Descriptions of the datasets

Datasets used for Supervised and Unsupervised Learning

- Iris Plants

This is perhaps the best known database to be found in the pattern recognition literature.
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.
Predicted attribute: class of iris plant.

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
   - Iris Setosa
   - Iris Versicolour
   - Iris Virginica

Datasets used for
Supervised and Unsupervised Learning

- Churn Modelling

The data set refers to customers who has exited a bank or not within the next 6 months from where the data was collected.
It contains 10000 instances and 12 relevant features (CreditScore, Geography, Age, Tenure, EstimatedSalary…).

- Social Network

The data set refers to customers who has purchased a product knowing some personal information.
It contains 400 instances and 3 relevant features (Gender, Age and EstimatedSalary).

Datasets used for
Supervised and Unsupervised Learning

- PIMA Indians Diabetes

This problem is comprised of 768 observations of medical details for Pima indians patients.

The records describe instantaneous measurements taken from the patient such as their age, the number of times pregnant and blood workup All patients are women aged 21 or older. All attributes are numeric, and their units vary from attribute to attribute.

Each record has a class value that indicates whether the patient suffered an onset of diabetes within 5 years of when the measurements were taken (1) or not (0).

This is a standard dataset that has been studied a lot in machine learning literature. A good prediction accuracy is 70%-76%.

Datasets used for
Supervised and Unsupervised Learning

- Wine recognition

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.
The analysis determined the quantities of 13 constituents found in each of the three types of wines. The aim is to predict the cultivar from the attributes.
In the data file, first column is the cultivar ID (1-3).
The attributes (from column 2 to 14) are:

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

Datasets used for
Supervised and Unsupervised Learning

- AutoMpg

The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 4 continuous attributes.

Attributes:

1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete

Dataset Loading (CSV Files) & Data Visualization

Exercises TP #1: reading CSV dataset files

Load CSV with Python Standard Library

The Python API provides the module *CSV* and the function *reader()* that can be used to load CSV files.
Once loaded, you convert the CSV data to a NumPy array and use it for machine learning.
For example, you can download the Pima Indians dataset into your local directory.
All fields are numeric and there is no header line.
Running the recipe below will load the CSV file and convert it to a NumPy array.

```
import csv
import numpy
filename = 'pima-indians-diabetes.data.csv'
raw_data = open(filename, 'rt')
reader = csv.reader(raw_data, delimiter=',', quoting=csv.QUOTE_NONE)
x = list(reader)
data = numpy.array(x).astype('float')
print(data,data.shape)
```

The example loads an object that can iterate over each row of the data and can easily be converted into a NumPy array. Running the example prints the shape of the array.

For more information on the csv.reader() function, see CSV File Reading and Writing in the Python API documentation.

Exercises TP #1: reading CSV dataset files

Load CSV File With NumPy

You can load your CSV data using NumPy and the numpy.loadtxt() function.

This function assumes no header row and all data has the same format. The example below assumes that the file pima-indians-diabetes.data.csv is in your current working directory.

```
import numpy
filename = pima-indians-diabetes.data.csv’
raw_data = open(filename, 'rt')
data = numpy.loadtxt(raw_data, delimiter=",")
print(data,data.shape)
```

Exercises TP #1: reading CSV dataset files

Load CSV File With Pandas

You can load your CSV data using Pandas and the pandas.read_csv() function.

This function is very flexible and is perhaps my recommended approach for loading your machine learning data. The function returns a pandas.DataFrame that you can immediately start summarizing and plotting.

The example below assumes that the ‘pima-indians-diabetes.data.csv‘ file is in the current working directory.

```
import pandas
filename = 'pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
data = pandas.read_csv(filename, names=names)
print(data,data.shape)
```

Exercises TP #1: reading CSV dataset files

Exercise 1:

Load the data of the following datasets using one of the three previous methods and verify you can access each data:

- PIMA Indians,
- Iris Plant,
- ChurnModelling,
- SocialNetwork

Exercises TP #1

Exercise 2:

Visualize as clearly as possible the data in the datasets:

- PIMA Indians,
- Iris Plant,
- ChurnModelling,
- SocialNetwork and
- MNIST

using one visualization method or comparing several methods we saw for Data Visualization and Dimensionality Reduction.

Supervised Learning

Exercises TP #2: Part 1

Problem 1
Study the python code *classification_template.py* where a logistic regression is used on the *Social Network* dataset, by taking into account the age and estimated salary to predict the output.

Problem 2
Apply a logistic regression on the *Pima Indians Diabetes* and on the *Iris Plant* dataset.

Exercises TP #2: Part 2

Problem 3

Study the python code *ann.py* where an MLP network is applied on the *Churn Modelling* dataset.

Used architecture: ((6,‘relu‘),(6,‘relu‘),(1,‘sigmoid‘))

Problem 4

With the trained MLP, predict the result for this person:

Geography: France
Credit Score: 600
Gender: Male
Age: 40
Tenure: 3
Balance: 60000
Number of Products: 2
Has Credit Card: Yes
Is Active Member: Yes
Estimated Salary: 50000

Problem 5

Apply an MLP on the *Pima Indians Diabetes* dataset.

Exercises TP #3

Problem 1
Study the python code *rnn_template.py* where a LSTM recurrent neural network is used to predict the stock value of Google.

Problem 2
Learn how to use a GRU recurrent network and apply it on the same dataset to compare the results of both kinds of recurrent networks.

Problem 3
Study the python code *cnn_template.py* where a Convolutional Neural Network is used to predict if an image is a cat or a dog.

Problem 4
Apply the CNN to predict the labels of the images in *dataset/single_prediction*.

Unsupervised Learning

Exercises TP #4

Problem 1
Study the code *'ul_template.py‘* and apply the GMM and Agglomerative Hierarchical Clustering on the random dataset. Compare the different results.

Problem 2
Apply the KMeans, MeanShift, DBScan, GMM and Agglomerative Hierarchical Clustering on the datasets *IrisPlant* and *PIMA Indians*. Determine the error in unsupervised classification compared to the correct targets.

Recommendation Systems

Exercises TP #5

Problem 1

Make sure you understand the codes *‘autoencoders.py‘* and *‘restricted_boltzmann_machines.py‘* .

Problem 2

Apply an Autoencoders or a Restricted Boltzmann Machine to the dataset *Anime*, available at:

https://github.com/caserec/Datasets-for-Recommneder-Systems/tree/master/Processed%20Datasets/Anime

Image Segmentation

Exercises TP #6

Problem 1
Download the file ‘ImageSegmentation.zip‘ on Moodle and test the python code about Region-Based Segmentation and Edge Detection.

Problem 2
Download an image and apply a neural network like Mask R-CNN to have an instance segmentation.