Chapter 5 : Part 1

**Association Rule Learning**

**Apriori**

**5.1.1 Association Rule Learning**

Association rule learning is a type of Unsupervised Learning Technique that checks for the DEPENDENCY of *one data-item* on *another data-item* and ***maps*** accordingly so that it can be more ***profitable***.

* It tries to find some interesting relations or associations among the variables of dataset. It is based on different rules to discover the *interesting* *relations* between variables in the database.

|  |  |
| --- | --- |
| * It is employed in-  1. Market Basket analysis: For example, if a customer buys *bread*, he most likely can also buy *butter*, *eggs*, or *milk*, so these products are stored *within a shelf* or *mostly* *nearby*. 2. Web usage mining 3. Recommendation system 4. Continuous production, etc.  * Here market basket analysis is a technique used by the various big retailer to discover the *associations between items*. We can understand it by taking an example of a supermarket, as in a supermarket, *all products* that are *purchased* *together* are *put* *together*. | Association Rule Learning |

* Association rule learning can be divided into three types of algorithms:

1. Apriori
2. Eclat
3. F-P Growth Algorithm

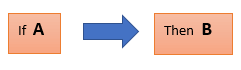
* Diapers and Beer Example: Very often during certain times of the day, when people shop in the afternoon between 6 and 9 p.m.



* People who buy *diapers* also buy *beer.* Which makes no sense at first but a plausible explanation might be: when the husband gets home and husband and the wife are taking care of their baby.
* They sometimes find that they run out of diapers and most of the time, the husband has to go pick up the diapers. While he's picking up the Diapers because it's really after hours after work, he also picks up some Beer.
* And based on that you can decide how to arrange products in your store.
* So some stores might decide to put these two items (Beer & Diapers) closer to *entice* *people* to *buy a beer when they're buying diapers*.
* But actually a lot of stores do the opposite. For example, you probably noticed this from your convenience store that they try to separate bread and milk as far as possible.
* Because that way they really know that *these two products are bought together*. And so you actually have to *walk through the whole store* to pick up.
* So you've picked up your bread and then to get to the milk you have to get all the way through the whole store to the completely opposite corner of the store.
* When you're *walking through* the store you *see more other products* and you're more likely to pick up an ***additional item*** that you ***weren't actually planning on buying*** when you got to the store in the first place.
* So there's a lot of interesting marketing tactics that are used based on this data.

**5.1.2 How does Association Rule Learning work?**

Association rule learning works on the concept of ***If*** and ***Else*** Statement, such as **if** A **then** B.

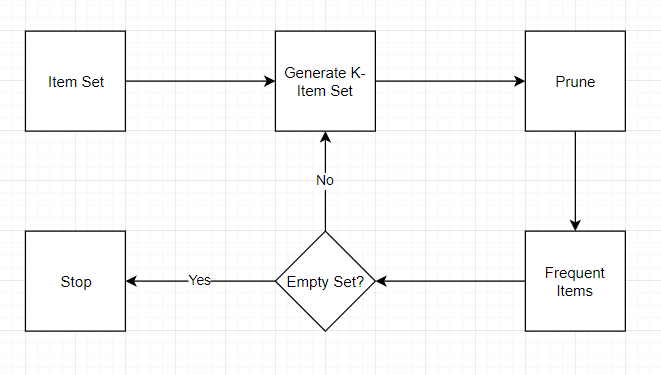


* Here the *If* *element* is called Antecedent, and *then* *statement* is called as Consequent. These types of relationships where we can find out some ***association or relation*** between ***two items*** is known as Single Cardinality.
* It is all about *creating rules*, and if the *number of items* *increases*, then *cardinality* also *increases* accordingly.

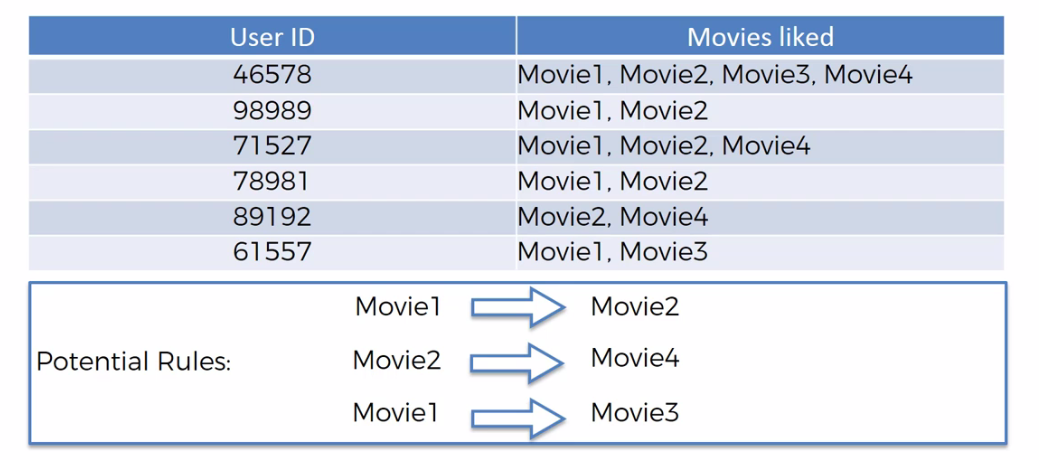
**5.1.3 Apriori Algorithm**

The *Apriori* algorithm uses *frequent itemsets* to generate *association* *rules*, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is the *iterative* process for finding the frequent itemsets from the large dataset.

* This algorithm was given by the *R. Agrawal* and *Srikant* in the year 1994. It is mainly used for *market basket analysis* and helps to ***find those products that can be bought together***. It can also be used in the ***healthcare*** field to find ***drug*** ***reactions*** for patients.
* The control flow diagram for the Apriori algorithm



* Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data.
* The algorithm terminates when no further successful extensions are found.
* Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length from item sets of length . Then it prunes the candidates which have an infrequent sub pattern.
* According to the downward closure lemma, the candidate set contains all frequent -length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.
* Apriori is all about:
* People who bought something also bought something else or
* Watched something also watched something else
* Did something also did something else
* This whole *association* *rule* *learning* part is all about analyzing when things come in *pairs* or in *triplicates* or in certain *sequence* but they are combined together for some reason looking for those rules (reasons, finding specific pattern) and those ways that this happens.
* Consider the following example:



* From above we can easily tell that there are some potential rules. For example everybody who watches Movie1 also like Movie2. People who like Movie2 also like Movie4. And people who like Movie1 are also quite likely to be like Movie3.
* But from those rules some rules are "**strong**" and some rules are "**weak**". We want to find the *very* *strong* *ones* in order to build our *business decisions* or other decisions on those rules.
* We don't want to go to the people and asking their opinions instead we extract those information/rules from our data. So if we have a large ***sample*** ***size*** say, ***50000*** or ***500000*** people then by we're analyzing that data we can come up with some quite solid rules.

**5.1.4 How Does Apriori algorithm works**

The apriori algorithm has three parts to it: the ***support***, the ***confidence*** and the ***lift***.

* Calculation metrices: So, to measure the associations between ***thousands*** of ***data-items***, there are ***several*** ***metrics***. These metrics are:

|  |  |  |
| --- | --- | --- |
| 1. Support | 1. Confidence | 1. Lift |

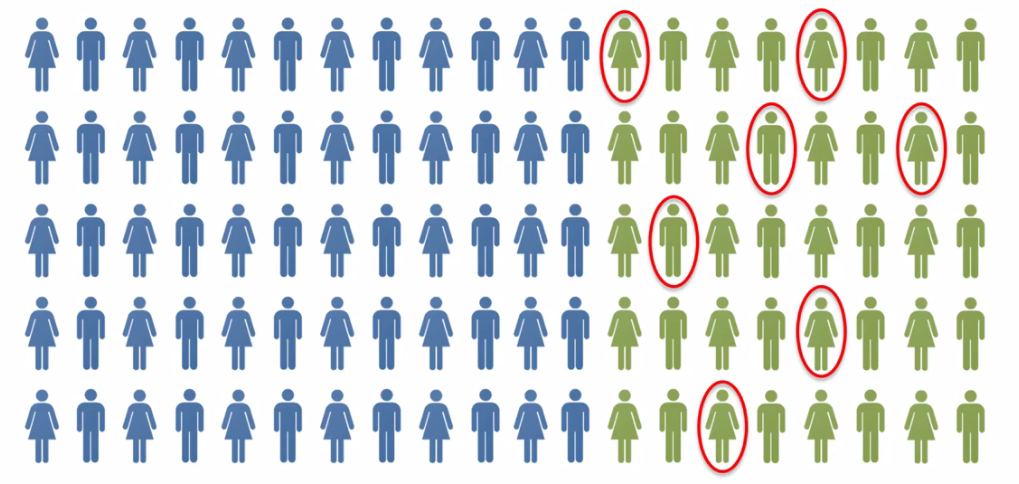
1. Support: We're going to start up with the support and you will see that it's very similar to the way we talked about the ***Naïve Bayes classifiers***.

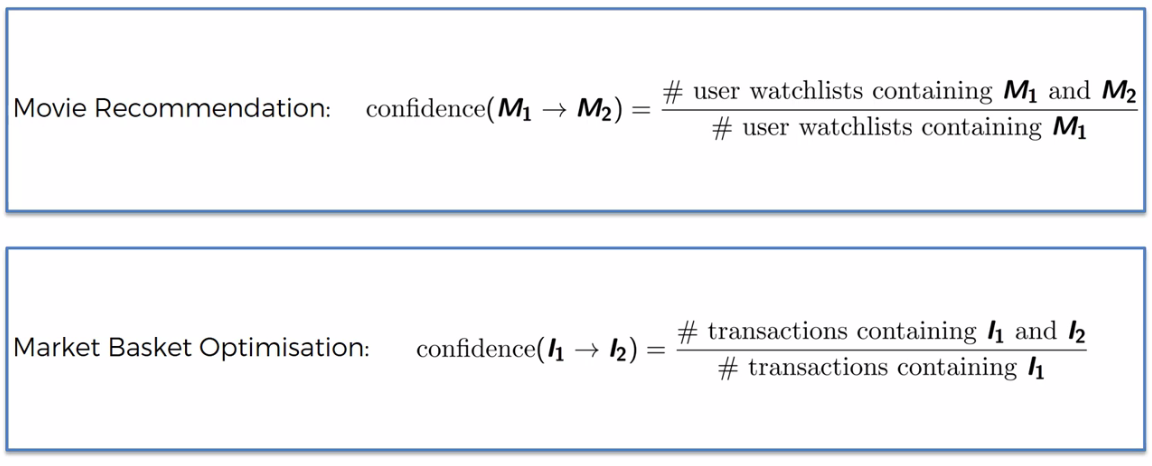
|  |  |
| --- | --- |
| * Support is the ***frequency*** of ***A*** or ***how frequently an item appears in the dataset***. It is defined as the fraction of the ***transaction T*** that contains the itemset ***X***. If there are ***X*** ***datasets***, then for ***transactions*** ***T***, it can be written as: |  |

* For example: Say from ***100 people***, ***10*** ***people*** watched ***Ex-Machina***, then **,**

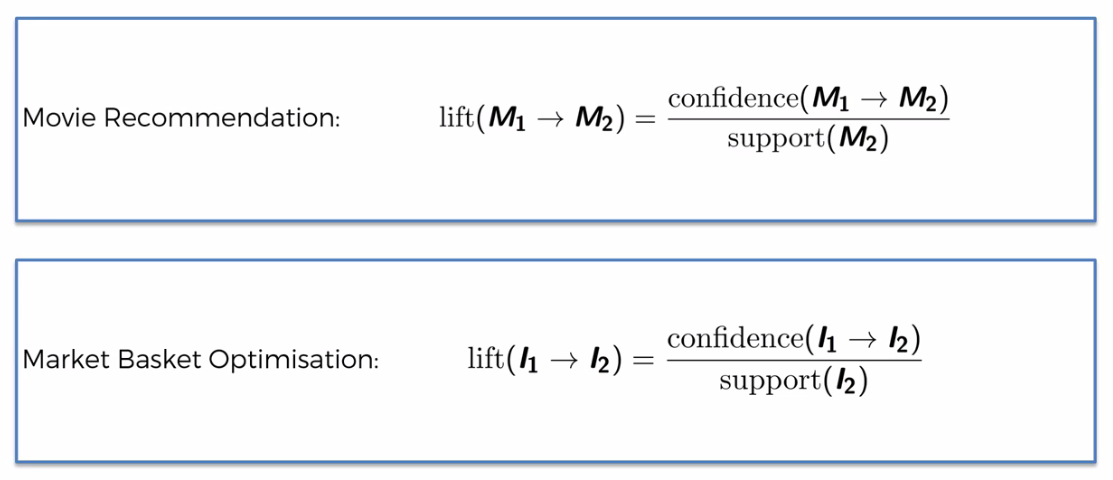
1. Confidence: Confidence indicates how often the rule has been found to be ***true***. Or, how often the ***items*** ***X*** and ***Y*** occur together in the dataset when the occurrence of ***X*** is already given. It is the ratio of the ***transaction*** that contains ***X*** and ***Y*** to the number of ***records*** that contain ***X***.

* For example: Now consider we are testing a rule, ***"People watched*** Intersteller ***are also watched Ex-Machina "***. So in the following diagram say ***green people watched*** Intersteller(the number is ***40 among 100***), and ***7*** of them watched ***Ex-Machina***, then the confidence is:

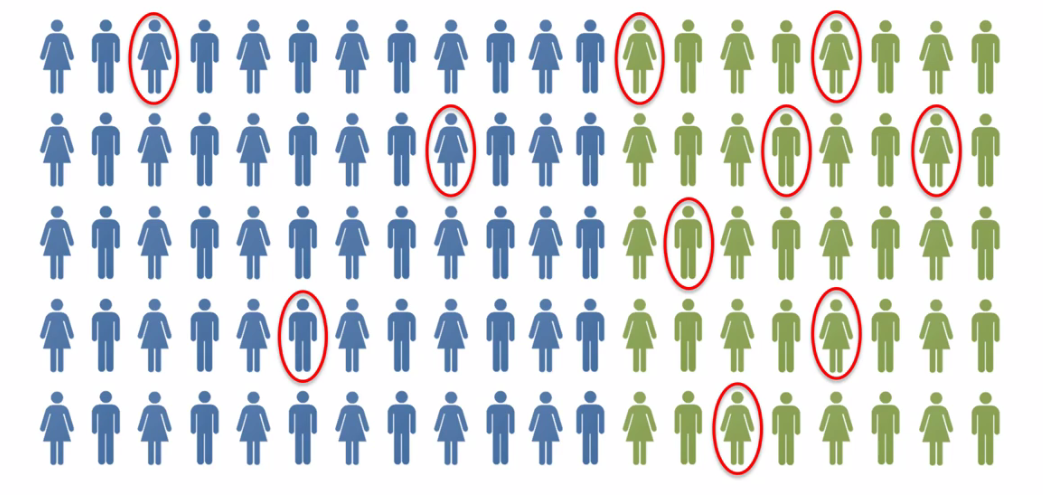




1. Lift: Lift is very similar to the ***Naïve Bayes classifiers***. Lift is basically is the ***ratio*** of ***Confidence*** and ***Support***. It is the *strength of any rule*, which can be defined as below formula:

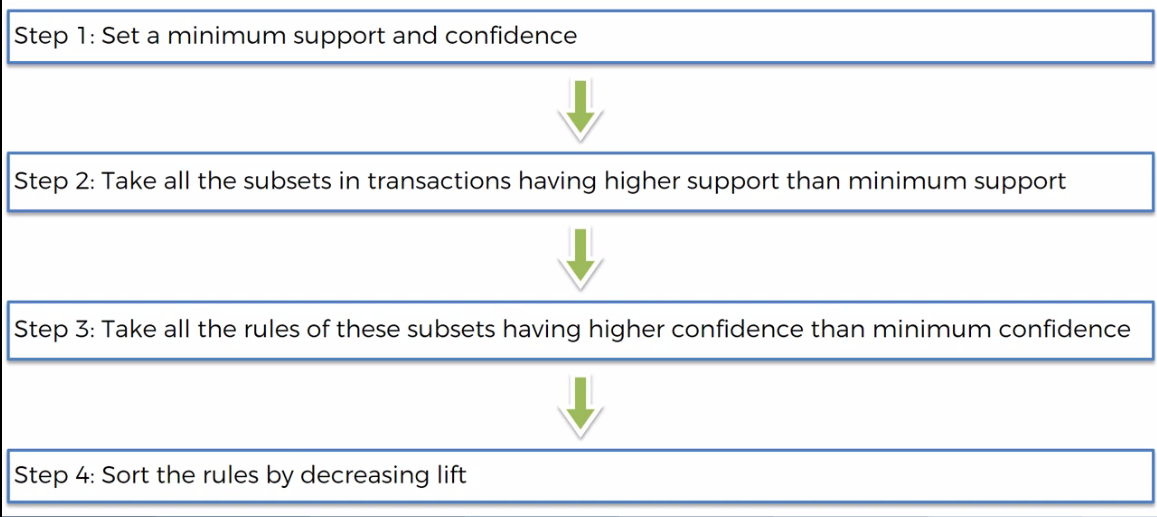


* Example: So in below ***Green*** people watched Interstellerand ***Red-circle*** represents the people watched ***Ex-Machina***.



* Out of this population we know that 10% actually likes ***Ex-Machina.*** So if we take another random population and then what is the ***likelihood that if we recommend to a random person*** in that brand ***new*** ***population*** will recommend the ***Ex-Machina*** movie, what is the likelihood that they will like it.
* Well the likelihood is 10%. But now the question is: Can we prove that result by using some prior knowledge. That's why the algorithm is called Apriori.
* In that new population let's only recommend ***Ex-Machina*** to people who have already seen ***Interstellar***. In that case the likelihood as we've calculated out of the Green people ***17.5%*** actually liked ***Ex-Machina***. So the ***Lift*** is the improvement in your prediction.
* So, out of your new population, if you first ask the question "Have you seen and liked interstellar?". If they say "yes" and then you recommend "***Ex-Machine***" the likelihood of a successful recommendation is ***17.5%***. So the lift is by definition is ***1.75***.

**5.1.5 Steps in Association Rule Learning**



* Apriori is actually quite a slow algorithm because it just goes through all of these different combinations of rules. For example it will calculate rules for *Movie1*, *Movie2*, *Movie3*, *Movie4*, … so on for pair (*Movie1* and *Movie2*) or triplet (*Movie1* and *Movie2 and Movie3*) so on.
* So there is so many different rules and we pick only strong rules. The rule with ***highest*** ***lift*** is the ***strongest*** rule.

**5.1.6 Types of Association Rule Learning**

1. Apriori Algorithm: This algorithm uses frequent datasets to generate association rules. It is designed to work on the databases that contain transactions. This algorithm uses a ***Breadth-First Search*** and ***Hash Tree*** to calculate the itemset efficiently.

* It is mainly used for market basket analysis and helps to understand the products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

1. Eclat Algorithm: *Eclat algorithm* stands for *Equivalence Class Transformation*. This algorithm uses a *depth-first search* technique to find *frequent* *itemsets* in a *transaction* database. It performs *faster* *execution* than *Apriori* Algorithm.
2. F-P Growth Algorithm: The ***F-P growth*** algorithm stands for ***Frequent Pattern***, and it is the ***improved*** ***version*** of the ***Apriori*** Algorithm. It represents the ***database*** in the form of a ***tree structure*** that is known as a ***frequent*** ***pattern*** or ***tree***. The purpose of this ***frequent tree*** is to ***extract*** the most ***frequent*** ***patterns***.

* Applications of Association Rule Learning: It has various applications in machine learning and data mining. Below are some popular applications of association rule learning:
* Market Basket Analysis: It is one of the popular examples and applications of association rule mining. This technique is commonly used by big *retailers* to determine the *association between items*.
* Medical Diagnosis: With the help of association rules, patients can be cured easily, as it helps in *identifying the probability of illness* for a particular disease.
* Protein Sequence: The association rules help in determining the synthesis of artificial Proteins.
* It is also used for the *Catalog* *Design* and *Loss-leader* *Analysis* and many more other applications.
* What is Frequent Itemset: Frequent itemsets are those items whose ***Support*** is greater than the ***Threshold*** ***Value*** or user-specified ***Minimum*** ***Support***. It means if ***A*** & ***B*** are the frequent ***itemsets*** ***together***, then ***individually*** ***A*** and ***B*** should also be the ***frequent*** ***itemset***.
* Suppose there are the two transactions: , and, in these two transactions, ***2*** and ***3*** are the ***frequent*** ***itemsets***.
* Advantages of Apriori Algorithm
* This is easy to understand algorithm
* The ***join*** and ***prune*** steps of the algorithm can be easily implemented on large datasets.
* Disadvantages of Apriori Algorithm
* The Apriori algorithm works Slow compared to other algorithms.
* The overall ***performance*** can be ***reduced*** as it scans the database for ***multiple times***.
* The timecomplexityand spacecomplexityof the Apriori algorithm is ***O(2D),*** which is ***very high***. Here ***D*** represents the horizontalwidthpresent in the database.