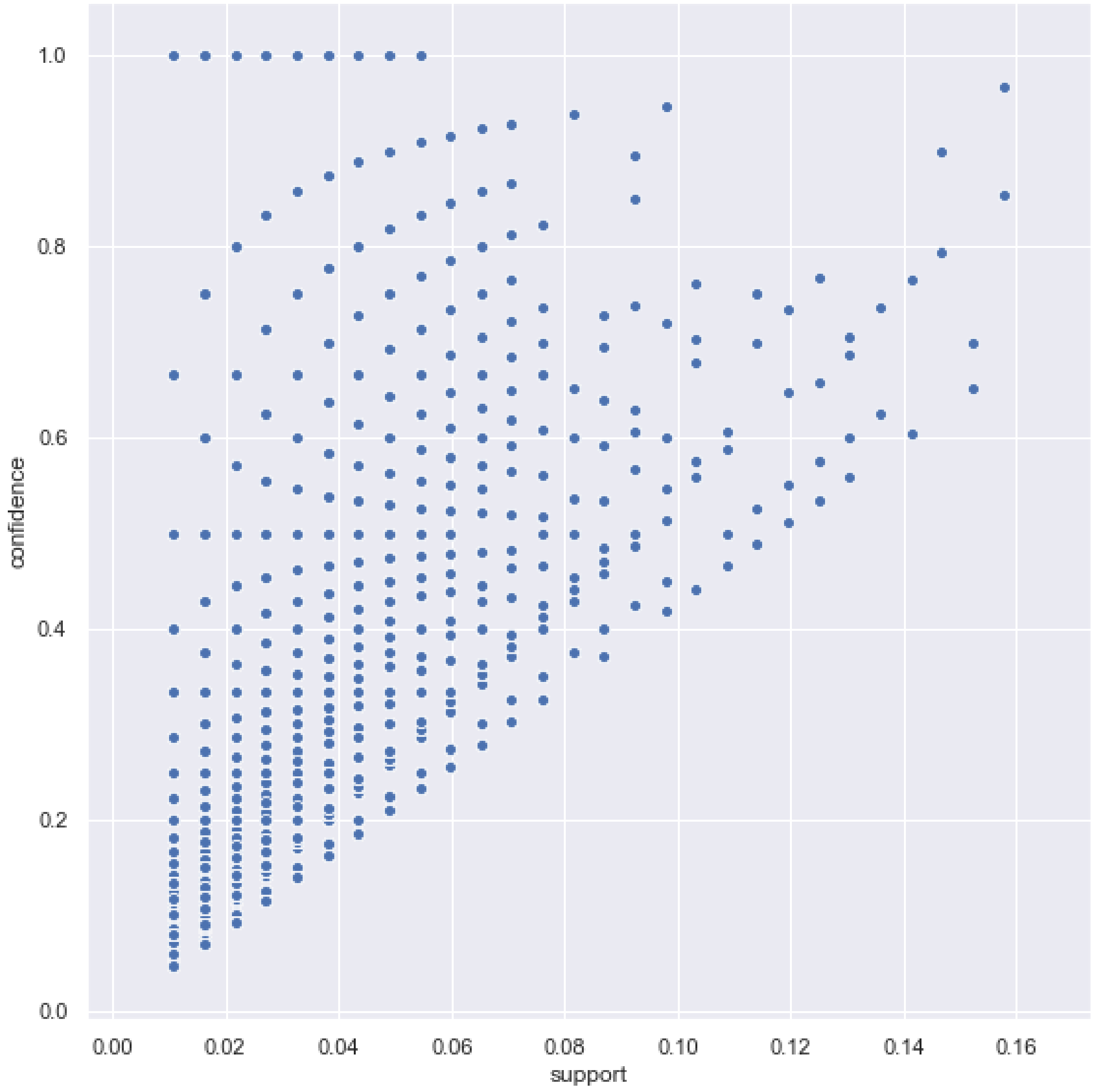
Introduction to scatterplots



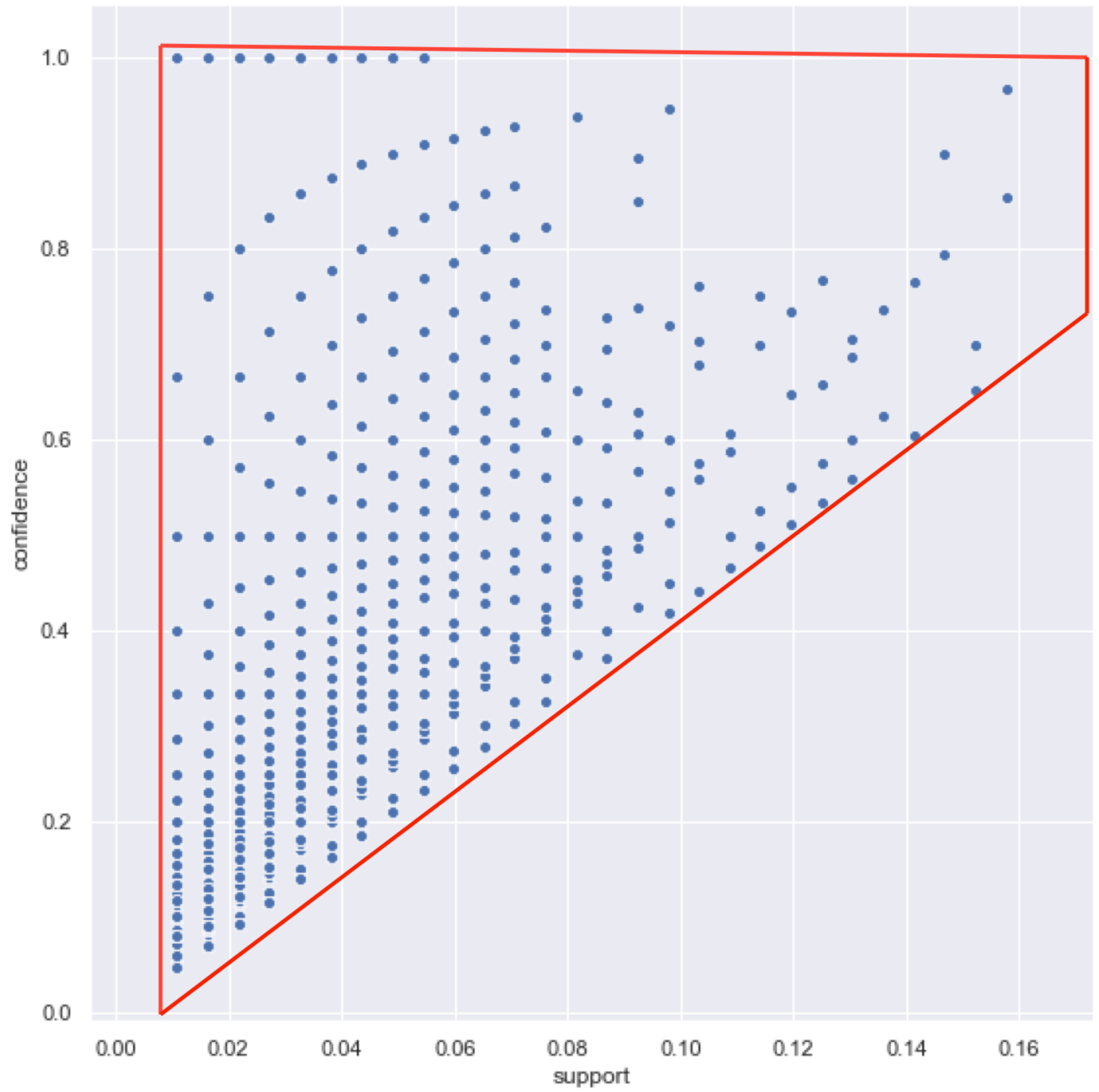
Introduction to scatterplots

- **A scatterplot displays pairs of values.**
  - Antecedent and consequent support.
  - Confidence and lift.
- **No model is assumed.**
  - No trend line or curve needed.
- **Can provide starting point for pruning.**
  - Identify patterns in data and rules.

Support versus confidence



Support versus confidence



[1] Bayardo Jr., R.J. and Agrawal, R. (1999). Mining the Most Interesting Rules. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 145-154).

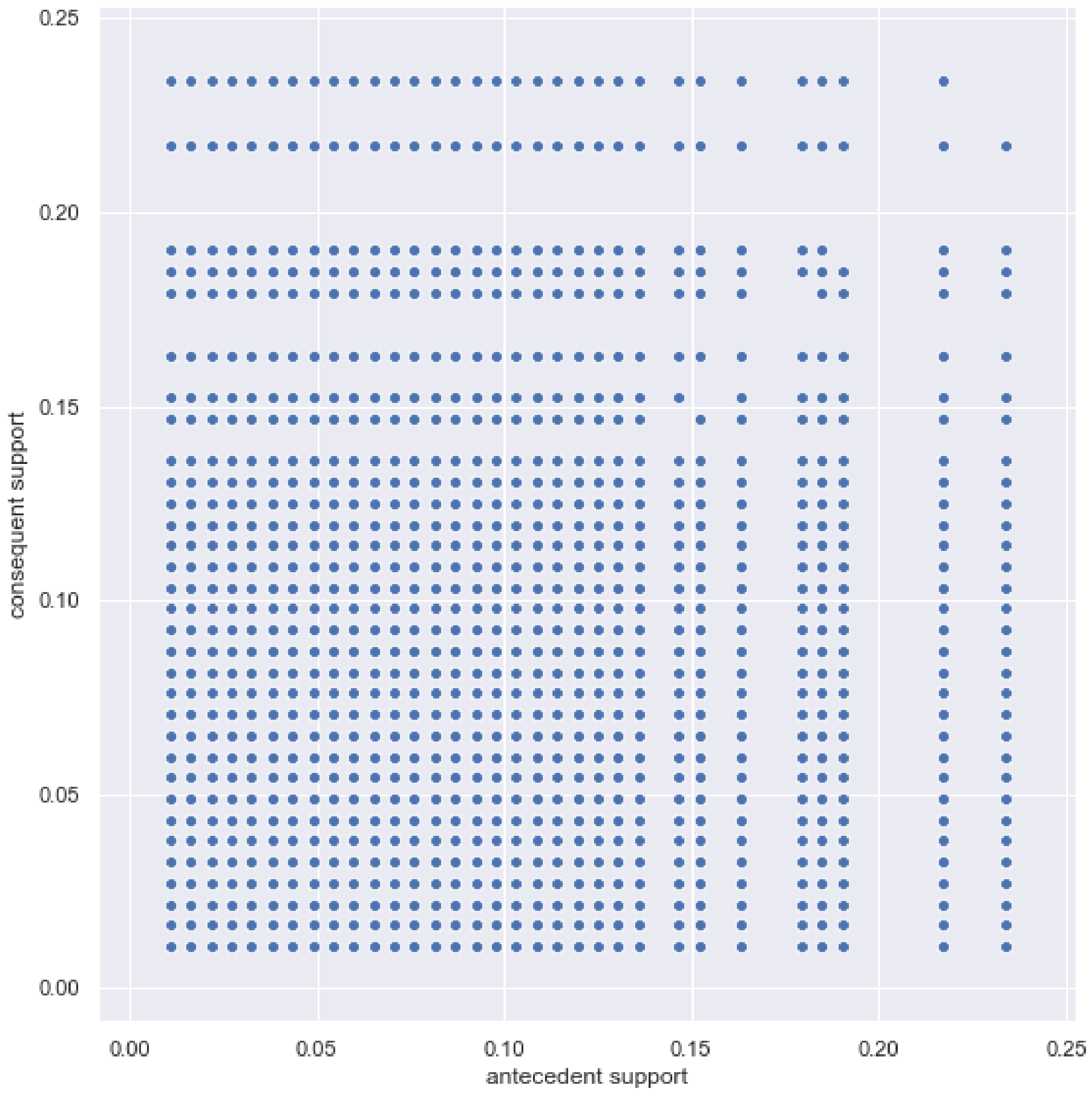
Generating a scatterplot

```python
import pandas as pd
import seaborn as sns
from mlxtend.frequent_patterns import association_rules, apriori

# Load one-hot encoded MovieLens data
onehot = pd.read_csv('datasets/movies_onehot.csv')
```

```python
# Generate frequent itemsets using Apriori
frequent_itemsets = apriori(onehot, min_support=0.01, use_colnames=True, max_len=2)

# Generate association rules
rules = association_rules(frequent_itemsets, metric='support', min_threshold=0.0)
```

```python
sns.scatterplot(x="antecedent support", y="consequent support", data=rules)
```

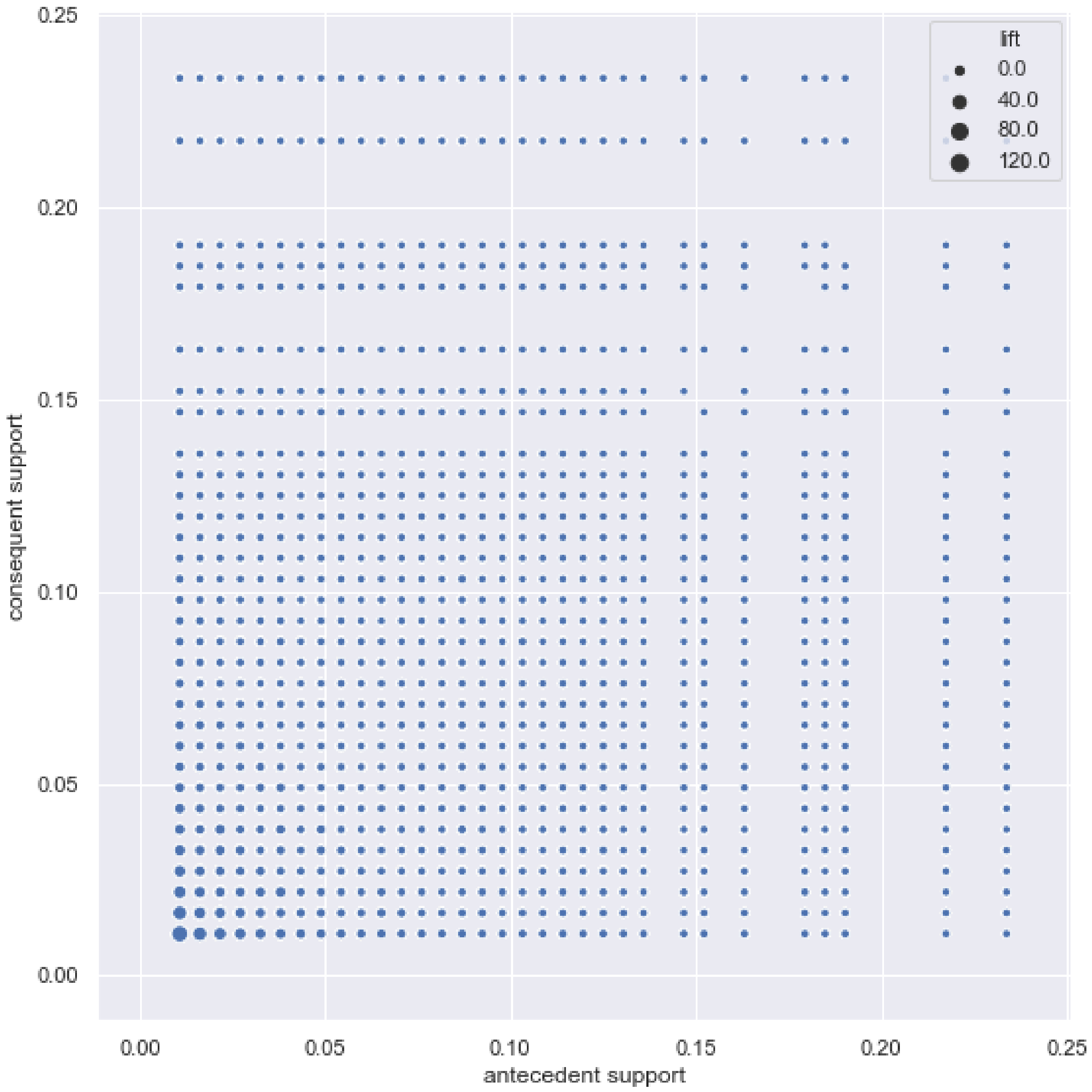
Generating a scatterplot



Adding a third metric

```
sns.scatterplot(x="antecedent support",
                y="consequent support",
                size="lift",
                data=rules)
```

Adding a third metric



What can we learn from scatterplots?

- **Identify natural thresholds in data.**
  - Not possible with heatmaps or other visualizations.
- **Visualize entire dataset.**
  - Not limited to small number of rules.
- **Use findings to prune.**
  - Use natural thresholds and patterns to prune.

Let's practice!

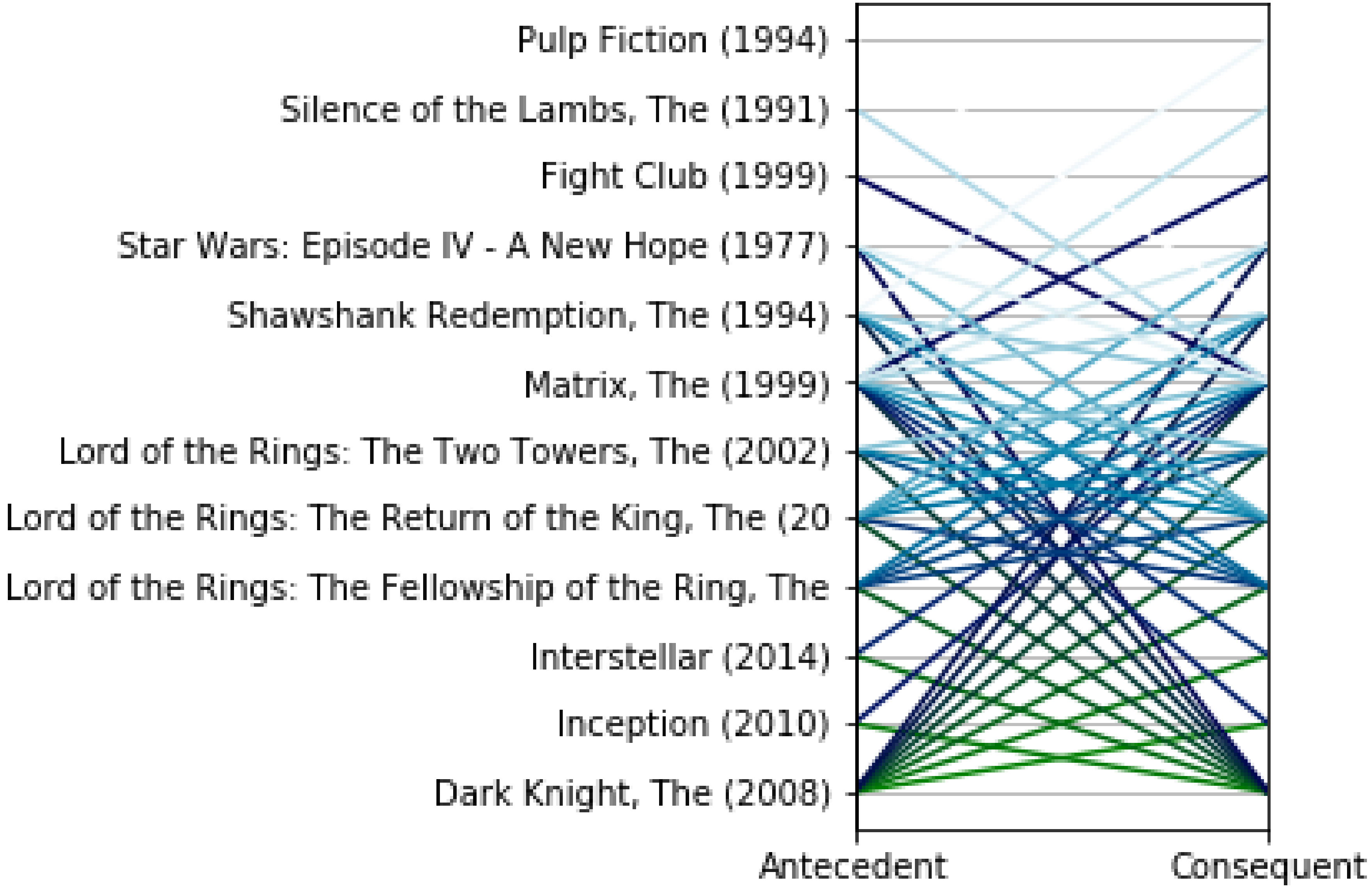
MARKET BASKET ANALYSIS IN PYTHON

Parallel coordinates plot

MARKET BASKET ANALYSIS IN PYTHON

Isaiah Hull
Economist

What is a parallel coordinates plot?



When to use parallel coordinate plots

- **Parallel coordinates vs. heatmap.**
  - Don't need intensity information.
  - Only want to know whether rule exists.
  - Want to reduce visual clutter.

- **Parallel coordinates vs. scatterplot.**
  - Want individual rule information.
  - Not interested in multiple metrics.
  - Only want to examine final rules.

Preparing the data

```python
from mlxtend.frequent_patterns import association_rules, apriori

# Load the one-hot encoded data
onehot = pd.read_csv('datasets/movies_onehot.csv')
```

```python
# Generate frequent itemsets
frequent_itemsets = apriori(onehot, min_support = 0.10, use_colnames = True, max_len = 2)

# Generate association rules
rules = association_rules(frequent_itemsets, metric = 'support', min_threshold = 0.00)
```

Converting rules to coordinates

```python
# Convert rules to coordinates.
rules['antecedent'] = rules['antecedents'].apply(lambda antecedent: list(antecedent)[0])
rules['consequent'] = rules['consequents'].apply(lambda consequent: list(consequent)[0])
rules['rule'] = rules.index
```

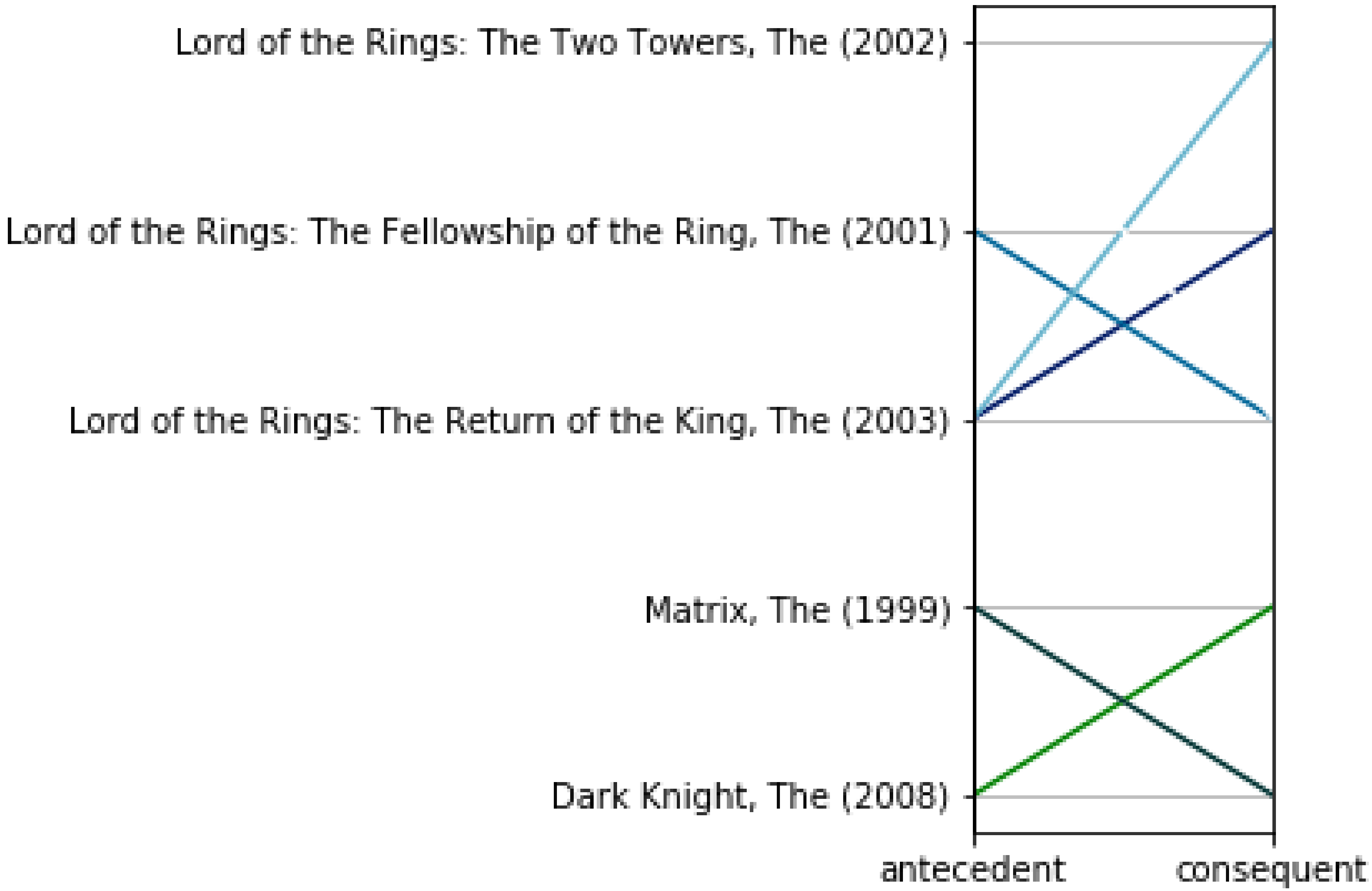
```python
# Define coordinates and label
coords = rules[['antecedent','consequent','rule']]

# Print example
print(coords.head(1))
```

```
                antecedent        consequent  rule
0  Dark Knight, The (2008)  Inception (2010)     0
```

Generating a parallel coordinates plot

```
from pandas.plotting import parallel_coordinates


# Generate parallel coordinates plot
parallel_coordinates(coords, 'rule', colormap = 'ocean')
```

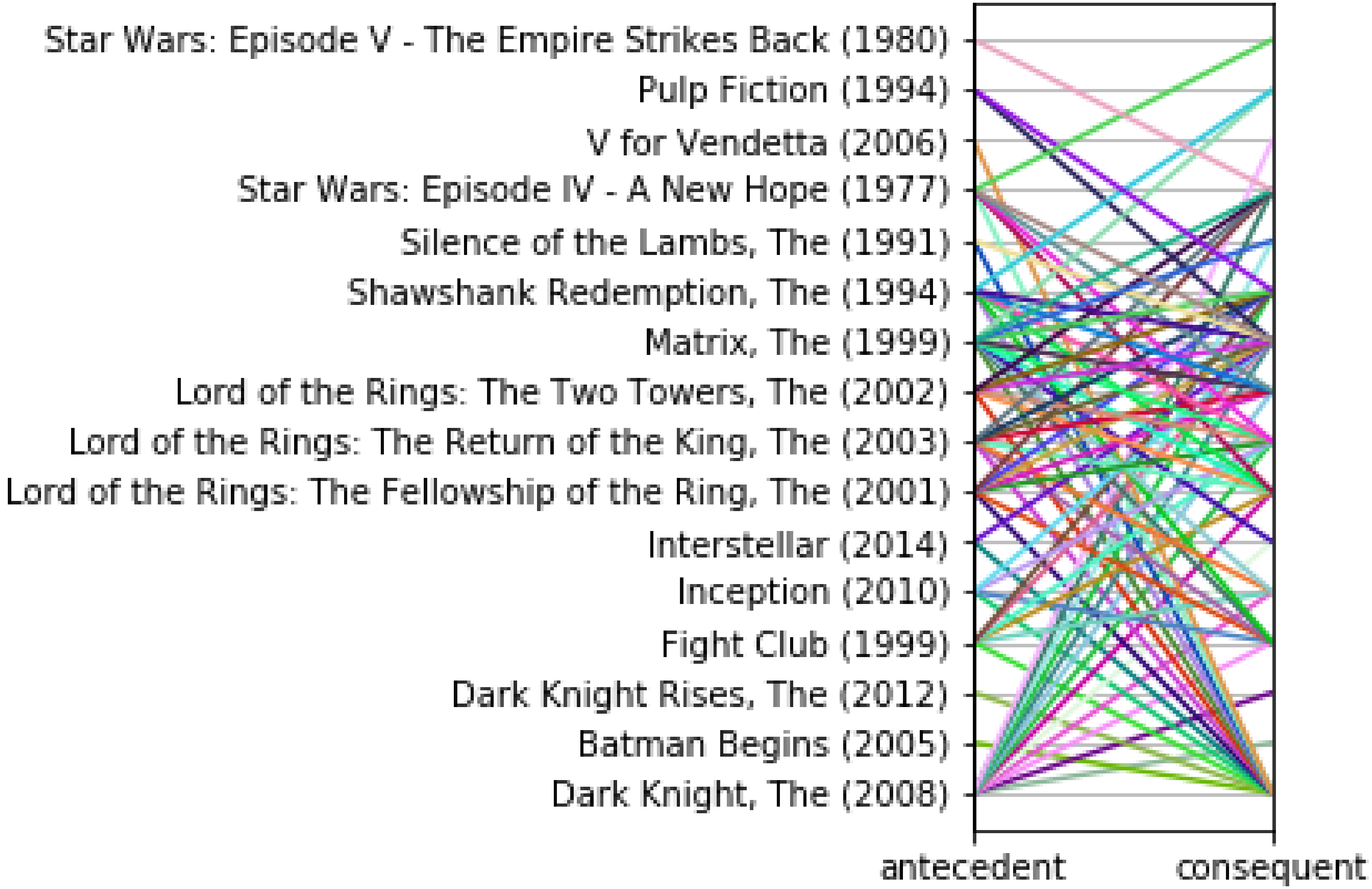
Generating a parallel coordinates plot



Refining a parallel coordinates plot

```python
# Generate frequent itemsets
frequent_itemsets = apriori(onehot, min_support = 0.01, use_colnames = True, max_len = 2)

# Generate association rules
rules = association_rules(frequent_itemsets, metric = 'lift', min_threshold = 1.00)

# Generate coordinates and print example
coords = rules_to_coordinates(rules)

# Generate parallel coordinates plot
parallel_coordinates(coords, 'rule')
```

Refining a parallel coordinates plot



Let's practice!

MARKET BASKET ANALYSIS IN PYTHON

Congratulations!

MARKET BASKET ANALYSIS IN PYTHON

Isaiah Hull
Instructor

Transactions and itemsets

- **Transactions**

TID	Transaction
1	MILK, BREAD, BISCUIT
...	...
20	TEA, MILK, COFFEE, CEREAL

- **Itemsets**
  - {MILK, BREAD}
  - {MILK, COFFEE, CEREAL}

Association rules and metrics

- **Association Rules**
  - Use if-then structure.
    - If A then B.
  - Have antecedent(s) and consequent(s).
  - Many association rules.

- **Metrics**
  - Measure strength of association.
    - Support, lift, confidence, conviction
  - Used to prune itemsets and rules.

Pruning and aggregation

~~Big Box~~

Red Bag

~~Purple Candle~~

~~Cloth Bag~~

Metal Sign

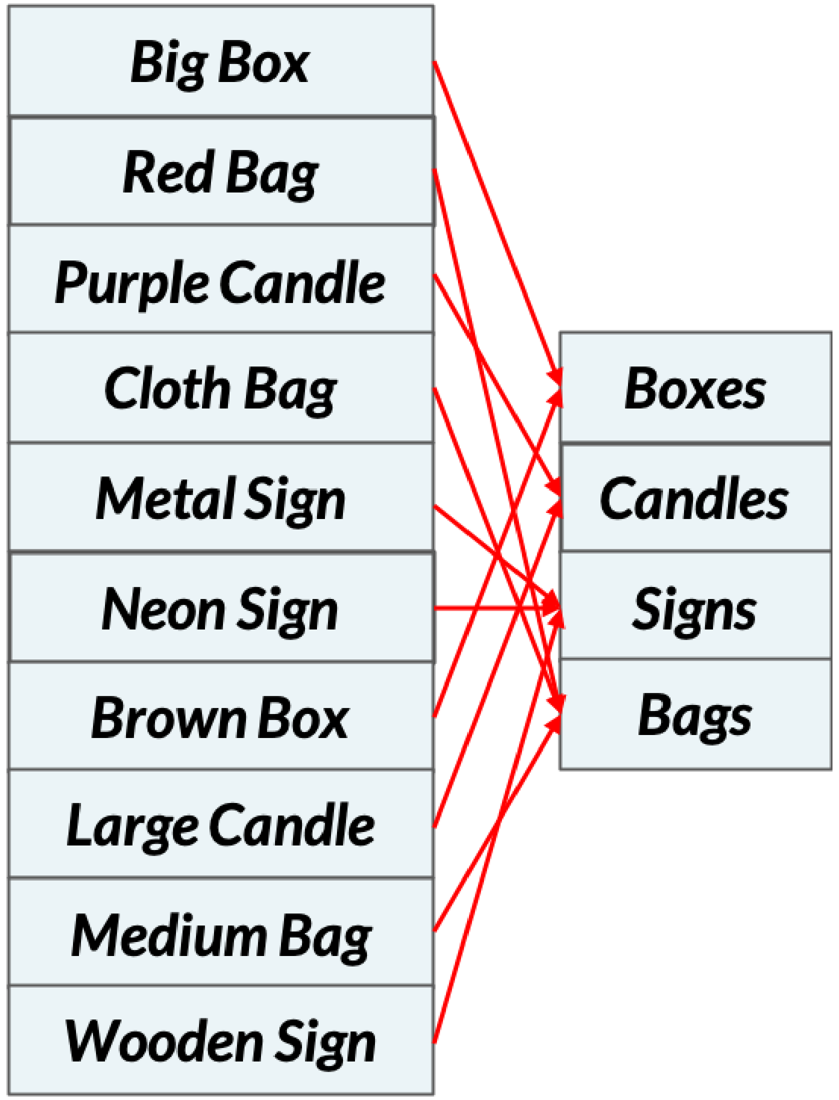
~~Neon Sign~~

Brown Box

~~Large Candle~~

~~Medium Bag~~

Wooden Sign



Big Box

Red Bag

Purple Candle

Cloth Bag

Metal Sign

Neon Sign

Brown Box

Large Candle

Medium Bag

Wooden Sign

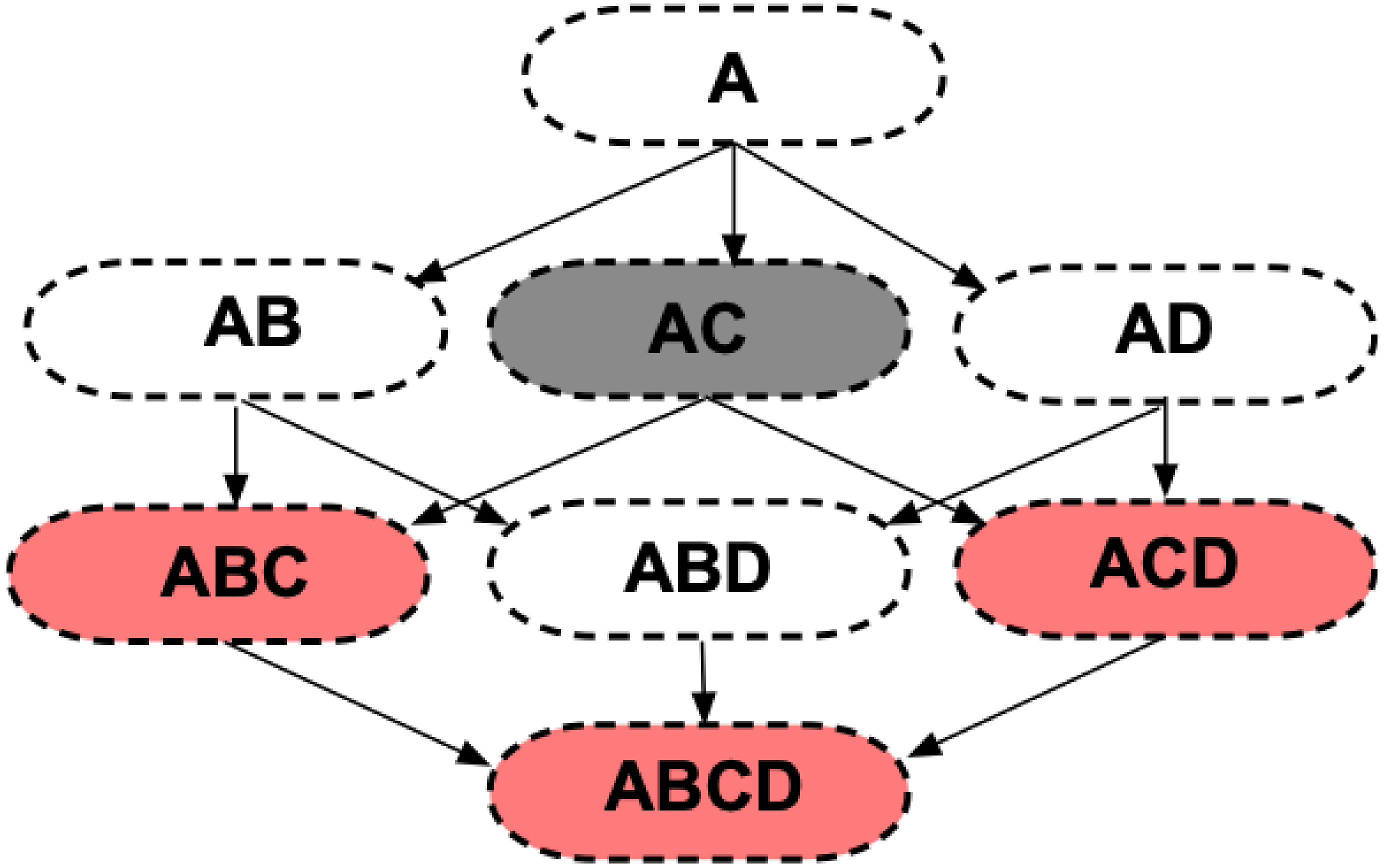
Boxes

Candles

Signs

Bags

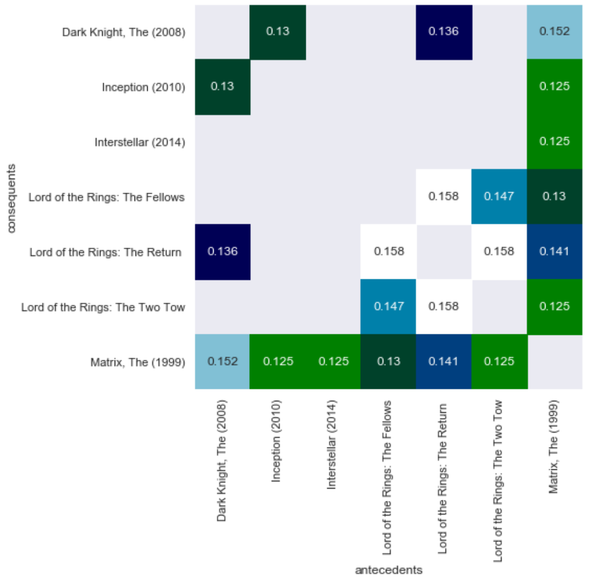
The Apriori algorithm

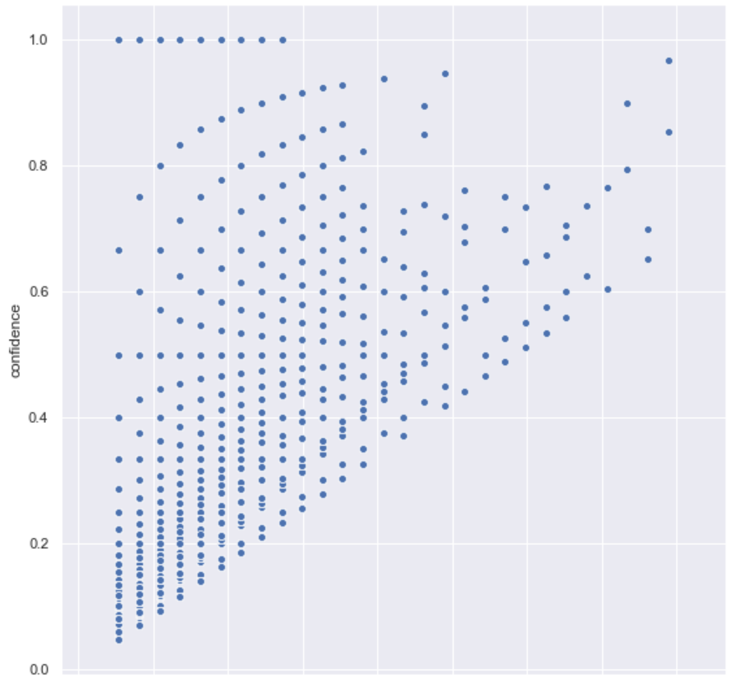


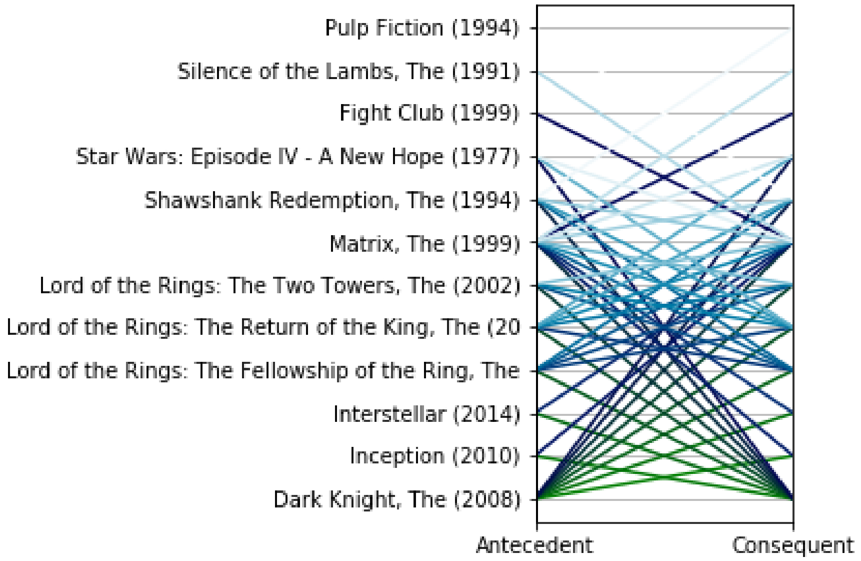
■ = not in frequent two-item sets

🟥 = eliminated by Apriori Principle

Visualizing rules







Congratulations!

MARKET BASKET ANALYSIS IN PYTHON