1 Laborator 1

1.1 Instalare Anaconda

Este explicata in detaliu la <u>alt laborator</u> (<a href="https://github.com/lmsasu/cursuri/tree/master/InteligentaArtificiala/laborator/laborater/laborator/laborater/lnteligentaArtificiala/laborater/lnteligentaArtificiala

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 notebook

In linia de comanda (command prompt sau powershell) scrieti comanda:

jupyter notebook

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

▼ 1.4 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 <u>aici</u>

(https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html).

Jupyter lab vs Jupyter notebook:

- <u>link1 (https://medium.com/@brianray_7981/jupyterlab-first-impressions-e6d70d</u>
- <u>link2 (https://towardsdatascience.com/jupyter-notebooks-are-breathtakingly-feabe858a67b59d)</u>

Se ruleaza din linia de comanda cu:

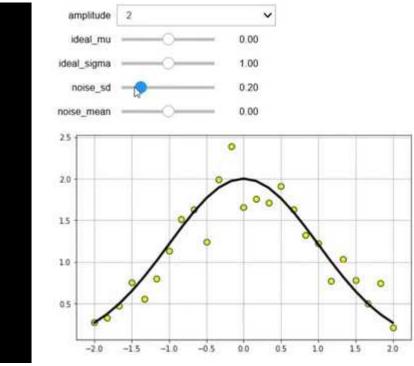
jupyter lab

1.5 Folosire de controale grafice

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

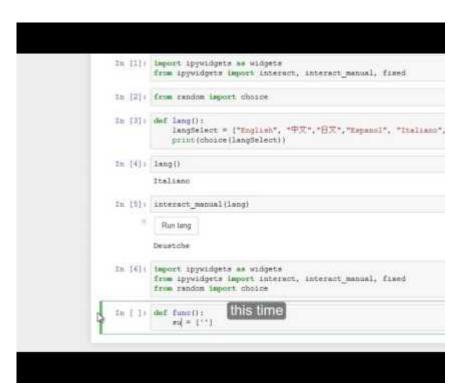
<u>ipywidgets (https://ipywidgets.readthedocs.io/en/stable/)</u> este o biblioteca de controi interactiune cu utilizatorul. Mai jos sunt cateva demo-uri de urmarit.

• Demo 1:



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

• Demo 2:



(https://www.youtube.com/watch?v=j5d7vOQBttl)

Demo 3:



IP[y]: IPython
Interactive Computing

Part 6 **IPython Widgets**

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

1.5.1 Exemple de utilizare

Documentatia completa si exemple <u>aici (https://ipywidgets.readthedocs.io/en/stable/linearcarea pachetului de ipywidgets</u> se face prin:

In [39]: 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 [40]: 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 ne doilea parametru dicteaza forma controlului: slider, combo box, checkbox etc:

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

```
In [42]:
          interact(n factorial, n=100)
         executed in 8.61s, finished 18:07:50 2019-02-21
           interactive(children=(IntSlider(value=100, description='n', max=300, min=-100), Output()), dom classes
           <function __main__.n_factorial(n)>
            Pentru limitarea domeniului in care n poate sa ia valori se va folosi:
In [43]:
          interact(n factorial, n=(0, 100))
         executed in 9.05s, finished 18:08:02 2019-02-21
           interactive(children=(IntSlider(value=50, description='n'), Output()), _dom_classes=('widget-interact',
           <function __main__.n factorial(n)>
            Pentru a evita actualizarea sacadata a valorilor afisate, se prefera inhibaarea feedb
            in <u>Disabling continuous updates</u>
            (https://ipywidgets.readthedocs.io/en/stable/examples/Using%20Interact.html#Disal
            Pentru alte tipuri de controale folosind interact, se poate folosi:
In [44]:
          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
           interactive(children=(Checkbox(value=True, description='x'), FloatSlider(value=5.5, description='y', max
           <function __main__.g(x, y, z, t)>
           Exemplu: Sa se deseneze graficul functiei f: [-10, 20] \rightarrow \mathcal{R}, \ f(x) = a \cdot x^4 + b
            a, b, c, d, e coeficienti reali.
            Rezolvare:
```

```
In [45]:
         # import de pachete numerice si grafice
          import matplotlib.pyplot as plt
         import numpy as np
        execution gueued 18:07:33 2019-02-21
In [46]:
         def f square(a=10, b=20, c=-10,left=-10, right=20):
              assert left < right</pre>
              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\cot x^2 + b\cdot x + c$')
              plt.plot(range_x, values_f, color='red')
              plt.show()
         interact(f_{\text{square}}, a=(-100, 100.0), b=(-100, 100.0), c=(-100, 100.0), d=(-100, 100.0)
        execution queued 18:07:33 2019-02-21
          interactive(children=(FloatSlider(value=10.0, description='a', min=-100.0), FloatSlider(value=20.0, des
           <function __main__.f_square(a=10, b=20, c=-10, left=-10, right=20)>
In [47]:
         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
          interactive(children=(FloatSlider(value=10.0, description='f', min=1.0, step=0.5), Output()), _dom_clas
           <function __main__.sinusoid(f=10)>
```

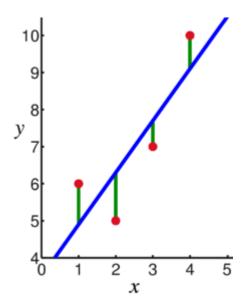
```
In [50]:
         def f(x):
             """calcul functie intr-un punct"""
             return x ** 2 - 10 * x + 50
         def f_values(left=-10, right=10):
             """calcul functie pe interval"""
             x = np.linspace(left, right, 100)
             return x, f(x)
         def f_prime(x):
             """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 = 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
          interactive(children=(IntSlider(value=0, description='x', max=20, min=-20), IntSlider(value=-30, descri
          <function __main__.graf_f_and_derived(x, left=-30, right=30)>
```

In []:



1.5.2 Exercitii:

- 1. Se considera urmatorul enunt: fie functia $f(x) = x^2$ si punctul P = (5,3), sa s minimul functiei sa sa afle in punctul P, fara sa se modifice forma grafica a funcaxelor OX si OY cu, reprezentarea puntului P printr-un dreptunghi, desenarea to 50 de puncte. Se vor determina coeficientii necesari mutarii functiei si se vor de acestea.
- 2. Reprezentati functia $a * x^4 + b * x^3 + c * x^2 + d * x +$ si derivata ei intr-ur
- 3. Incarcati fisierul de date <u>iris (https://archive.ics.uci.edu/ml/machine-learning-dat</u> functie de alegerile exprimate de un utilizator, afisati intr-un grafic bidimensiona (de exemplu, coloana 0 si coloana 2).
- 4. Generati o lista de 20 de perechi de valori $\{x_i, y_i\}_{i=0,19}$, afisati aceste valori per dreapta definita de o functie liniara y=a*x+b Intr-un alt plot afisati, ca hist punct de coordona punct fata de dreapta. Dreapta trebuie sa fie controlabila dir coeficienti. Constatati modificarea histogramei in functie de pozitia dreptei si ca $\sum_{i=0}^{19} (y_i (a \cdot x_i + b))^2$, adica suam patratelor lungimilor segmentelor verzi d





1.6 Ciclari, siruri de caractere

Se recomanda ca urmatoarele exercitii sa le lucrati in Jupyter notebook/lab. Incerca ipywidgets.

1. (fizz-buzz test) Sa se scrie numerele de la 1 la n; pentru fiecare multiplu de 3 se 'Fizz', pentru multiplu al lui 5 se va scrie 'Buzz'; daca numarul este multiplu de 'FizzBuzz'.

- 2. Sa se verifice <u>conjectura Collatz (https://en.wikipedia.org/wiki/Collatz_conjectur</u> 1000.
- 3. * Sa se creeze o functie care preia un numar n si returneaza un alt numar pe ba 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 procesi 15. Exemplu: 20->1, 2, 4, 5, 10, 20 -> suma cifrelor: 15->1, 3, 5, 15-> suma cifr

In []: