# Podstawy Python - przetwarzanie danych

## Przygotowywanie rozwiązań w notebookach IPYNB

Rozwiązania ćwiczeń, które będą Państwo wykonywać w ramach laboratoriów nalezy wysyłać na UPEL, w postaci zeszytów .ipynb, które można wygenerować korzystając np. z Jupyter Notebook lub Google Colab. Szczegółówa instrukcja znajduje sie w poniższym linku:

Preparing results - Jupyter Notebook or Google Colab?

## Pakiet Pandas

Jednym z podstawowych pakietów do analizy danych w języku Python jest pakiet Pandas. W trakcie wykonywania zadań proszę wykorzystać podstawowe operacje związane z analizą danych za pomocą funkcji dostępnych w pakiecie Pandas.

Proszę zapoznać się z Pandas Cheat Sheet

## Import Pakietów

Pierwszym krokiem do rozpoczęcia pracy z Python i Pandas jest zaimportowanie niezbędnych pakietów. W proponowanym przykładzie należy zaimportować pakiet NumPy do obliczeń numerycznych, zwłaszcza obliczeń wektorowo-macierzowych, pakiet Matplotlib do tworzenia wykresów oraz oczywiście sam pakiet Pandas. Wymienione pakiety zostały domyślnie umieszczone w popularnej dystrybucji Anaconda Python używanej w Data Science i są domyślnie instalowane

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

## Zadanie - baza danych Iris

Proszę zapoznać się z bazą danych Iris Iris flowers
Zestaw pomiarów kwiatów irysa, udostępniony po raz pierwszy przez Ronalda Fishera w roku 1936. Jeden z najbardziej znanych zbiorów, a zarazem bardzo prosty i użyteczny. Zbiór irysów składa się z 4 wartości pomiarów jego płatków (szerokości i długość) oraz klasy do jakiej należy. Przykład kilku wybranych rekordów:

```
sepal length,sepal width,petal length,petal width,class
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
```

Import danych:

```
from sklearn.datasets import load_iris
# load the famous iris data
irisRaw = load_iris()
```

Konwersja danych do pakietu Pandas

```
# read iris.data into a pandas DataFrame (df), including column names
iris = pd.DataFrame(data= np.c_[irisRaw['data'], irisRaw['target']],
                    columns= irisRaw['feature_names'] + ['target'])
```

Ćwiczenia do wykonania:

1. Proszę wyświetlić dane i zapoznać się z posczególnymi informacjami.
2. Wyświetl liczbę wierszy oraz kolumn.
3. Wyświetl podstawowe informacje dla poszczególnych kolumn. Skorzystaj z metod `describe`.
4. Zapoznaj się z działaniem metody `groupby`.
5. Wykorzystując metodę head proszę wyświelić 5 pierwszych wierszy
6. Korzystając z `dropna` sprawdź, czy baza zawiera brakujące dane
7. Korzystając z metody `sort_values` posortuj dane rosnąco względem drugiej kolumny.
8. Wyznacz minimalną i maksymalną długość płatka (kolumna 3 - petal length). Podaj indeks tych wartości (`min`, `idxmin`).
9. Oblicz odchylenie standardowe dla każdej kolumny (`std`)
10. Wyodrębnij wiersze dla których długość kielicha kwiatów (sepal length) jest większa od średniej długości (cały zbiór)
11. Wyświetl histogramy dla poszczególnych parametrów z uwzglednieniem przynależności do danej klasy

Proszę zapoznać się z możliwościami generowaniem wykresów:
PyPlot tutorial

**Rozwiązane zadania, w formie jednego pliku .ipynb, należy umieścić na platformie UPEL.**

## Introduction

Preparing results - Jupyter Notebooks and Google Colab How to prepare results

For each exercise please prepare results in Jupyter Notebook format (.ipynb file). If you never used it before, Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. You don't have to use specifically Jupyter Notebook application. There are other notebook-like software (such as Google Colab) that allow you to save work in the .ipynb format. You can also use classic Python editor (like Spyder or PyCharm), which allows some extended developer options and only copy and run the final code in a notebook.

After you finish each lab, create a notebook containing all the code, training progress bars, results metrics (like accuracy), plots, graphics, discussion and answers to questions and upload it on UPEL

platform. Why do we like .ipynb format?

We see your results, not only code.

- No need to spend time for training your models again each time.
- No problem if we use different libraries and versions.
- No need to change paths to input data each time.
- No “but it worked on my computer” problem ;)
- Your answers and comments are harder to miss
- If you fail to complete the task, because of some errors, we can see error message and try to help.

## Libraries Needed

For the next 11 labs you will need Python and apropriate libaries installed. In this course you will mostly use: `keras`, `numpy`, `matplotlib` and in some rare cases also `sklearn`, `cv2`, `os`, `shutil`.

## Installing Jupyter Notebook

Jupyter Notebook can be run on a PC/Laptop without Internet access, or it can be installed on a remote server, where you can access it through the Internet.

You can install Jupyter Notebook using *anaconda* environment, using *pip* or with a *docker*, whichever you prefer. Commands for installation can be found in the Installation Guide. Detailed instructions are in tutorials at the end of this section.

Below you can find links to some useful tutorials and documentation for installing and using Jupyter Notebook:
Jupyter Documentation
Tutorial 1
Tutorial 2
Tutorial 3

## Using Google Colab

Google Colab is a free notebook environment, based on Jupyter Notebook, that runs entirely in the cloud. It can be very useful for training larger neural network models if you don't have access to a fast PC. Furthermore, Google Colab provides free access to a GPU - graphics processing units, which largely speed up machine learning tasks. So, if you don't have a fast processor or the NVIDIA graphic card, Google Colab is a way to go. It will save you some time.

Most of the libraries you will use in this course are already pre-installed, but if you need something that is not, you can always add it via `pip`.

Google Colab works on Drive, so you may sometimes need to copy some files into the cloud. You will also need an internet connection.

In the links below, you will find some tutorials:
Tutorial 1
Tutorial 2 (Video)

To sum things up, if you don't mind working in the cloud and want to train networks faster, use Google Colab. If you prefer to work offline, the basic Jupyter Notebook will be for you. Most exercises won't be that computationally expensive so you can finish the course on CPU, without access to GPU.

**Your task is to choose and configure the environment you will work on. Try to write something to make yourself familiar with notebooks**

# Introduction Python

## Excercise 1 - database Iris

Please get familiar with the database `Iris` Iris flowers
Iris flower measurement kit, first made available by Ronald Fisher in 1936. One of the most famous collections, yet very simple and useful. The set of irises consists of 4 measurement values of its petals (width and length) and the class to which it belongs. An example of a few selected records:

```
sepal length,sepal width,petal length,petal width,class
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
```

Import danych:

```
from sklearn.datasets import load_iris
# load the famous iris data
irisRaw = load_iris()

#Convert data to Panda package
# read iris.data into a pandas DataFrame (df), including column names
iris = pd.DataFrame(data= np.c_[irisRaw['data'], irisRaw['target']],
                    columns= irisRaw['feature_names'] + ['target'])
```

1. Please view the data and refer to the specific information.
2. Display the number of rows and columns.
3. View basic information for individual columns. Use the 'describe' and 'groupby' methods
4. Using the 'head' method please display the first 5 rows
5. Using “dropna” check if the database contains missing data
6. Sorting the dataset according to specific criteria is another important part of the pandas package. In order to sort by rows or columns, use the sort_values method, which returns a new, sorted object.
7. Sort the data in ascending order relative to the second column. (Sorting by index Panda-sort_index())

8. Find the minimum and maximum petal length (column 3 - petal length). Give the index of each of these values.
9. Calculate the standard deviation for each column
10. Identify rows for which the length of the flower cup (sepal length) is greater than the average length (the whole set)

Please see the charts generation options:
PyPlot tutorial

1. Display histograms for individual parameters with regard to belonging to a given class

From:
https://home.agh.edu.pl/~mdig/dokuwiki/ - **MVG Group**

Permanent link:
**https://home.agh.edu.pl/~mdig/dokuwiki/doku.php?id=teaching:data_science:ml_pl:topics:intro**

Last update: **2023/03/01 13:54**