**Data Transformation - Sample and Split**

**Sample and Split**

Splitting and sampling datasets are both important tasks in machine learning. For example, it's a common practice to divide data into training and testing sets to help you evaluate a model on a holdout dataset. Sampling is also increasingly important in the era of big data, to ensure that there's a fair distribution of classes in your training data. Sampling also helps ensure that you're not processing more data than is needed.

You can use Machine Learning Studio (classic) modules to customize the way you split or sample datasets:

* Filter training data based on an attribute in the data.
* Perform stratified sampling to divide the class variable equally among n number of groups.
* Divide source data into a training and testing data set by using a custom ratio.
* Apply regular expressions to the data to filter out invalid values.

**Choosing the right operation: Split or sampling**

Machine Learning Studio (classic) provides two modules that encapsulate tasks. The modules sound similar, but they have different uses, and provide complementary functionality. It's likely that you'll use both modules in an experiment, to get the right amount and the right mix of data.

Next, we compare the Split Data module and the Partition and Sample module by seeing which tasks each module is commonly used for.

**Uses of the Split Data module**

* Divide data into two groups. Use the Split Data module. The module produces exactly two splits of the data. You can specify the condition on which the data is split, and the proportion of the data to put into each subset. Split Data always saves the subset of data that doesn’t meet the conditions.
* Allocate label values equally to datasets. The option to stratify on a specified column is supported by both modules. However, if you want to create two datasets and are mostly interested in the label column, the Split Data module is a quick solution.

**Example of using the Split Data module**

Suppose you imported a very large dataset from a CSV file. The dataset contains customer demographics. You want to create different models for customers in different countries, so you decide to split the data by using the value of the Country-Region column. Here are the steps you take to complete this task:

1. Add the Split Data module, and then specify an expression on the Country-Region field. The remainder of the data is available on the secondary output.
2. Add another instance of the Split Data module.
3. Repeat steps 1 and 2. Specify a different country in the expression for each iteration.

The Split Data module supports both regular expressions, for text data, and relative expressions, for numeric data.The Split Data module also provides sophisticated functionality that you can use to divide specialized datasets. Use the functionality to create recommendation models, and to generate predictions.

**Uses of the Partition and Sample module**

* ***Sampling.*** Always use the Partition and Sample module. The module provides multiple customizable sampling methods, including several options for stratified sampling.
* ***Assign cases to multiple groups.*** Use the Assign to Fold or Pick Fold options in the Partition and Sample module.
* ***Return only a subset of the data***. Use the Partition and Sample module. The module gives you the specified subset on the primary output. The remaining data is available on a secondary output.
* ***Get only the top 2,000 rows of a dataset.*** Use the Partition and Sample module. Select the Head option. This is particularly handy when you are testing a new experiment and want to run short trials of a workflow.

**Example of using the Partition and Sample module**

The Partition and Sample module can generate multiple partitions of the data, not just two. At the same time, it can perform various sampling operations.

For example, suppose you need to get just 10 percent of your data, while ensuring that the distribution of the target attribute is the same as in the source data. Here are the steps you take to complete this task:

1. Add the Partition and Sample module.
2. Choose the Sampling mode, and then specify 10%.
3. Select the stratified sampling option, and then pick the column that contains the target attribute.

If you don't need to keep all the data, use the Partition and Sample module. The remaining data is still present in the workspace, but it doesn't need to be processed further as part of the experiment.

Partition and Sample

To perform sampling on a dataset or to create partitions from your dataset. Sampling is an important tool in machine learning because it lets you reduce the size of a dataset while maintaining the same ratio of values. This module supports several related tasks that are important in machine learning:

* Dividing your data into multiple subsections of the same size.

You might use the partitions for cross-validation, or to assign cases to random groups.

* Separating data into groups and then working with data from a specific group.

After randomly assigning cases to different groups, you might need to modify the features that are associated with only one group.

* Sampling.

You can extract a percentage of the data, apply random sampling, or choose a column to use for balancing the dataset and perform stratified sampling on its values.

* Creating a smaller dataset for testing.

If you have a lot of data, you might want to use only the first n rows while setting up the experiment, and then switch to using the full dataset when you build your model. You can also use sampling to create s smaller dataset for use in development.

**How to configure Partition and Sample**

This module supports multiple methods for dividing your data into partitions or for sampling. Choose the method first, and then set additional options required by the method.

* Get the top number of rows

**Get TOP N rows from a dataset**

Use this mode to get only the first n rows. This option is useful if you want to test an experiment on a small number of rows, and don't need the data to be balanced or sampled in any way.

1. Add the Partition and Sample module to your experiment in Studio (classic), and connect the dataset.
2. Partition or sample mode: Set this option to Head.
3. Number of rows to select: Type the number of rows to return.

The number of rows you specify must be a non-negative integer. If the number of selected rows is larger than the number of rows in the dataset, the entire dataset is returned.

1. Run the experiment

The module outputs a single dataset containing only the specified number of rows. The rows are always read from the top of the dataset.

**Create a sample of data**

This option supports simple random sampling or stratified random sampling. This is useful if you want to create a smaller representative sample dataset for testing.

1. Add the Partition and Sample module to your experiment in Studio (classic), and connect the dataset
2. Partition or sample mode: Set this to Sampling.
3. Rate of sampling: Type a value between 0 and 1. this value specifies the percentage of rows from the source dataset that should be included in the output dataset.

For example, if you want only half of the original dataset, type 0.5 to indicate that the sampling rate should be 50%.

The rows of the input dataset are shuffled and selectively put into the output dataset, according to the specified ratio.

1. Random seed for sampling: Optionally, type an integer to use as a seed value.

This option is important if you want the rows to be divided the same way every time. The default value is 0, meaning that a starting seed is generated based on the system clock. This can lead to slightly different results each time you run the experiment.

1. Stratified split for sampling: Select this option if the rows in the dataset must be divided evenly by some key column before sampling.

For the Stratification key column for sampling, select a single strata column to use when dividing the dataset. The rows in the dataset are then divided as follows:

1. All input rows are grouped (stratified) by the values in the specified strata column.
2. Rows are shuffled within each group.
3. Each group is selectively added to the output dataset to meet the specified ratio.
4. Run the experiment.

with this option, the module outputs a single dataset that contains a representative sampling of the data.

The remaining, unsampled portion of the dataset is not output. However, you can create join on the datasets, using the Apply SQL Transformation module, to determine which rows were unused.

**Split data into partitions**

Use this option when you want to divide the dataset into subsets of the data. This option is also useful when you want to create a custom number of folds for cross-validation or to split rows into several groups.

1. Add the Partition and Sample module to your experiment in Studio (classic), and connect the dataset.
2. For Partition or sample mode, select Assign to Folds.
3. Use replacement in the partitioning: Select this option if you want the sampled row to be put back into the pool of rows for potential reuse. As a result, the same row might be assigned to several folds.

If you do not use a replacement (the default option), the sampled row is not put back into the pool of rows for potential reuse. As a result, each row can be assigned to only one fold.

1. Randomized split: Select this option if you want rows to be randomly assigned to folds.

If you do not select this option, rows are assigned to folds using the round-robin method.

1. Random seed: Optionally, type an integer to use as the seed value. This option is important if you want the rows to be divided the same way every time. Otherwise, the default value of 0 means that a random starting seed will be used.
2. Specify the partitioner method: Indicate how you want data to be apportioned to each partition, using these options:

* Partition evenly: Use this option to place an equal number of rows in each partition. To specify the number of output partitions, type a whole number in the Specify number of folds to split evenly into text box.
* Partition with customized proportions: Use this option to specify the size of each partition as a comma-separated list.

For example, if you want to create three partitions, with the first partition containing 50% of the data, and the remaining two partitions each containing 25% of the data, click the List of proportions separated by a comma text box, and type these numbers: .5, .25, .25

The sum of all partition sizes must add up to exactly 1.

* If you enter numbers that add up to less than 1, an extra partition is created to hold the remaining rows. For example, if you type the values .2 and .3, a third partition is created that holds the remaining 50 percent of all rows.
* If you enter numbers that add up to more than 1, an error is raised when you run the experiment.

1. Stratified split: Select this option if you want the rows to be stratified when split, and then choose the strata column.
2. Run the experiment.

With this option, the module outputs multiple datasets, partitioned using the rules you specified.

**Use data from a predefined partition**

This option is used when you have divided a dataset into multiple partitions and now want to load each partition in turn for further analysis or processing.

1. Add the Partition and Sample module to the experiment in Studio (classic).
2. Connect it to the output of a previous instance of Partition and Sample. That instance must have used the Assign to Folds option to generate some number of partitions.
3. Partition or sample mode: Select Pick Fold.
4. Specify which fold to be sampled from: Select a partition to use by typing its index. Partition indices are 1-based. For example, if you divided the dataset into three parts, the partitions would have the indices 1, 2, and 3.

If you type an invalid index value, a design-time error is raised: "Error 0018: Dataset contains invalid data."

In addition to grouping the dataset by folds, you can separate the dataset into two groups: a target fold, and everything else. To do this, type the index of a single fold, and then select the option, Pick complement of the selected fold, to get everything but the data in the specified fold.

1. If you are working with multiple partitions, you must add additional instances of the Partition and Sample module to handle each partition.For example, let's say previously partitioned patients into five folds using age. To work with each individual fold, you need five copies of the Partition and Sample module, and in each, you select a different fold.
2. Run the experiment.

With this option, the module outputs a single dataset containing only the rows assigned to that fold.

**Split Data**

This topic describes how to use the Split Data module in Machine Learning Studio (classic), to divide a dataset into two distinct sets.

This module is particularly useful when you need to separate data into training and testing sets. You can customize the way that data is divided as well. Some options support randomization of data; others are tailored for a certain data type or model type.

**How to configure Split Data**

1. Add the Split Data module to your experiment in studio. You can find this module under Data Transformation, in the Sample and Split category.
2. Splitting mode: Choose one of the following modes, depending on the type of data you have, and how you want to divide it. Each splitting mode has different options. Click the following topics for detailed instructions and examples.

* Split Rows: Use this option if you just want to divide the data into two parts. You can specify the percentage of data to put in each split, but by default, the data is divided 50-50.You can also randomize the selection of rows in each group, and use stratified sampling. In stratified sampling, you must select a single column of data for which you want values to be apportioned equally among the two result datasets.
* Recommender Split: Always choose this option if you are preparing data for use in a recommender system. It helps you divide data sets into training and testing groups while ensuring that important values such as user-item pairs or ratings are evenly divided among the groups.
* Regular Expression Split: Choose this option when you want to divide your dataset by testing a single column for a value. For example, if you are analyzing sentiment, you could check for the presence of a particular product name in a text field, and then divide the dataset into rows with the target product name, and those without.
* Relative Expression Split: Use this option whenever you want to apply a condition to a number column. The number could be a date/time field, a column containing age or dollar amounts, or even a percentage. For example, you might want to divide your data set depending on the cost of the items, group people by age ranges, or separate data by a calendar date.

**Split Data using Split Rows**

The Split Rows option supports both random and stratified splits. For example, you can specify a 70-30 split, or a 10-90 split with your target variable equally represented in both datasets.

**Divide a dataset into two groups**

1. Add the Split Data module to your experiment in Studio (classic), and connect the dataset you want to split.
2. For Splitting mode, choose Split rows.
3. Fraction of rows in the first output dataset. Use this option to determine how many rows go into the first (left-hand) output. All other rows will go to the second (right-hand) output.

The ratio represents the percentage of rows sent to the first output dataset, so you must type a decimal number between 0 and 1.For example, if you type 0.75 as the value, the dataset would be split by using a 75:25 ratio, with 75% of the rows sent to the first output dataset, and 25% sent to the second output dataset.

1. Select the Randomized split option if you want to randomize selection of data into the two groups. This is the preferred option when creating training and test datasets.
2. Random Seed: Type a non-negative integer value to initialize the pseudorandom sequence of instances to be used. This default seed is used in all modules that generate random numbers.

Specifying a seed makes the results generally reproducible. If you need to repeat the results of a split operation, you should specify a seed for the random number generator. Otherwise the random seed is set by default to 0, which means the initial seed value is obtained from the system clock. As a result, the distribution of data might be slightly different each time you perform a split.

1. Stratified split: Set this option to True to ensure that the two output datasets contain a representative sample of the values in the strata column or stratification key column.

With stratified sampling, the data is divided such that each output dataset gets roughly the same percentage of each target value. For example, you might want to ensure that your training and testing sets are roughly balanced with regard to the outcome, or with regard to some other column such as gender.

1. Run the experiment, or right-click the module and select Run selected.

**Split Data using Recommender Split**

The Recommender split option makes this process easier by asking for the type of recommendation model you are working with: for example, are you recommending items, suggesting a rating, or finding related users? It then divides the dataset by criteria you specify, such as how to handle cold users or cold items.

When you split the datasets, the module returns two datasets, one intended for training and the other for testing or model evaluation. If the input dataset contains any extra data per instance (such as ratings), it is preserved in the output.

**Divide a dataset used by a recommendation model**

The Recommender Split option is provided specifically for data used to train recommendation systems.

Before you use this option, make sure that your data is in a compatible format. The recommender splitter works under the assumption the dataset consists only of user-item pairs or user-item-rating triples. For details, see Input data requirements in this article.

1. Add the Split Data module to your experiment, and connect it as input to the dataset you want to split.
2. For Splitting mode, select Recommender split.
3. Set the following options to control how values are divided. Specify a percentage represented as a number between 0 and 1.

* Fraction of training only users: Specify the fraction of users that should be assigned only to the training data set. This means the rows would never be used to test the model.
* Fraction of test user ratings for training: Specify that some portion of the user ratings you have collected can be used for training.
* Fraction of cold users: Cold users are users that the system has not previously encountered. Typically, because the system has no information on these users, they are valuable for training, but predictions might be less accurate.
* Fraction of cold items: Cold items are items that the system has not previously encountered. Because the system has no information about these items, they are valuable for training, but predictions might be less accurate.
* Fraction of ignored users: This option allows the recommender to ignore some users, which lets you train the model on a subset of data. This might be useful for performance reasons. You specify the percentage of users that should be ignored.
* Fraction of ignored items: The recommender splitter can ignore some items and train the model on a subset of data. This might be useful for performance reasons. You specify the percentage of items to ignore.

1. Remove occasionally produced cold items: This option is typically set to zero, to ensure that all entities in the test set are included in the training set.

An item is said to be "occasionally cold" if it is covered only by the test set and it wasn't explicitly chosen as cold. Such items can be produced by steps (4) and (6) in the algorithm described in the How Recommender Data is Split section.

1. Random seed for recommender: Specify a seed value if you want to split the data the same way every time. Otherwise, by default the input data is randomly split, using a system clock value as the seed.
2. Run the experiment.

**Split Data using Regular Expression**

You can use a regular expression split on a single text column. You define a regular expression that includes the text column name, and then set conditions that apply to the column, such as "begins with", ""contains", or "does not contain".

**Use a regular expression to divide a dataset**

* Add the Split Data module to your experiment, and connect it as input to the dataset you want to split.
* For Splitting mode, select Regular expression split.
* In the Regular expression box, type a valid regular expression. Some examples are provided here.The regular expression is applied only to the specified column, which must be a string data type.
* Run the experiment, or right-click the module and select Run selected.

Based on the regular expression you provide, the dataset is divided into two sets of rows: rows with values that match the expression and all remaining rows.

**Examples**

The following examples demonstrate how to divide a dataset using the **Regular Expression** option.

**Single whole word**

This example puts into the first dataset all rows that contain the text Gryphon in the column Text, and puts other rows into the second output of **Split Data**:

*\"Text" Gryphon*

**Substring**

This example looks for the specified string in any position within the second column of the dataset, denoted here by the index value of 1. The match is case-sensitive.

*(\1) ^[a-f]*

The first result dataset contains all rows where the index column begins with one of these characters: a, b, c, d, e, f. All other rows are directed to the second output.

**String match on IP addresses**

This example divides some server log data into two categories for analysis: connections behind the firewall and connections with IP addresses outside the firewall. The regular expression is applied to the IP\_Address field (a **string** data type).

*(\IP\_Address) ^[10]*

The first output contains all addresses that begin with 10.

**Split a dataset using a relative expression**

To divide your data, you choose a single numeric column in your data, and define an expression to use in evaluating each row. The relative expression must include the column name, the value, and an operator such as greater than and less than, equal and not equals.This option divides the dataset into two groups. **Use a relative expression to divide a dataset**

1. Add the Split Data module to your experiment in Stuio, and connect it as input to the dataset you want to split.
2. For Splitting mode, select relative expression split.
3. In the Relational expression text box, type an expression that performs a numeric comparison operation, on a single column:

* The column contains numbers of any numeric data type, including date/time data types.
* The expression can reference a maximum of one column name.
* Use the ampersand character (&) for the AND operation and use the pipe character (|) for the OR operation.
* The following operators are supported: <, >, <=, >=, ==, !=
* You cannot group operations by using ( and ).

1. Run the experiment, or right-click the module and select Run selected.

The expression divides the dataset into two sets of rows: rows with values that meet the condition, and all remaining rows.

If you need to perform additional split operations, you can either add a second instance of \*Split Data or use the Apply SQL Transformation module and define a CASE statement

**Examples of relative expressions**

The following examples demonstrate how to divide a dataset using the **Relative Expression** option in the **Split Data** module:

**Using calendar year**

A common scenario is to divide a dataset by years. The following expression selects all rows where the values in the column Year are greater than 2010.

*\"Year" > 2010*

The date expression must account for all date parts that are included in the data column, and the format of dates in the data column must be consistent.

For example, in a date column using the format mmddyyyy, the expression should be something like this:

*\"Date" > 1/1/2010*

**Using column indices**

The following expression demonstrates how you can use the column index to select all rows in the first column of the dataset that contain values less than or equal to 30, but not equal to 20.

*(\0)<=30 & !=20*

**Compound operation on time values using multiple splits**

Suppose you want to split a table of log data, to group queries that run too long. You could use the following relative expression on the column, Elapsed, to get the queries that ran over 1 minute.

*\"Elapsed" >00:01:00*

To get the queries with response times under one minute but more than 30 seconds, add another instance of Split Data on the right-hand output, and use an expression like this:

\"Elapsed" <:00:01:00 & >00:00:30

**Split dataset on date values**

The following relative expression divides the dataset by using the date values in the column dt1.

*\"dt1" > 10-08-2015*

Rows with a date greater than 10-08-2015 are added to the first (left) output dataset.

Rows with a date of 10-08-2015 or earlier are added to the second (right) output dataset.

**Restrictions**

The following restrictions apply to relative expressions on a dataset:

* Relative expressions can be applied only to numeric data types and date/time data types.
* Relative expressions can reference a maximum of one column name.
* Use the ampersand character (&) for the AND operation and the pipe character (|) for the OR operation.
* The following operators are allowed for relative expressions: <, >, <=, >=, ==, !=
* Grouping operations with parentheses is not supported.