**APRIORI ALGORITHM**

Nowadays we purchase all things online leading to enormous growth of data only in months not even in years. Apriori algorithm is a classical algorithm used in the field of machine learning, which is used for mining frequent itemsets and relevant association rule from the database.

**Association rule:**

It is used in identifying the relationship among dataset variables. For example, if the shops keeps bread and jam as a combo with a discount, there is a possibility of buying both rather than keeping them in separate or different places. Likewise association rule mining picks up items from the database based on the frequency of items purchased and the related item bought along with the previous items. This is can be technically termed as follows:

Let I=\{i_1, i_2, i_3, ..., i_n\}be a set of n attributes called items and D=\{t_1, t_2, ..., t_n\}be the set of transactions. It is called database. Every transaction, t_iin has a unique transaction ID, and it consists of a subset of itemsets in I

**Supermarket dataset example:**

Let us consider a simple dataset containing the following items and six transactions since in real case there might millions of transaction. Each transaction has tuples (record) with 0’s (absence) and 1’s (presence) of items.

I = {Onion, Potato, Burger, Milk, Beer}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Transaction ID** | **Onion** | **Potato** | **Burger** | **Milk** | **Beer** |
| t1 | 1 | 1 | 1 | 0 | 0 |
| t2 | 0 | 1 | 1 | 1 | 0 |
| t3 | 0 | 0 | 0 | 1 | 1 |
| t4 | 1 | 1 | 0 | 1 | 0 |
| t5 | 1 | 1 | 1 | 0 | 1 |
| t6 | 1 | 1 | 1 | 1 | 1 |

From the above table we can form a rule like {onion, potato} => {burger}, it tells if a person buys onion and potato there is high chance of buying burger bun. Another rule can be {potato, burger} => {milk}, it depicts if a person buys potato and burger bun he will surely buy milk. Various rules can be devised based on various measures and constraints. The following are the various measures used:

* Support
* Confidence
* Lift
* Conviction

**Support:**

It implies that, for how many transactions a particular itemset (X) from the dataset has occurred. It gives the proportion of a transaction of an itemset (X) under consideration.

**Supp(X) = Number of transactions in which X appears**

**Total number of transactions**

From the above table we can perform the support measure as follows:

Supp (onion) = 4/6 = 0.66667

Supp (potato) = 5/6 = 0.83333

Supp (Burger) = 4/6 = 0.66667

Supp (Milk) = 4/6 = 0.66667

Supp (Beer) = 3/6 = 0.5

From the above support measure we can clearly understand that potato has more popularity than other items. Hence it is a profit and can be set as a support threshold for identifying significant itemsets.

**Support threshold = 0.66667**

**Working of Apriori algorithm with illustration:**

**Step 1:** Create a frequency table containing all items that occurs in all transactions

|  |  |
| --- | --- |
| **Items** | **Frequency** |
| Onion (O) | 4 |
| Potato (P) | 5 |
| Burger (B) | 4 |
| Milk (M) | 4 |
| Beer (Be) | 2 |

**Step 2:** For our example we have set a threshold value as 0.7. Hence from the above frequency table eliminate items that have threshold value less than 0.7. Here Beer has 0.5 as threshold value. So we are eliminating beer from our table.

|  |  |
| --- | --- |
| **Items** | **Frequency** |
| Onion (O) | 4 |
| Potato (P) | 5 |
| Burger (B) | 4 |
| Milk (M) | 4 |

**Step 3:** Now pair all possible combinations from the table as follows:

OP, OB, OM, PB, PM, BM. Since AB and BA are same no need to pair up PO, BO, MO, BP, MP, and MB. This will result items purchased in pair.

|  |  |
| --- | --- |
| **Items** | **Frequency** |
| OP | 4 |
| OB | 3 |
| OM | 2 |
| PB | 4 |
| PM | 3 |
| BM | 2 |

**Step 4:** Now eliminate the item pairs having minimum threshold from the above table. Here OM and BM are only having minimum threshold. Hence we eliminate those pairs as follows:

|  |  |
| --- | --- |
| **Items** | **Frequency** |
| OP | 4 |
| OB | 3 |
| PB | 4 |
| PM | 3 |

**Step 5:** Now we have to create 3 itemset rules, to find out which three items are purchased together. In order to do that we have to consider pairs having first letter as same.

**O**P AND **O**B => OPB

**P**B AND **P**M => PBM

Now find the frequency of those three items set.

|  |  |
| --- | --- |
| **Items** | **Frequency** |
| OPM | 4 |
| PBM | 3 |

**Step 6:** Eliminate the item having minimum threshold which from our Table is PBM. **Hence the OPB combination is mostly bought by the Customers when three itemsets are considered.**