

ASSOCIATION ANALYSIS

UNIT-II

Basic Concepts

Mining Frequent Patterns:

Item Set:

An Item set is collection or set of items

Examples:

{Computer, Printer, MSOffice} is **3 item set**

{Milk, Bread} is **2 item set**

Similarly,

Set of K items is called **k item set**

Frequent patterns/Itemset:

These are patterns that appear **frequently** in a data set. Patterns may be item sets, or sub sequences.

Example: Transaction Database (Dataset)

TID	Items
T1	Bread, Coke, Milk
T2	Popcorn, Bread
T3	Bread, Egg, Milk
T4	Egg, Bread, Coke, Milk

A set of items, such as **Milk & Bread** that appear together in a *transaction data set* (Also called as **Frequent Item set**).

Frequent item set mining leads to the discovery of *associations* and *correlations* among items in large transactional (or) relational data sets.

Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data **classification**, **clustering**, and other data mining tasks.

Market Based Analysis:

Association rule mining (or) Frequent item set mining finds interesting *associations* and relationships (*correlations*) in large transactional or relational data sets. This rule shows how frequently an item set occurs in a transaction. A typical example is *Market Based Analysis*.

Market Based Analysis is one of the key techniques used by large relations to show associations between items. It allows retailers to identify relationships between the items that people buy together frequently.

This process analyzes customer buying habits by finding associations between the different items that customers place in their “*shopping baskets*.”

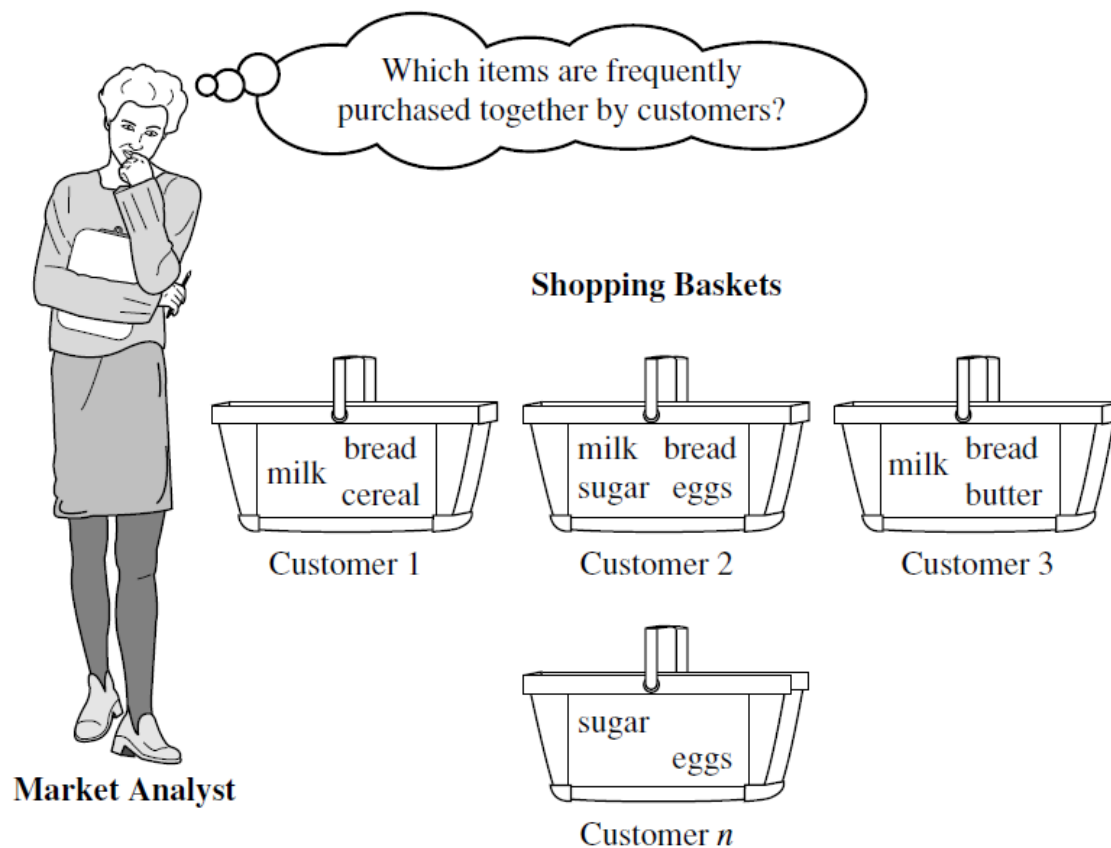


Fig: Market basket analysis

The discovery of these associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. *For instance*, if customers are buying **milk**, how likely are they to also buy **bread** (and what kind of bread) on the same trip to the supermarket? This information can lead to increased sales by helping retailers do *selective marketing* and *plan their shelf space*.

Understanding these buying patterns can help to increase sales in several ways. If there is a pair of items, X and Y, which are frequently bought together:

- Both X and Y can be placed on the same shelf, so that buyers of one item would be prompted to buy the other.
- Promotional discounts could be applied to just one out of the two items.
- Advertisements on X could be targeted at buyers who purchase Y.
- X and Y could be combined into a new product, such as having Y in flavours of X.

Association rule: If there is a pair of items, X and Y, which are frequently bought together then association rule is represented as $X \Rightarrow Y$.

For example, the information that customers who purchase *computers* also tend to buy *antivirus software* at the same time is represented as

Computer $\Rightarrow$ Antivirus_Software

Measures to discover interestingness of association rules:

Association rules analysis is a technique to discover how items are associated to each other. There are three measure to discover interestingness of association rules. Those are:

Support: The support of an item / item set is the *number of transactions* in which the item / item set appears, divided by the *total number of transactions*.

Formula:

$$\text{Support}\{A\} = \frac{\text{Frequency}\{A\}}{N}$$

(or)

$$\text{Support}\{A,B\} = \frac{\text{Frequency}\{A,B\}}{N}$$

Where, *A*, *B* are items and *N* is the total number of transactions.

Example:

TID	ITEMS
T1	Bread, Coke, Milk
T2	Popcorn, Bread
T3	Bread, Egg, Milk
T4	Egg, Bread, Coke, Milk
T5	Egg, Apple

Table 1: Example Transactions

Eg: Support of item *Coke*:

$$\begin{aligned}\text{Support } \{\text{Coke}\} &= \text{frequency } \{\text{Coke}\} / N \\ &= 2 / 5 = 0.4 \text{ i.e., } 40\%\end{aligned}$$

Eg: Support of item set *Bread, Milk*:

$$\begin{aligned}\text{Support } \{\text{Bread, Milk}\} &= \text{frequency } \{\text{Bread, Milk}\} / N \\ &= 3 / 5 = 0.6 \text{ i.e., } 60\%\end{aligned}$$

Confidence: This says that how likely item **B** is purchased when item **A** is purchased, expressed as $\{A \rightarrow B\}$. The Confidence of items (**A** and **B**) is the *frequency or number of transactions* in which the items (**A** and **B**) appear, divided by the *frequency or number of transactions* in which the item (**A**) appears.

Formula:

$$\text{Confidence}\{A \rightarrow B\} = \frac{\text{Frequency}\{A, B\}}{\text{Frequency}\{A\}}$$

(or)

$$\text{Confidence}\{A \rightarrow B\} = \frac{\text{Support}\{A, B\}}{\text{Support}\{A\}}$$

Example:

From the *Table 1*, the confidence of $\{\text{Bread} \rightarrow \text{Milk}\}$ is

$$\begin{aligned}\text{Confidence } \{\text{Bread} \rightarrow \text{Milk}\} &= \text{frequency } \{\text{Bread, Milk}\} / \text{frequency } \{\text{Bread}\} \\ &= 3 / 4 = 0.75 \text{ i.e., } 75\%\end{aligned}$$

(Or)

The **confidence** of $\{Bread \rightarrow Milk\}$ is

$$\text{Support } \{Bread, Milk\} = 3 / 5 = 0.6$$

$$\text{Support } \{Bread\} = 4 / 5 = 0.8$$

$$\begin{aligned}\text{Confidence } \{Bread \rightarrow Milk\} &= \text{Support } \{Bread, Milk\} / \text{Support } \{Bread\} \\ &= 0.6 / 0.8 \\ &= 0.75 \text{ i.e., } 75\%\end{aligned}$$

Lift: This says that how likely item **B** is purchased when item **A** is purchased, expressed as an association rule $\{A \rightarrow B\}$. The *lift* is a measure to *predict* the performance of an association rule (targeting model).

If *lift value* is:

- **greater than 1** means that item **B** is **likely** to be bought if item **A** is bought,
- **less than 1** means that item **B** is **unlikely** to be bought if item **A** is bought,
- **equals to 1** means there is **no association** between items (**A** and **B**).

Formula:

$$\text{Lift}\{A \rightarrow B\} = \frac{\text{Support}\{A, B\}}{\text{Support}\{A\} * \text{Support}\{B\}}$$

Example:

From the *Table 1*, the *lift* of $\{Bread \rightarrow Milk\}$ is

$$\text{Support } \{Bread, Milk\} = 3 / 5 = 0.6$$

$$\text{Support } \{Bread\} = 4 / 5 = 0.8$$

$$\text{Support } \{Milk\} = 3 / 5 = 0.6$$

$$\begin{aligned}\text{Lift } \{Bread \rightarrow Milk\} &= \text{Support } \{Bread, Milk\} / \text{Support } \{Bread\} * \text{Support } \{Milk\} \\ &= 0.6 / 0.8 * 0.6 \\ &= 0.6 / 0.42 \\ &= 1.25\end{aligned}$$

The Lift value is **greater than 1** means that item **Milk** is **likely** to be bought if item **Bread** is bought.

Example: To find **Support**, **Confidence** and **Lift** measures on the following transactional data set.

TID	ITEMS
T1	Bread, Milk
T2	Bread, Diaper, Burger, Eggs
T3	Milk, Diaper, Burger, Coke
T4	Bread, Milk, Diaper, Burger
T5	Bread, Milk, Diaper, Coke

Table 2: Example Transactions

Number of transactions = **5**.

Support:

1 – Item Set:

$$\text{Support } \{\text{Bread}\} = 4 / 5 = 0.8 = 80\%$$

$$\text{Support } \{\text{Diaper}\} = 4 / 5 = 0.8 = 80\%$$

$$\text{Support } \{\text{Milk}\} = 4 / 5 = 0.8 = 80\%$$

$$\text{Support } \{\text{Burger}\} = 3 / 5 = 0.6 = 60\%$$

$$\text{Support } \{\text{Coke}\} = 2 / 5 = 0.4 = 40\%$$

$$\text{Support } \{\text{Eggs}\} = 1 / 5 = 0.2 = 20\%$$

2 – Item Set:

$$\text{Support } \{\text{Bread, Milk}\} = 3 / 5 = 0.6 = 60\%$$

$$\text{Support } \{\text{Milk, Diaper}\} = 3 / 5 = 0.6 = 60\%$$

$$\text{Support } \{\text{Milk, Burger}\} = 2 / 5 = 0.4 = 40\%$$

$$\text{Support } \{\text{Burger, Coke}\} = 1 / 5 = 0.2 = 20\%$$

$$\text{Support } \{\text{Milk, Eggs}\} = 0 / 5 = 0.0 = 0\%$$

3 – Item Set:

$$\text{Support } \{\text{Bread, Milk, Diaper}\} = 2 / 5 = 0.4 = 40\%$$

$$\text{Support } \{\text{Milk, Diaper, Burger}\} = 2 / 5 = 0.4 = 40\%$$

Confidence:

$$\begin{aligned}\text{Confidence } \{Bread \rightarrow Milk\} &= \text{Support } \{Bread, Milk\} / \text{Support } \{Bread\} \\ &= 0.6 / 0.8 \\ &= 0.75 \text{ i.e., } 75\%.\end{aligned}$$

$$\begin{aligned}\text{Confidence } \{Milk \rightarrow Diaper\} &= \text{Support } \{Milk, Diaper\} / \text{Support } \{Milk\} \\ &= 0.6 / 0.8 \\ &= 0.75 \text{ i.e., } 75\%.\end{aligned}$$

$$\begin{aligned}\text{Confidence } \{Milk \rightarrow Burger\} &= \text{Support } \{Milk, Burger\} / \text{Support } \{Milk\} \\ &= 0.4 / 0.8 \\ &= 0.5 \text{ i.e., } 50\%.\end{aligned}$$

$$\begin{aligned}\text{Confidence } \{Burger \rightarrow Coke\} &= \text{Support } \{Burger, Coke\} / \text{Support } \{Burger\} \\ &= 0.2 / 0.6 \\ &= 0.33 \text{ i.e., } 33.3\%.\end{aligned}$$

$$\begin{aligned}\text{Confidence } \{Bread, Milk \rightarrow Diaper\} &= \text{Support } \{Bread, Milk, Diaper\} / \text{Support } \{Bread, Milk\} \\ &= 0.4 / 0.6 \\ &= 0.66 \text{ i.e., } 66.6\%.\end{aligned}$$

$$\begin{aligned}\text{Confidence } \{Milk, Diaper \rightarrow Burger\} &= \text{Support } \{Milk, Diaper, Burger\} / \text{Support } \{Milk, Diaper\} \\ &= 0.4 / 0.6 \\ &= 0.66 \text{ i.e., } 66.6\%.\end{aligned}$$

Lift:

$$\text{Lift } \{Bread \rightarrow Milk\} =$$

$$\begin{aligned} & \text{Support } \{Bread, Milk\} / (\text{Support } \{Bread\} * \text{Support } \{Milk\}) \\ &= 0.6 / (0.8 * 0.8) \\ &= 0.93 \end{aligned}$$

While lift value less than 1 means that item '**Milk**' is unlikely to be bought if item '**Bread**' is bought.

$$\text{Lift } \{Milk \rightarrow Burger\} =$$

$$\begin{aligned} & \text{Support } \{Milk, Burger\} / (\text{Support } \{Milk\} * \text{Support } \{Burger\}) \\ &= 0.4 / (0.8 * 0.6) \\ &= 0.83 \end{aligned}$$

While lift value less than 1 means that item '**Burger**' is unlikely to be bought if item '**Milk**' is bought.

$$\text{Lift } \{Bread, Milk \rightarrow Diaper\} =$$

$$\begin{aligned} & \text{Support } \{Bread, Milk, Diaper\} / (\text{Support } \{Bread, Milk\} * \text{Support } \{Diaper\}) \\ &= 0.4 / (0.6 * 0.8) \\ &= 0.4 / 0.48 \\ &= 0.83 \end{aligned}$$

While lift value less than 1 means that item '**Diaper**' is unlikely to be bought if itemset '**Bread, Milk**' is bought.

$$\text{Lift } \{Milk, Diaper \rightarrow Burger\} =$$

$$\begin{aligned} & \text{Support } \{Milk, Diaper, Burger\} / (\text{Support } \{Milk, Diaper\} * \text{Support } \{Burger\}) \\ &= 0.4 / (0.6 * 0.6) \\ &= 0.4 / 0.36 \\ &= 1.11 \end{aligned}$$

While lift value greater than 1 means that item '**Burger**' is likely to be bought if itemset '**Milk, Diaper**' is bought.

Frequent Itemset Mining Methods:

The most famous story about association rule mining is the “**beer and diaper.**” Researchers discovered that customers who buy diapers also tend to buy beer. This classic example shows that there might be many **interesting association rules** hidden in our daily data.

Association rules help to predict the occurrence of one item based on the occurrences of other items in a set of transactions.

Examples

- People who buy *bread* will also buy *milk*; represented as { **bread** → ~~milk~~ }
- People who buy *milk* will also buy *eggs*; represented as { **milk** → ~~eggs~~ }
- People who buy *bread* will also buy *jam*; represented as { **bread** → ~~jam~~ }

Association Rules discover the relationship between two or more attributes. It is mainly in the form of- **If** antecedent **than** consequent. For example, a supermarket sees that there are 200 customers on Friday evening. Out of the 200 customers, 100 bought chicken, and out of the 100 customers who bought chicken, 50 have bought Onions. Thus, the association rule would be- If customers buy chicken then buy onion too, with a support of $50/200 = 25\%$ and a confidence of $50/100=50\%$.

Association rule mining is a technique to *identify interesting relations* between different items. Association rule mining has to:

- Find all the frequent items.
- Generate association rules from the above frequent itemset.

There are many *methods or algorithms* to perform Association Rule Mining or Frequent Itemset Mining, those are:

- ❖ Apriori algorithm
- ❖ FP-Growth algorithm

Apriori algorithm:

Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. It was introduced by R Agarwal and R Srikant.

Name of the algorithm is *Apriori* because it uses *prior knowledge* of frequent itemset properties.

Frequent Item Set

- Frequent Itemset is an itemset whose **support** value is **greater than a threshold value**(support).

Apriori algorithm uses frequent itemsets to generate association rules. To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called **Apriori property** which helps by reducing the search space.

Apriori Property:

- All subsets of a frequent itemset must be frequent (Apriori property).
- If an itemset is infrequent, all its supersets will be infrequent.

Steps in Apriori algorithm:

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. A **minimum support threshold** is given in the problem or it is assumed by the user.

The steps followed in the Apriori Algorithm of data mining are:

Join Step: This step generates (K+1) itemset from K-itemsets by joining each item with itself.

Prune Step: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as **infrequent** and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

The above **join** and the **prune** steps iteratively until the most frequent itemsets are achieved.

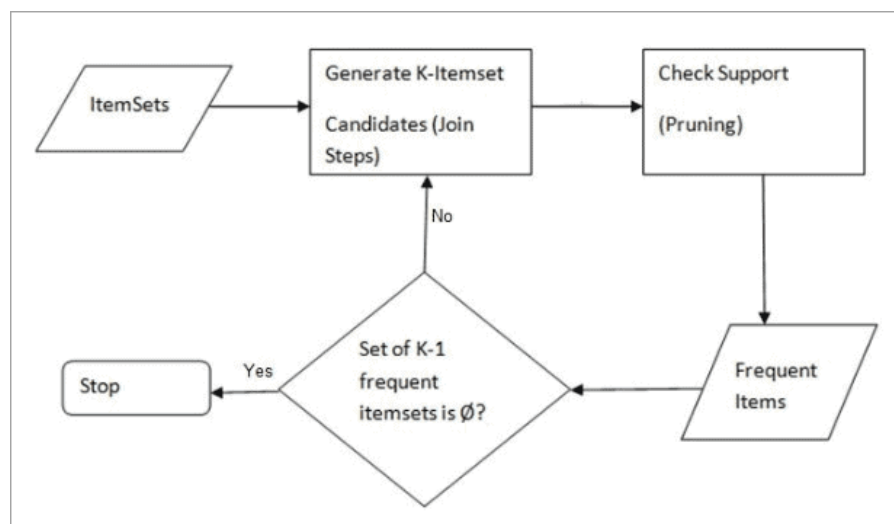


Fig: Apriori algorithm

Apriori Algorithm Example:

Consider the following dataset and find *frequent item sets* and generate *association rules* for them.

Assume that minimum *support threshold* ($s = 50\%$) and minimum *confident threshold* ($c = 80\%$).

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution:

Finding frequent item sets:

Support threshold=50% $\Rightarrow 0.5 \times 6 = 3 \Rightarrow \text{min_sup}=3$

Step-1:

- (i) Create a table containing support count of each item present in dataset – Called **C1** (candidate set).

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

- (ii) **Prune Step:** Compare candidate set item's support count with minimum support count. The above table shows that **I5** item does not meet **min_sup=3**, thus it is removed, only I1, I2, I3, I4 meet min_sup count.

This gives us the following item set **L1**.

Item	Count
I1	4
I2	5
I3	4
I4	4

Step-2:

- (i) **Join step:** Generate candidate set **C2** (2-itemset) using **L1**. And find out the occurrences of 2-itemset from the given dataset.

Item	Count
I1,I2	4
I1,I3	3
I1,I4	2
I2,I3	4
I2,I4	3
I3,I4	2

- (ii) **Prune Step:** Compare candidate set item's support count with minimum support count. The above table shows that **item sets {I1, I4}** and **{I3, I4}** does not meet **min_sup=3**, thus those are removed.

This gives us the following item set **L2**.

Item	Count
I1,I2	4
I1,I3	3
I2,I3	4
I2,I4	3

Step-3:

- (i) **Join step:** Generate candidate set **C3** (3-itemset) using **L2**. And find out the occurrences of 3-itemset from the given dataset.

Item	Count
I1,I2,I3	3
I1,I2,I4	2
I1,I3,I4	1
I2,I3,I4	2

- (ii) **Prune Step:** Compare candidate set item's support count with minimum support count. The above table shows that **itemset {I1, I2, I4}, {I1, I3, I4}** and **{I2, I3, I4}** does not meet **min_sup=3**, thus those are removed. Only the item set **{I1, I2, I3}** meet min_sup count.

So, finally the item set **{I1, I2, I3}** is a **frequent item set**

Generate Association Rules:

Thus, we have discovered all the frequent item-sets. Now we need to generate **strong association rules** (satisfies the minimum confidence threshold) from frequent item sets. For that we need to calculate **confidence** of each rule.

The given Confidence threshold is **80%**.

The all possible association rules from the frequent item set {I1, I2, I3} are:

{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)* 100 = 75% **(Rejected)**

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)* 100 = 100% **(Selected)**

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)* 100 = 75% **(Rejected)**

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)* 100 = 75% **(Rejected)**

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2} = (3/ 5)* 100 = 60% **(Rejected)**

{I3} => {I1, I2}

Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)* 100 = 75% **(Rejected)**

This shows that the association rule **{I1, I3} => {I2}** is strong if minimum confidence threshold is **80%**.

Exercise1: Apriori Algorithm

TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Dataset

Consider the above dataset and find frequent item sets and generate association rules for them.
Assume that Minimum support count is 2 and minimum confident threshold ($c = 50\%$).

Exercise2: Apriori Algorithm

TID	Items
T1	{milk, bread}
T2	{bread, sugar}
T3	{bread, butter}
T4	{milk, bread, sugar}
T5	{milk, bread, butter}
T6	{milk, bread, butter}
T7	{milk, sugar}
T8	{milk, sugar}
T9	{sugar, butter}
T10	{milk, sugar, butter}
T11	{milk, bread, butter}

Dataset

Consider the above dataset and find frequent item sets and generate association rules for them.
Assume that Minimum support count is 3 and minimum confident threshold ($c = 60\%$).

FP Growth algorithm:

The FP Growth algorithm is a popular method for frequent pattern mining in data mining. It works by constructing a frequent pattern tree (**FP-tree**) from the input dataset. The **FP-tree** is a *compressed representation* of the dataset that captures the frequency and association information of the items in the data.

FP Growth algorithm was developed by Han in 2000 and is a powerful tool for frequent pattern mining in data mining. It is widely used in various applications such as market basket analysis, bioinformatics, and web usage mining.

The FP Growth algorithm in data mining has several advantages over other frequent pattern mining algorithms, such as Apriori. The Apriori algorithm is *not suitable* for handling large datasets because it generates a large number of candidates and requires *multiple scans* of the database to mine frequent items.

In comparison, the FP Growth algorithm requires only a single scan of the data and a small amount of memory to construct the FP tree. It can also be *parallelized* to improve performance.

FP Tree:

The FP-tree (Frequent Pattern tree) is a data structure used in the FP Growth algorithm that stores the frequent item sets and their support counts.

Working on FP Growth Algorithm:

The working of the FP Growth algorithm in data mining can be summarized in the following steps:

Step 1: Scan the database

In this step, the algorithm scans the input dataset to determine the frequency of each item.

Step 2: Sort items

In this step, the items in the dataset are sorted in descending order of frequency. The infrequent items that do not meet the minimum support threshold are removed from the dataset. This helps to reduce the dataset's size and improve the algorithm's efficiency.

Step 3: Construct the FP-tree

In this step, the FP-tree is constructed. The FP-tree is a compact data structure that stores the frequent itemsets and their support counts. In this items are added to the FP tree, with the most frequent items added first.

Step 4: Generate frequent itemsets

Once the FP-tree has been constructed, frequent itemsets can be generated by recursively mining the tree. Starting at the bottom of the tree, the algorithm finds all combinations of frequent item sets that satisfy the minimum support threshold.

Step 5: Generate association rules

Once all frequent item sets have been generated, the algorithm post-processes the generated frequent item sets to generate association rules, which can be used to identify interesting relationships between the items in the dataset.

FP Growth algorithm Example:

Consider the below dataset and find frequent item sets and generate association rules for them. Assume that Minimum support count is 2 and minimum confident threshold ($c = 70\%$).

Transaction ID	Items
T1	A, C, D
T2	B, C, E, F
T3	A, B, C, E
T4	B, E
T5	A, C, E

Solution:

The above dataset contains 5 transactions and 6 items.

Step1: Scan the dataset

In this step, we will first calculate the *support count* of *all the items* in the transaction dataset. The results have been tabulated below.

Item	Support Count
A	3
B	3
C	4
D	1
E	4
F	1

Now, let's sort the above table by the support count.

Item	Support Count
C	4
E	4
A	3
B	3
D	1
F	1

Step2: Sort items

After calculating the support count of each item, we will *sort the items* in *each transaction* according to their support count in *descending order*. If two items have the same support count, we will keep their order in topological order.

The given minimum support *count is 2*. In the above table Items **D** and **F** doesn't meet minimum support count. So we need to remove Items **D** and **F** from each transaction the above dataset.

The resulting dataset is as follows.

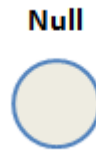
Transaction ID	Items
T1	C, A
T2	C, E ,B
T3	C, E ,A, B
T4	E,B
T5	C, E ,A

Step 3: Construct the FP-tree

After sorting the items in each transaction in the dataset by their support count, we need to create an *FP Tree* using the dataset.

To create an **FP-Tree** in the FP growth algorithm, we use the following steps.

1. First, we create a **root node** and name it **Null or None**. This node contains no data.

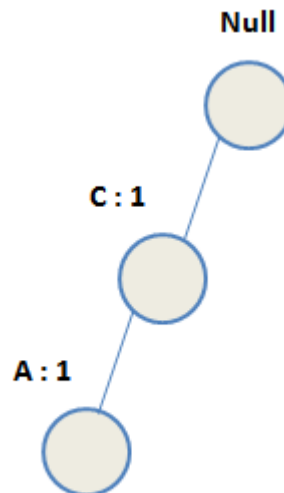


2. Next, for each transaction in the database, we insert the items in descending order of their support count into the FP-tree as follows:

❖ **T1: C, A**

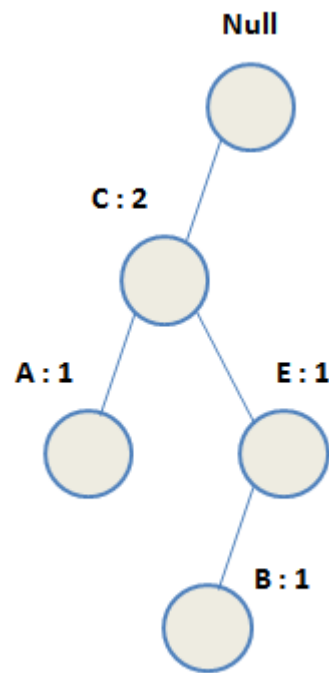
The transaction **T1: C, A** contains two items **C** and **A**, where **C** is linked as a child to **root node** and **A** is linked to **C**.

Here, all the items are simply linked one after the other in the order of occurrence in the set and initialize the support count for each item as **1**. i.e. **{C: 1, A: 1}**.



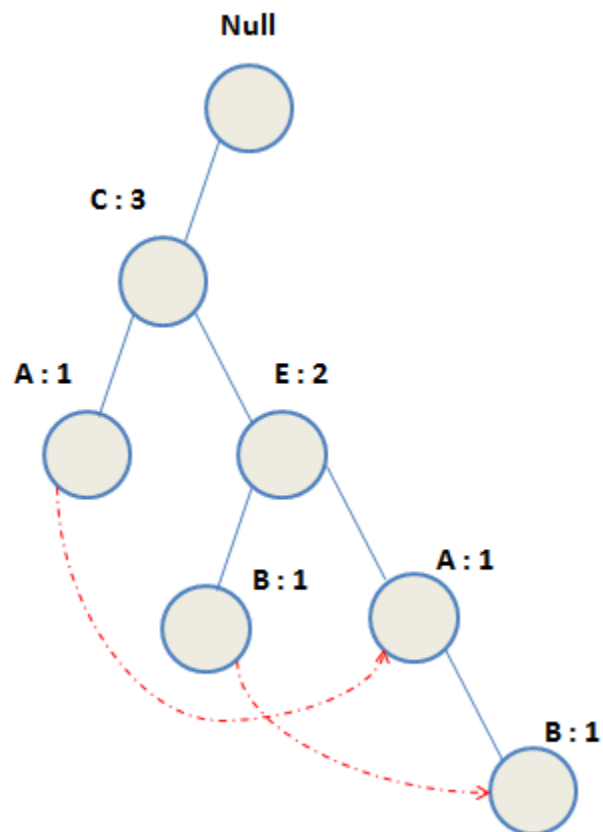
❖ **T2: C, E, B**

The transaction **T1: C, E, B** wherein **C** already linked with **root node**, So increment the support count of **C** by 1 and **E** is linked to **C** and **B** is linked to **E**. Hence initialize the support count for **E** and **B** as **1**.



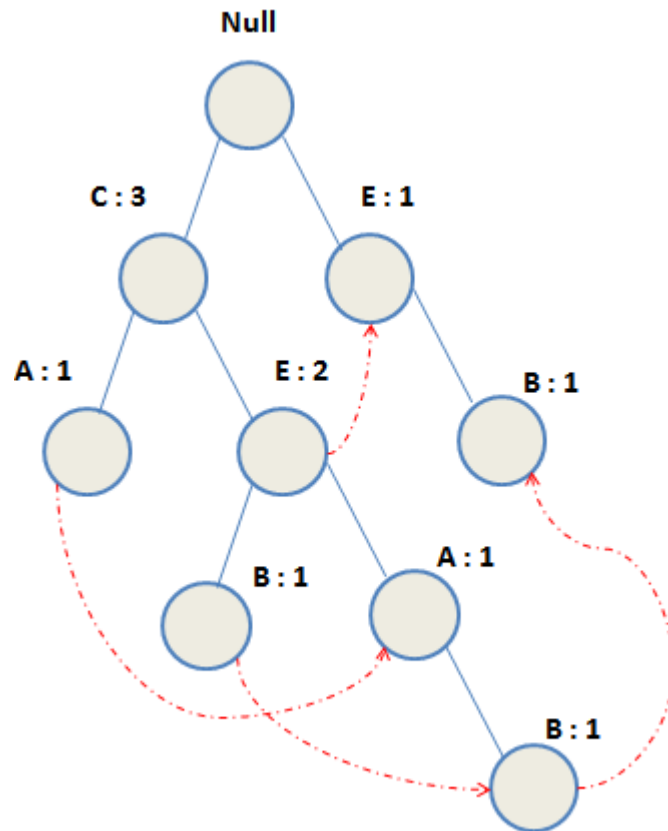
❖ **T3:** C, E, A, B

The transaction **T1:** C, E, A, B where C and E already linked, So increment the support count of C and E by 1 and A is linked to E and B is linked to A. Hence initialize the support count for A and B as 1.



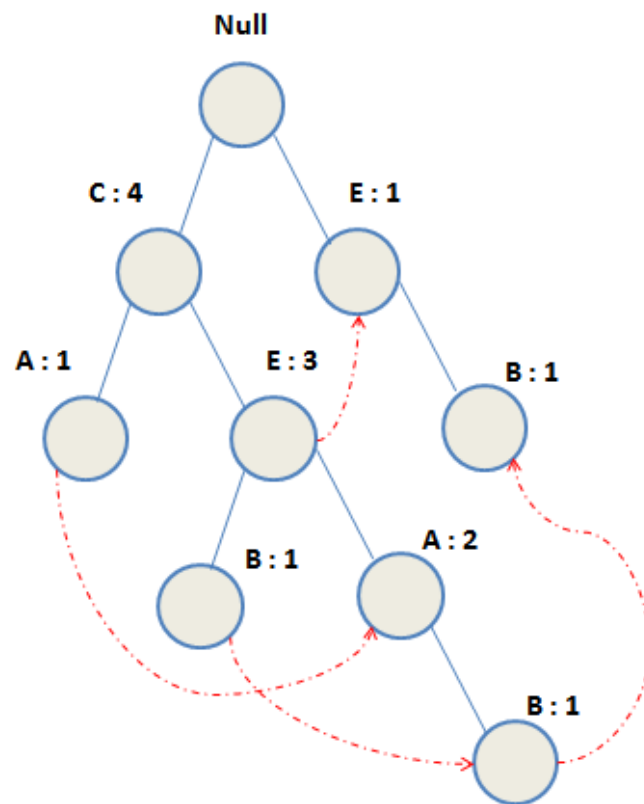
❖ **T4:** E, B

The transaction **T4:** E, B where **E** is linked with *root node* and **B** is linked with **E**, so set the support count for **E** and **B** as **1**.

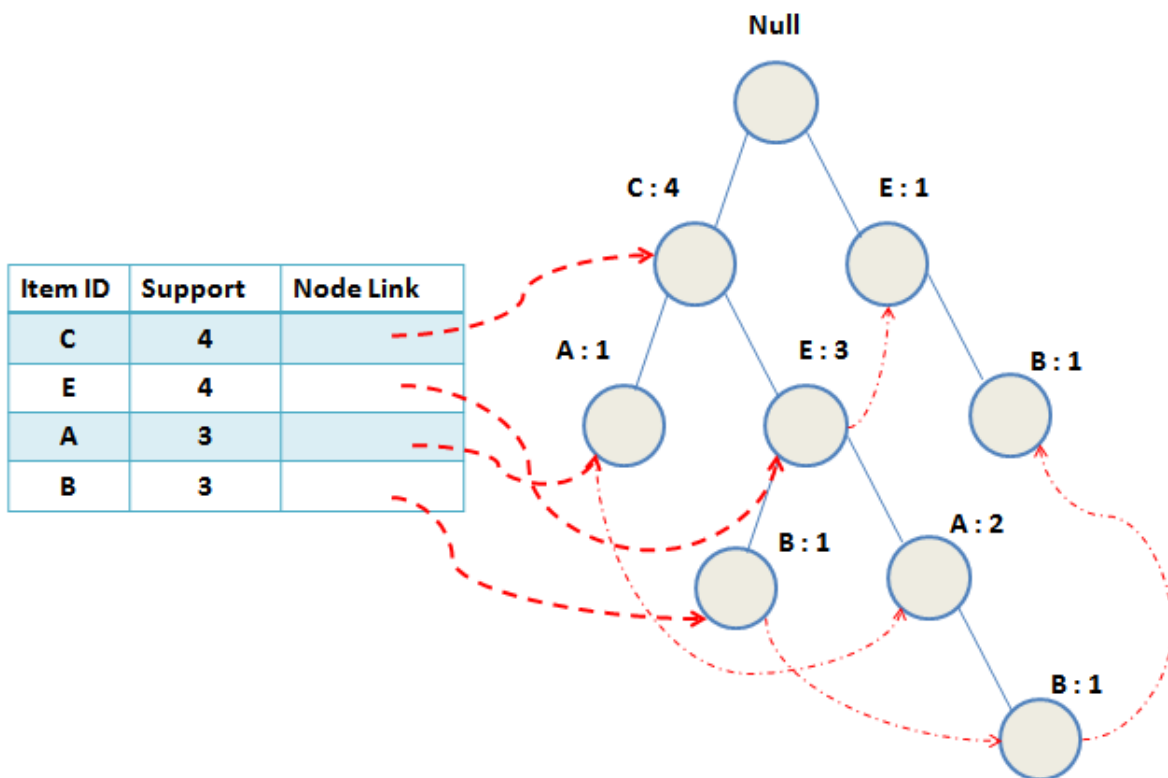


❖ **T5:** C, E, A

The transaction **T4:** C, E, A where C, E and A already linked, so increment the support count of C, E and A by 1.



The following is the complete *FP-Tree* along with support count:



Step 4: Generate frequent item sets

In this step, we need to generate frequent item sets by the help of the above **FP-Tree**.

To generate the frequent item sets, **first** we need to create a **pattern base** for all the items in the transaction dataset.

Create a pattern base:

Item	Pattern Base
A	{C:1},{C,E:2}
B	{C,E:1},{C,E,A:1},{E:1}
C	{}
E	{C:3}

Next, we need to create a **conditional fp-tree** for each frequent item. Here, we will calculate the **count of items** in the conditional fp tree of each item.

Item	Pattern Base	Conditional FP Tree
A	{C:1},{C,E:2}	{C:3},{E:2}
B	{C,E:1},{C,E,A:1},{E:1}	{C:2},{E:3}
C	{}	
E	{C:3}	{C:3}

After creating the **conditional fp-tree**, we need to generate frequent item sets for each item. To generate the frequent item sets, we take all the combinations of the items in the conditional fp-tree with their count.

Item	Pattern Base	Conditional FP Tree	Frequent item sets
A	{C:1},{C,E:2}	{C:3},{E:2}	{A,C}:3,{A,E}:2,{A,C,E}:2
B	{C,E:1},{C,E,A:1},{E:1}	{C:2},{E:3}	{B,C}:2,{B,E}:3,{B,C,E}:2
E	{C:3}	{C:3}	{E,C}:3

Step 5: Generate association rules

Thus, we have discovered all the frequent item-sets. Now we need to generate strong association rules (satisfies the minimum confidence threshold) from frequent item sets. For that we need to calculate confidence of each rule.

The given Confidence threshold is 70%.

(i) {A, C}

The all possible association rules from the above frequent item set are:

$$\{A\} \Rightarrow \{C\}$$

$$\text{Confidence} = \text{support } \{A, C\} / \text{support } \{A\} = (3 / 3) * 100 = 100\% \text{ (Selected)}$$

$$\{C\} \Rightarrow \{A\}$$

$$\text{Confidence} = \text{support } \{A, C\} / \text{support } \{C\} = (3 / 4) * 100 = 75\% \text{ (Selected)}$$

(ii) {A, E}

The all possible association rules from the above frequent item set are:

$$\{A\} \Rightarrow \{E\}$$

$$\text{Confidence} = \text{support } \{A, E\} / \text{support } \{A\} = (2 / 3) * 100 = 66.6\% \text{ (Rejected)}$$

$$\{E\} \Rightarrow \{A\}$$

$$\text{Confidence} = \text{support } \{A, E\} / \text{support } \{E\} = (2 / 4) * 100 = 50\% \text{ (Rejected)}$$

(iii) {A, C, E}

The all possible association rules from the above frequent item set are:

$$\{A, C\} \Rightarrow \{E\}$$

$$\text{Confidence} = \text{support } \{A, C, E\} / \text{support } \{A, C\} = (2 / 3) * 100 = 66.6\% \text{ (Rejected)}$$

$$\{A, E\} \Rightarrow \{C\}$$

$$\text{Confidence} = \text{support } \{A, C, E\} / \text{support } \{A, E\} = (2 / 2) * 100 = 100\% \text{ (Selected)}$$

$$\{C, E\} \Rightarrow \{A\}$$

$$\text{Confidence} = \text{support } \{A, C, E\} / \text{support } \{C, E\} = (2 / 3) * 100 = 66.6\% \text{ (Rejected)}$$

FP Growth algorithm Exercise 1:

Consider the below dataset and find frequent item sets and generate association rules for them.
Assume that Minimum support count is 3 and minimum confident threshold ($c = 60\%$).

Transaction ID	Items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

FP Growth algorithm Exercise 2:

Consider the below dataset and find frequent item sets Assume that Minimum support count is 3.

Transaction ID	Items
T100	{N, O, M, K, E, Y}
T200	{E, O, N, K, D Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{ K, O, O, C, I, E}

FP Growth algorithm Exercise 3:

Consider the below dataset and find frequent item sets. Assume that Minimum support count is 2.

<i>TID</i>	<i>List of item IDs</i>
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Mining various kinds of Association Rules:

An association rule is a statement of the form "If A, **then** B," where A and B are sets of items. The strength of an association rule is measured using two measures: **support** and **confidence**. Support measures the frequency of the occurrence of the items in the rule, and confidence measures the reliability of the rule.

There are various types of association rules in data mining:-

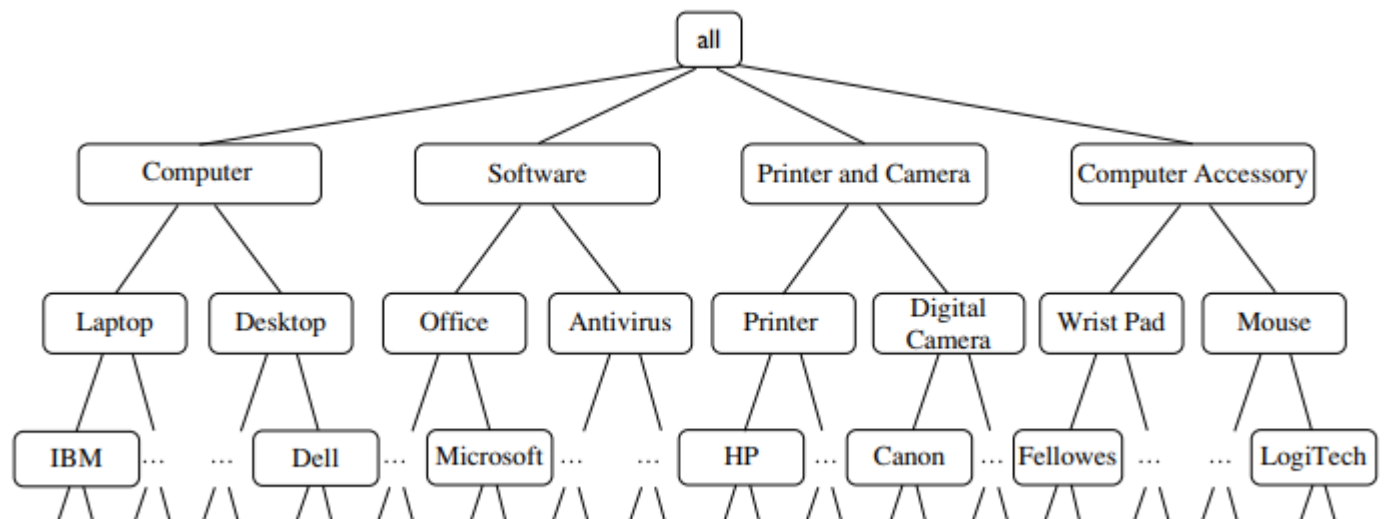
- Multi-level association rules
- Multidimensional association rules
- Quantitative association rules

Multi-level association rules:

Association rules generated from mining data at *multiple levels of abstraction* are called multiple-level or multilevel association rules. Multilevel association rules can be mined efficiently using **concept hierarchies** under a support-confidence framework.

Multilevel Association Rule mining is a technique that extends Association Rule mining to discover relationships between items at different levels of granularity.

For example, in a retail dataset, multi-level Association Rule mining can be used to find relationships between individual items and categories of items.



Multidimensional association rules:

Association rules that involve single dimension or predicate can be referred to as a ***single dimensional*** or intra dimensional association rule.

Example:

$$\text{buys}(X, \text{"Computer"}) \Rightarrow \text{buys}(X, \text{"printer"})$$

The above rule is a single dimensional or intra dimensional association rule because it contains a single distinct predicate (e.g., *buys*) with multiple occurrences (i.e., the predicate occurs more than once within the rule).

Association rules that involve two or more dimensions or predicates can be referred to as ***multidimensional association rules***.

Example:

$$\text{age}(X, \text{"20...29"}) \wedge \text{occupation}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"laptop"})$$

The above rule contains three predicates (*age*, *occupation*, and *buys*), each of which occurs only once in the rule.

Multidimensional association rules with ***no repeated predicates*** are called ***inter dimensional association rules***. The above rule is an example of inter dimensional association rule.

Multidimensional association rules ***with repeated predicates***, which contain multiple occurrences of some predicates. These rules are called ***hybrid-dimensional association rules***.

Example:

$$\text{age}(X, \text{"20...29"}) \wedge \text{buys}(X, \text{"laptop"}) \Rightarrow \text{buys}(X, \text{"Printer"})$$

Quantitative association rules:

Quantitative association rules are multidimensional association rules in which the numeric attributes are dynamically determined during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined.

Quantitative association rules having two quantitative attributes on the left-hand side of the rule and one categorical attribute on the right-hand side of the rule. That is,

$$A_{\text{quant1}} \wedge A_{\text{quant2}} \Rightarrow A_{\text{categ}}$$

where A_{quant1} and A_{quant2} are tests on quantitative attribute intervals (where the intervals are dynamically determined), and A_{categ} tests a categorical attribute from the task-relevant data.

For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as high-definition TV, i.e., HDTV) that customers like to buy. An example of such a 2-D quantitative association rule is

$$age(X, "30....39") \wedge income(X, "42K...48K") \Rightarrow buys(X, "HDTV")$$

Correlation Analysis:

Association and **Correlation** in Data Mining are two of the most widely used techniques. They are used to identify *patterns* and *relationships* between variables in a dataset.

Association refers to the discovery of co-occurrences or relationships between items in a dataset. On the other hand, correlation measures the *strength of the relationship* between two variables (attributes).

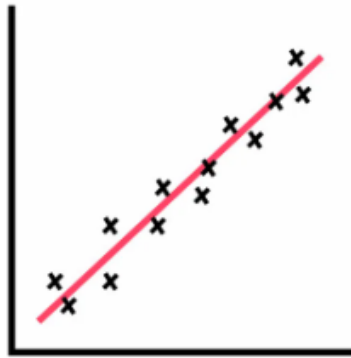
Correlation Analysis is a data mining technique used to identify the degree of relationships i.e. in which two or more variables are related or associated with each other.

Correlation refers to the statistical relationship between two or more variables, where the variation in one variable is associated with the variation in another variable.

In other words, it measures how changes in one variable are related to changes in another variable. Correlation can be *positive, negative, or zero*, depending on the direction and strength of the relationship between the variables.

A **positive correlation** is a relationship between two variables in which both variables move in the same direction. Therefore, when one variable increases as the other variable increases or one variable decreases while the other decreases.

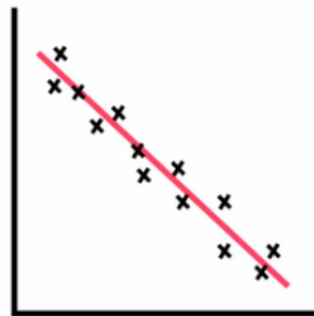
An example of a positive correlation would be *height* and *weight*. Taller people tend to be heavier.



Positive correlation

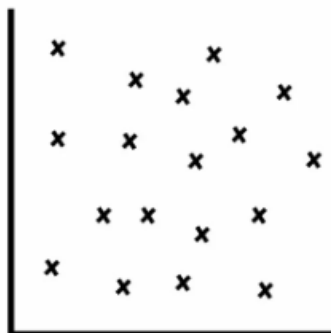
A **negative correlation** is a relationship between two variables in which an increase in one variable is associated with a decrease in the other.

An example of a negative correlation would be the **height** above sea level and temperature. As you climb the mountain (increase in height), it gets colder (decrease in temperature).



Negative correlation

A **zero correlation** exists when there is no relationship between two variables. For example, there is no relationship between the amount of tea drunk and the level of intelligence.



Zero correlation