Prepare Data for Machine Processing

Contents:

Depending on your downstream system, you may need to convert your data into numeric values of the expected form or to standardize the distribution of numeric values. This section summarizes some common statistical transformations that can be applied to columnar data to prepare it for use in downstream analytic systems.

Scaling

You can scale the values within a column using either of the following techniques.

Scale to zero mean and unit variance

Zero mean and unit variance scaling renders the values in the set to fit a normal distribution with a mean of `0` and a variance of `1`. This technique is a common standard for normalizing values into a normal distribution for statistical purposes.

In the following example, the values in the `POS_Sales` column have been normalized to average `0`, variance `1`.

- **Remove mean:** When selected, the existing mean (average) of the values is used as the center of the distribution curve.

  > **NOTE:** Re-centering sparse data by removing the mean may remove sparseness.

- **Scale to unit variance:** When selected, the range of values are scaled such that their variance is `1`. When deselected, the existing variance is maintained.

  > **NOTE:** Scaling to unit variance may not work well for managing outliers. Some additional techniques for managing outliers are outlined below.

Transformation Name	`Scale column`
Parameter: Column	`POS_Sales`
Parameter: Scaling method	`Scale to zero mean and unit variance`
Parameter: Remove mean	`false`

Parameter: Scale to unit variance	`true`
Parameter: Output options	`Create new column`
Parameter: New column name	`scale_POS_Sales`

Scale to min-max range

You can scale column values fitting between a specified minimum and maximum value. This technique is useful for distributions with very small standard deviation values and for preserving 0 values in sparse data.

The following example scales the `TestScores` column to a range of `0` and `1`, inclusive.

Transformation Name	`Scale column`
Parameter: Column	`TestScores`
Parameter: Scaling method	`Scale to a given min-max range`
Parameter: Minimum	`0`
Parameter: Maximum	`1`
Parameter: Output options	`Replace current column`

Outliers

You can use several techniques for identifying statistical outliers in your dataset and managing them as needed.

Identify outliers

Suppose you need to remove the outliers from a column. Assuming a normal bell distribution of values, you can use the following formula to calculate the number of standard deviations a column value is from the column mean (average). In this case, the source column is `POS_Sales`.

Transformation Name	`New formula`
Parameter: Formula type	`Multiple row formula`
Parameter: Formula	`(ABS(POS_Sales - AVERAGE(POS_Sales))) / STDEV (POS_Sales)`
Parameter: New column name	`stdevs_POS_Sales`

Remove outliers

The new `stdevs_POS_Sales` column now contains the number of standard deviations from the mean for the corresponding value in `POS_Sales`. You can use the following transformation to remove the rows that contain outlier values for this column.

> **Tip:** An easier way to select these outlier values is to select the range of values in the `stdevs_POS_Sales` column histogram. Then, select the suggestion to delete these rows. You may want to edit the actual formula before you add it to your recipe.

In the following transformation, all rows that contain a value in `POS_Sales` that is greater than four standard deviations from the mean are deleted:

Transformation Name	`Filter rows`
Parameter: Condition	`Custom formula`
Parameter: Type of formula	`Custom single`
Parameter: Condition	`4 <= stdevs_POS_Sales`
Parameter: Action	`Delete matching rows`

Change outliers to mean values

You can also remove the effects of outliers be setting their value to the mean (average), which preserves the data in other columns in the row.

Transformation Name	`Edit with formula`
Parameter: Columns	`POS_Sales`
Parameter: Formula	`IF(stdevs_POS_Sales > 4, AVERAGE(POS_Sales), POS_Sales)`

Binning

You can modify your data to fit into bins of equal or custom size. For example, the lowest values in your range would be marked in the `0` bin, with larger values being marked with larger bin numbers.

Bins of equal size

You can bin numeric values into bins of equal size. Suppose your column contains numeric values `0-1000`. You can bin values into equal ranges of `100` by creating `10` bins.

Transformation Name	`Bin column`
Parameter: Column	`MilleBornes`
Parameter: Select Option	`Equal Sized Bins`
Parameter: Number of Bins	`10`
Parameter: New column name	`MilleBornesRating`

Bins of custom size

You can also create custom bins. In the following example, the `TestScores` column is binned into the following bins. In a later step, these bins are mapped to grades:

Bins	Bin Range	Bin Number	Grade
59	0-59	0	F
69	60-69	1	D
79	70-79	2	C
89	80-89	3	B
	90+	4	A
(no value)			I

First, you bin values into the bin numbers listed above:

Transformation Name	`Bin column`
Parameter: Column	`TestScores`
Parameter: Select option	`Custom bin size`
Parameter: Bins	`59,69,79,89`
Parameter: New column name	`Grades`

You can then use the following transformation to assign letters in the `Grades` column:

Transformation Name	`Conditions`
Parameter: Condition type	`Case on single column`
Parameter: Column to evaluate	`Grades`
Parameter: Case - 0	`'F'`
Parameter: Case - 1	`'D'`
Parameter: Case - 2	`'C'`
Parameter: Case - 3	`'B'`
Parameter: Case - 4	`'A'`
Parameter: Default value	`'I'`
Parameter: New column name	`Grades_letters`

One-Hot Encoding

One-hot encoding refers to distributing the listed values in a column into individual columns. Within each row of each individual column is a `0` or a `1`, depending on whether the value represented by the column appears in the corresponding source column. The source column is untouched. This method of encoding allows for easier consumption of data in target systems.

> **Tip:** This transformation is particularly useful for columns containing a limited set of enumerated values.

In the following example, the values in the `BrandName` column are distributed into separate columns of binary values, with a maximum limit of `50` new columns.

> **NOTE:** Be careful applying this to a column containing a wide variety of values, such as Decimal values. Your dataset can expand significantly in size. Use the max columns setting to constrain the upper limit on dataset expansion.

Transformation Name	`One-hot encoding of values to columns`
Parameter: Column	`BrandName`
Parameter: Max number of columns to create	`50`

Tip: If needed, you can rename the columns to prepend the names with a reference to the source column.