Preparing Data for Machine Learning

Sam Scott, Mohawk College, 2019.

What you Get

Usually the data will come to you as a big table of **Delimited Values** (usually comma-separated). Each row is an **item**, **example**, or **data point** and each column represents a **feature** or **attribute** of the data. Sometimes the first row has the **feature names**, and usually the **class labels** or **targets** will be included alongside the features.

It might look a bit like this in **Excel**:

Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica

The above comes from the **Iris** data set found in the **Appendix** folder. **Species** is the target.

Separate Data from Targets

The first thing you must do is remove the header line, if any, and separate the feature **data** from the class label **targets** to form two **related arrays**, like this:

Data Array			
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1

Target Array
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica

You can do this separation in code, or you can do it using tools like **Excel** or **Notepad++**.

Clean Up the Data

You may also need to clean up the data to remove or recode non-numeric features, and remove or repair items that have missing values.

The **Iris** data is very clean, but for some implementations of machine learning algorithms, you need to replace the class labels with integers. Again, you can do this in code or you can do a simple search and replace in **Excel** or **Notepad++**. Here's what the result might look like:

Data Array			
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1

Targets Array
0
0
0
1
1
1
2
2
2

If you use pre-loaded data from **sklearn.datasets**, then the separation and cleanup steps are already done for you. See **data_preparation_example.py**.

Create Training and Testing Sets

To test your learning algorithm, you need a portion of the data for **training** (usually 75% or so) and the rest for **testing**. Data sets are often sorted in some kind of order, so it's a good idea to shuffle the arrays first, and then chop them in two. You could also just select random items and move them to the test array, but some algorithms are sensitive to sorting, so it's best to shuffle.

The **data_preparation_example.py** file shows you how to make effective use of **Numpy** to do the shuffling and splitting in just a few lines. Here's what the final arrangement might look like:

Training Data Array			
5.8	2.7	5.1	1.9
4.9	3	1.4	0.2
6.4	3.2	4.5	1.5
4.7	3.2	1.3	0.2
6.9	3.1	4.9	1.5
7.1	3	5.9	2.1

Training Targets Array
2
0
1
0
1
2

Testing Data Array			
5.1	3.5	1.4	0.2
6.3	3.3	6	2.5
7	3.2	4.7	1.4

Testing Targets Array
0
2
1