**Appriori Algorithm**

Find strong/interesting rules/patterns from the following transaction table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL | Item | | | |
| 1 | 2 | 3 | 4 |
| 1 | Chicken | Ghee | Onion | Rice |
| 2 | Egg | Onion | Rice |  |
| 3 | Chicken | Ghee | Onion |  |
| 4 | Chicken | Egg | Rice |  |
| 5 | Chicken | Ghee | Rice |  |
| 6 | Egg | Rice |  |  |
| 7 | Chicken | Egg | Onion | Rice |
| 8 | Chicken | Onion | Rice |  |
| 9 | Chicken | Rice |  |  |
| 10 | Egg | Ghee | Rice |  |

Consider Minimum Support 50% and Minimum Confidence 60%.

**1st Iteration:**

Itemset, I = {Chicken, Egg, Ghee, Onion, Rice}

Calculating the Support for 1-itemsets:

|  |  |  |  |
| --- | --- | --- | --- |
| **SL** | **1-ItemSet** | **Count** | **Support** |
| 1 | Chicken | 7 | 0.7 |
| 2 | Egg | 5 | 0.5 |
| 3 | Ghee | 4 | 0.4 |
| 4 | Onion | 5 | 0.5 |
| 5 | Rice | 9 | 0.9 |

The items Chicken, Egg, Onion and Rice qualify.

**2nd Iteration:**

Itemset, I = {Chicken, Egg, Onion, Rice}

Calculating the support for 2-itemsets:

|  |  |  |  |
| --- | --- | --- | --- |
| **SL** | **2-ItemSet** | **Count** | **Support** |
| 1 | Chicken, Egg | 2 | 0.2 |
| 2 | Chicken, Onion | 4 | 0.4 |
| 3 | Chicken, Rice | 6 | 0.6 |
| 4 | Egg, Onion | 2 | 0.2 |
| 5 | Egg, Rice | 5 | 0.5 |
| 6 | Onion, Rice | 3 | 0.3 |

The items Chicken, Egg and Rice qualify.

**3rd Iteration:**

Itemset, I = {Chicken, Egg, Rice}

Calculating the support for 2-itemsets:

|  |  |  |  |
| --- | --- | --- | --- |
| **SL** | **3-ItemSet** | **Count** | **Support** |
| 1 | Chicken, Egg, Rice | 2 | 0.2 |

None of the items qualify. So, we will use the item-sets from the 2nd iteration to generate association rules.

Association Rule generation:

Association Rules are generated from the item-sets that qualifies the minimum support condition. Item-set I3 and I5 qualifies from the 2nd iteration. We need to prepare subsets for I3 and I5.

Subsets of I3, S1 = {{Chicken, Rice}, {Chicken}, {Rice}, {}}

Subsets of I5, S2 = {{Egg, Rice}, {Egg}, {Rice}, {}}

The all-element set and empty set will not be considered for rule generation.

An association rule generated from a subset S is represented as L->R, where L is an element of S and R = S-L. We need to generate rules for all possible value of L.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **L** | **->** | **R=S-L** | **Count(S)** | **Support(S)** | **Count(L)** | **Support(L)** | **Confidence** | **Strong** |
| Chicken | **->** | Rice | 6 | 0.6 | 7 | 0.7 | 0.857142857 | Yes |
| Rice | **->** | Chicken | 6 | 0.6 | 9 | 0.9 | 0.666666667 | Yes |
| Egg | **->** | Rice | 5 | 0.5 | 5 | 0.5 | 1 | Yes |
| Rice | **->** | Egg | 5 | 0.5 | 9 | 0.9 | 0.555555556 | No |

**Frequent Pattern Growth Tree Algorithm**

Find strong/interesting rules/patterns from the following transaction table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL | Item | | | |
| 1 | 2 | 3 | 4 |
| 1 | Chicken | Ghee | Onion | Rice |
| 2 | Egg | Onion | Rice |  |
| 3 | Chicken | Ghee | Onion |  |
| 4 | Chicken | Egg | Rice |  |
| 5 | Chicken | Ghee | Rice |  |
| 6 | Egg | Rice |  |  |
| 7 | Chicken | Egg | Onion | Rice |
| 8 | Chicken | Onion | Rice |  |
| 9 | Chicken | Rice |  |  |
| 10 | Egg | Ghee | Rice |  |

Consider Minimum Support 50% and Minimum Confidence 60%.

**1st Step:**

Itemset, I = {Chicken, Egg, Ghee, Onion, Rice}

Calculating the Support for 1-itemsets:

|  |  |  |  |
| --- | --- | --- | --- |
| **SL** | **1-ItemSet** | **Count** | **Support** |
| 1 | Chicken | 7 | 0.7 |
| 2 | Egg | 5 | 0.5 |
| 3 | Ghee | 4 | 0.4 |
| 4 | Onion | 5 | 0.5 |
| 5 | Rice | 9 | 0.9 |

The items Chicken, Egg, Onion and Rice qualify.

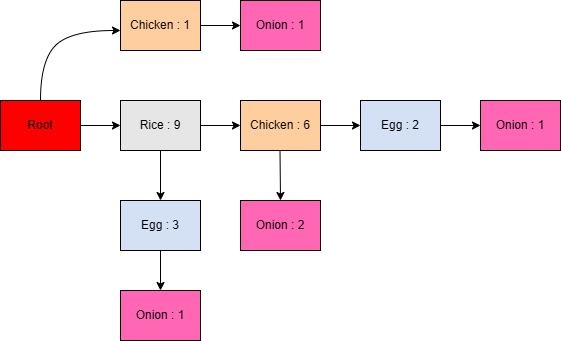
**2nd Step:**

Sorted Itemset, I = {Rice: 9, Chicken: 7, Egg: 5, Onion: 5}

Update the transaction table according to the sorted itemset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL | Item | | | |
| 1 | 2 | 3 | 4 |
| 1 | Rice | Chicken | Onion |  |
| 2 | Rice | Egg | Onion |  |
| 3 | Chicken | Onion |  |  |
| 4 | Rice | Chicken | Egg |  |
| 5 | Rice | Chicken |  |  |
| 6 | Rice | Egg |  |  |
| 7 | Rice | Chicken | Egg | Onion |
| 8 | Rice | Chicken | Onion |  |
| 9 | Rice | Chicken |  |  |
| 10 | Rice | Egg |  |  |

Draw FP Tree from the updated transaction table:



**3rd Step:**

For each item, create conditional frequent pattern and conditional frequent pattern tree:

|  |  |  |
| --- | --- | --- |
| **Item** | **Conditional Frequent Pattern** | **Conditional Frequent Pattern Tree** |
| Onion | {Rice, Chicken: 2}, {Rice, Egg: 1},  {Rice, Chicken, Egg: 1}, {Chicken: 1} |  |
| Egg | {Rice, Chicken: 2}, {Rice: 3} | {Rice: 5} |
| Chicken | {Rice: 6} | {Rice: 6} |
| Rice |  |  |

[Note: Conditional Frequent Pattern for an item is the path to item from root and the value is the count of the item. Conditional Frequent Pattern Tree for an item is the common element among the patterns of that item and the value is the sum of the item.]

|  |  |
| --- | --- |
| **Item** | **Generated Frequent Pattern** |
| Onion |  |
| Egg | {<Rice, Egg : 5>} |
| Chicken | {<Rice, Chicken : 6>} |
| Rice |  |

[Note: ]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **L** | **->** | **R=S-L** | **Count(S)** | **Support(S)** | **Count(L)** | **Support(L)** | **Confidence** | **Strong** |
| Chicken | **->** | Rice | 6 | 0.6 | 7 | 0.7 | 0.857142857 | Yes |
| Rice | **->** | Chicken | 6 | 0.6 | 9 | 0.9 | 0.666666667 | Yes |
| Egg | **->** | Rice | 5 | 0.5 | 5 | 0.5 | 1 | Yes |
| Rice | **->** | Egg | 5 | 0.5 | 9 | 0.9 | 0.555555556 | No |