Unit – III - Data Visualization Techniques

Data visualization is a graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. This blog on data visualization techniques will help you understand detailed techniques and benefits.

Data Visualization Techniques

Box plots

Histograms

Heat maps

Charts

Tree maps

Word Cloud/Network diagram

Box Plots

A box plot is a graph that gives you a good indication of how the values in the data are spread out. Although box plots may seem primitive in comparison to a histogram or density plot, they have the advantage of taking up less space, which is useful when comparing distributions between many groups or datasets. For some distributions/datasets, you will find that you need more information than the measures of central tendency (median, mean, and mode). You need to have information on the variability or dispersion of the data.

List of Methods to Visualize Data

Column Chart: It is also called a vertical bar chart where each category is represented by a rectangle. The height of the rectangle is proportional to the values that are plotted.

Bar Graph: It has rectangular bars in which the lengths are proportional to the values which are represented.

Stacked Bar Graph: It is a bar style graph that has various components stacked together so that apart from the bar, the components can also be compared to each other.

Stacked Column Chart: It is similar to a stacked bar; however, the data is stacked horizontally.

Area Chart: It combines the line chart and bar chart to show how the numeric values of one or more groups change over the progress of a viable area.

Dual Axis Chart: It combines a column chart and a line chart and then compares the two variables.

Line Graph: The data points are connected through a straight line; therefore, creating a representation of the changing trend.

Mekko Chart: It can be called a two-dimensional stacked chart with varying column widths.

Pie Chart: It is a chart where various components of a data set are presented in the form of a pie which represents their proportion in the entire data set.

Waterfall Chart: With the help of this chart, the increasing effect of sequentially introduced positive or negative values can be understood.

Bubble Chart: It is a multi-variable graph that is a hybrid of Scatter Plot and a Proportional Area Chart.

Scatter Plot Chart: It is also called a scatter chart or scatter graph. Dots are used to denote values for two different numeric variables.

Bullet Graph: It is a variation of a bar graph. A bullet graph is used to swap dashboard gauges and meters.

Funnel Chart: The chart determines the flow of users with the help of a business or sales process.

Heat Map: It is a technique of data visualization that shows the level of instances as color in two dimensions.

Histograms

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges. Taller bars show that more data falls in that range. A histogram displays the shape and spread of continuous sample data.

Heat Maps

A heat map is data analysis software that uses colour the way a bar graph uses height and width: as a data visualization tool. If you're looking at a web page and you want to know which areas get the most attention, a heat map shows you in a visual way that's easy to assimilate and make decisions from. It is a graphical representation of data where the individual values contained in a matrix are represented as colours. Useful for two purposes: for visualizing correlation tables and for visualizing missing values in the data. In both cases, the information is conveyed in a two-dimensional table.

Charts

Line Chart

The simplest technique, a line plot is used to plot the relationship or dependence of one variable on another. To plot the relationship between the two variables, we can simply call the plot function.

Bar Charts

Bar charts are used for comparing the quantities of different categories or groups. Values of a category are represented with the help of bars and they can be configured with vertical or horizontal bars, with the length or height of each bar representing the value.

Pie Chart

It is a circular statistical graph which decides slices to illustrate numerical proportion. Here the arc length of each slide is proportional to the quantity it represents. As a rule, they are used to compare the parts of a whole and are most effective when there are limited components and when text and percentages are included to describe the content. However, they can be difficult to interpret because the human eye has a hard time estimating areas and comparing visual angles.

Scatter Charts

Another common visualization technique is a scatter plot that is a two-dimensional plot representing the joint variation of two data items. Each marker (symbols such as dots, squares and plus signs) represents an observation. The marker position indicates the value for each observation. When you assign more than two measures, a scatter plot matrix is produced that is a series scatter plot displaying every possible pairing of the measures that are assigned to the visualization. Scatter plots are used for examining the relationship, or correlations, between X and Y variables.

Bubble Charts

It is a variation of scatter chart in which the data points are replaced with bubbles, and an additional dimension of data is represented in the size of the bubbles.

Timeline Charts

Timeline charts illustrate events, in chronological order — for example the progress of a project, advertising campaign, acquisition process — in whatever unit of time the data was recorded — for example week, month, year, quarter. It shows the chronological sequence of past or future events on a timescale.

Tree Maps

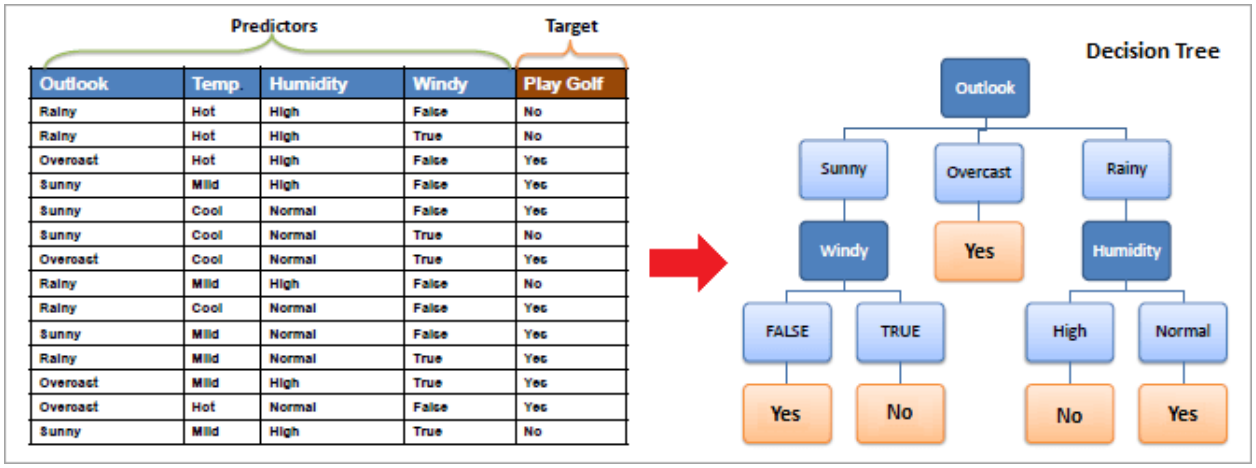
A treemap is a visualization that displays hierarchically organized data as a set of nested rectangles, parent elements being tiled with their child elements. The sizes and colours of rectangles are proportional to the values of the data points they represent. A leaf node rectangle has an area proportional to the specified dimension of the data. Depending on the choice, the leaf node is coloured, sized or both according to chosen attributes. They make efficient use of space, thus display thousands of items on the screen simultaneously.

Decision Tree Algorithm Examples

Decision Tree Mining is a type of data mining technique that is used to build Classification Models. It builds classification models in the form of a tree-like structure, just like its name. This type of mining belongs to supervised class learning.

In supervised learning, the target result is already known. Decision trees can be used for both categorical and numerical data. The categorical data represent gender, marital status, etc. while the numerical data represent age, temperature, etc.

An example of a decision tree with the dataset is shown below.



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

ID3 Algorithm

ID3 algorithm, stands for Iterative Dichotomiser 3, is a classification algorithm that follows a greedy approach of building a decision tree by selecting a best attribute that yields maximum Information Gain (IG) or minimum Entropy (H).

What is an ID3 Algorithm?

ID3 stands for Iterative Dichotomiser 3

It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain(IG) or minimum Entropy(H).

What is Entropy and Information gain?

Entropy is a measure of the amount of uncertainty in the dataset S. Mathematical Representation of Entropy is shown here -

$$H(S) = \sum c \in C - p(c) \log 2\, p(c)$$

Where,

S - The current dataset for which entropy is being calculated(changes every iteration of the ID3 algorithm).

C - Set of classes in S {example - C ={yes, no}}

p(c) - The proportion of the number of elements in class c to the number of elements in set S.

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on that particular iteration.

Entropy = 0 implies it is of pure class, that means all are of same category.

Information Gain IG(A) tells us how much uncertainty in S was reduced after splitting set S on attribute A. Mathematical representation of Information gain is shown here -

$IG(A, S) = H(S) - \sum t \in T\ p(t)\ H(t)$

Where,

H(S) - Entropy of set S.

T - The subsets created from splitting set S by attribute A such that

$S = \bigcup t \in T\ t$

p(t) - The proportion of the number of elements in t to the number of elements in set S.

H(t) - Entropy of subset t.

In ID3, information gain can be calculated (instead of entropy) for each remaining attribute. The attribute with the largest information gain is used to split the set S on that particular iteration.

The steps in ID3 algorithm are as follows:

Calculate entropy for dataset.

For each attribute/feature.

2.1. Calculate entropy for all its categorical values.

2.2. Calculate information gain for the feature.

Find the feature with maximum information gain.

Repeat it until we get the desired tree.

Use ID3 algorithm on a data

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

First Attribute - Outlook

Categorical values - sunny, overcast and rain

H(Outlook=sunny) = -(2/5)*log(2/5)-(3/5)*log(3/5) =0.971

H(Outlook=rain) = -(3/5)*log(3/5)-(2/5)*log(2/5) =0.971

H(Outlook=overcast) = -(4/4)*log(4/4)-0 = 0

Average Entropy Information for Outlook -

I(Outlook) = p(sunny) * H(Outlook=sunny) + p(rain) * H(Outlook=rain) + p(overcast) * H(Outlook=overcast)

= (5/14)*0.971 + (5/14)*0.971 + (4/14)*0

= 0.693

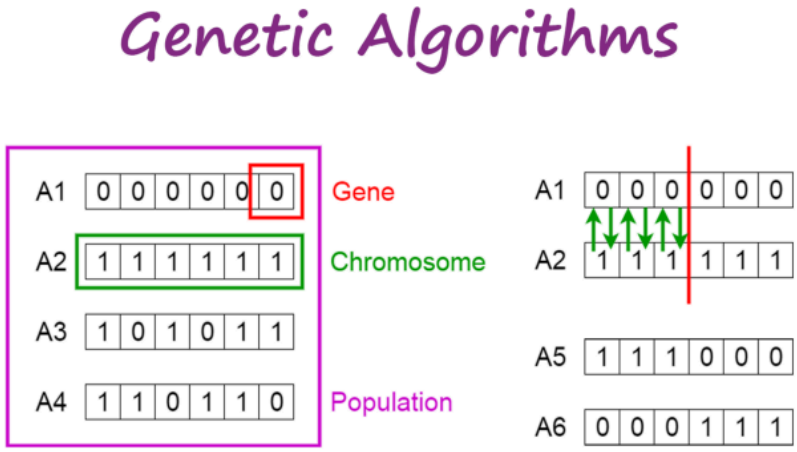
Information Gain = H(S) - I(Outlook)

= 0.94 - 0.693

= 0.247

Genetic Algorithm

A **genetic algorithm** is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.



Notion of Natural Selection

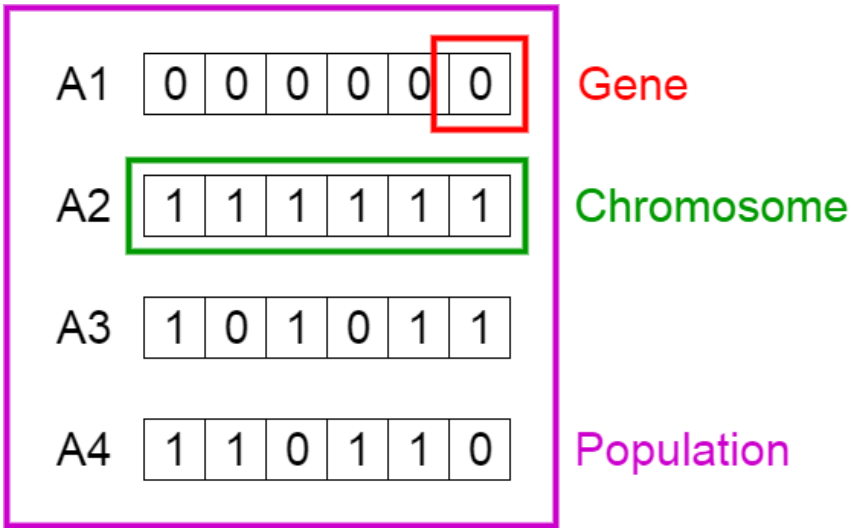
The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found. This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.

Five phases are considered in a genetic algorithm.

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

Initial Population

The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve. An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution). In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome.



Fitness Function

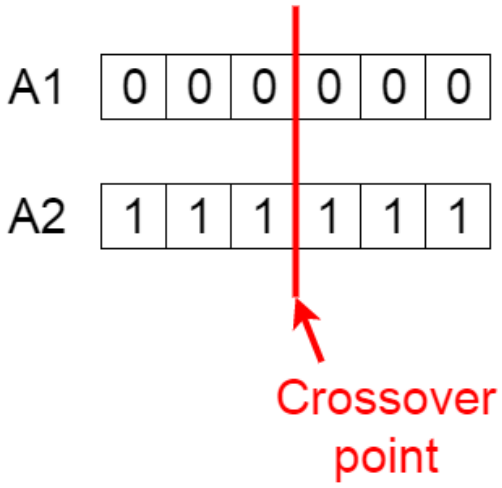
The **fitness function** determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

Selection

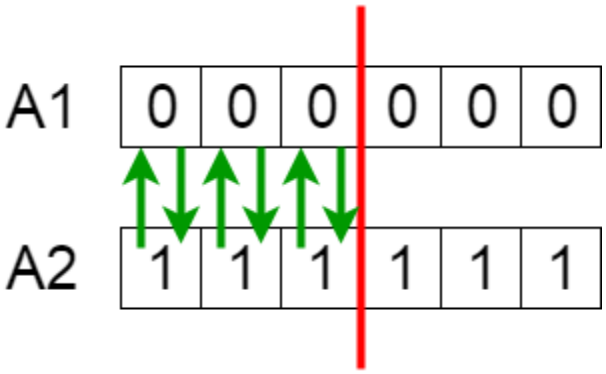
The idea of **selection** phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (**parents**) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction.

Crossover

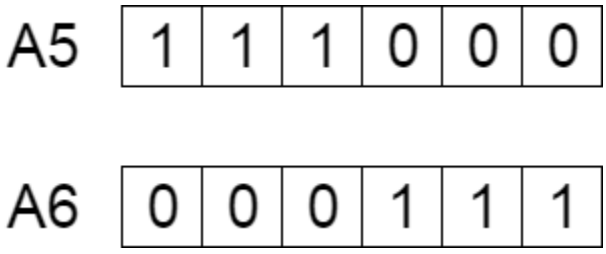
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a **crossover point** is chosen at random from within the genes. For example, consider the crossover point to be 3 as shown below.



Crossover point - Offspring are created by exchanging the genes of parents among themselves until the crossover point is reached.

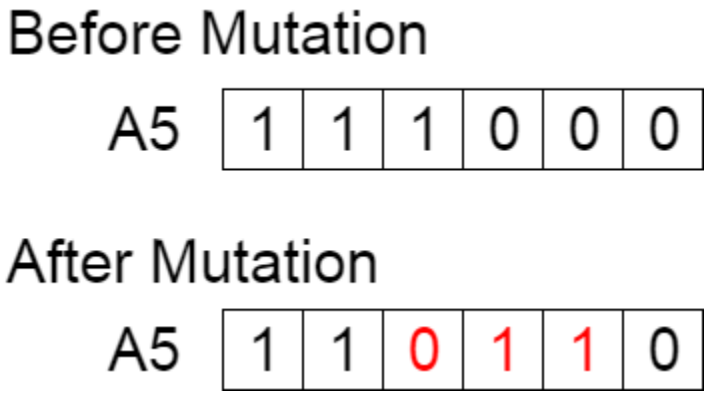


Exchanging genes among parents - The new offspring are added to the population.



Mutation

In certain new offspring formed, some of their genes can be subjected to a **mutation** with a low random probability. This implies that some of the bits in the bit string can be flipped.



Mutation: Before and After Mutation occurs to maintain diversity within the population and prevent premature convergence.

Termination

The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the genetic algorithm has provided a set of solutions to our problem.

Introduction to K-Means Algorithm

The K-means clustering algorithm computes centroids and repeats until the optimal centroid is found. It is presumptively known how many clusters there are. It is also known as the flat clustering algorithm. The number of clusters found from data by the method is denoted by the letter 'K' in K-means. In this method, data points are assigned to clusters in such a way that the sum of the squared distances between the data points and the centroid is as small as possible. It is essential to note that reduced diversity within clusters leads to more identical data points within the same cluster.

Working of K-Means Algorithm

The following stages will help us understand how the K-Means clustering technique works-

Step 1: First, we need to provide the number of clusters, K, that need to be generated by this algorithm.

Step 2: Next, choose K data points at random and assign each to a cluster. Briefly, categorize the data based on the number of data points.

Step 3: The cluster centroids will now be computed.

Step 4: Iterate the steps below until we find the ideal centroid, which is the assigning of data points to clusters that do not vary.

4.1 The sum of squared distances between data points and centroids would be calculated first.

4.2 At this point, we need to allocate each data point to the cluster that is closest to the others (centroid).

4.3 Finally, compute the centroids for the clusters by averaging all of the cluster's data points.

K-means implements the Expectation-Maximization strategy to solve the problem. The Expectation-step is used to assign data points to the nearest cluster, and the Maximization-step is used to compute the centroid of each cluster.

When using the K-means algorithm, we must keep the following points in mind:

It is suggested to normalize the data while dealing with clustering algorithms such as K-Means since such algorithms employ distance-based measurement to identify the similarity between data points.

Because of the iterative nature of K-Means and the random initialization of centroids, K-Means may become stuck in a local optimum and fail to converge to the global optimum. As a result, it is advised to employ distinct centroids' initializations.

Implementation of K Means Clustering Graphical Form

STEP 1: Let us pick k clusters, i.e., K=2, to separate the dataset and assign it to its appropriate clusters. We will select two random places to function as the cluster's centroid.

STEP 2: Now, each data point will be assigned to a scatter plot depending on its distance from the nearest K-point or centroid. This will be accomplished by establishing a median between both centroids. Consider the following illustration:

STEP 3: The points on the line's left side are close to the blue centroid, while the points on the line's right side are close to the yellow centroid. The left Form cluster has a blue centroid, whereas the right Form cluster has a yellow centroid.

STEP 4: Repeat the procedure, this time selecting a different centroid. To choose the new centroids, we will determine their new center of gravity, which is represented below:

STEP 5: After that, we'll re-assign each data point to its new centroid. We shall repeat the procedure outlined before (using a median line). The blue cluster will contain the yellow data point on the blue side of the median line.

STEP 6: Now that reassignment has occurred, we will repeat the previous step of locating new centroids.

STEP 7: We will repeat the procedure outlined above for determining the center of gravity of centroids, as shown below.

STEP 8: Similar to the previous stages, we will draw the median line and reassign the data points after locating the new centroids.

STEP 9: We will finally group points depending on their distance from the median line, ensuring that two distinct groups are established and that no dissimilar points are included in a single group.

Apriori Algorithm

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

#1) In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.

#2) Let there be some minimum support, min_sup (eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to min_sup, are taken ahead for the next iteration and the others are pruned.

#3) Next, 2-itemset frequent items with min_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.

#4) The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 –itemsets with min-sup only.

#5) The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.

#6) Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.

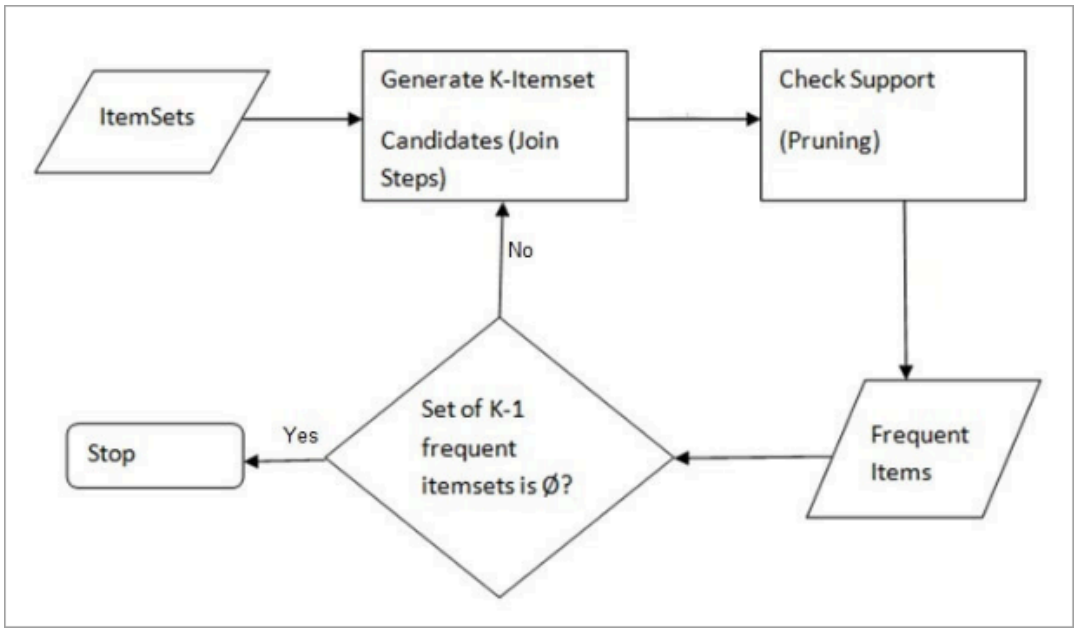


TABLE-1

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:

Support threshold=50% => 0.5*6= 3 => min_sup=3

1. Count Of Each Item

TABLE-2

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

2. Prune Step: TABLE -2 shows that I5 item does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

TABLE-3

Item	Count
I1	4
I2	5
I3	4
I4	4

3. Join Step: Form 2-itemset. From TABLE-1 find out the occurrences of 2-itemset.

TABLE-4

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

4. Prune Step: TABLE -4 shows that item set {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted.

TABLE-5

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

5. Join and Prune Step: Form 3-itemset. From the TABLE- 1 find out occurrences of 3-itemset. From TABLE-5, find out the 2-itemset subsets which support min_sup.

We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3} are occurring in TABLE-5 thus {I1, I2, I3} is frequent.

We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in TABLE-5 thus {I1, I2, I4} is not frequent, hence it is deleted.

TABLE-6

Item

I1,I2,I3

I1,I2,I4

I1,I3,I4

I2,I3,I4

Only {I1, I2, I3} is frequent.

6. Generate Association Rules: From the frequent itemset discovered above the association could be:

{I1, I2} => {I3}

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

{I1, I3} => {I2}

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

{I2, I3} => {I1}

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

{I1} => {I2, I3}

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

{I2} => {I1, I3}

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

{I3} => {I1, I2}

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

This shows that all the above association rules are strong if minimum confidence threshold is 60%.