

Devanshu Surana
 PC-23, 1032210755
 Panel C, Batch C1

DEC Lab Assignment - 6

Aim - Apply a-priori algorithm to find frequently occurring items from given data and generate strong association rules.

Objective:

- Learn frequent itemsets, closed itemsets, Market basket analysis.
- Calculate Support, Confidence and lift
- To generate the association rules.

Theory:-

- A-priori Algorithm.

F_k - Frequent k itemsets

L_k - Candidate k itemsets.

Let $k = 1$

Generate $F_1 = \{ \text{Frequent 1-itemsets} \}$

Repeat until F_k is empty

- Candidate Generation (Join Step): Generate L_{k+1} From F_k
- Support Counting Count the support of each candidate in L_{k+1} by scanning the DB

- Candidate pruning: Prime Candidate itemsets in L_{k+1} containing subsets of length k that are infrequent → Eliminate candidates in L_{k+1} that are infrequent leaving only those that are frequent $\Rightarrow F_{k+1}$

Rule: Merge two Frequent itemsets with size n if their first $n-1$ items are identical

- $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$

Merge $(ABC, ABD) = ABCD$

Merge $(ABC, ABE) = ABCE$

Merge $(ABD, ABE) = ABDE$

We can't merge (ABD, ACD) because they share only prefix of length 1 instead of length 2

-- Association Rule Generation

- Given a Frequent itemset L , Find all non-empty subsets $F \subseteq L$ such that $F \rightarrow L$

- F satisfies the minimum confidence requirement.

If $\{A, B, C, D\}$ is a Frequent itemset. Then Candidate rules:

$ABCD \rightarrow D$ $ABD \rightarrow C$ $ACD \rightarrow B$

$A \rightarrow BCD$ $B \rightarrow ACD$ $C \rightarrow ABD$

$AB \rightarrow CD$ $AC \rightarrow BD$ $AD \rightarrow BC$

$BD \rightarrow AC$ $CD \rightarrow AB$

If $|L| = k$ then there are $2^k - 2$ Candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Input: Datasets

Output: Generating association Rules

Platform: Windows

Conclusion: Thus, we have learned to perform
a-priori Algorithm in python Pandas

FAQs.

1. What is the a-priori principle?

Ans. "If an itemset is frequent, then all of its subsets must also be frequent."

Apriori principle holds due to the following property of the support measure:

$$\forall X, Y: (X \subseteq Y) \Rightarrow S(X) \geq S(Y)$$

2. What are the different steps involved in the a priori algorithm?

Ans. Itemset generation Create candidate itemsets of $L=1$ by scanning the database for individual items.

- Frequent itemset generation Count the support of these candidate itemsets in the db and retain only those itemsets that meet the min. support

- Join: Generate candidate itemsets of length $(k+1)$ ^{by joining} pairs of frequent itemsets of $L=k$

- Prune: Eliminate candidate itemsets that contain subsets of length k with low support as they are not frequent.

Repeat: Repeat the process by incrementing the length of itemsets until no more frequent itemsets can be found

3. What is a minimum Support threshold? What is a minimum Confidence threshold?

Ans. Minimum Support Threshold: Sets the minimum occurrence Frequency for itemsets to be considered significant.

— Minimum Confidence Threshold: Establishes the minimum confidence level for association rules to be considered interesting.

4. How are frequent itemsets identified using a-priori algorithm?

Ans. Frequent itemsets are identified using A-priori algorithm by iteratively generating and checking candidate itemsets of increasing lengths, keeping only those that meet the minimum support threshold. This process continues until no more frequent itemsets can be found.

5. How are association rules generated using a-priori algorithm?

Ans. To generate association rules A-priori Algo —

1. Find Frequent itemsets in the database.
2. Create candidate rules, from these itemsets.
3. Calculate the Confidence for each rule.
4. keep rules that meet a minimum confidence threshold.
5. The retained rules are the generated association

~~Q~~ rules, Showing item associations in the database.

6. How can be a-priori algorithm be used to detect credit card fraud in real time?

Ans. Detecting credit card fraud in real time involves

1 - Data Collection: Gather transaction data in real time, including transaction amounts, user details.

2 - Feature Extraction: Extract relevant features from the transaction data.

3 - Model Building: - utilize machine learning models for rules based systems and current transaction data.

4 - Real time monitoring: Continuously monitoring incoming transactions and compare them to the established models and rules.

5 - Anomaly Detection: - Identify transactions that deviate significantly from established patterns.

6 - Alert Generation: Flag suspicious transactions for review.

7 - Fraud Prevention: - Take actions like blocking, notifying or encharing security.

~~Ans~~

```
In [1]: !pip install apyori
```

Requirement already satisfied: apyori in c:\users\91902\anaconda3\lib\site-packages
(1.1.2)

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from apyori import apriori
```

Read data and Display

```
In [3]: store_data = pd.read_csv("store_data.csv", header=None)
display(store_data.head())
print(store_data.shape)
```

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN
(7501, 20)											



```
In [4]: records = []
for i in range(1, 7501):
    records.append([str(store_data.values[i, j]) for j in range(0, 20)])
```

```
In [5]: print(type(records))

<class 'list'>
```

```
In [6]: association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lif
```

```
In [7]: #print(association_results)
```

How many relation derived

```
In [8]: print("There are {} Relation derived.".format(len(association_results)))
```

There are 48 Relation derived.

Association Rules Derived

```
In [9]: for i in range(0, len(association_results)):
    print(association_results[i][0])
```

```
frozenset({'chicken', 'light cream'})
frozenset({'mushroom cream sauce', 'escalope'})
frozenset({'pasta', 'escalope'})
frozenset({'herb & pepper', 'ground beef'})
frozenset({'tomato sauce', 'ground beef'})
frozenset({'olive oil', 'whole wheat pasta'})
frozenset({'shrimp', 'pasta'})
frozenset({'chicken', 'light cream', 'nan'})
frozenset({'shrimp', 'frozen vegetables', 'chocolate'})
frozenset({'cooking oil', 'spaghetti', 'ground beef'})
frozenset({'mushroom cream sauce', 'nan', 'escalope'})
frozenset({'pasta', 'nan', 'escalope'})
frozenset({'frozen vegetables', 'spaghetti', 'ground beef'})
frozenset({'frozen vegetables', 'olive oil', 'milk'})
frozenset({'shrimp', 'frozen vegetables', 'mineral water'})
frozenset({'frozen vegetables', 'olive oil', 'spaghetti'})
frozenset({'shrimp', 'frozen vegetables', 'spaghetti'})
frozenset({'frozen vegetables', 'spaghetti', 'tomatoes'})
frozenset({'grated cheese', 'spaghetti', 'ground beef'})
frozenset({'mineral water', 'herb & pepper', 'ground beef'})
frozenset({'herb & pepper', 'nan', 'ground beef'})
frozenset({'spaghetti', 'herb & pepper', 'ground beef'})
frozenset({'olive oil', 'milk', 'ground beef'})
frozenset({'tomato sauce', 'nan', 'ground beef'})
frozenset({'shrimp', 'spaghetti', 'ground beef'})
frozenset({'olive oil', 'spaghetti', 'milk'})
frozenset({'soup', 'mineral water', 'olive oil'})
frozenset({'olive oil', 'whole wheat pasta', 'nan'})
frozenset({'shrimp', 'pasta', 'nan'})
frozenset({'pancakes', 'olive oil', 'spaghetti'})
frozenset({'shrimp', 'frozen vegetables', 'chocolate', 'nan'})
frozenset({'cooking oil', 'spaghetti', 'nan', 'ground beef'})
frozenset({'frozen vegetables', 'spaghetti', 'nan', 'ground beef'})
frozenset({'frozen vegetables', 'mineral water', 'spaghetti', 'milk'})
frozenset({'olive oil', 'frozen vegetables', 'nan', 'milk'})
frozenset({'shrimp', 'frozen vegetables', 'mineral water', 'nan'})
frozenset({'frozen vegetables', 'olive oil', 'spaghetti', 'nan'})
frozenset({'shrimp', 'frozen vegetables', 'spaghetti', 'nan'})
frozenset({'frozen vegetables', 'spaghetti', 'nan', 'tomatoes'})
frozenset({'grated cheese', 'spaghetti', 'nan', 'ground beef'})
frozenset({'mineral water', 'herb & pepper', 'nan', 'ground beef'})
frozenset({'spaghetti', 'herb & pepper', 'nan', 'ground beef'})
frozenset({'olive oil', 'nan', 'milk', 'ground beef'})
frozenset({'shrimp', 'spaghetti', 'nan', 'ground beef'})
frozenset({'olive oil', 'nan', 'spaghetti', 'milk'})
frozenset({'soup', 'mineral water', 'olive oil', 'nan'})
frozenset({'pancakes', 'olive oil', 'spaghetti', 'nan'})
frozenset({'spaghetti', 'frozen vegetables', 'mineral water', 'milk', 'nan})
```

Rules Generated

In [10]:

```
for item in association_results:
    # first index of the inner list
    # Contains base item and add item
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + " -> " + items[1])

    # second index of the inner list
    print("Support: " + str(item[1]))

    # third index of the list located at 0th
```

```
# of the third index of the inner list

print("Confidence: " + str(item[2][0][2]))
print("Lift: " + str(item[2][0][3]))
print("=====")
```

```
Rule: chicken -> light cream
Support: 0.00453333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844
=====
Rule: mushroom cream sauce -> escalope
Support: 0.00573333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845
=====
Rule: pasta -> escalope
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287
=====
Rule: herb & pepper -> ground beef
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096
=====
Rule: tomato sauce -> ground beef
Support: 0.00533333333333333
Confidence: 0.37735849056603776
Lift: 3.840147461662528
=====
Rule: olive oil -> whole wheat pasta
Support: 0.008
Confidence: 0.2714932126696833
Lift: 4.130221288078346
=====
Rule: shrimp -> pasta
Support: 0.005066666666666666
Confidence: 0.3220338983050848
Lift: 4.514493901473151
=====
Rule: chicken -> light cream
Support: 0.00453333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844
=====
Rule: shrimp -> frozen vegetables
Support: 0.00533333333333333
Confidence: 0.23255813953488372
Lift: 3.260160834601174
=====
Rule: cooking oil -> spaghetti
Support: 0.0048
Confidence: 0.5714285714285714
Lift: 3.281557646029315
=====
Rule: mushroom cream sauce -> nan
Support: 0.00573333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845
=====
Rule: pasta -> nan
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287
=====
Rule: frozen vegetables -> spaghetti
Support: 0.008666666666666666
Confidence: 0.3110047846889952
Lift: 3.164906221394116
```

```
=====
Rule: frozen vegetables -> olive oil
Support: 0.0048
Confidence: 0.20338983050847456
Lift: 3.094165778526489
=====

Rule: shrimp -> frozen vegetables
Support: 0.0072
Confidence: 0.3068181818181818
Lift: 3.2183725365543547
=====

Rule: frozen vegetables -> olive oil
Support: 0.005733333333333333
Confidence: 0.20574162679425836
Lift: 3.1299436124887174
=====

Rule: shrimp -> frozen vegetables
Support: 0.006
Confidence: 0.21531100478468898
Lift: 3.0183785717479763
=====

Rule: frozen vegetables -> spaghetti
Support: 0.006666666666666667
Confidence: 0.23923444976076555
Lift: 3.497579674864993
=====

Rule: grated cheese -> spaghetti
Support: 0.005333333333333333
Confidence: 0.3225806451612903
Lift: 3.282706701098612
=====

Rule: mineral water -> herb & pepper
Support: 0.006666666666666667
Confidence: 0.390625
Lift: 3.975152645861601
=====

Rule: herb & pepper -> nan
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096
=====

Rule: spaghetti -> herb & pepper
Support: 0.0064
Confidence: 0.3934426229508197
Lift: 4.003825878061259
=====

Rule: olive oil -> milk
Support: 0.004933333333333333
Confidence: 0.22424242424242424
Lift: 3.411395906324912
=====

Rule: tomato sauce -> nan
Support: 0.005333333333333333
Confidence: 0.37735849056603776
Lift: 3.840147461662528
=====

Rule: shrimp -> spaghetti
Support: 0.006
Confidence: 0.5232558139534884
Lift: 3.004914704939635
=====

Rule: olive oil -> spaghetti
Support: 0.0072
Confidence: 0.20300751879699247
```

Lift: 3.0883496774390333

=====

Rule: soup -> mineral water

Support: 0.0052

Confidence: 0.2254335260115607

Lift: 3.4295161157945335

=====

Rule: olive oil -> whole wheat pasta

Support: 0.008

Confidence: 0.2714932126696833

Lift: 4.130221288078346

=====

Rule: shrimp -> pasta

Support: 0.005066666666666666

Confidence: 0.3220338983050848

Lift: 4.514493901473151

=====

Rule: pancakes -> olive oil

Support: 0.0050666666666666

Confidence: 0.20105820105820105

Lift: 3.0586947422647217

=====

Rule: shrimp -> frozen vegetables

Support: 0.0053333333333333

Confidence: 0.23255813953488372

Lift: 3.260160834601174

=====

Rule: cooking oil -> spaghetti

Support: 0.0048

Confidence: 0.5714285714285714

Lift: 3.281557646029315

=====

Rule: frozen vegetables -> spaghetti

Support: 0.0086666666666666

Confidence: 0.3110047846889952

Lift: 3.164906221394116

=====

Rule: frozen vegetables -> mineral water

Support: 0.0045333333333333

Confidence: 0.28813559322033905

Lift: 3.0224013274860737

=====

Rule: olive oil -> frozen vegetables

Support: 0.0048

Confidence: 0.20338983050847456

Lift: 3.094165778526489

=====

Rule: shrimp -> frozen vegetables

Support: 0.0072

Confidence: 0.3068181818181818

Lift: 3.2183725365543547

=====

Rule: frozen vegetables -> olive oil

Support: 0.0057333333333333

Confidence: 0.20574162679425836

Lift: 3.1299436124887174

=====

Rule: shrimp -> frozen vegetables

Support: 0.006

Confidence: 0.21531100478468898

Lift: 3.0183785717479763

=====

Rule: frozen vegetables -> spaghetti

Support: 0.006666666666666667

Confidence: 0.23923444976076555

Lift: 3.49759674864993

=====

Rule: grated cheese -> spaghetti

Support: 0.005333333333333333

Confidence: 0.3225806451612903

Lift: 3.282706701098612

=====

Rule: mineral water -> herb & pepper

Support: 0.006666666666666666

Confidence: 0.390625

Lift: 3.975152645861601

=====

Rule: spaghetti -> herb & pepper

Support: 0.0064

Confidence: 0.3934426229508197

Lift: 4.003825878061259

=====

Rule: olive oil -> nan

Support: 0.004933333333333333

Confidence: 0.2242424242424242

Lift: 3.411395906324912

=====

Rule: shrimp -> spaghetti

Support: 0.006

Confidence: 0.5232558139534884

Lift: 3.004914704939635

=====

Rule: olive oil -> nan

Support: 0.0072

Confidence: 0.20300751879699247

Lift: 3.0883496774390333

=====

Rule: soup -> mineral water

Support: 0.0052

Confidence: 0.2254335260115607

Lift: 3.4295161157945335

=====

Rule: pancakes -> olive oil

Support: 0.005066666666666666

Confidence: 0.20105820105820105

Lift: 3.0586947422647217

=====

Rule: spaghetti -> frozen vegetables

Support: 0.00453333333333334

Confidence: 0.28813559322033905

Lift: 3.0224013274860737

=====