1 Laborator 1

1.1 Instalare Anaconda

Este explicata in detaliu la alt laborator (https://github.com/lmsasu/cursuri/tree/master/InteligentaArtificiala/laborator/laborat

Exercitiu: creati un mediu virtual numit *ids*, folosind conda, si actualizati-i pachetele

1.2 Manifestul Python

Lansati interpretorul interactiv Python in linia de comanda: `ipython` si apoi rulati cor afisa pe ecran manifestul Python the Zen of Python (https://www.python.org/dev/pe

1.3 Jupyter lab

Unii prefera folosirea de Jupyter lab in loc de Jupyter notebook; cel din urma este re sugerandu-se sa se treaca ulterior la Jupyter lab. Jupyter lab este deja disponibil in Anaconda, sau poate fi instalat conform instructiunilor de aici (https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html).

Jupyter lab vs Jupyter notebook:

- link1 (https://medium.com/@brianray_7981/jupyterlab-first-impressions-e6d70d
- link2 (https://towardsdatascience.com/jupyter-notebooks-are-breathtakingly-fea be858a67b59d)

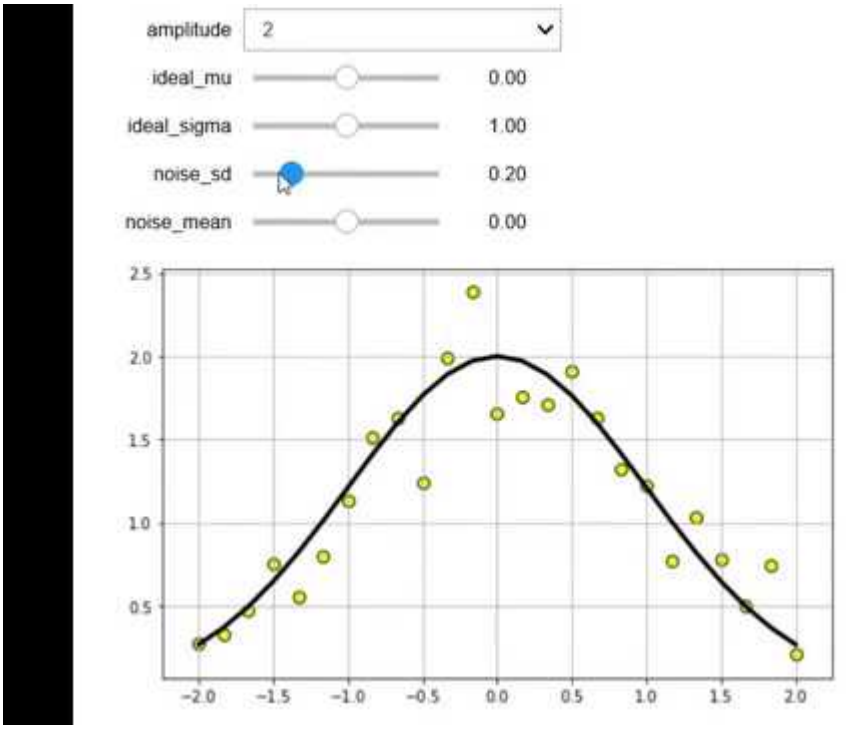
Exercitiu: Instalati si porniti Jupyter lab.

1.4 Folosire de controale grafice

Notebook-urile - indiferent ca se ruleaza in Jupyter lab sau Jupyter notebook - se pc interactive. O varianta este modificarea codului ijn timpul demo-ului si rularea celule rapid de facut. O alta varianta eset folosirea de controale grafice care sa permita uti valori de parametri etc.

ipywidgets (https://ipywidgets.readthedocs.io/en/stable/) este o biblioteca de contro interactiune cu utilizatorul. Mai jos sunt cateva demo-uri de urmarit.

- Demo 1:

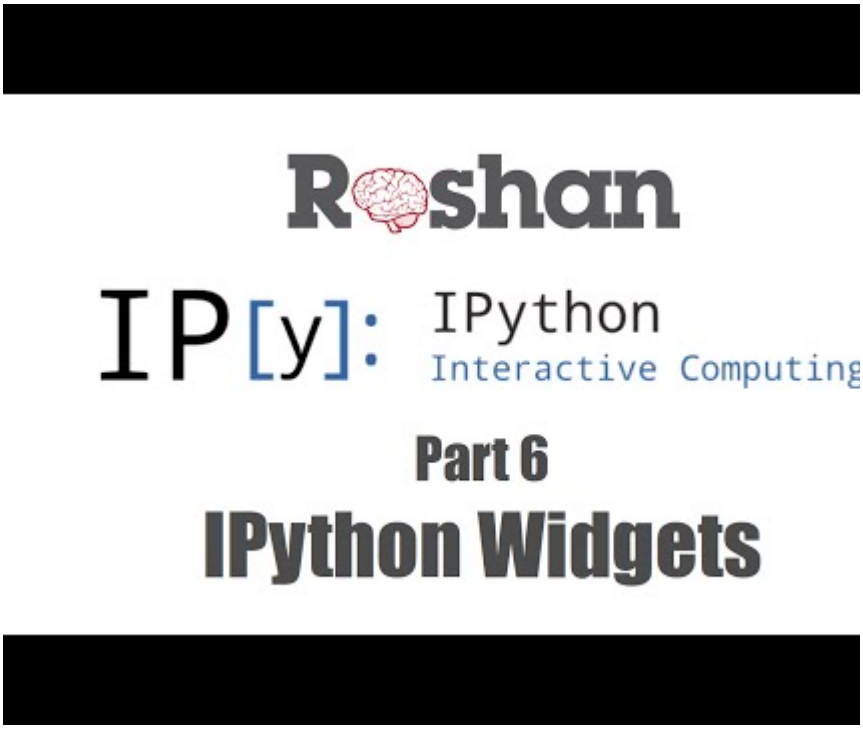


(https://www.youtube.com/watch?v=nRmkS_6ngCU)

- Demo 2:

(https://www.youtube.com/watch?v=j5d7vOQBttI)

- Demo 3:



(https://www.youtube.com/watch?v=wxVx54ax47s)

1.4.1 Exemple de utilizare

Un tutorial amplu se gaseste aici (https://ipywidgets.readthedocs.io/en/stable/user_

Incarcarea pachetului de `ipywidgets` se face prin:

In [1]:
```
import ipywidgets as widgets
```

executed in 878ms, finished 18:07:34 2019-02-21

De regula, e nevoie si de alte pachete, de exemplu:

In [2]:
```
from ipywidgets import interact, interactive, fixed, interact_manual
```

executed in 1.04s, finished 18:07:38 2019-02-21

Cel mai simplu control utilizabil este `interact` . El poate prelua ca prim parametru n
doilea parametru dicteaza forma controlului: slider, combo box, checkbox etc:

In [3]:
```
def n_factorial(n):
    """Calculeaza n factorial"""
    p = 1
    for i in range(1, n+1):
        p *= i
    return str(n) + "!= " + str(p)
```

executed in 1.09s, finished 18:07:40 2019-02-21

In [4]:
```
interact(n_factorial, n=100)
```

executed in 8.61s, finished 18:07:50 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found
kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.n_factorial(n)>
```

Pentru limitarea domeniului in care n poate sa ia valori se va folosi:

In [5]:
```
interact(n_factorial, n=(0, 100))
```

executed in 9.05s, finished 18:08:02 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found
kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.n_factorial(n)>
```

Pentru a evita actualizarea sacadata a valorilor afisate, se prefera inhibaarea feedb
in Disabling continuous updates
(https://ipywidgets.readthedocs.io/en/stable/examples/Using%20Interact.html#Disab

Pentru alte tipuri de controale folosind interact, se poate folosi:

In [ ]:
```
def g(x, y, z, t):
    return (x, y, z, t)

interact(g, x=True, y=(1.0, 10.0, 0.5), z='Un text',
         t={'English':'Hello', 'Romanian':'Salut', 'Spanish':'Hola'})
```

execution queued 18:07:33 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found
kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.g(x, y, z, t)>
```

Exemplu: Sa se deseneze graficul functiei $f : [-10, 20] \to \mathcal{R}$, $f(x) = a \cdot x^4 + b$
a, b, c, d, e coeficienti reali.

Rezolvare:

In [ ]:
```
# import de pachete numerice si grafice

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

execution queued 18:07:33 2019-02-21

In [ ]:
```
def f_square(a=10, b=20, c=-10,left=-10, right=20):
    assert left < right
    range_x = np.linspace(left, right, 100)
    values_f = a * range_x ** 2 + b * range_x + c
    plt.figure(figsize=(20, 10))
    plt.xlabel('x')
    plt.ylabel('$a\cdot x^2 + b\cdot x + c$')
    plt.plot(range_x, values_f, color='red')
    plt.show()

interact(f_square, a=(-100, 100.0), b=(-100, 100.0), c=(-100, 100.0), d=(-100, 10
```

execution queued 18:07:33 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found
kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.f_square(a=10, b=20, c=-10, left=-10, right=20)>
```

In [ ]:
```
def sinusoid(f=10):
    range_x = np.linspace(-5, 5, 100)
    values_f = np.sin(2 * np.pi * f * range_x)
    plt.xlabel('x')
    plt.ylabel(f'$2 \cdot \pi \cdot {f} \cdot x$')
    plt.plot(range_x, values_f)

interact(sinusoid, f = (1, 100.0, 0.5))
```

execution queued 18:07:33 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found
kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.sinusoid(f=10)>
```

In [ ]:

```
def f(x):
    """calcul functie intr-un punct"""
    return x ** 2 - 10 * x + 50

def f_values(left=-10, right=10):
    """calcul functie pe interval"""
    x = np.linspace(left, right, 100)
    return x, f(x)

def f_prime(x):
    """Calcul derivata f
    :param x: punctul in care se calculeaza derivata
    :return: f'(x)
    """
    return 2 * x - 10

def graf_f_and_derived(x, left=-30, right=30):
    # calcul valoare functie f
    x_range, fx = f_values(left, right)

    # intervalul pe care se reprezinta tangenta
    x_segment = np.linspace(x-10, x+10, 100)
    # panta tangentei la grafic este derivata functiei in pctul de tangenta
    slope = f_prime(x)

    #calcul puncte de tangenta
    y_segment = f(x) + slope * (x_segment - x)

    plt.figure(figsize=(20, 10))
    plt.plot(x_range, fx, color='red')
    plt.plot(x_segment, y_segment, color='blue')

# graf_f_and_derived(10, left=-30, right=30)

interact(graf_f_and_derived, x = (-20, 20))
```

execution queued 18:07:33 2019-02-21

A Jupyter widget could not be displayed because the widget state could not be found

kernel storing the widget is no longer available, or if the widget state was not saved i
able to create the widget by running the appropriate cells.

```
<function __main__.graf_f_and_derived(x, left=-30, right=30)>
```

1.4.2 Exercitii:

1. Reprezentati functia $a * x^4 + b * x^3 + c * x^2 + d * x +$ si derivata ei intr-ur
2. Incarcati fisierul de date iris (https://archive.ics.uci.edu/ml/machine-learning-dat functie de alegerile exprimate de un utilizator, afisati intr-un grafic bidimensiona (de exemplu, coloana 0 si coloana 2).
3. Arpad, poti sa mai adaugi 2 exercitii?

1.5 Ciclari, siruri de caractere

Se recomanda ca urmatoarele exercitii sa le lucrari in Jupyter notebook. In linia de c sau PowerShell) scrieti comanda:

```
jupyter notebook
```

Se va deschide automa browserul implicit la adresa localhost:8888 (daca portul 888 automat un alt port).

1. (fizz-buzz test) Sa se scrie numerele de la 1 la n; pentru fiecare multiplu de 3 s 'Fizz', pentru multiplu al lui 5 se va scrie 'Buzz'; daca numarul este multiplu de 'FizzBuzz'.
2. Sa se verifice conjectura Collatz (https://en.wikipedia.org/wiki/Collatz_conjectur 1000.
3. * Sa se creeze o functie care preia un numar n si returneaza un alt numar pe b calculeaza numarul cifrelor pare din n (fie el si 0), numarul de cifre impare (poa formeaza numarul din acestea 3; daca exista vreun zero nesemnificativ, acesta obtinut se va supune aceleiasi transformari. Exemplu: 3->11->22->202->303-> dupa un numar finit de transformari se ajunge la numarul 123; faceti aceste ver intevalul 1, 1000.
4. * Se pleaca de la un numar intreg. Fiecare cifra a sa se scrie cu litere, in limba >five). Claculati numarul total de caractere rezultate, iar pentru numarul obtinut pentru numerele de la 1 la n ca se obtine intr-un numar finit de pasi numarul 4. >11->oneone->6->six->3->three->5->five->4->four->4->four....
5. * Este rezultatul de mai sus valabil si pentru transcriere in limba romana?
6. * Se pleaca de la un numar n; se scriu toti divizorii sai, inclusiv 1 si n; se aduna divizori; pentru numarul obtinut se aplica acelasi procedeu. Verificati ca procesu 15. Exemplu: 20->1, 2, 4, 5, 10, 20 -> suma cifrelor: 15->1, 3, 5, 15-> suma cifr

In [ ]: