Data Preprocessing

Trung Nguyen

Contents

1 Introduction

Machine learning “gives computers the ability to learn without being explicitly programmed” (Arthur Samuel, 1959).

There are five main tasks in the machine learning work flow.

- Data collection and preparation.
- Feature selection and feature engineering.
- Choosing the machine learning algorithm (choosing the model) and training the model.
- Evaluating the model
- Model tweaking, parameter tuning and prediction.

Data Prepossessing is an important step. It transforms the raw data into an understandable format for the machine.The phrase “Garbage In ,Garbage Out” is particularly applicable to machine learning and data mining.

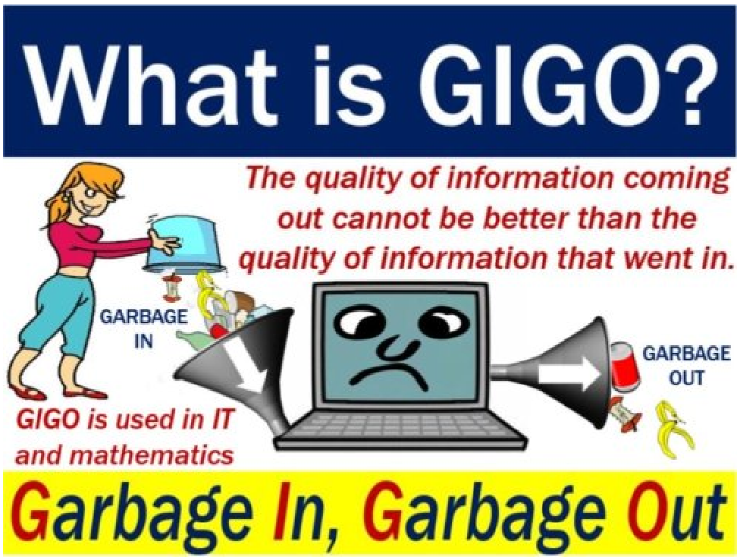

What is GIGO?
The quality of information coming out cannot be better than the quality of information that went in.
GARBAGE IN
GARBAGE OUT
GIGO is used in IT and mathematics
Garbage In, Garbage Out

Figure 1: GIGO
source:[1]

If it is much irrelevant and redundant data or noisy and unreliable data then it is difficult to understand the pattern and our machine learning algorithm will not fit well to predict the correct estimations. (No Quality data, no correct predictions). In the next chapter, the discussion on the basic step of preparing the data will be further discussed.

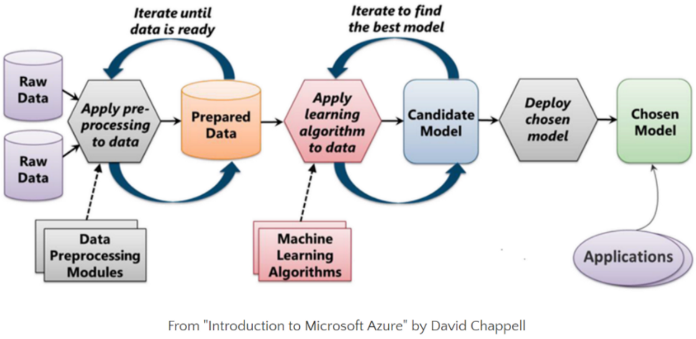

Iterate until data is ready
Iterate to find the best model
Raw Data
Raw Data
Apply pre-processing to data
Prepared Data
Apply learning algorithm to data
Candidate Model
Deploy chosen model
Chosen Model
Data Preprocessing Modules
Machine Learning Algorithms
Applications
From "Introduction to Microsoft Azure" by David Chappell

Figure 2: Machine Learning Process
source:[2]

2 Data Reprocessing and Data Visualization

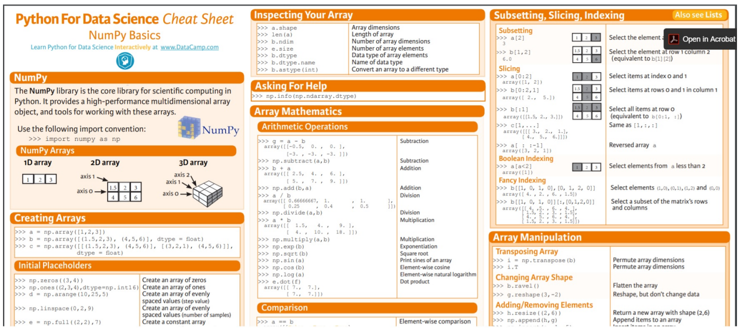
The visualization is not only important in the training steps but also important in the pre-processing steps.
The raw data are often incomplete, noisy, and inconsistent. Visualizing the data help users understand more in-depth of the data sets.
These are the most commonly used libraries in python.

- **Numpy:** the fundamental package for computing and mathematical operations with Python.
- textbfPandas: it is used for data manipulation and analysis.
- **Matplotlib :** python 2D plotting library.
- **Seaborn:** another python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Try an notebook example of data pre processing here.

3 Tips and Recommendation

A cheat sheet is a concise set of notes used for quick reference (wikipedia). Looking into a cheat sheet will give a quick overview of the libraries function. An example of Numpy cheat sheet is showing below:



Python For Data Science *Cheat Sheet*

NumPy Basics

Learn Python for Data Science Interactively at www.DataCamp.com

NumPy

The **NumPy** library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:

```
>>> import numpy as np
```

NumPy Arrays

Creating Arrays

```
>>> a = np.array([1,2,3])
>>> b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
>>> c = np.array([[(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]],
                 dtype = float)
```

Initial Placeholders

`>>> np.zeros((3,4))`	Create an array of zeros
`>>> np.ones((2,3,4),dtype=np.int16)`	Create an array of ones
`>>> d = np.arange(10,25,5)`	Create an array of evenly spaced values (step value)
`>>> np.linspace(0,2,9)`	Create an array of evenly spaced values (number of samples)
`>>> e = np.full((2,2),7)`	Create a constant array

Inspecting Your Array

`>>> a.shape`	Array dimensions
`>>> len(a)`	Length of array
`>>> b.ndim`	Number of array dimensions
`>>> e.size`	Number of array elements
`>>> b.dtype`	Data type of array elements
`>>> b.dtype.name`	Name of data type
`>>> b.astype(int)`	Convert an array to a different type

Asking For Help

```
>>> np.info(np.ndarray.dtype)
```

Array Mathematics

Arithmetic Operations

```
>>> g = a - b                                              Subtraction
 array([[-0.5,  0. ,  0. ],
        [-3. , -3. , -3. ]])
>>> np.subtract(a,b)                                       Subtraction
>>> b + a                                                  Addition
 array([[ 2.5,  4. ,  6. ],
        [ 5. ,  7. ,  9. ]])
>>> np.add(b,a)                                            Addition
>>> a / b                                                  Division
 array([[ 0.66666667,  1.        ,  1.        ],
        [ 0.25      ,  0.4       ,  0.5       ]])
>>> np.divide(a,b)                                         Division
>>> a * b                                                  Multiplication
 array([[  1.5,   4. ,   9. ],
        [  4. ,  10. ,  18. ]])
>>> np.multiply(a,b)                                       Multiplication
>>> np.exp(b)                                              Exponentiation
>>> np.sqrt(b)                                             Square root
>>> np.sin(a)                                              Print sines of an array
>>> np.cos(b)                                              Element-wise cosine
>>> np.log(a)                                              Element-wise natural logarithm
>>> e.dot(f)                                               Dot product
 array([[ 7.,  7.],
        [ 7.,  7.]])
```

Comparison

`>>> a == b`	Element-wise comparison

Subsetting, Slicing, Indexing Also see Lists

```
Subsetting
>>> a[2]                          Select the element at index 2
 3
>>> b[1,2]                        Select the element at row 1 column 2
 6.0                               (equivalent to b[1][2])
Slicing
>>> a[0:2]                        Select items at index 0 and 1
 array([1, 2])
>>> b[0:2,1]                      Select items at rows 0 and 1 in column 1
 array([ 2.,  5.])
>>> b[:1]                         Select all items at row 0
 array([[1.5, 2., 3.]])            (equivalent to b[0:1, :])
>>> c[1,...]                      Same as [1,:,:]
 array([[[ 3.,  2.,  1.],
         [ 4.,  5.,  6.]]])
>>> a[ : :-1]                     Reversed array a
 array([3, 2, 1])
Boolean Indexing
>>> a[a<2]                        Select elements from a less than 2
 array([1])
Fancy Indexing
>>> b[[1, 0, 1, 0],[0, 1, 2, 0]]  Select elements (1,0),(0,1),(1,2) and (0,0)
 array([ 4. , 2. , 6. , 1.5])
>>> b[[1, 0, 1, 0]][:,[0,1,2,0]]  Select a subset of the matrix's rows
 array([[ 4. ,5. , 6. , 4. ],      and columns
        [ 1.5, 2. , 3. , 1.5],
        [ 4. , 5. , 6. , 4. ],
        [ 1.5, 2. , 3. , 1.5]])
```

Array Manipulation

```
Transposing Array
>>> i = np.transpose(b)           Permute array dimensions
>>> i.T                           Permute array dimensions
Changing Array Shape
>>> b.ravel()                     Flatten the array
>>> g.reshape(3,-2)               Reshape, but don't change data
Adding/Removing Elements
>>> h.resize((2,6))               Return a new array with shape (2,6)
>>> np.append(h,g)                Append items to an array
```

Figure 3: Numpy cheat sheet

Some useful cheat sheets for data manipulation:

- Numpy basic cheat sheet
- Panda basic cheat sheet

- Pandas Data Wrangling Cheat Sheet
- Matplotlib Cheat Sheet
- Seaborn Cheat Sheet
- Bokeh Cheat Sheet

There are a lot of tutorial from basic to advance with step to steps guideline which available from Medium and Towardsdatascience blog spot. Read and try out those examples not only help to improve knowledge in the field but also get up to with the current development.

4 Link to relevant information

Some more example of data preprocessing on Medium and Kaggle notebook.

Medium and towardsdatascience:

- Data Preprocessing- A significant step in Machine Learning
- Data Preprocessing : Concepts
- Data Pre Processing Techniques You Should Know

.

Kaggle Notebook:

- Full Preprocessing Tutorial Notebook

5 List of Documentation

1. Python Guideline for Beginner.
2. Setting up Online Python Notebook
3. Data Preprocessing
4. update...

6 Recommendation Courses

Machine learning by Andrew Ng.
Deep Learning using FastAI by Jeremy Howard.

References

[1] URL: https://marketbusinessnews.com/financial-glossary/gigo-garbage-in-garbage-out/.

[2] David Chappelle. *Introducing Azure Machine Learning*.