Clean Missing Data

Updated: October 12, 2015

Specifies how to handle the values missing from a dataset

Category: Data Transformation / Manipulation (https://msdn.microsoft.com/en-us/library/azure/dn905863.aspx)

Module Overview

You can use the **Clean Missing Data** module to remove missing values, to replace them with a placeholder, mean, or other value, or to completely remove rows and columns with missing values.

Clean Missing Data does not change your source dataset—it creates a new dataset in your workspace that you can use in the subsequent workflow. However, you have the option to save the new, cleaned dataset for reuse.

Clean Missing Data also outputs a definition of the transformation that you used to clean the missing values. You can re-use this transformation on other datasets with the same schema, by using the Apply Transformation (https://msdn.microsoft.com/en-us/library/azure/dn913055.aspx) module and connecting the saved transformation and the dataset to clean.

How to Use Clean Missing Data

You can define how individual columns are cleaned, or you can save a definition for cleaning columns as a transform to reuse with the same dataset. Saving a cleaning transformation is useful if you must frequently re-import and then clean data that has the same schema.

To replace missing values

1. Connect as the input the dataset that has missing values.

2. Use the column selector to choose the columns that have missing values.

   You can choose multiple columns, but you must use the same replacement method in all selected columns.

   For example, if you want to replace some missing values with zeroes in some columns but insert a placeholder in other columns, you should use separate instances of **Clean Missing Data**.

3. Select one of the many options for handling missing values. These include:

  - Replace missing values with a placeholder.
  - Replace missing values with a calculated value, such as a mean or imputed value.
  - Delete rows or columns that contain any missing values, or that are completely empty.

  The complete list of options depends on the data type of the column you are cleaning. See the options section for detail.

4. The module returns two outputs:

   - **Cleaned dataset**. A dataset comprised of the selected columns, with missing values handled as specified.

     In addition to the dataset containing the replacement values, **Clean Missing Data** can output a column that indicates which columns met the specified criteria for cleaning.

   - **Cleaning transformation**. The definition of the data transformation used for cleaning can be saved in your workspace and applied to new data later.

To apply a cleaning transformation to new data

1. Add the Apply Transformation (https://msdn.microsoft.com/en-us/library/azure/dn913055.aspx) module to your experiment.

2. Add the dataset you want to clean, and connect the dataset to the right-hand input port.

3. Expand the **Transforms** group in the left-hand navigation pane. Locate the saved transformation and drag it into the experiment.

4. Connect the saved transformation to the left input port of Apply Transformation (https://msdn.microsoft.com/en-us/library/azure/dn913055.aspx).

5. Note that when you apply a saved transformation, you do not select the columns to apply it to. That is because the transformation has been already defined and applies automatically to the data types specified in the original operation.

   One benefit is that, if you created a transformation on a subset of numeric columns, you can later apply this transformation to a dataset of mixed column types without raising an error, and the missing values are changed only in the matching numeric columns.

6. Run the experiment.

Note these restrictions when applying a saved transformation to new data:

- A saved transformation cannot generate indicator values, even if this option was used in the original cleaning operation.

  Consider the indicator values as most useful when testing a new transformation.

- The transformation does not calculate new values for the new dataset. In other words, if you used **Clean Missing Data** on Dataset A and generated a mean value of 0.5, that same value would be applied as the mean for replacing missing values when applied to Dataset B, regardless of the actual values in Dataset B.

- The data type of the columns in the new dataset must match the data type of the column that the transformation was originally created on.

  If any other operations are performed on the column that implicitly change the data type, you will get an error.

  For example, suppose you create a mean for an integer data column [Col1], and save the transformation. Now you want to apply the cleanup transformation to a copy of [Col1] that has been adjusted using a formula, such as *([Col1] /1.5)*. To ensure that the result is an integer, you round up the result, but still get an error when you apply the transformation. However, no error is raised when you adjust the value using a formula such as *([Col 1] * 10)*. You can use Metadata Editor (https://msdn.microsoft.com/en-us/library/azure/dn905986.aspx) to explicitly reset the data type to integer. Note that, in general, operations in Apply Math Operation (https://msdn.microsoft.com/en-us/library/azure/dn905975.aspx) module implicitly change numeric columns to **double**.)

Options

Use the following parameters to customize the behavior of the **Clean Missing Data** module:

Columns to be cleaned

Select the columns that have missing values to fix by using the column selector.

Note that any method you choose will be applied to **all** columns in the selection; moreover, the module will return an error and stop the experiment if the data in any column is incompatible with the specified operation.

Therefore, typically you will need to clean string columns and numeric columns separately. For example, if you specify that missing values be replaced by a mean, the module will return an error message if you selected any string columns.

Minimum missing value ratio

Specify the minimum number of missing values required for the operation to be performed. You define the number as the ratio of missing values to all values in the column.

By default, the **Minimum missing value ratio** property is set to 0. This means that missing values will be cleaned even if there is only one missing value.

⚠ **Warning**

This condition must be met by every column in order for the specified operation to apply. For example, assume you selected three columns and then set the minimum ratio of missing values to .2 (20%); however, only one column actually has 20% missing values. In this case, the cleanup operation would apply only to the column with over 20% missing values. Therefore, the other columns would be unchanged.

Tip: If you have any doubt about whether missing values were changed, select the option, **Generate missing value indicator column** option. A column will be appended to the dataset to indicate whether or not each column met the specified criteria for the minimum and maximum ranges.

Maximum missing value ratio

Specify the maximum number of missing values required for the operation to be performed. You define the number as the ratio of missing values to all values in the column.

By default, the **Maximum missing value ratio** is set to 1. This means that missing values will be cleaned even if 100% of the values in the column are missing.

If you specify multiple columns for missing-value cleaning, but some columns do not meet the criteria you set for maximum missing value ratio, those columns will not be changed.

Cleaning Mode

Specify how you want missing values handled.

Clean Missing Data provides the following methods for replacing missing values:

Replace using MICE

For each missing value, this option assigns a new value, which is calculated by using a method described in the statistical literature as *Multivariate Imputation using Chained Equations* or *Multiple Imputation by Chained Equations*.

In a *multiple imputation method*, each variable with missing data is modeled conditionally using the other variables in the data before filling in the missing values. In contrast, in a *single imputation method* (such as replacing a missing value with a column mean) a single pass is made over the data to determine the fill value.

All imputation methods introduce some error or bias, but multiple imputation better simulates the process generating the data and the probability distribution of the data.

For a general introduction to methods for handling missing values, see Missing Data: the state of the art. Schafer and Graham, 2002 (http://www.academia.edu/1045565/Missing_Data_Our_View_of_the_State_of_the_Art).

⚠ **Warning**

The **Replace using MICE** option cannot be applied to completely empty columns. Such columns must be removed or passed to the output as is.

Custom substitution value

Use this option to specify a placeholder value (such as a 0 or NA) that applies to all missing values.

The value that you specify as a replacement must be compatible with the data type of the column.

Replace with mean

Calculates the column mean and uses the mean as the replacement value for each missing value in the column.

Applies only to columns that have Integer, Double, or Boolean data types. See the Technical Notes section for more information.

Replace with median

Calculates the column median value and assigns that as the replacement for any missing value in the column.

Applies only to columns that have Integer or Double data types. See the Technical Notes section for more information.

Replace with mode

Calculates the mode for the column and uses that as the replacement value for every missing value in the column.

Applies to columns that have Integer, Double, Boolean, or Categorical data types. See the Technical Notes section for more information.

Remove entire row

Completely removes any row in the dataset that has one or more missing values. This is useful if the missing value can be considered randomly missing.

Remove entire column

Completely removes any column in the dataset that has one or more missing values.

Replace using Probabilistic PCA

Replaces the missing values by using a linear model that analyzes the correlations between the columns and estimates a low-dimensional approximation of the data, from which the full data is reconstructed. The underlying dimensionality reduction is a probabilistic form of Principal Component Analysis (PCA), and it implements a variant of the model proposed in the Journal of the Royal Statistical Society, Series B 21(3), 611–622 by Tipping and Bishop.

Compared to other options, such as Multiple Imputation using Chained Equations (MICE), this option has the advantage of not requiring the application of predictors for each column. Instead, it approximates the covariance for the full dataset. It may therefore offer better performance for datasets that have missing values in many columns.

The key limitations of this method are that it expands categorical columns into numerical indicators and computes a dense covariance matrix of the resulting data. It also is not optimized for sparse representations. For these reasons, datasets with large numbers of columns and/or large categorical domains (tens of thousands) are not supported due to prohibitive space consumption.

Replacement value

After you have selected one of the methods that support a user-specified replacement value, you can type the new value, and it will be used as the replacement value for all missing values in the column.

Note that you can use this option only in columns with the Integer, Double, Boolean, or Date data types. For date columns, the replacement value can also be entered as the number of 100-nanosecond ticks since 1/1/0001 12:00 A.M.

Examples

Sample Results

For example, the following table shows some columns from the automobile prices sample dataset, which is relatively small but has many columns with missing values. The table shows the number of missing values and the ratio of missing values overall.

If the minimum ratio of missing values is set to 0.019 and the maximum ratio of missing values is set to 0.020, the missing values in the Normalized-losses column will not be cleaned and will be left as is.

Column name	**Count of missing values**	**Ratio of missing values**
Normalized-losses	41	0.2
Bore	4	0.019512195
Stroke	4	0.019512195

If you are uncertain about whether missing value cleanup worked, select the **Generate missing value indicator column** option. A column will be appended to the dataset for each column that met the criteria for the minimum and maximum ranges. Columns that have missing values but that did not meet the criteria will not have an indicator column.

Gallery Examples

You can see examples of how this module is used by exploring these sample experiments in the Model Gallery (http://gallery.azureml.net/):

- In the Prediction of student performance (http://go.microsoft.com/fwlink/?LinkId=525727) sample, zeros are inserted for missing values.

- In the Cross Validation for Binary Classifier sample (http://go.microsoft.com/fwlink/?LinkId=525734), zeros are used to fill-in for missing values, but an indicator column is created to track the changes. Columns with all missing values are also retained.

- In the Dataset Processing and Analysis (http://go.microsoft.com/fwlink/?LinkId=525733) sample, different branches of the experiment use different methods for missing value substitution, and the datasets are then evaluated by using Descriptive Statistics (https://msdn.microsoft.com/en-us/library/azure/dn905933.aspx) and Linear Correlation (https://msdn.microsoft.com/en-us/library/azure/dn905819.aspx).

- In the Flight delay prediction sample (http://go.microsoft.com/fwlink/?LinkId=525725), empty rows are removed.

Technical Notes

- An error occurs if the mean or median option is used when any string columns are selected. If you need to process columns of different data types, create two instances of **Clean Missing Data**.
- When replacing missing values with a mean value in columns with the Boolean, Integer, DateTime, or TimeSpan data types, the column is first converted to floating point numbers, the mean is calculated, and then the result is rounded to the nearest value of the original data type.
- When you type a replacement value, the value must be compatible with the data type in the selected column.
- Values of NaN, Inf, and –Inf are allowed for columns where the data type is Double.
- When using the MICE method, the replacement value is predicted by using the trained MICE model.
- Using **Clean Missing Data** can reset column types to feature. If your data contains other types of columns, such as labels, use Metadata Editor (https://msdn.microsoft.com/en-us/library/azure/dn905986.aspx) to correct the column types.

Expected Input

Name	Type	Description
Dataset	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	Dataset to be cleaned

Module Parameters

Name	Range	Type	Default	Description
	any	ColumnSelection	All	

Columns to be cleaned				Select columns for the missing values clean operation.
Minimum missing value ratio	[0.0;1.0]	Float	0.0	Clean only column with missing value ratio above the specified value, out of a set of all selected columns.
Maximum missing value ratio	[0.0;1.0]	Float	1.0	Clean only columns with missing value ratio below the specified value out of a set of all selected columns.
Cleaning mode	List (subset)	Handling policy	Custom substitution value	Choose an algorithm to use when cleaning missing values.
Replacement value	Any	String	"0"	Type a value to take the place of missing values. This value is optional.
Cols with all missing values	Any	ColumnsWithAllValuesMissing	Remove	Indicate if columns of all missing values should be preserved in the output.
Generate missing value indicator column	Any	Boolean	false	Generate a column that indicates which rows were cleaned.
Number of iterations	[1;10]	Integer	5	Specify the number of iterations when

				using MICE.
Number of iterations for PCA prediction	[1;50]	Integer	10	Specify the number of iterations when using a PCA prediction.

Outputs

Name	Type	Description
Cleaned dataset	Data Table (https://msdn.microsoft.com/en-us/library/azure/dn905851.aspx)	Cleaned dataset
Cleaning transformation	ITransform interface (https://msdn.microsoft.com/en-us/library/azure/dn905982.aspx)	Transformation that is to be passed to the **Apply Transformation** module to clean new data.

Exceptions

For a list of all exceptions, see Machine Learning Module Error Codes (https://msdn.microsoft.com/en-us/library/azure/dn905910.aspx).

Exception	Description
Error 0002 (https://msdn.microsoft.com/en-us/library/azure/dn906011.aspx)	An exception occurs if one or more parameters could not be parsed or converted from the specified type into the type required by the target method.
Error 0003 (https://msdn.microsoft.com/en-us/library/azure/dn906003.aspx)	An exception occurs if one or more input datasets are null or empty.
Error 0008 (https://msdn.microsoft.com/en-us/library/azure/dn905856.aspx)	An exception occurs if a parameter is not in range.
	An exception occurs if the leaner passed to the module has an invalid type.

Error 0013 (https://msdn.microsoft.com/en-us/library/azure/dn906041.aspx)	
Error 0018 (https://msdn.microsoft.com/en-us/library/azure/dn905809.aspx)	An exception occurs if the input dataset is not valid.
Error 0039 (https://msdn.microsoft.com/en-us/library/azure/dn906051.aspx)	An exception occurs if the operation fails.

See Also

Data Transformation / Manipulation (https://msdn.microsoft.com/en-us/library/azure/dn905863.aspx)
Data Transformation (https://msdn.microsoft.com/en-us/library/azure/dn905834.aspx)
A-Z List of Machine Learning Studio Modules (https://msdn.microsoft.com/en-us/library/azure/dn906033.aspx)