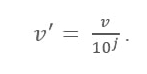
**Normalization:**

One of the common challenges is that, usually, databases contain attributes of different units, range, and scales. Applying algorithms to such drastically ranging data may not deliver accurate results. This calls for data normalization in data mining.

It is a necessary process required to normalize heterogeneous data. Data can be put into a smaller range, such as 0.0 to 1.0 or -1.0 to 1.0. In simple words, data normalization makes data easier to classify and understand.

Decimal scaling is another technique for normalization in data mining. It functions by converting a number to a decimal point.

Decimal Scaling Formula



Here:

* V’ is the new value after applying the decimal scaling
* V is the respective value of the attribute

Now, integer J defines the movement of decimal points. So, how to define it? It is equal to the number of digits present in the maximum value in the data table. Here is an example:

Suppose a company wants to compare the salaries of the new joiners. Here are the data values:

| Employee Name | Salary |
| --- | --- |
| ABC | 10,000 |
| XYZ | 25,000 |
| PQR | 8,000 |
| MNO | 15,000 |

Now, look for the maximum value in the data. In this case, it is 25,000. Now count the number of digits in this value. In this case, it is ‘5’. So here ‘j’ is equal to 5, i.e 100,000. This means the V (value of the attribute) needs to be divided by 100,000 here.

After applying the zero decimal scaling formula, here are the new values:

| Name | Salary | Salary after Decimal Scaling |
| --- | --- | --- |
| ABC | 10,000 | 0.1 |
| XYZ | 25, 000 | 0.25 |
| PQR | 8, 000 | 0.08 |
| MNO | 15,000 | 0.15 |

Thus, decimal scaling can tone down big numbers into easy to understand smaller decimal values. Also, data attributed to different units becomes easy to read and understand once it is converted into smaller decimal values.

**Regression**

is a **data mining** (machine learning) technique used to fit an equation to a dataset. The simplest form of **regression**, linear **regression** [2], uses the formula of a straight line (y = mx + b) and determines the appropriate values for m and b to predict the value of y based upon a given value of x.

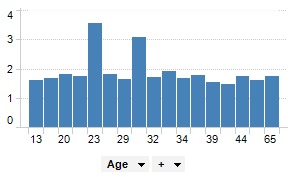
**Binning**

is a way to group a number of more or less continuous values into a smaller number of "bins". For example, if you have data about a group of people, you might want to arrange their ages into a smaller number of age intervals. Numeric columns can also be temporarily grouped by right-clicking on a column selector and clicking Auto-bin Column.

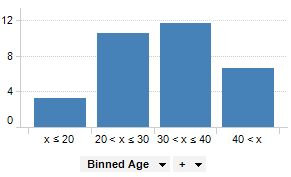
There is also an option to group categorical values into bins. This is useful when you have more categorical values in a column than you find necessary. Your visualization may for example show sales of apples, pears, oranges and limes, but you are interested in citrus fruit sales compared to

apples and pears sales. Then oranges and limes can be grouped into a bin.

**Note:** A special use case of this binning method is grouping values that are misspelt or differ due to other reasons. For example, if a column contains values like “apple” and “appel”, or “UK” and “United Kingdom”, you can group these values into bins.

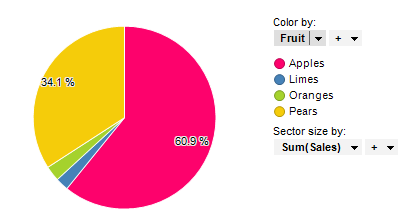


By binning the age of the people into a new column, data can be visualized for the different age groups instead of for each individual.

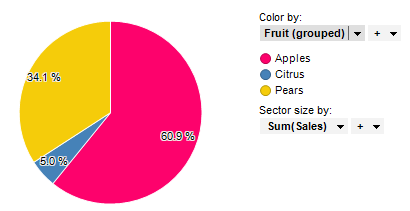


**Example of binning categorical data**

The pie chart shows sales per apples, limes, oranges and pears.



Below oranges and limes have been grouped into a bin called “Citrus”.



**Discretization:**

Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy.

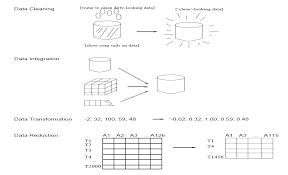
## **Data discretization example**

we have an attribute of age with the following values.

| **Age** | 10,11,13,14,17,19,30, 31, 32, 38, 40, 42,70 , 72, 73, 75 |
| --- | --- |

**Table:** Before discretization

| **Attribute** | **Age** | **Age** | **Age** |
| --- | --- | --- | --- |
|  | 10,11,13,14,17,19, | 30, 31, 32, 38, 40, 42 | 70 , 72, 73, 75 |
| After Discretization | Young | Mature | Old |

****

**Pre-processing of data**