This Weka Lab has 3 sections:

1. Understand Your Machine Learning Data in Weka
2. How to Normalize and Standardize Your Machine Learning Data
3. Transforming Your Machine Learning Data

1. Understand Your Machine Learning Data in Weka

It is important to take your time to learn about your data when starting on a new machine learning problem. There are key things that you can look at to quickly learn about your dataset, such as descriptive statistics and data visualizations.

Descriptive Statistics

The Weka Explorer will automatically calculate descriptive statistics for numerical attributes.

1. Open the Weka GUI Chooser.
2. Click Explorer to open the Weka Explorer.
3. Load the Pima Indians datasets from data/diabetes.arff.

The Pima Indians dataset contains numeric input variables that we can use to demonstrate the calculation of descriptive statistics. You can learn more about this dataset in Section 8.2.1. Firstly, note that the dataset summary in the Current Relation section.

- What is the name of the dataset (relation)?
- The number of rows (instances)?
- How many features are there?

4. Click on the first attribute in the dataset in the Attributes pane.

Take note of the details in the Selected attribute pane. It lists a lot of information about the selected attribute, such as:

- The name of the attribute.
- The number of missing values and the ratio of missing values across the whole dataset.
- The number of distinct values.
- The data type.
- The key statistics

You can learn a lot from this information. i.e.

- The presence and ratio of missing data can give you an indication of whether or not you need to remove or impute values.
- The mean and standard deviation give you a quantified idea of the spread of data for each attribute.
- The number of distinct values can give you an idea of the granularity of the attribute distribution.

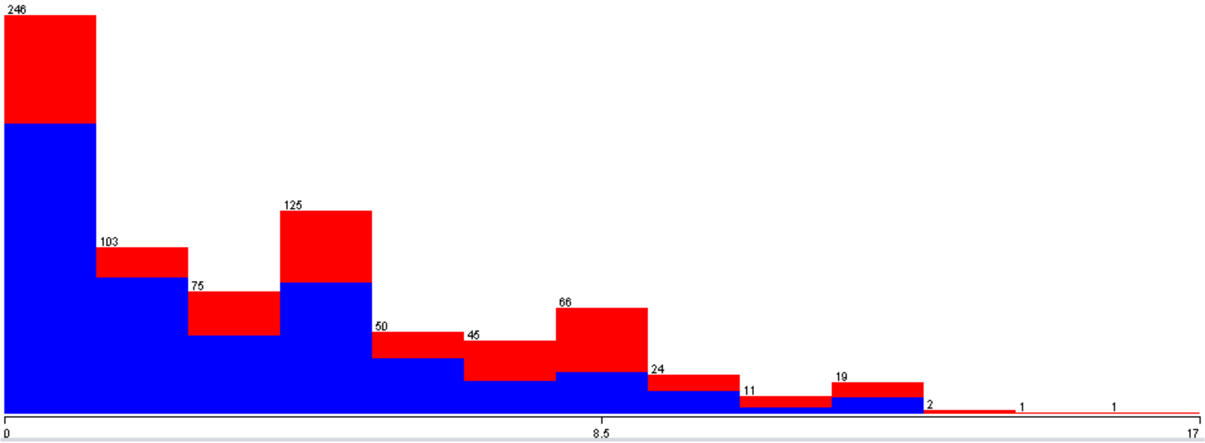
4. Click the class attribute. This attribute has a nominal type. Review the Selected attribute pane.
    - How many tested positive?

Univariate Attribute Distributions

The distribution of each attribute can be plotted to give a visual qualitative understanding of the distribution. Weka provides these plots automatically when you select an attribute in the Preprocess tab. We can follow on from the previous section where we already have the Pima Indians dataset loaded.

1. Click on the preg attribute in the Attributes pane and note the plot below the Selected attribute pane.

You will see the distribution of preg values between 0 and 17 along the x-axis. The y-axis shows the count or frequency of values with each preg value.



Note the red and blue colours overlay referring to the positive and negative classes respectively. The colours are assigned automatically to each categorical value. If there were three categories for the class value, we would see the breakdown of the preg distribution by three colours rather than two.

This is useful to get a quick idea of whether the problem is easily separable for a given attribute, e.g. all the red and blue are cleanly separated for a single attribute.

Clicking through each attribute in the list of Attributes and reviewing the plots, we can see that there is no such easy separation of the classes. We can quickly get an overview of the distribution of all attributes in the dataset and the breakdown of distributions by class by clicking the Visualize All button above the univariate plot.

Looking at these plots we can see a few interesting things about this dataset.

- Is there any observations that you can make about the distributions?

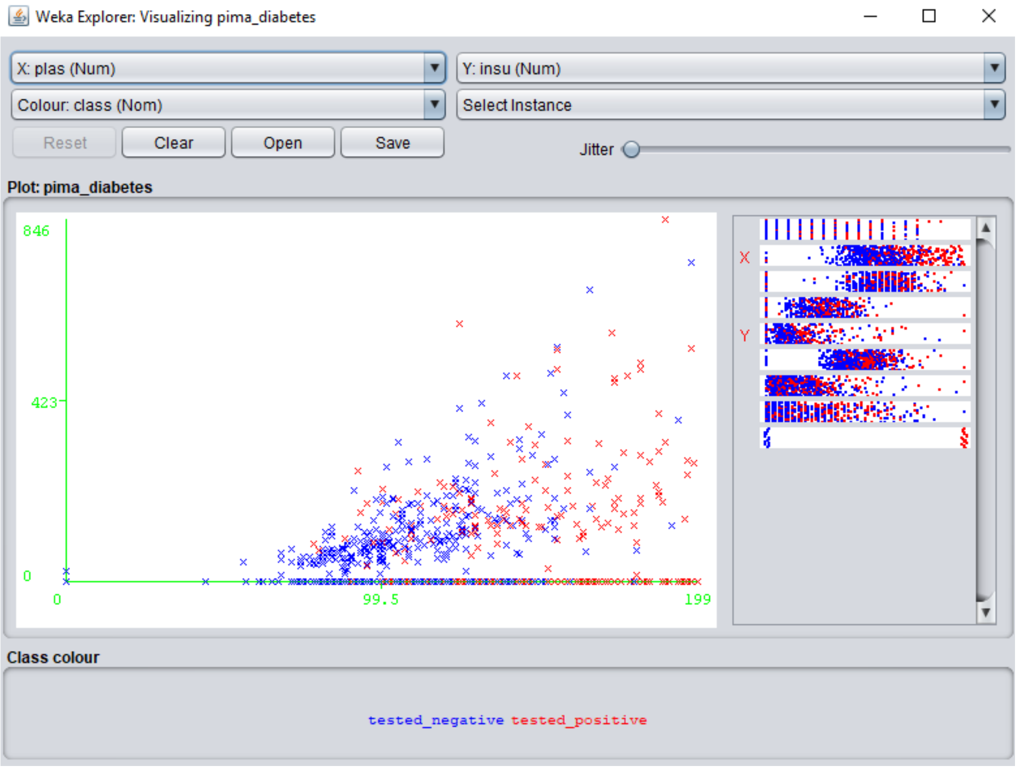
Visualize Attribute Interactions

When attributes are numeric we can create a scatter plot of one attribute against another. This is useful as it can highlight any patterns in the relationship between the attributes, such as positive or negative correlations. We can create scatter plots for all pairs of input attributes. Weka provides a scatter plot matrix for review by default in the Visualise tab.

1. Click the Visualize tab, and make the window large enough to review all of the individual scatter plots.

You can see that all combinations of attributes are plotted in a systematic way. You can also see that each plot appears twice, first in the top left triangle and again in the bottom right triangle with the axes flipped. You can also see a series of plots starting in the bottom left and continuing to the top right where each attribute is plotted against itself. These can be ignored. Finally, notice that the dots in the scatter plots are coloured by their class value. Can you see any obvious correlations (positive or negative)?

2. Click on a plot and it will give you a new window with the plot that you can further play with.



Weka Explorer: Visualizing pima_diabetes

X: plas (Num)

Y: insu (Num)

Colour: class (Nom)

Select Instance

Plot: pima_diabetes

Class colour

tested_negative tested_positive

Note the controls at the top of the screen. They let you increase the size of the plots (select Rectangle and draw it on the plot; then select submit) and add jitter. This last point about jitter is useful when you have a lot of dots overlaying each other and it is hard to see what is going on. Jitter will add some random noise to the data in the plots, spread out the points a bit and help you see what is going on. When you make a change to these controls, click the Update button to apply the changes.

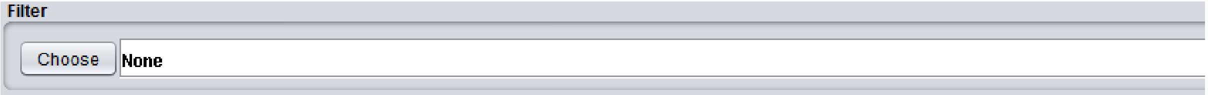
Click on one of the points on the plot. A new window should open giving you the values for that instance.

2. How to Normalize and Standardize Your Machine Learning Data

Often, raw data is comprised of attributes with varying scales. For example, one attribute may be in kilograms and another may be a count. Although not required, you can often get a boost in performance by carefully choosing methods to rescale your data. You will learn how you can rescale your data so that all of the data has the same scale.

1 About Data Filters in Weka

Weka provides filters for transforming your dataset. The best way to see what filters are supported and to play with them on your dataset is to use the Weka Explorer. The Filter pane allows you to choose a filter.



None

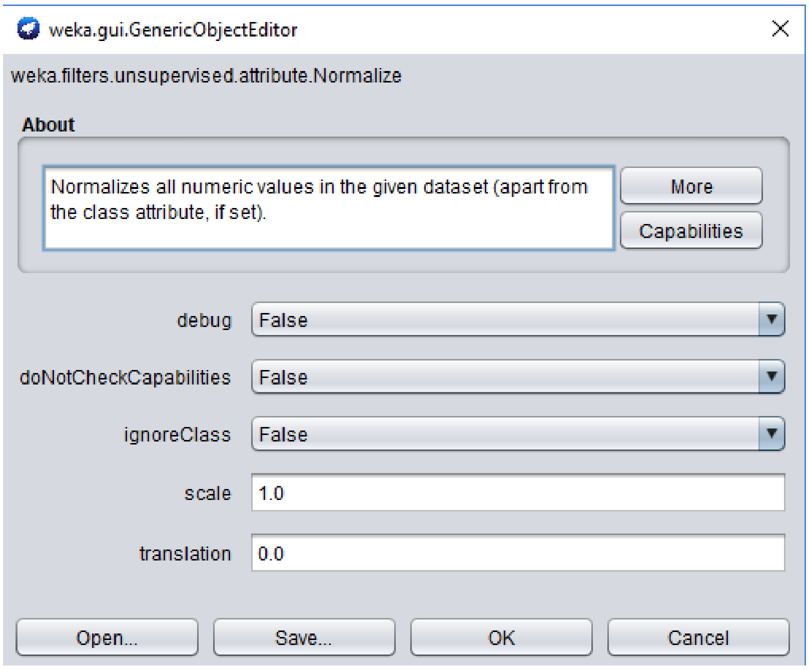
Filters are divided into two types:

- **Supervised Filters:** That can be applied but require user control in some way. Such as rebalancing instances for a class.
- **Unsupervised Filters:** That can be applied in an undirected manner. For example, rescale all values to the range 0-to-1.

Within these two groups, filters are further divided into filters for Attributes and Instances:

- **Attribute Filters:** Apply an operation on attributes or one attribute at a time.
- **Instance Filters:** Apply an operation on instance or one instance at a time.

This distinction makes a lot more sense. After you have selected a filter, its name will appear in the box next to the Choose button. You can configure a filter by clicking its name which will open the configuration window. You can change the parameters of the filter and even save or load the configuration of the filter itself. This is great for reproducibility.



weka.filters.unsupervised.attribute.Normalize

Normalizes all numeric values in the given dataset (apart from the class attribute, if set).

debug False

doNotCheckCapabilities False

ignoreClass False

scale 1.0

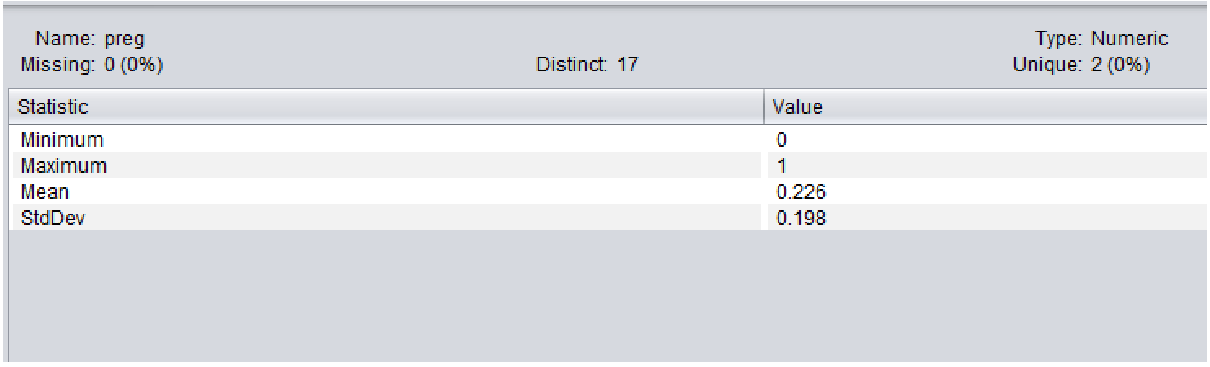
translation 0.0

You can learn more about each configuration option by hovering over it and reading the tooltip. You can also read all of the details about the filter including the configuration, papers and books for further reading and more information about the filter works by clicking the More button.

Normalize Your Numeric Attributes

Data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0. Normalization is a good technique to use when you do not know the distribution of your data or when you know the distribution is not Gaussian (a bell curve). The dataset used for this example is the Pima Indians onset of diabetes dataset. You can learn more about this dataset in Section 8.2.1. You can normalize all of the attributes in your dataset with Weka by choosing the Normalize filter and applying it to your dataset. You can use the following recipe to normalize your dataset:

1. Open the Weka Explorer.
2. Load the data/diabetes.arff dataset.
3. Click the Choose button and select the unsupervised.attribute.Normalize filter.
4. Click the Apply button to normalize your dataset.
5. Click the Save button and type a filename to save the normalized copy of your dataset.
6. Reviewing the details of each attribute in the Selected attribute window will give you confidence that the filter was successful and that each attribute was rescaled to the range of 0 to 1.



Name: preg
Type: Numeric
Missing: 0 (0%)
Distinct: 17
Unique: 2 (0%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.226
StdDev	0.198

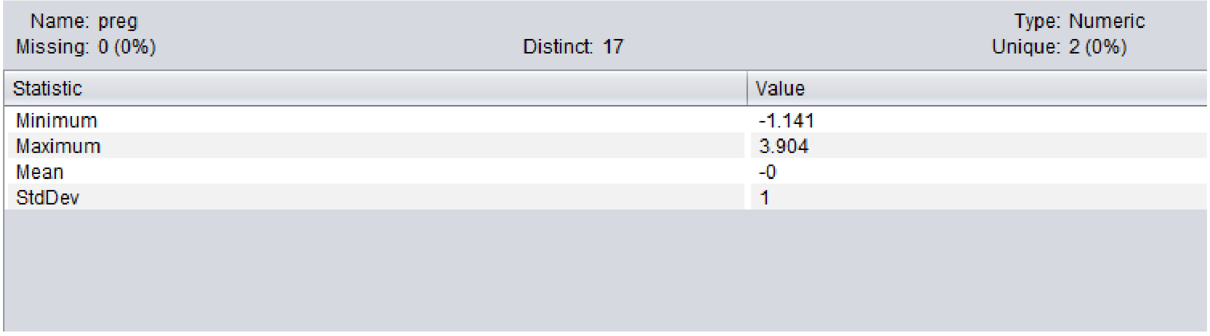
You can use other scales such as -1 to 1, which is useful when using Support Vector Machines and AdaBoost. Normalization is useful when your data has varying scales and the algorithm you are using does not make assumptions about the distribution of your data, such as k-Nearest Neighbors and Artificial Neural Networks.

Standardize Your Numeric Attributes

Data standardization is the process of rescaling one or more attributes so that they have a mean value of 0 and a standard deviation of 1. Standardization assumes that your data has a Gaussian (bell curve) distribution. This does not strictly have to be true, but the technique is more effective if your attribute distribution is Gaussian. You can standardize all of the attributes in your dataset with Weka by choosing the Standardize filter and applying it your dataset. You can use the following recipe to standardize your dataset:

1. Open the Weka Explorer.
2. Load the data/diabetes.arff dataset.
3. Click the Choose button to and select the unsupervised.attribute.Standardize filter.
4. Click the Apply button to normalize your dataset.
5. Click the Save button and type a filename to save the standardized copy of your dataset.

Reviewing the details of each attribute in the Selected attribute window will give you confidence that the filter was successful and that each attribute has a mean of 0 and a standard deviation of 1.



Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)

Statistic	Value
Minimum	-1.141
Maximum	3.904
Mean	-0
StdDev	1

Standardization is useful when your data has varying scales and the algorithm you are using does make assumptions about your data having a Gaussian distribution, such as linear regression, logistic regression and linear discriminant analysis.

3. Transforming Your Machine Learning Data

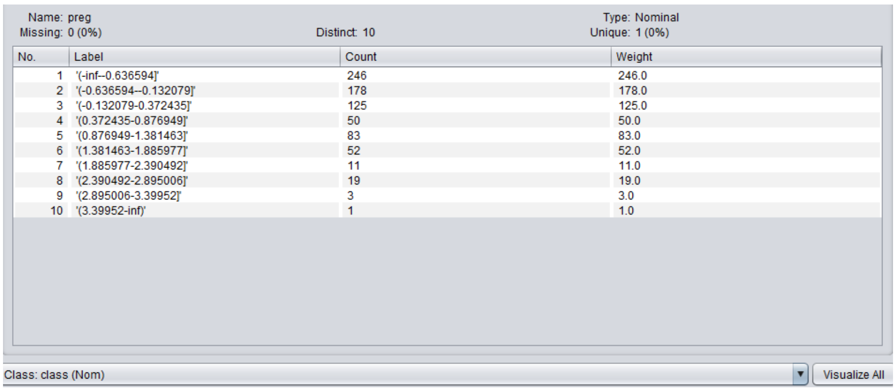
Often your raw data for machine learning is not in an ideal form for modelling. You need to prepare or reshape it to meet the expectations of different machine learning algorithms.

Discretize Numerical Attributes

Some machine learning algorithms prefer or find it easier to work with discrete attributes. For example, decision tree algorithms can choose split points in real valued attributes, but are much cleaner when split points are chosen between bins or predefined groups in the real-valued attributes. Discrete attributes are those that describe a category, called nominal attributes. Those attributes that describe a category that where there is a meaning in the order for the categories are called ordinal attributes. The process of converting a real-valued attribute into an ordinal attribute or bins is called discretization. You can discretize your real valued attributes in Weka using the Discretize filter. The tutorial below demonstrates how to use the Discretize filter. The Pima Indians onset of diabetes dataset is used to demonstrate this filter because of the input values are real-valued and grouping them into bins may make sense.

1. Open the Weka Explorer.
2. Load the data/diabetes.arff dataset

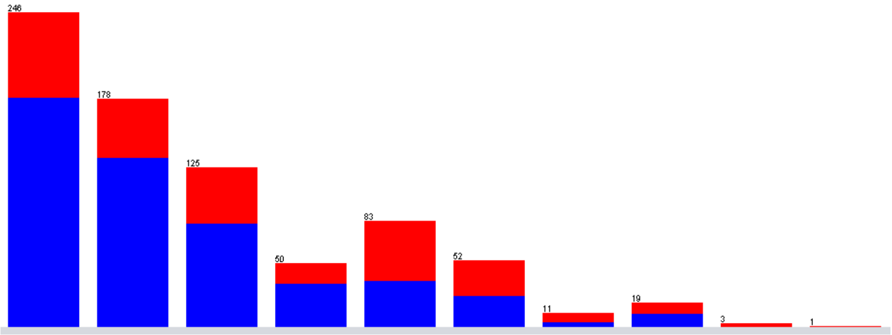
3. Click the Choose button for the Filter and select the unsupervised.attribute.Discretize filter.
4. Click on the filter to configure it. You can select the indices of the attributes to discretize, the default is to discretize all attributes, which is what we will do in this case. Click the OK button.
5. Click the Apply button to apply the filter.



Name: preg
Missing: 0 (0%)
Distinct: 10
Type: Nominal
Unique: 1 (0%)

No.	Label	Count	Weight
1	'(-inf--0.636594]'	246	246.0
2	'(-0.636594--0.132079]'	178	178.0
3	'(-0.132079-0.372435]'	125	125.0
4	'(0.372435-0.876949]'	50	50.0
5	'(0.876949-1.381463]'	83	83.0
6	'(1.381463-1.885977]'	52	52.0
7	'(1.885977-2.390492]'	11	11.0
8	'(2.390492-2.895006]'	19	19.0
9	'(2.895006-3.39952]'	3	3.0
10	'(3.39952-inf)'	1	1.0

Class: class (Nom)



You can click on each attribute and review the details in the Selected attribute pane to confirm that the filter was applied successfully.

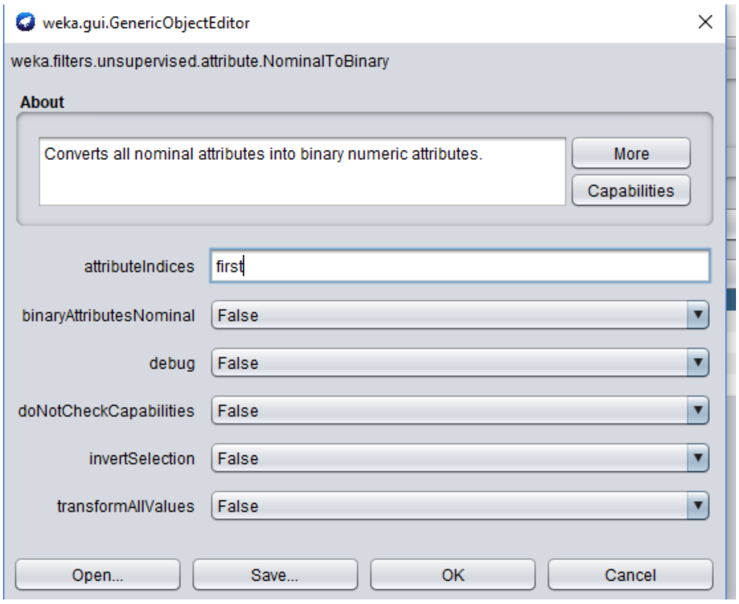
Convert Nominal Attributes to Dummy Variables

Some machine learning algorithms prefer to use real valued inputs and do not support nominal or ordinal attributes. Nominal attributes can be converted to real values. This is done by creating one new binary attribute for each category. For a given instance that has a category for that value, the binary attribute is set to 1 and the binary attributes for the other categories is set to 0. This process is called creating dummy variables.

You can create dummy binary variables from nominal attributes in Weka using the NominalToBinary filter. The instructions below demonstrates how to use the NominalToBinary filter. The Contact Lenses dataset is used to demonstrate this filter because the attributes are all nominal and provide plenty of opportunity for creating dummy variables. You can access the dataset directory in your installation of Weka under the data/ directory by loading the file contact-lenses.arff.

1. Open the Weka Explorer.
2. Load the data/contact-lenses.arff dataset.
3. Click the Choose button the unsupervised.attribute.NominalToBinary filter.

4. Click on the filter to configure it. You can select the indices of the attributes to convert to binary values, the default is to convert all attributes. Change it to only the first attribute. Click the OK button.



weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.NominalToBinary

About

Converts all nominal attributes into binary numeric attributes.

attributeIndices: first

binaryAttributesNominal: False

debug: False

doNotCheckCapabilities: False

invertSelection: False

transformAllValues: False

5. Click the Apply button to apply the filter.

Reviewing the list of attributes will show that the age attribute has been removed and replaced with three new binary attributes: age=young, age=pre-presbyopic and age=presbyopic.

Creating dummy variables is useful for techniques that do not support nominal input variables like linear regression and logistic regression. It can also prove useful in techniques like k-nearest neighbors and artificial neural networks.

Exercise

Load the *weather.nominal* dataset. Use the filter *weka. unsupervised.instance.RemoveWithValues* to remove all instances in which the *humidity* attribute has the value *high*. To do this, first make the field next to the *Choose* button show the text *RemoveWithValues*.

Exercise

Undo the change to the dataset that you just performed, and verify that the data has reverted to its original state.

Exercise

Read **Section 2.3 Filtering Algorithms** (pages 27-34) of the Weka Workbench Manual. Try out some of the filters below:

Some Unsupervised attribute filters.

Name	Function
Add	Add a new attribute, whose values are all marked as *missing.*
AddCluster	Add a new nominal attribute representing the cluster assigned to each instance by a given clustering algorithm.
AddExpression	Create a new attribute by applying a specified mathematical function to existing attributes.
AddNoise	Change a percentage of a given nominal attribute's values.
ClusterMembership	Use a clusterer to generate cluster membership values, which then form the new attributes.
Copy	Copy a range of attributes in the dataset.
Discretize	Convert numeric attributes to nominal: Specify which attributes, number of bins, whether to optimize the number of bins, and output binary attributes. Use equal-width (default) or equal-frequency binning.
FirstOrder	Apply a first-order differencing operator to a range of numeric attributes.
MakeIndicator	Replace a nominal attribute with a Boolean attribute. Assign value 1 to instances with a particular range of attribute values; otherwise, assign 0. By default, the Boolean attribute is coded as numeric.
MergeTwoValues	Merge two values of a given attribute: Specify the index of the two values to be merged.
NominalToBinary	Change a nominal attribute to several binary ones, one for each value. *Normalize* Scale all numeric values in the dataset to lie within the interval [0,1].
NumericToBinary	Convert all numeric attributes into binary ones: Nonzero values become 1.
NumericTransform	Transform a numeric attribute using any Java function.
Obfuscate	Obfuscate the dataset by renaming the relation, all attribute names, and nominal and string attribute values.
PKIDiscretize	Discretize numeric attributes using equal-frequency binning, where the number of bins is equal to the square root of the number of values (excluding missing values).
RandomProjection	Project the data onto a lower-dimensional subspace using a random matrix. *Remove* Remove attributes.
RemoveType	Remove attributes of a given type (nominal, numeric, string, or date).
RemoveUseless	Remove constant attributes, along with nominal attributes that vary too much.
ReplaceMissingValues	Replace all missing values for nominal and numeric attributes with the modes and means of the training data.
Standardize	Standardize all numeric attributes to have zero mean and unit variance.
StringToNominal	Convert a string attribute to nominal.

StringToWordVector	Convert a string attribute to a vector that represents word occurrence frequencies; you can choose the delimiter(s)—and there are many more options.
SwapValues	Swap two values of an attribute.
TimeSeriesDelta	Replace attribute values in the current instance with the difference between the current value and the value in some previous (or future) instance.
TimeSeriesTranslate	Replace attribute values in the current instance with the equivalent value in some previous (or future) instance.

Some Unsupervised instance filters.

Name	Function
NonSparseToSparse	Convert all incoming instances to sparse format
Normalize	Treat numeric attributes as a vector and normalize it to a given length
Randomize	Randomize the order of instances in a dataset
RemoveFolds	Output a specified cross-validation fold for the dataset
RemoveMisclassified	Remove instances incorrectly classified according to a specified classifier—useful for removing outliers
RemovePercentage	Remove a given percentage of a dataset
RemoveRange	Remove a given range of instances from a dataset
RemoveWithValues	Filter out instances with certain attribute values
Resample	Produce a random subsample of a dataset, sampling with replacement
SparseToNonSparse	Convert all incoming sparse instances into nonsparse format

Exercise

Examine Some of Supervised attribute filters and Supervised instance filters and try one or two out