

## 1. Introduction

The transmission of information from one machine to another, from one software to another or from one database to another is a recurring task.

The simplest format for data is csv format. It works well for tables but it does not easily transmit unstructured data.

The purpose of this lab is to show and try how we can manipulate unstructured (semi-structured) data formats with the python language and its libraries.

## 2. Lab datasets:

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

### Question 1 :

Download the dataset and show part of it.

#### Indications for use :

- use sklearn.datasets library

### Question 2: write the data in CSV format

#### Indications for use :

- use StringIO , to\_csv().

### Q3: write data in JSON format

#### Indications for use :

- use StringIO , to\_csv().

## 3. XML to JSON

Q4 : write the Python code that converts the XML file alongside to a JSON language code.

#### Indications for use :

- use `xml.etree.ElementTree` library
- `E.parse()`, `tree.getroot()`..

```
8 <?xml version="1.0" encoding="UTF-8"
9 <?xml-stylesheet type="text/xsl"?>
10 <employees>
11   <details>
12     <firstname>Shiv</firstname>
13     <lastname>Mishra</lastname>
14     <title>Engineer</title>
15     <division>computer</division>
16     <building>301</building>
17     <room>11</room>
18   </details>
19   <details>
20     <firstname>Yuh</firstname>
21     <lastname>Datta</lastname>
22     <title>developer</title>
23     <division>computer</division>
24     <building>303</building>
25     <room>02</room>
26   </details>
27 </employees>
28   <details>
29     <firstname>Rahil</firstname>
30     <lastname>Khan</lastname>
31     <title>Tester</title>
32     <division>computer</division>
33     <building>304</building>
34     <room>10</room>
35   </details>
36   <details>
37     <firstname>Deep</firstname>
38     <lastname>Paraph</lastname>
39     <title>Designer</title>
40     <division>computer</division>
41     <building>305</building>
42     <room>14</room>
43   </details>
44 </employees>
```

## 4. XML to JSON

Q5 : write the Python code that converts the JSON file next to a XML language code.

#### Indications for use :

- use `json` library

```
read_activations.py x file.xml x nouveau 5 x data.json x Tnsnames.ora x 2020_json_xml.py x a.xml x
1 {
2   "Name" : "Rani",
3   "Designation" : "PHP Developer",
4   "Salary" : 98000,
5   "Age":27,
6   "Projects" : [
7     {
8       "Topic":"Smart Ambulance","Category":"Android Application","Months":2},
9     {
10      "Topic":"AST","Category":"Embedded System","Months":1},
11     {
12      "Topic":"Plant Nursery","Category":"Website","Months":3}
13   ]
14 }
```

## Comparison between JSON and XML

JSON	XML
JSON object has a type	XML data is typeless
JSON types: string, number, array, Boolean	All XML data should be string
Data is readily accessible as JSON objects	XML data needs to be parsed
JSON is supported by most browsers	Cross-browser XML parsing can be tricky
JSON has no display capabilities	XML offers the capability to display data because it is a markup language
JSON supports only text and number data type.	XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data.
Retrieving value is easy	Retrieving value is difficult
Supported by many Ajax toolkit	Not fully supported by Ajax toolkit
A fully automated way of deserializing/serializing JavaScript	Developers have to write JavaScript code to serialize/de-serialize from XML
Native support for object	The object has to be express by conventions – mostly missed use of attributes and elements.
It supports only UTF-8 encoding.	It supports various encoding
It doesn't support comments.	It supports comments.
JSON files are easy to read as compared to XML.	XML documents are relatively more difficult to read and interpret.
It does not provide any support for namespaces	It supports namespaces.
It is less secured.	It is more secure than JSON.