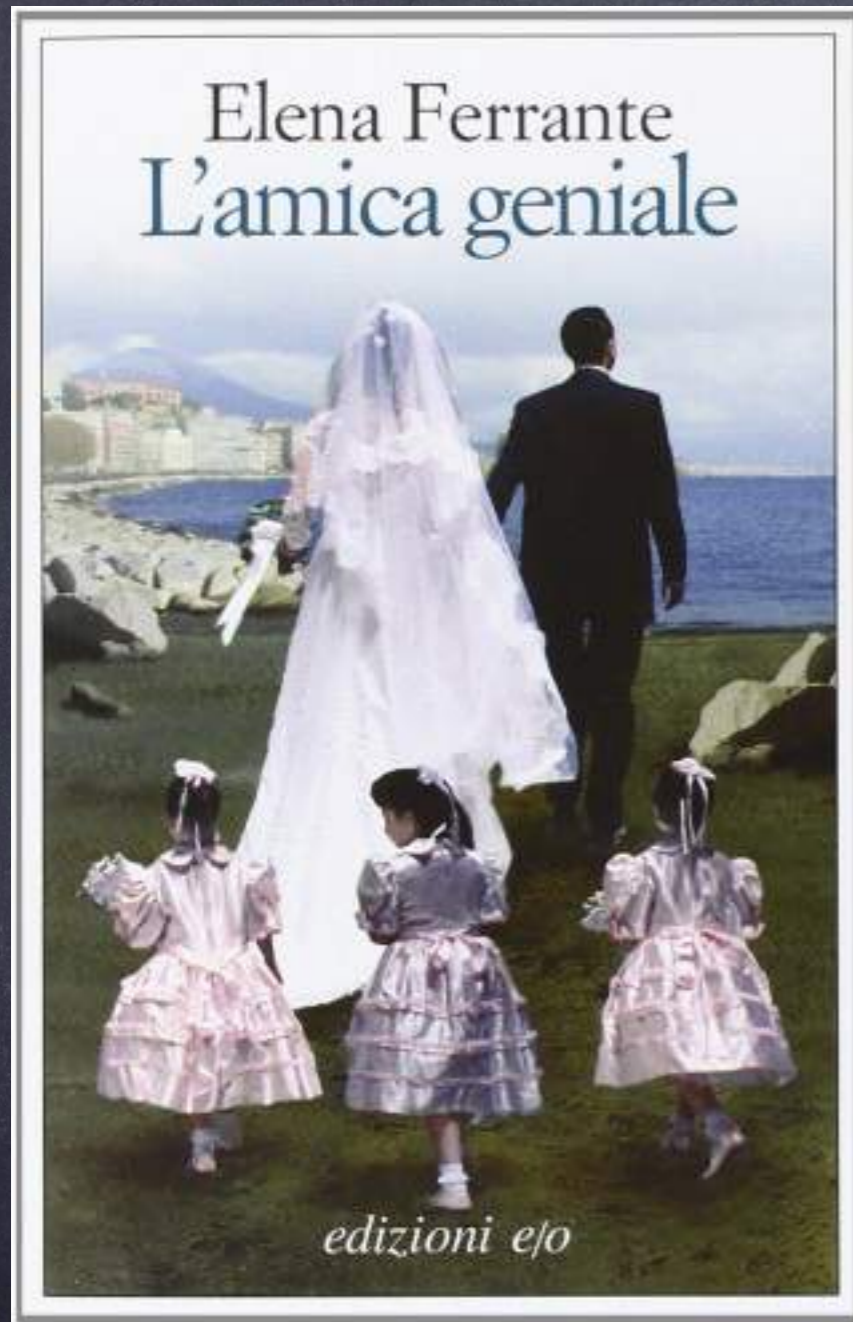


Premessa interdisciplinare



Chi è la
scrittrice (o lo
scrittore) dietro
lo pseudonimo di
Elena Ferrante?

Premessa interdisciplinare



È possibile attribuire a Gramsci alcuni articoli usciti in forma anonima sui quotidiani "Avanti!" e "Grido del Popolo"?

Risposta "Letteraria"

Analisi qualitativa di tipo filologico



Lorenzo Valla
LA FALSA
DONAZIONE
DI COSTANTINO

a cura di Olga Pugliese
TESTO LATINO A FRONTE

BUR

CLASSICI

Un esempio noto:
L'umanista Lorenzo Valla
e la dimostrazione
filologica della falsa
donazione di
Costantino

Risposta "matematica"

Si sono sviluppate delle tecniche per effettuare una analisi quantitativa dei testi individuando delle "misure di similarità" fra testi

Cosa vedremo durante il seminario

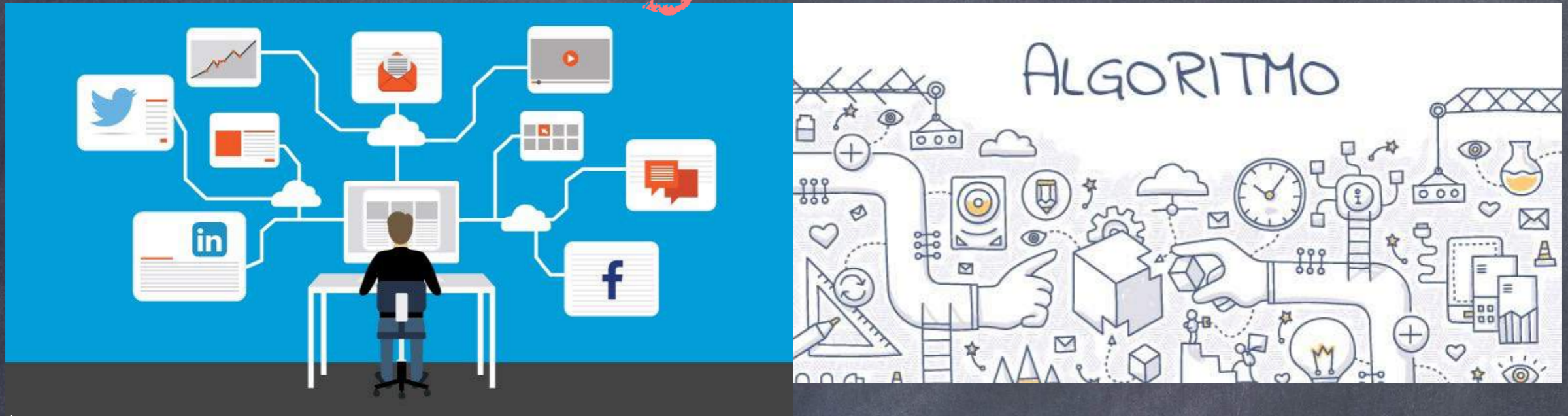
- il coding con il linguaggio Python come strumento per apprendere algoritmi;
- Python è perfetto per fare percorsi interdisciplinari

Esperienze interdisciplinari realizzate



- Matematica e Fisica: analisi con Python dei dati misurati con smartphone
- Matematica e Letteratura: analisi di similarità fra testi letterari

Perché il coding: mondo di algoritmi



- Viviamo in un mondo di algoritmi. Per gli studenti più giovani è una esperienza in cui sono immersi sin da piccoli.
- Si sviluppa un pensiero "magico" secondo cui le cose le fa il cellulare/computer/tablet
- Sono importanti le competenze matematiche di cittadinanza per capire cosa ci sia dietro agli algoritmi in cui si è "immersi".

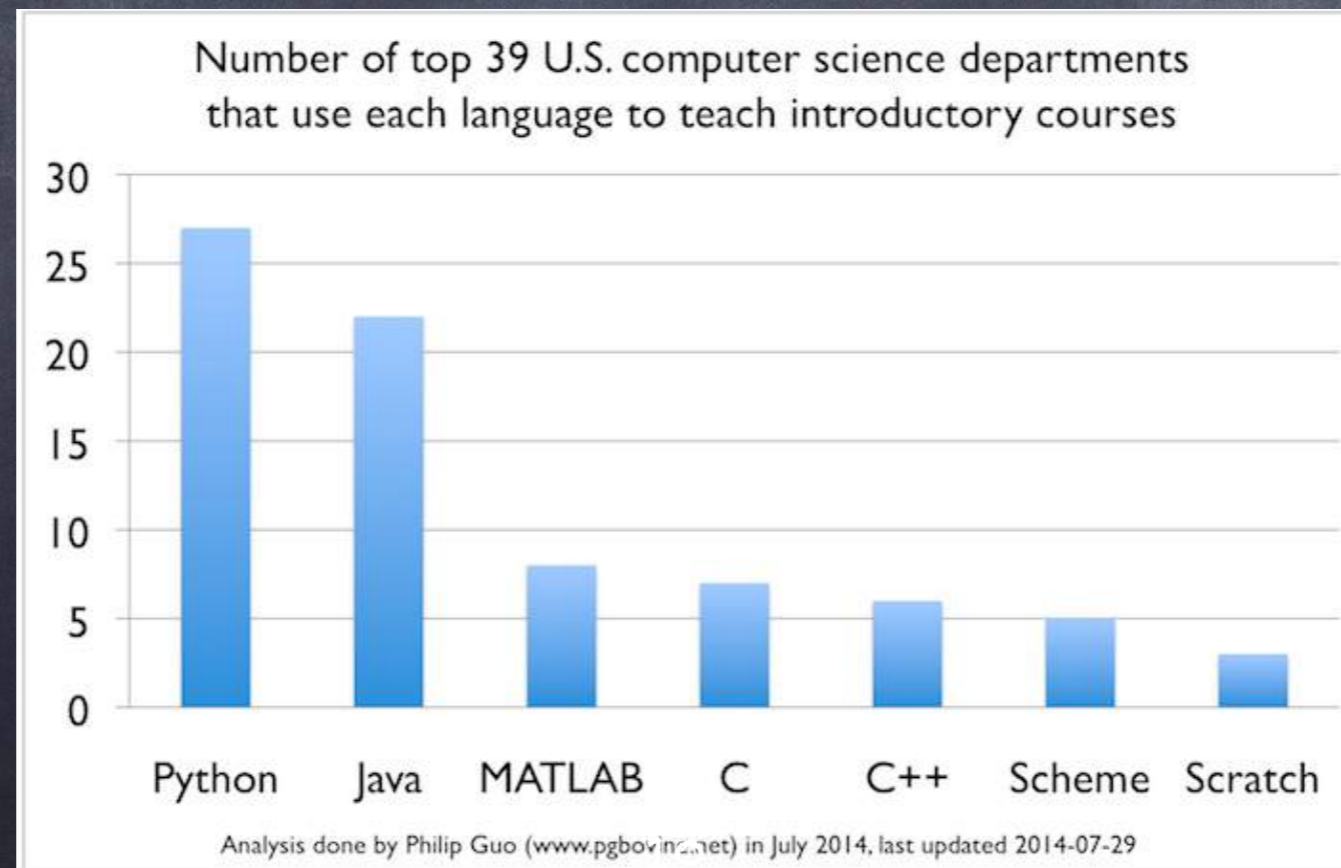
Perché con Python

Il linguaggio Python per la sua semplicità si presta ad essere un linguaggio introduttivo con una curva di apprendimento bassa

- un migliore sviluppo di capacità logiche e di *problem solving* a partire da problematiche concrete;
- una maggiore comprensione di strutture logiche attraverso istruzioni tipiche della programmazione;
- una conoscenza approfondita di cosa sia un algoritmo, concetto altrimenti poco affrontato in concreto al Liceo;
- attenzione al rigore nella sintassi e nel ragionamento a seguito della necessaria precisione derivante dalla scrittura di un programma.

Python

- Linguaggio di programmazione semplice da usare, di nuova creazione
- Con molte librerie utili in molti campi
- **CONTRO:** è un linguaggio interpretato



Semplicità di Python

```
# Primo programma
print("Ciao Mondo!")

# Input dati
dato= input("Dammi un numero")
print(dato)

#tipi dati
a=5
b='ciao'
c=[2,3,4,5]
d=['ciao', 'casa', 'terra']
print(a)
print(b)
print(c)
print(d[0])
```

Primo programma,
input dati e tipi di
dato

Semplicità di Python

```
a=3
b=5
if a<b:
    print('minore')
elif a==b:
    print("uguale")
else:
    print("maggiore")

for i in range(5):
    print(i)
```

Istruzione if, ciclo
for, uso range e
operatori == < >

Semplicità di Python

```
def funzione(a,b): #definisco funzione  
    c=a*b  
    return c
```

```
risultato=funzione(4,5) #chiamo funzione  
print(risultato)
```

Definire e chiamare
una funzione

```
#fattoriale  
def fattoriale(n):  
    .... if n==0:  
    ..... return 1  
    .... else:  
    ..... return n*fattoriale(n-1)
```

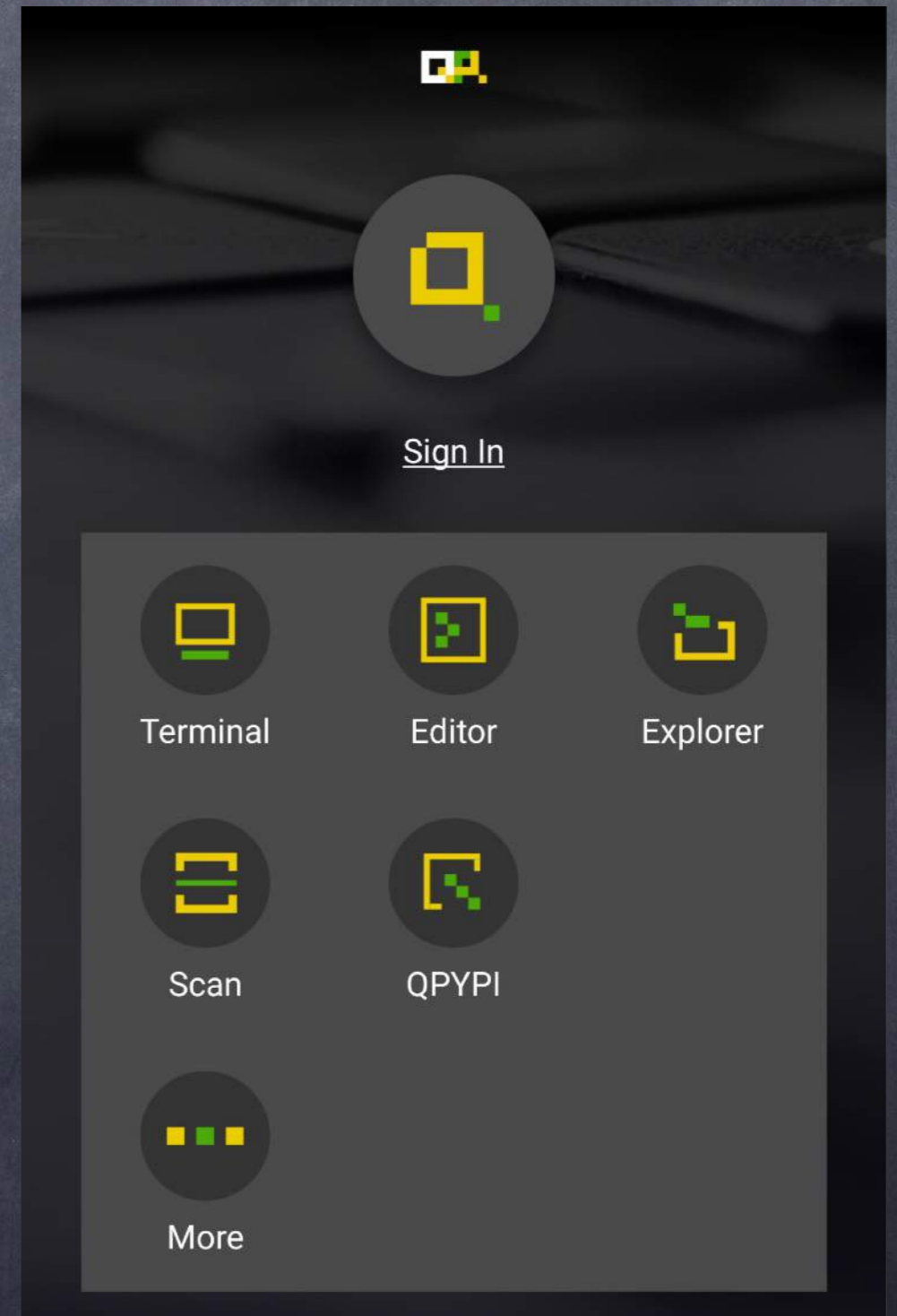
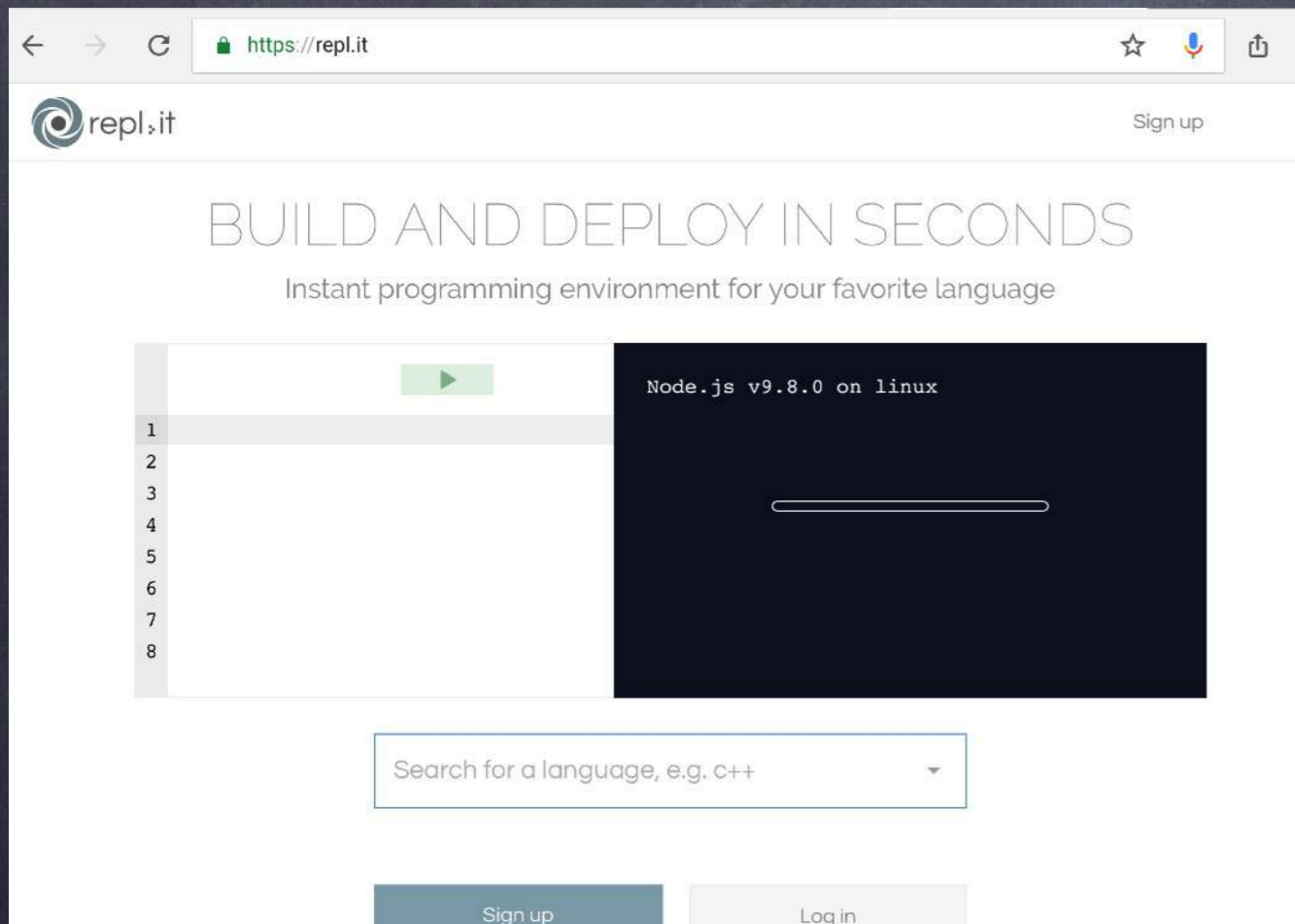
Semplicità di Python

```
import math
radice = math.sqrt(2)
print(radice)
x=math.sin(math.pi/2)
print(x)

import random
dado=random.randint(1,6)
print(dado)
```

Uso semplicissimo
dei moduli di
sistema e aggiuntivi

Python si può usare anche online e da smartphone e tablet



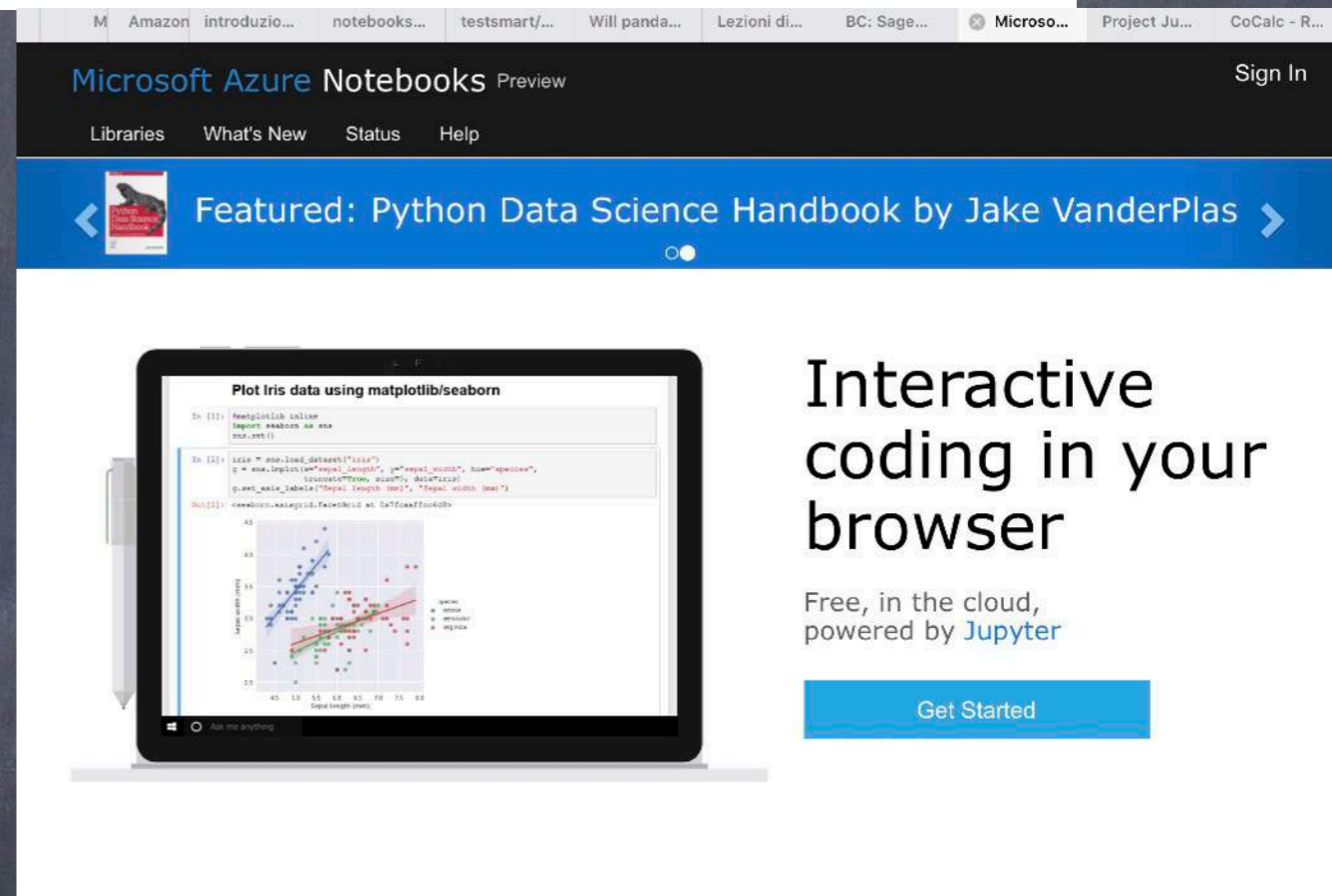
- repl.it
- trinket.io
- <http://www.pythontutor.com>
- Qpython per android

Utilissimi sono i moduli...

```
1 pip install nomepacchetto
```



The screenshot shows the Anaconda website homepage. At the top left is the Anaconda logo. The navigation menu includes 'What is Anaconda?', 'Products', 'Support', 'Resources', 'About', and a 'Downloads' button. The main heading is 'Download Anaconda Distribution', with 'Version 5.2 | Release Date: May 30, 2018' below it. There are icons for Windows, macOS, and Linux. Below this, three columns describe the distribution: 'High-Performance Distribution' (easily install 1,000+ data science packages), 'Package Management' (manage packages, dependencies and environments with conda), and 'Portal to Data Science' (uncover insights in your data and create interactive visualizations).



The screenshot shows the Microsoft Azure Notebooks interface. The top navigation bar includes 'Libraries', 'What's New', 'Status', 'Help', and a 'Sign In' button. A featured banner for 'Python Data Science Handbook by Jake VanderPlas' is visible. The main content area shows a Jupyter Notebook titled 'Plot Iris data using matplotlib/seaborn'. The notebook code includes imports for matplotlib and seaborn, data loading from sklearn, and a plot of the Iris dataset with linear regression lines for each species. To the right of the notebook, the text reads 'Interactive coding in your browser' and 'Free, in the cloud, powered by Jupyter', with a 'Get Started' button.

- numpy e scipy
- matplotlib
- Jupyter e Pandas

Python a scuola

Tantissimi possono essere gli argomenti delle Indicazioni Nazionali che potrebbe giovare dell'utilizzo di Python

LA LEVA DI ARCHIMEDE Matematica e programmazione: usare Python al Liceo di Davide Passaro

L'obiettivo di quest'articolo è quello di fornire alcuni spunti per realizzare un semplice progetto di *programmazione* con il linguaggio *Python* all'interno dell'insegnamento della matematica al Liceo. Da più parti, infatti, si sta diffondendo l'esigenza di introdurre la programmazione a tutti i livelli di scuola. In quest'ultimo periodo, inoltre, sembra che si stiano superando alcune obiezioni all'insegnamento della programmazione nelle scuole. In particolare è sempre più diffusa l'idea che:

- la conoscenza di un linguaggio di programmazione non sia una conoscenza di tipo «tecnico» destinata a pochi che la apprenderanno, se necessario, a livello universitario;
- il computer non sia un semplice strumento per tutte le discipline; quello che serve, per intenderci, ad uno studente è utilizzare pacchetti software come Office, saper navigare in Internet, utilizzare semplici programmi di grafica.

L'affermazione dell'*informatica* e l'utilizzo sempre più diffuso delle potenzialità offerte da essa nelle discipline scientifiche, conferma l'importanza culturale che queste conoscenze hanno per uno studente di scuola superiore e in particolare del Liceo. La questione è ovviamente complessa e dibattuta e l'impressione è che in Italia arrivino solo gli echi di queste riflessioni. Ultimamente, però, si sta facendo strada l'idea che un linguaggio di programmazione possa supportare [1]:

- un migliore sviluppo di capacità logiche e di *problem solving* a partire da problematiche concrete;
- una maggiore comprensione di strutture logiche attraverso istruzioni tipiche della programmazione;
- una conoscenza approfondita di cosa sia un algoritmo, concetto altrimenti poco affrontato in concreto al Liceo;
- attenzione al rigore nella sintassi e nel ragionamento a seguito della necessaria precisione derivante dalla scrittura di un programma.

A seguito della Riforma Gelmini, che ha eliminato le sperimentazioni (si pensi, per esempio, al Piano Nazionale Informatica), l'insegnamento dell'informatica nei Licei è una disciplina a sé solo nel Liceo delle Scienze Applicate, mentre nel Liceo Scientifico, che è attualmente la scelta più diffusa fra gli studenti in Italia, è, in base alle Indicazioni Nazionali, parte integrante dell'insegnamento della matematica. Nelle Indicazioni Nazionali è scritto, infatti: «lo studente diverrà familiare con gli strumenti informatici, al fine precipuo di rappresentare e manipolare oggetti mate-

- algoritmi (trova max, min, algoritmo Euclide)
- La logica AND OR
- Grafici
- Statistica
- ecc..

Alcuni esempi

```
numero = int(input("Dammi un numero intero: "))
```

```
if (numero%2)==0:
```

```
    print("Il numero dato è pari")
```

```
else:
```

```
    print("Il numero dato è dispari")
```

Algoritmo di Euclide per MCD

```
# algoritmo non ricorsivo
```

```
def mcd(a, b):  
    while(b != 0):  
        r=a%b  
        a=b  
        b=r  
    return a
```

```
# algoritmo ricorsivo
```

```
def mcd_ricorsivo(a,b):  
    if b==0:  
        return a  
    else:  
        return mcd(b, (a%b))
```

Stima radice quadrata

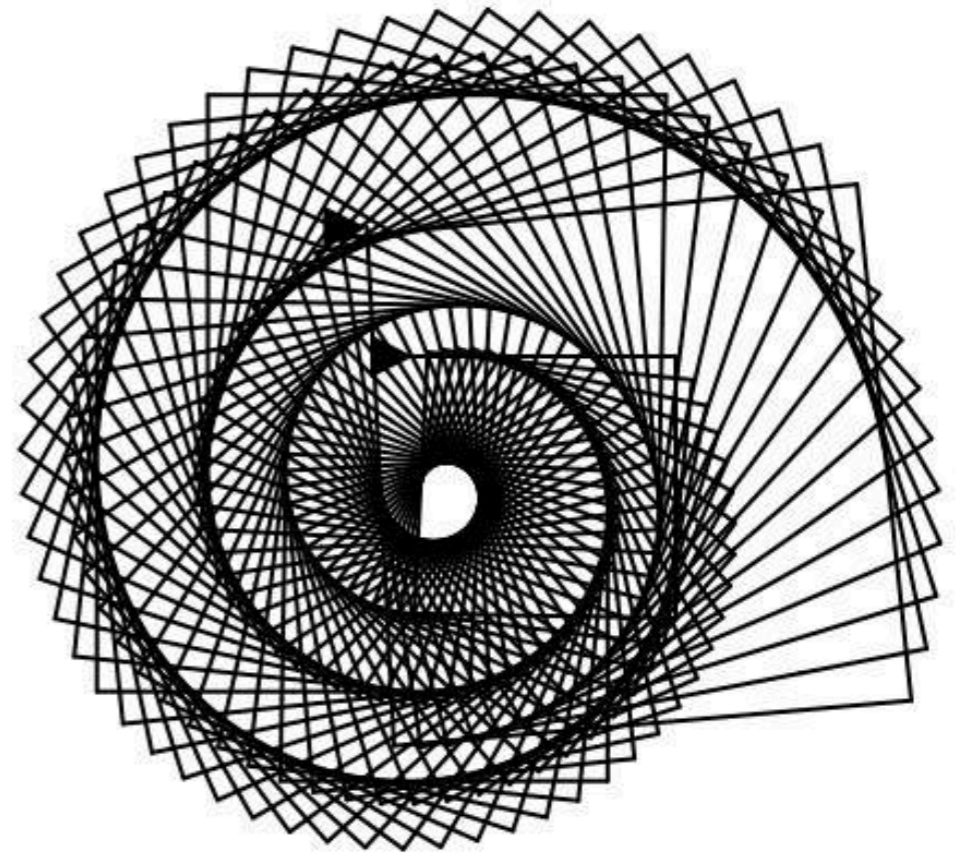
```
def radicequadrata(n, n_iterazioni):  
    approssimazione = 0.5 * n  
    for i in range(n_iterazioni):  
        nuova_approssimazione = 0.5 * (approssimazione + n/  
            approssimazione)  
        approssimazione = nuova_approssimazione  
    return nuova_approssimazione
```

Alcuni esempi

```
import turtle

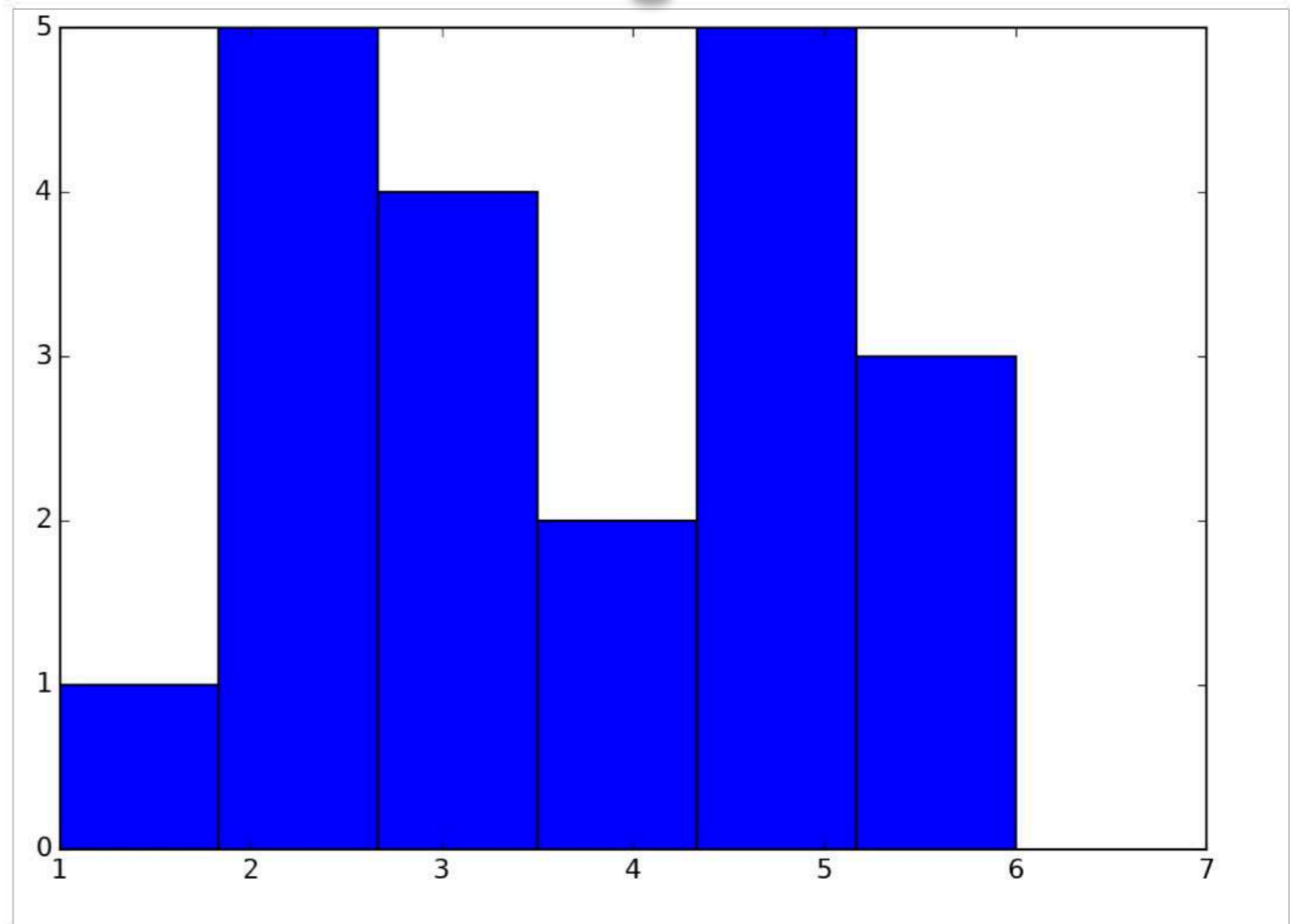
tarta = turtle.Turtle()

for j in range(60):
    for i in range(4):
        tarta.forward(60+j)
        tarta.right(90+i)
    ...
turtle.done()
```



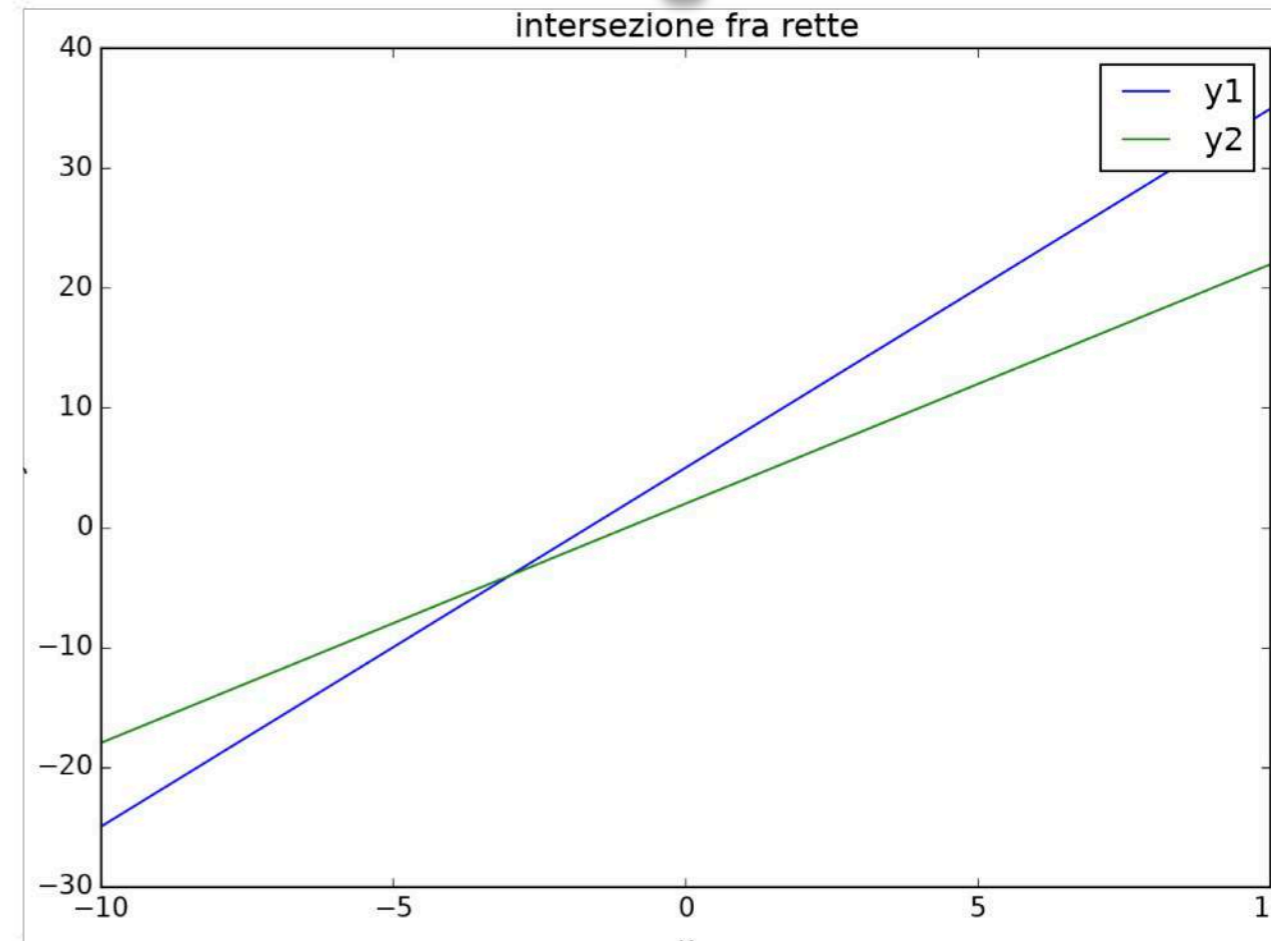
Alcuni esempi

```
istogrammi1.py ▾  
1 import matplotlib.pyplot as plt  
2  
3 a=[2,3,2,2,2,3,4,5,5,5,6,1,3,2,6,6,3,4,5,5]  
4 #plt.hist(a,[1,2,2,3,4,5,6,7])  
5 #plt.show()  
6 plt.hist(a,6)  
7 plt.show()
```



Alcuni esempi

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x= np.linspace(-10,10,20)
4 y1=3*x+5
5 y2=2*x+2
6 fig=plt.figure()
7 plt.plot(x,y1)
8 plt.plot(x,y2)
9
10 #plt.fill_between(x,y1,y2)
11 plt.title("intersezione fra rette")
12 plt.xlabel("x")
13 plt.ylabel("y")
14 plt.legend(('y1', 'y2'))
15 plt.show()
```



```
: # importiamo pyplot "sotto-modulo" di "matplotlib chiamandolo plt
import matplotlib.pyplot as plt

# importiamo la libreria numeric python che serve a creare più facilmente
sequenze di numeri
import numpy as np

x = np.linspace(0, 5, 10) # creiamo i valori x del grafico fra 0 e 5
y = x ** 2 # creiamo i valori delle y
plt.figure()
plt.plot(x, y, 'r')
plt.xlabel('x')
plt.ylabel('y')
plt.title('grafico $y=x^2$')
plt.show()

# oppure si possono fare subplot
plt.figure()
plt.subplot(1,2,1)
plt.title('grafico $y=x^2$')
plt.plot(x, y, 'r--')
plt.subplot(1,2,2)
plt.title('$x=\sqrt{y}$')
plt.plot(y, x, 'g*-')
plt.show()
```

grafico $y = x^2$

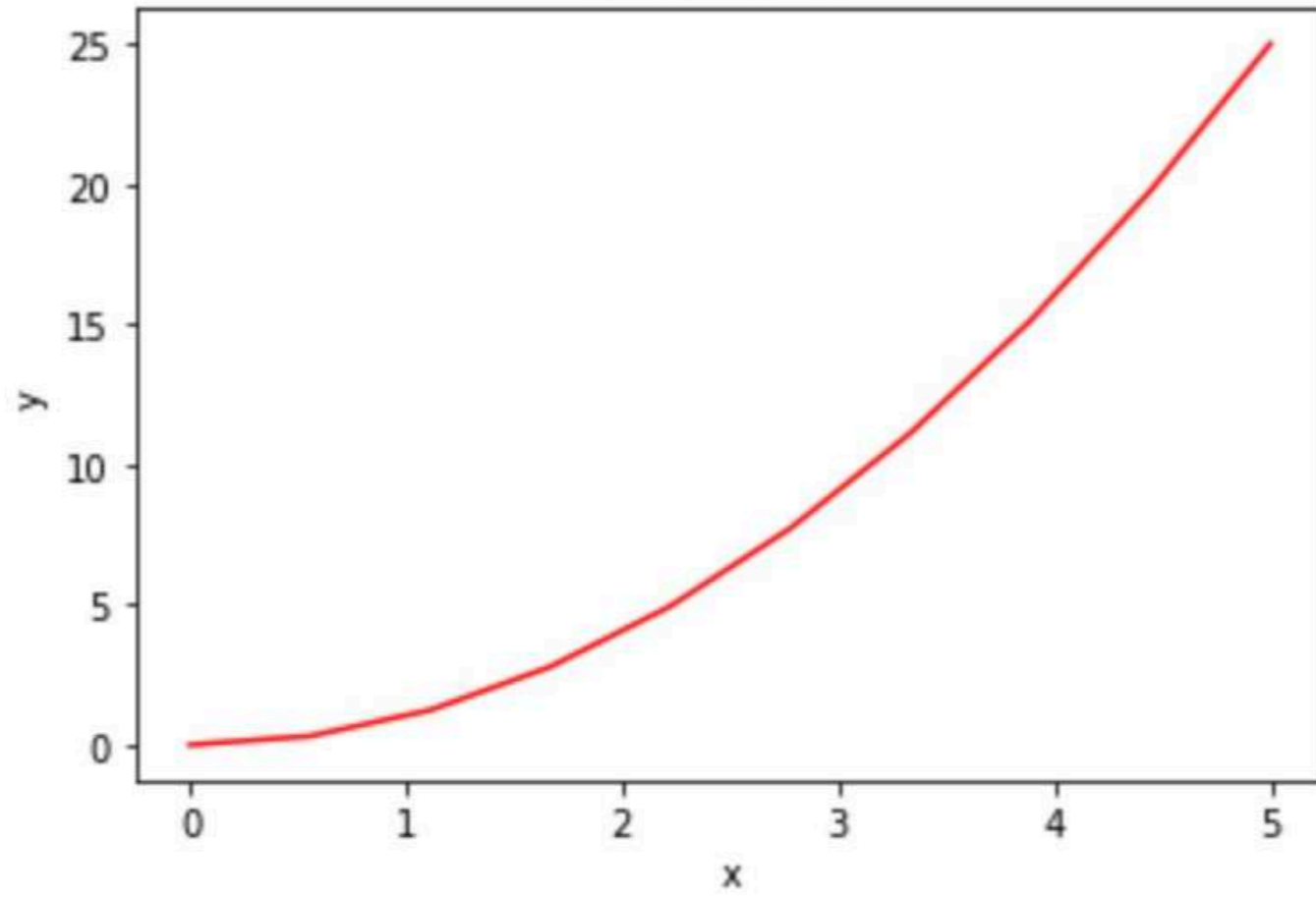
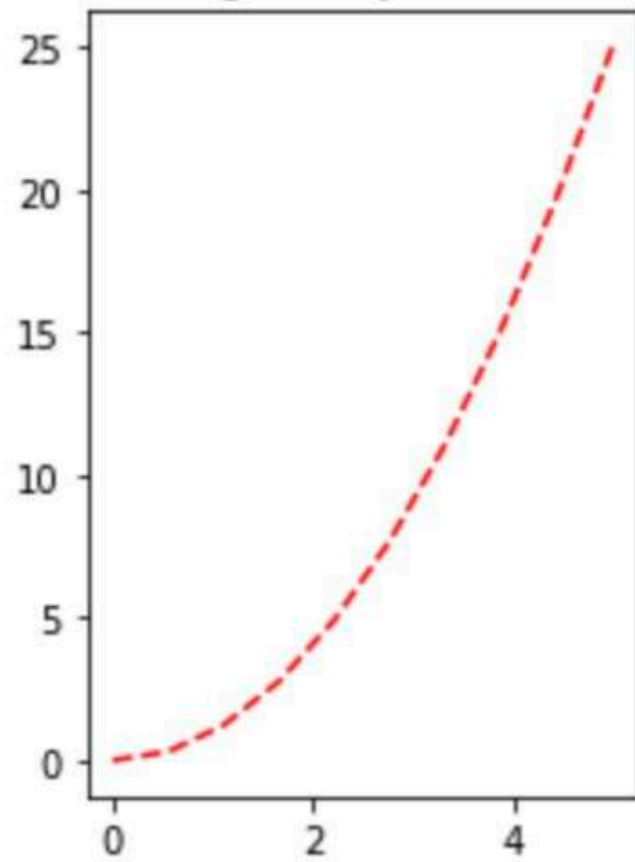
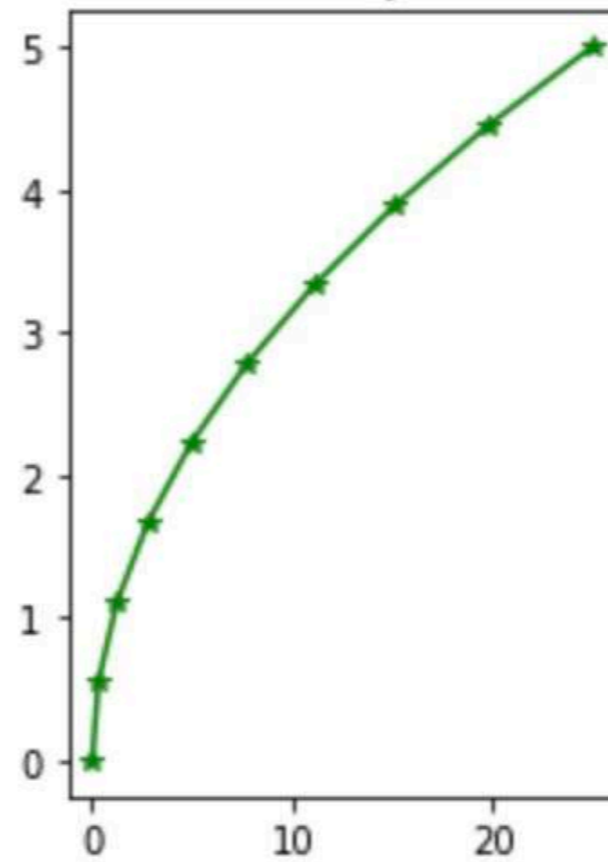


grafico $y = x^2$



$x = \sqrt{y}$



Python e Notebook Jupyter

Strumento ottimo per inserire nello stesso luogo: testo, formule con LaTeX, immagini, codice, tabelle.

Grafici interattivi con slider

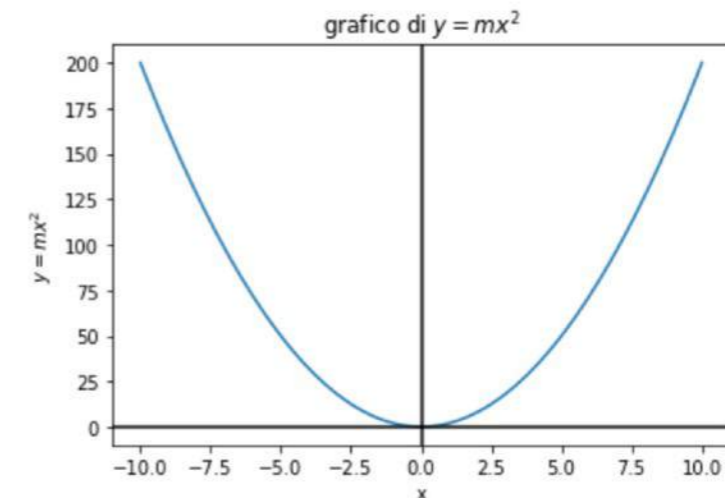
In questa parte si mostra come inserire uno slider nei grafici in modo interattivo

N.B. la parte interattiva è visibile solo accedendo all'account di azure

Iniziamo da come fare un grafico

```
In [3]: from matplotlib import pyplot as plt # modulo per far
import numpy as np # modulo numerical python che ser
locemente vettori di x
x= np.linspace(-10,10,100) # creo vettore da -10 a 1
ri
plt.plot(x,2*x*x) # faccio il grafico della funzione
plt.axhline(0, color= 'k')# retta asse x
plt.axvline(0, color= 'k') # retta asse y
plt.xlabel('x') #asse x
plt.ylabel('$y=mx^2$') #asse y
plt.title('grafico di $y=mx^2$') # titolo grafico
```

Out[3]: Text(0.5,1,u'grafico di $y=mx^2$ ')



Primo percorso interdisciplinare



Matematica e fisica: Python come strumento
per analizzare i dati acquisiti con uno
smartphone

Lo smartphone: un laboratorio tascabile



Esistono diverse App gratuite in grado di salvare i dati registrati dai sensori del cellulare (accelerometri, sensori di luminosità e di prossimità...)

L'uso di queste App permette di realizzare diversi esperimenti sia a scuola che a casa anche in assenza di un laboratorio attrezzato

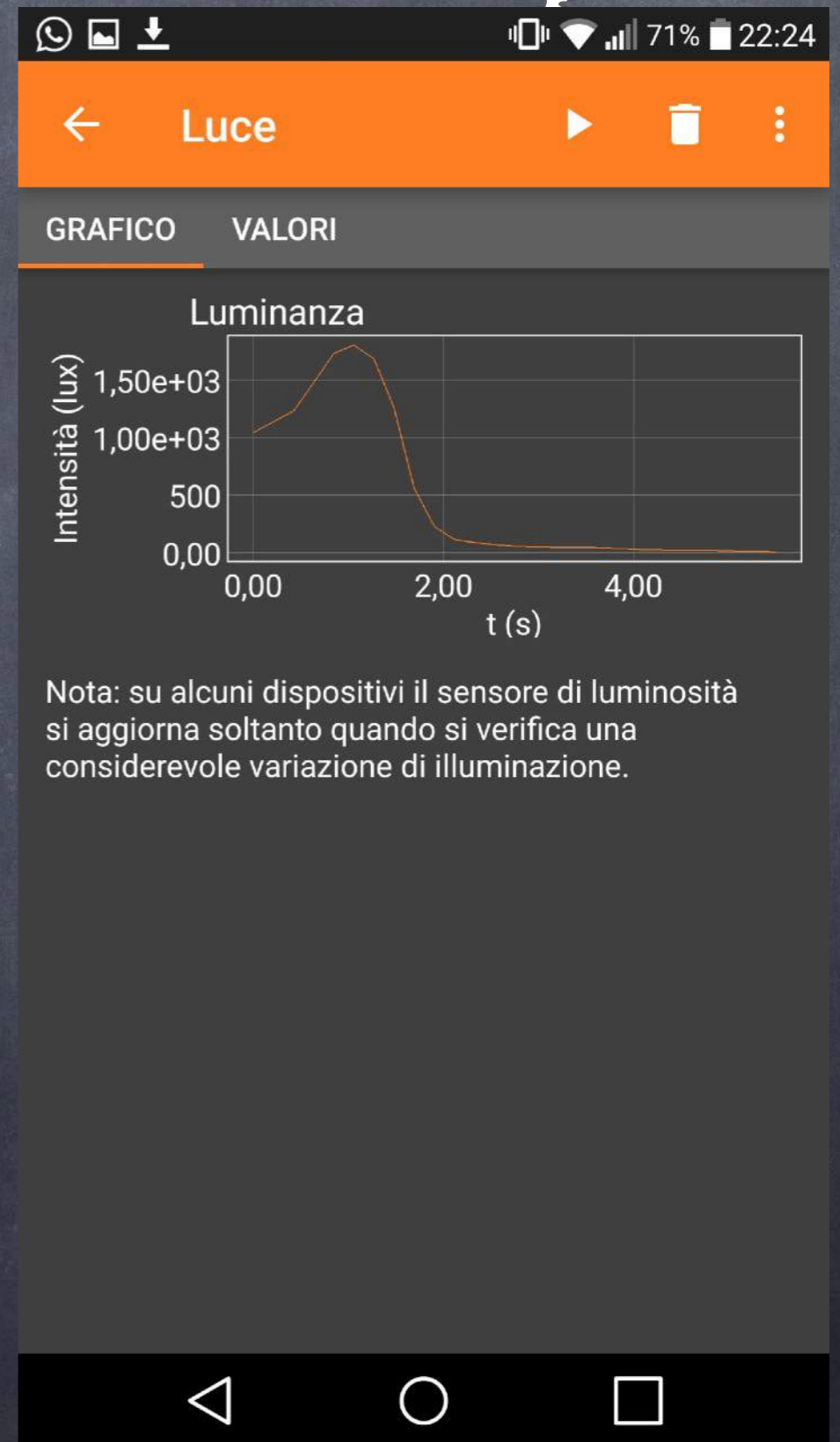
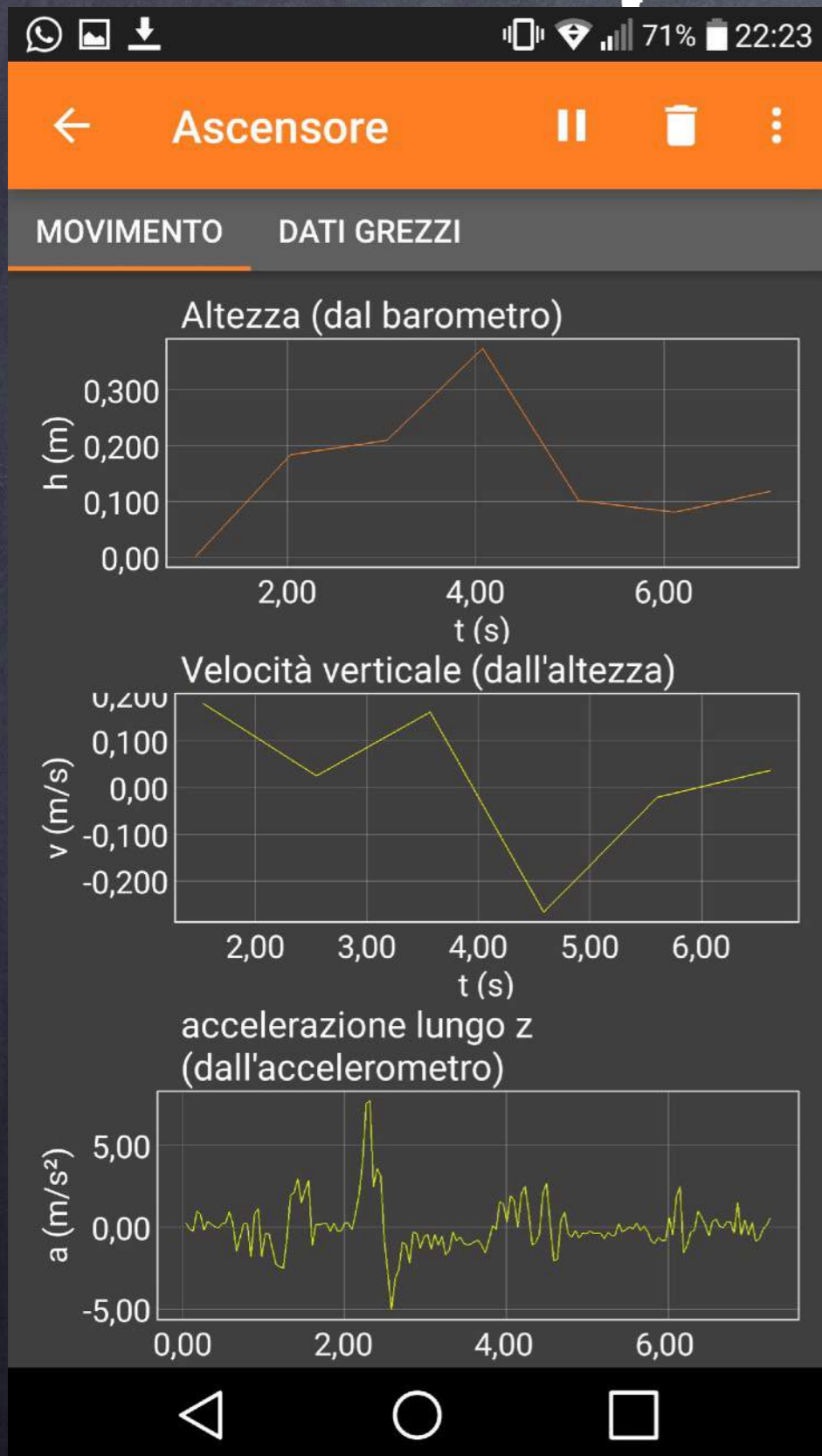
In particolare: Phypbox



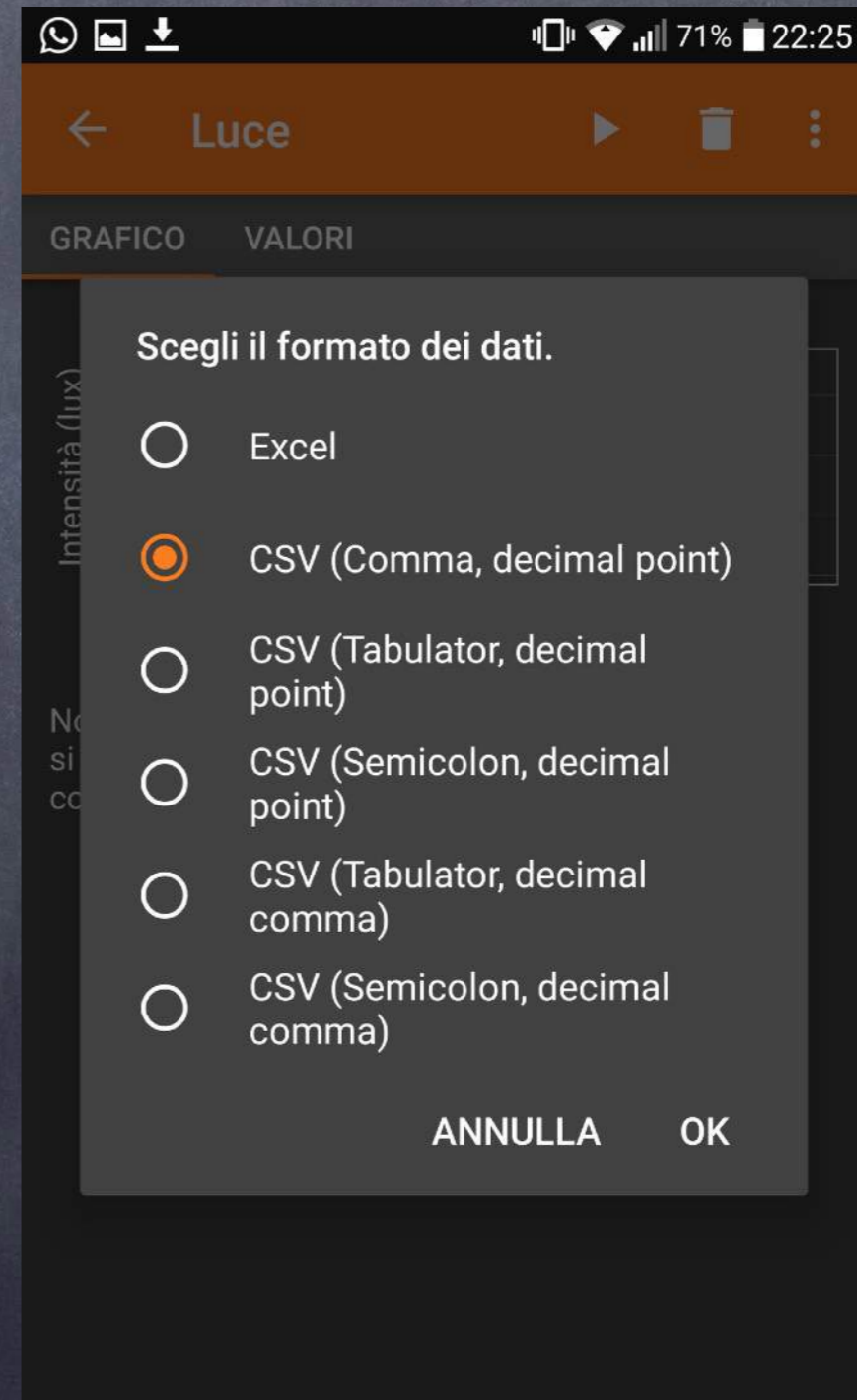
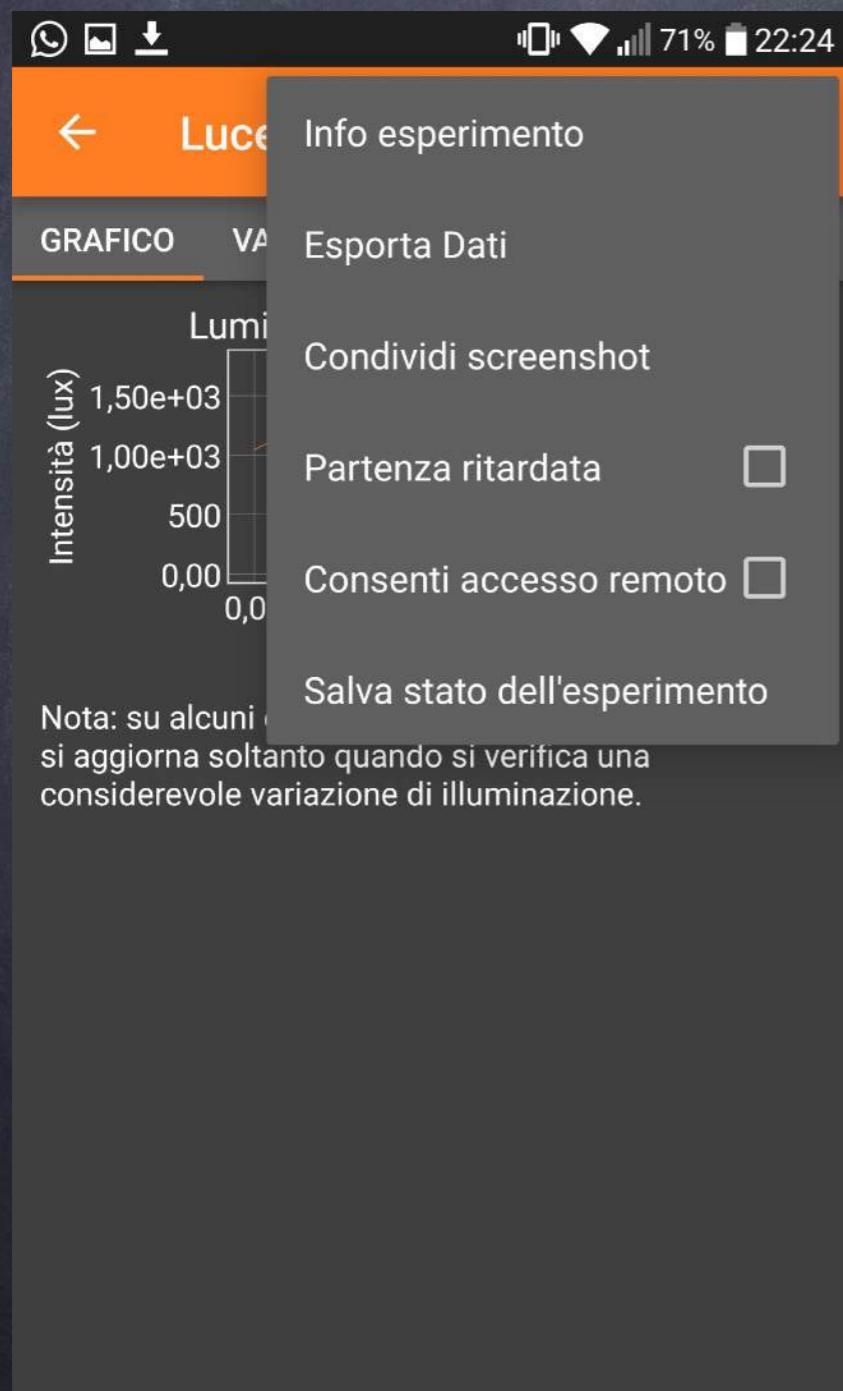
Permette di accedere e registrare i valori di tutti i sensori presenti nel cellulare

Presenta anche degli esempi di esperimenti suggeriti con Tutorial su youtube

Due possibili esempi



Come inviare i dati registrati in un file csv



Importare i dati CSV

In questo notebook mostreremo come importare i dati presenti da un file csv e come farne i grafici. Mostreremo inoltre come applicare una semplice ed iniziale analisi dei dati, calcolando media, deviazione standard e realizzando dei "fit"

```
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
dati=pd.read_csv("Amplitudes.csv")
print dati
tempo =dati["Time (s)"]
pressione= dati["Sound pressure level (dB)"]
plt.plot(tempo,pressione,'o')
plt.xlabel('Tempo(s)')

tempo.mean()
print tempo.std()
```

Time (s) Sound pressure level (dB)

Esempi esperimenti

Grafici

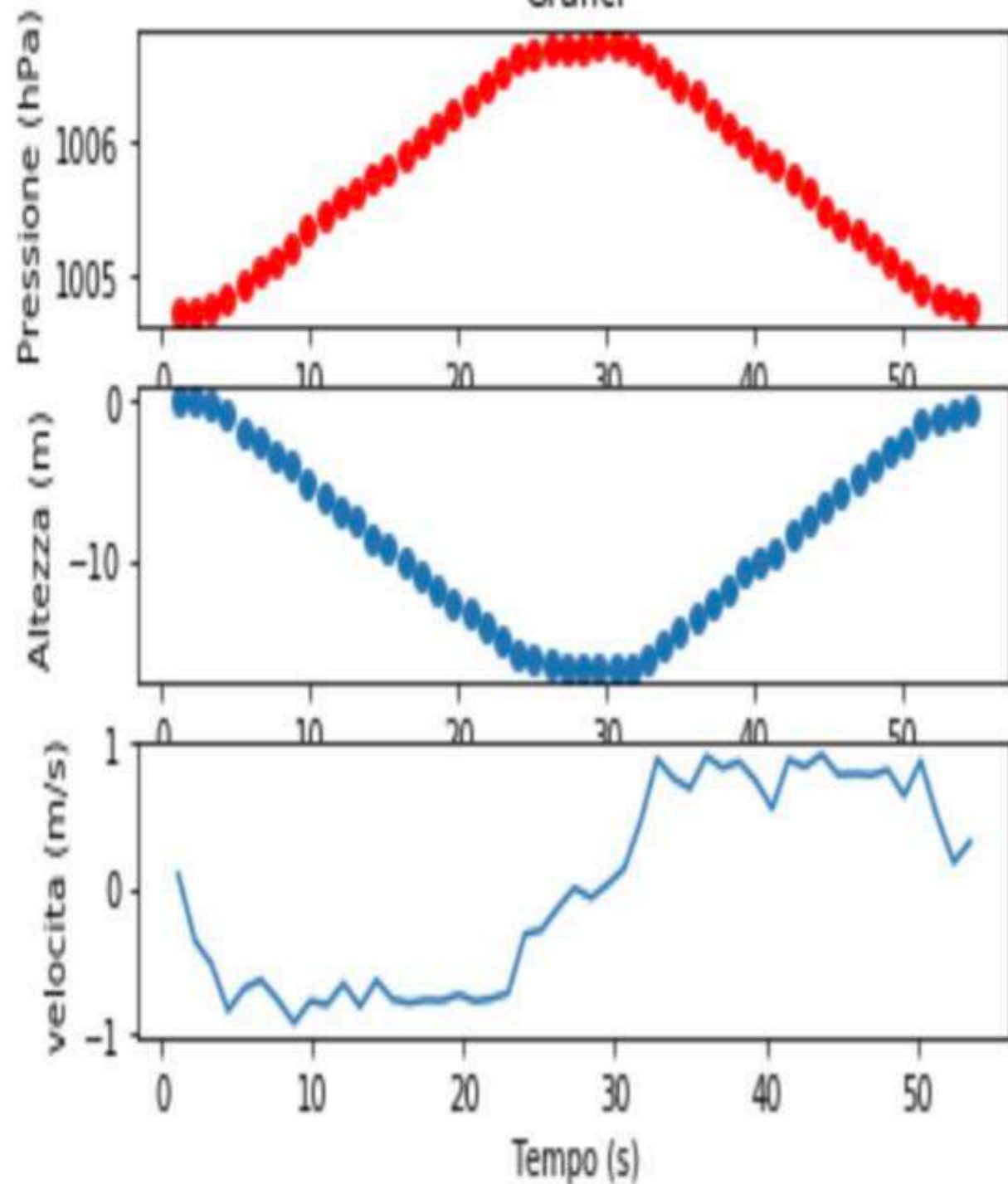


Fig. 1: Esperimento di uno studente che scende in ascensore e poi sale.

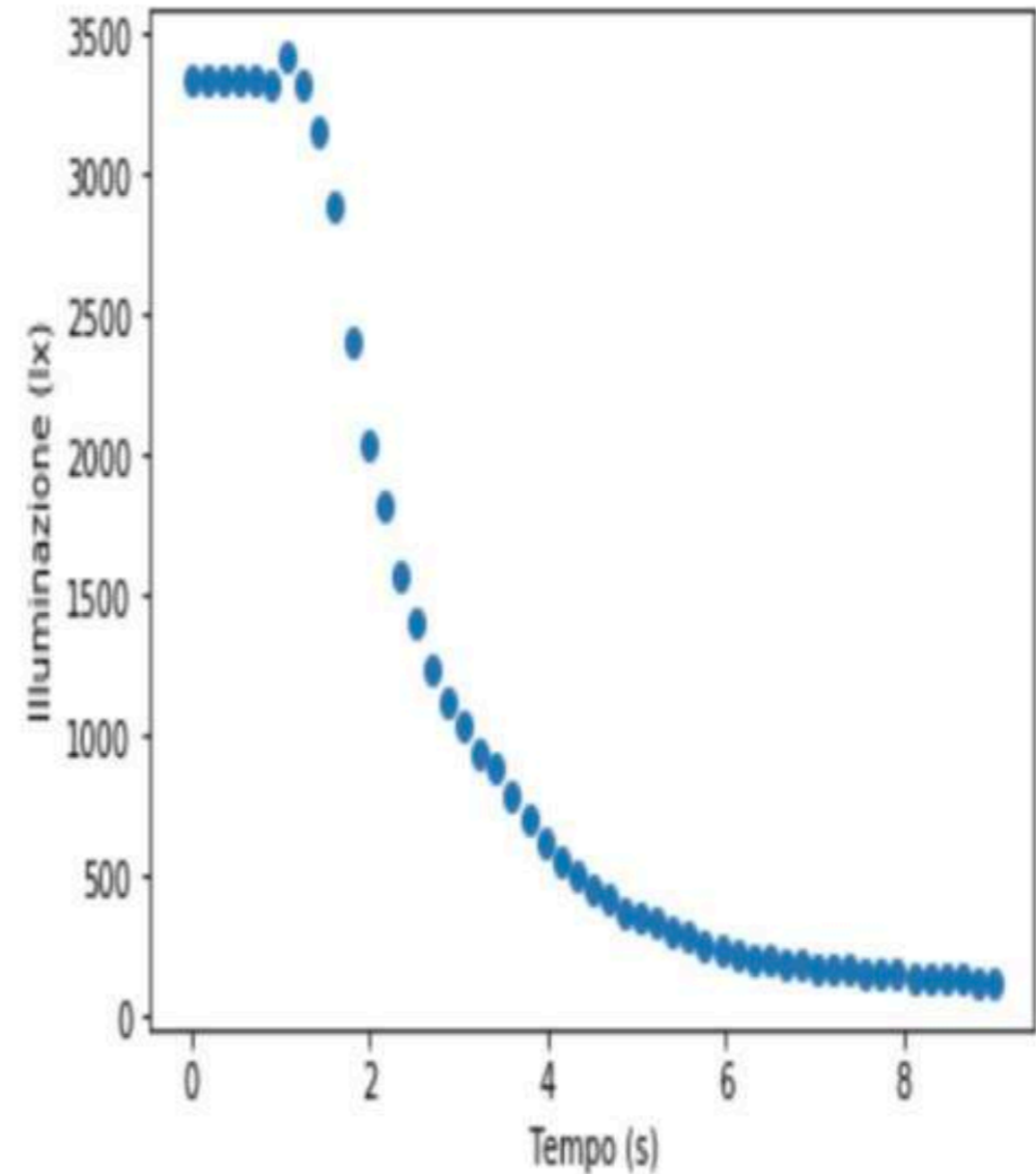


Fig.2: Esperimento ottenuto allontanando una sorgente luminosa dal sensore di luce dello smartphone

Esempi esperimenti

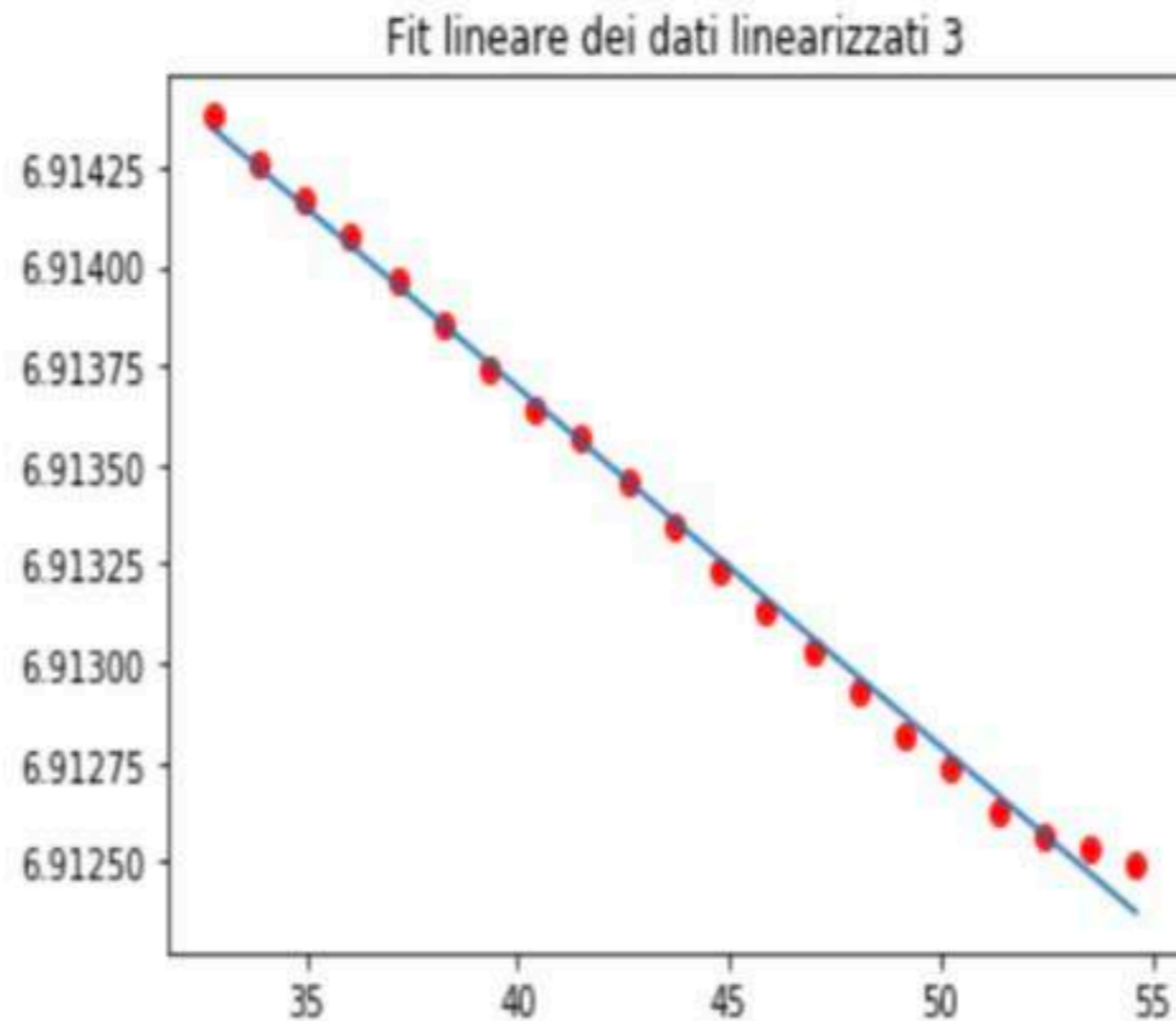


Fig. 3: Fit lineare della parte di discesa dell'ascensore

Secondo percorso interdisciplinare



Matematica e Letteratura: testi analizzati
usando Python

Premessa "storica"

- Percorso di matematica e letteratura con collega lettere
- Si sono sviluppate alcune misure di similarità fra testi: jaccard, cosine similarity, distanza euclidea.
- Idea che un testo può diventare un vettore di parole nello spazio n dimensionale

Analisi testuale

- Percorso di matematica e letteratura con collega lettere
- Si sono sviluppate alcune misure di similarità fra testi: jaccard, cosine similarity, distanza euclidea.
- Idea che un testo può diventare un vettore di parole nello spazio n dimensionale

Borsa delle parole

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Borsa delle parole: esempio

Se per esempio si dovessero confrontare due testi come: “Andrea mangia il gelato ” e “Il cane gioca con il sasso” si potrebbero per prima cosa trasformare le frasi in sequenza di parole e considerare la frequenza di ognuna ottenendo:

$Testo1 = \{“Andrea”: 1, “mangia”:1, “il”:1, “gelato”:1\}$

$Testo2 = \{“il”: 2, “cane”:1, “gioca”:1, “con”:1, “sasso”:1 \}$

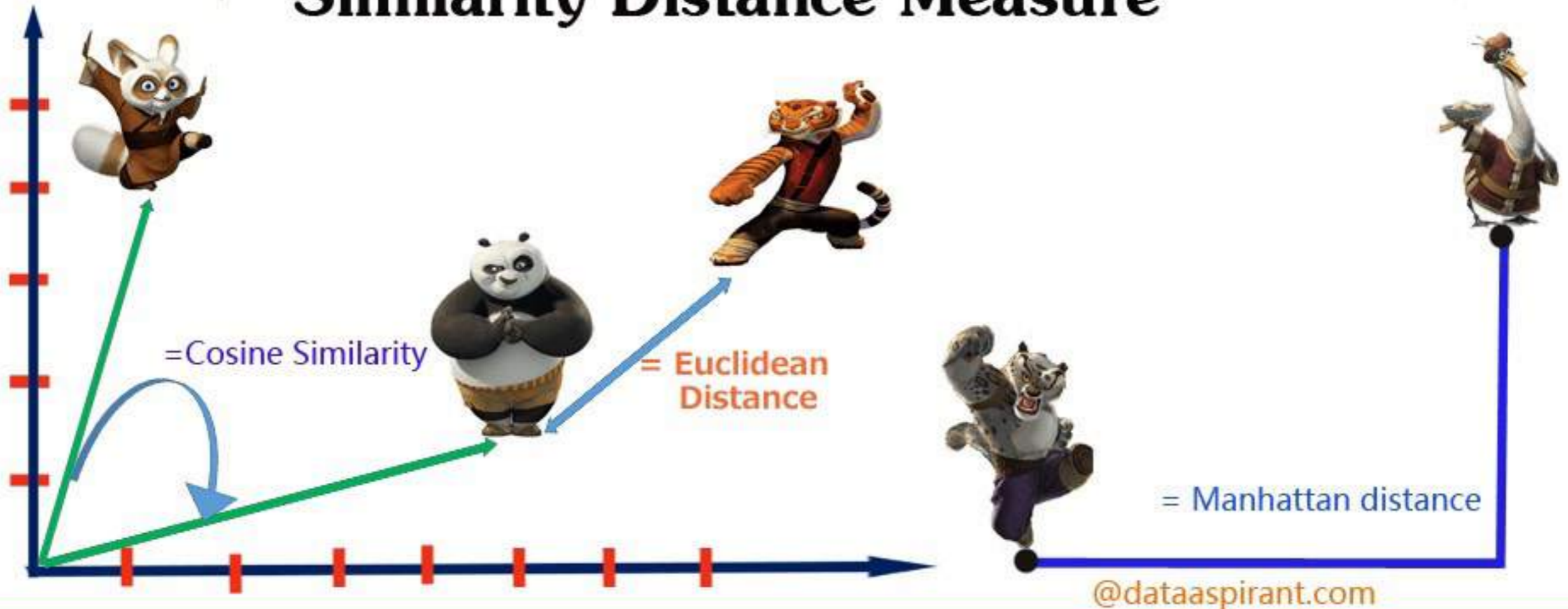
In seguito, si conta il numero n di parole diverse presenti in entrambi le frasi seguendo un ordine deciso arbitrariamente e si rappresentano i testi come vettori in uno spazio a n dimensioni riportando ordinatamente la frequenza di ciascuna parola. Nel nostro esempio, il numero n è otto; scegliendo l'ordine: [Andrea, mangia, il, gelato, cane, gioca, con, sasso], si ha:

$Testo1 = [1,1,1,1,0,0,0,0]$

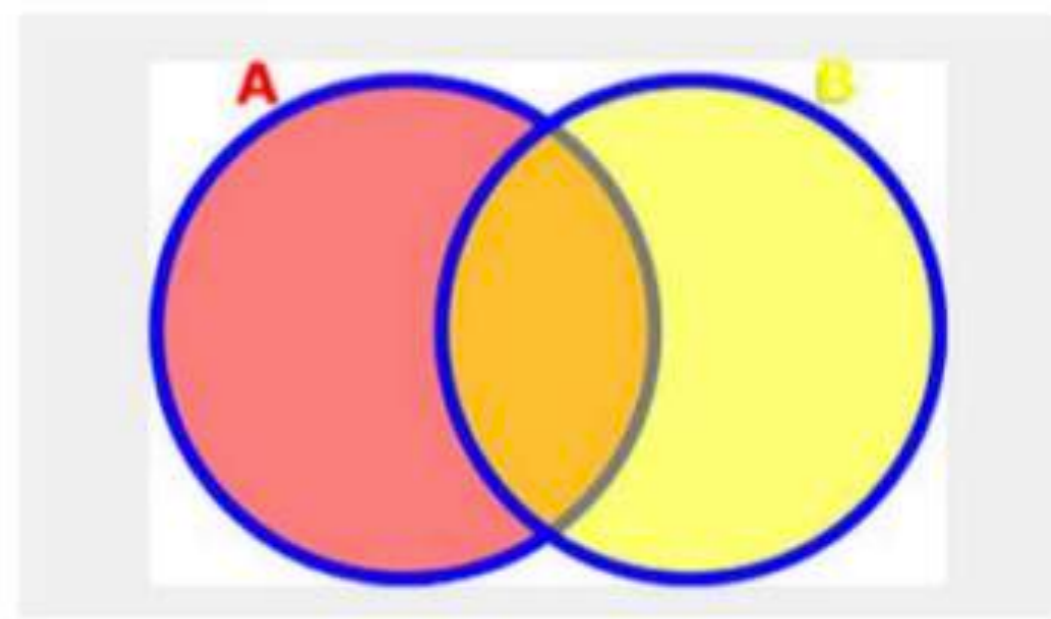
$Testo2 = [0,0,2,0,1,1,1,1]$

Misure di similarità

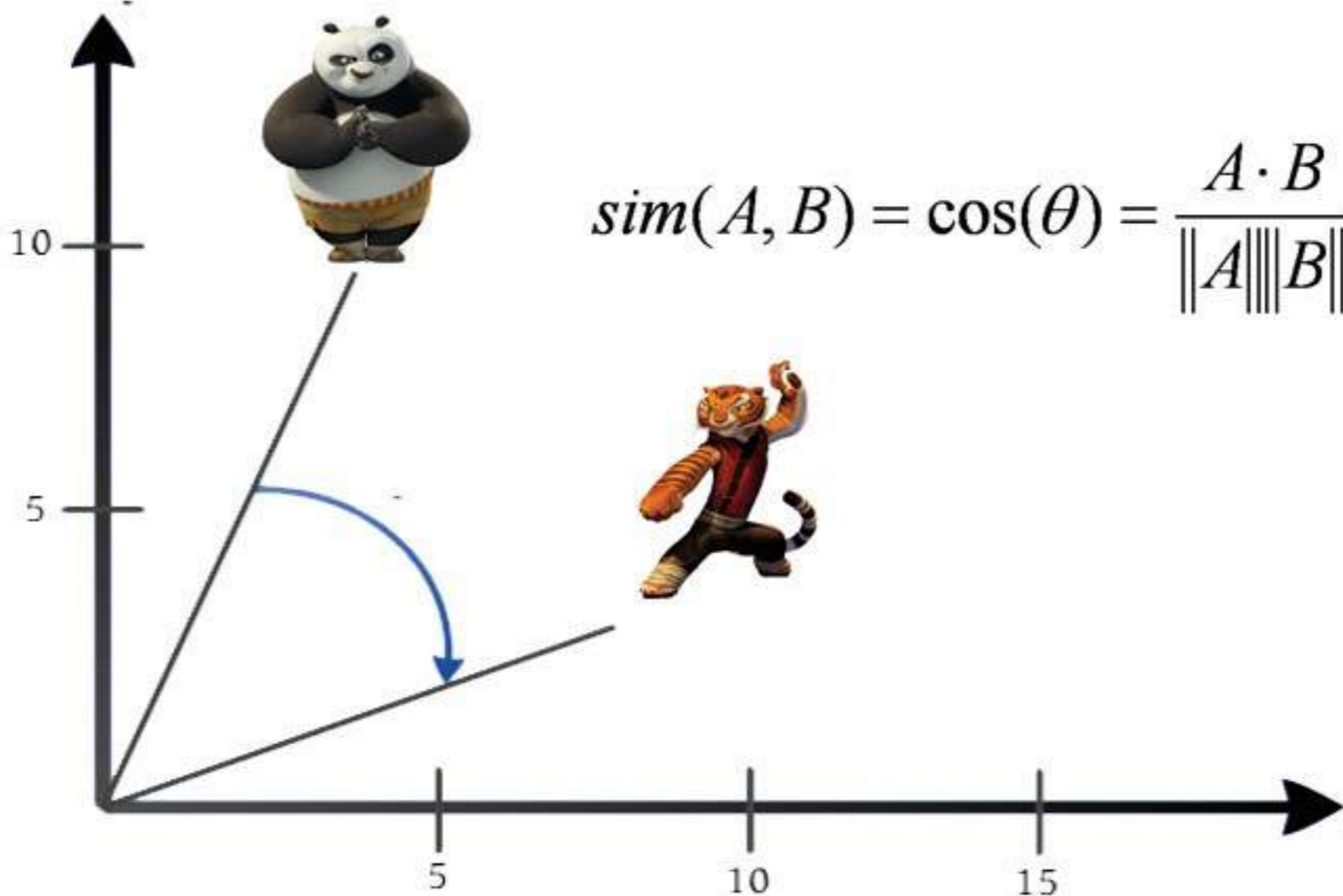
Similarity Distance Measure



$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



Cosine Similarity



Descrizione procedimento

- acquisizione di un dataset e sua normalizzazione: questa prima fase consiste nell'acquisizione di un corpus di testi (per esempio dal web) eliminando tutto il campo non testuale (immagini, didascalie, menù di navigazione, ecc...);
- tokenizzazione: a partire dal testo preprocessato nella precedente fase, si procede suddividendolo in sequenze di caratteri di unità minime di analisi, dette token. I token possono essere di diverso tipo e comprendono svariate categorie di parti del testo (parole, punteggiatura, numeri, date) la cui individuazione può essere più o meno semplice;
- analisi: questa fase consiste nell'applicazione di algoritmi di tipo statistico e di apprendimento automatico; in questa fase, è talvolta utile rielaborare e modificare i token individuati in precedenza, o con l'obiettivo, per esempio, di eliminare le parole più comuni di una lingua (dette "stop words" che non sono caratterizzanti un testo) ed eventualmente

individuando i vari lemmi (per esempio riconoscendo "tavoli" e "tavolo" come un'unica parola) o considerando l'importanza dei termini rispetto ad una più ampia collezione di documenti utilizzando una funzione di peso (metodo in genere indicato con il termine tf-idf ovvero term frequency-inverse document frequency).

Confronto I Malavoglia, Promessi Sposi e Pinocchio

In questo programma confronto il testo dei Promessi Sposi di Alessandro Manzoni con il libro 'I Malavoglia' di Giovanni Verga e 'Pinocchio' di Carlo Collodi e fra questi ultimi due. Effettuando tale analisi si nota che i testi più simili sono I Malavoglia e Pinocchio.

```
def Similarita(primo_file_da_confrontare, secondo_file_da_confrontare):  
    def get_jaccard_sim(str1, str2): # fonte: https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50  
        a = set(stringa1.split())  
        b = set(stringa2.split())  
        c = a.intersection(b)  
        return float(len(c) / (len(a) + len(b) - len(c))) # trovo  
un indice che mi dice quanto sono simili tra di loro i due insiemi  
creati precedentemente facendo l'intersezione dei due insiemi divi  
so l'unione meno l'intersezione  
  
ps = open(primo_file_da_confrontare) #apro il file contenente
```

```
Similarità testi i_promessi sposi.txt e I Malavoglia - Giovanni V  
erga.txt : 17.8675940455 %  
Similarità testi i_promessi sposi.txt e pinocchio.txt : 15.101933  
6775 %  
Similarità testi pinocchio.txt e I Malavoglia - Giovanni Verga.tx  
t : 22.8322937714 %
```

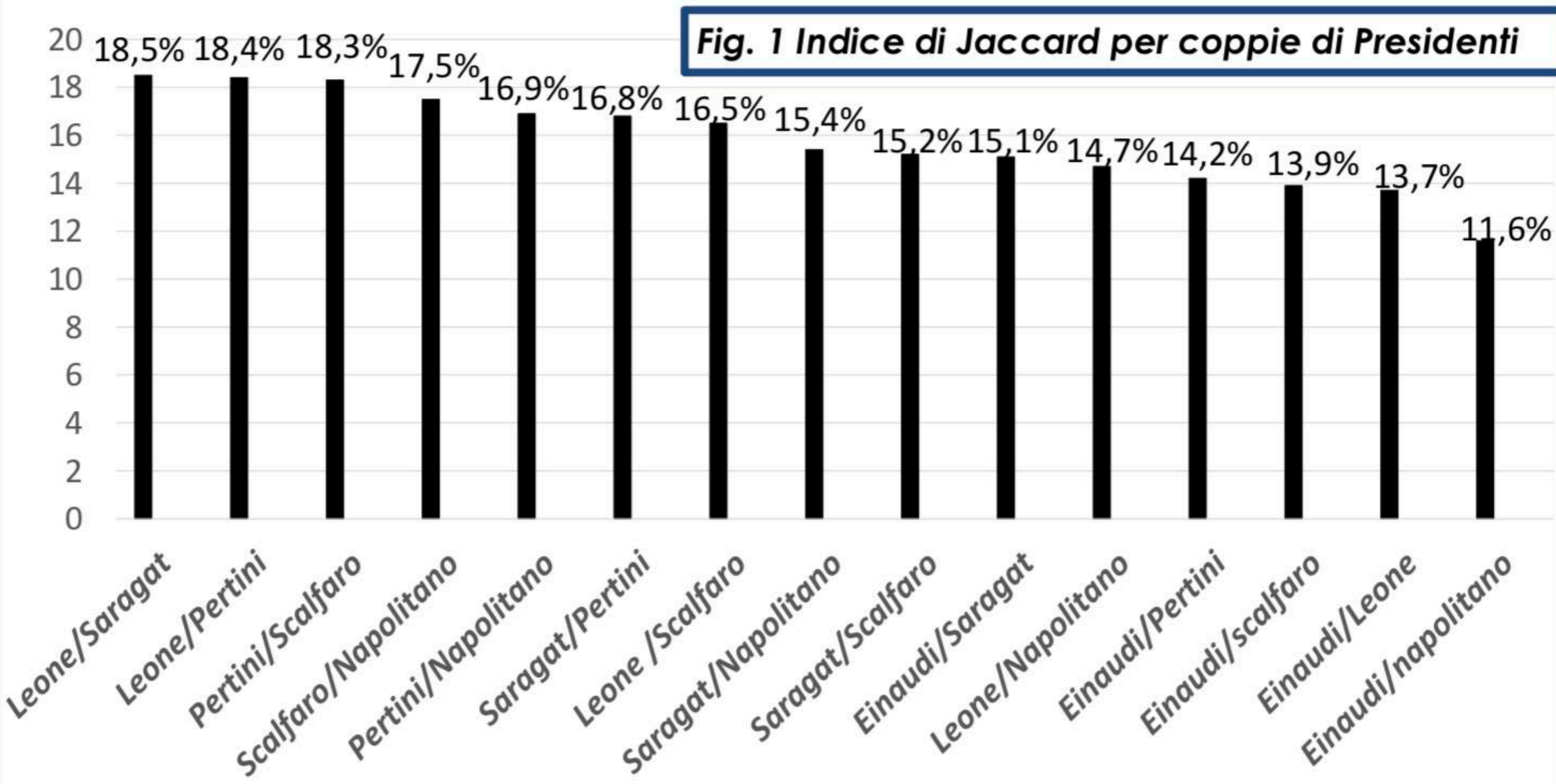
Poster concorso di statistica sui discorsi dei Presidenti della Repubblica

Statistica Presidenziale

Un'analisi statistica sul linguaggio dei Presidenti della Repubblica Italiana nel tempo

Nell'analisi abbiamo preso in considerazione 6 (su 12) Presidenti della Repubblica Italiana, uno per ogni decennio a partire dal 1948. Dall'analisi è stato escluso l'attuale decennio e dunque l'ultimo Presidente della Repubblica (Mattarella). Sono stati selezionati per ciascuno, in formato digitale dal sito del Quirinale, un discorso per l'inizio dell'anno scolastico, uno di fine anno solare, uno in occasione della festa della Repubblica e un discorso per altre feste nazionali (o 4/11 o 25/04). Per i primi due Presidenti non è stato possibile trovare tali discorsi e abbiamo usato altri discorsi pubblici. Abbiamo scansionato i discorsi digitali mediante i tool offerti gratuitamente dal sito web Lexicool.com e abbiamo successivamente elaborato i dati estrapolati con programmi scritti da noi in Python.

Poster concorso di statistica sui discorsi dei Presidenti della Repubblica



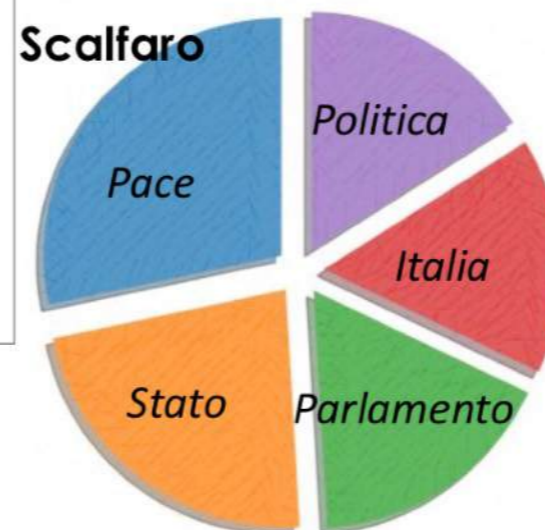
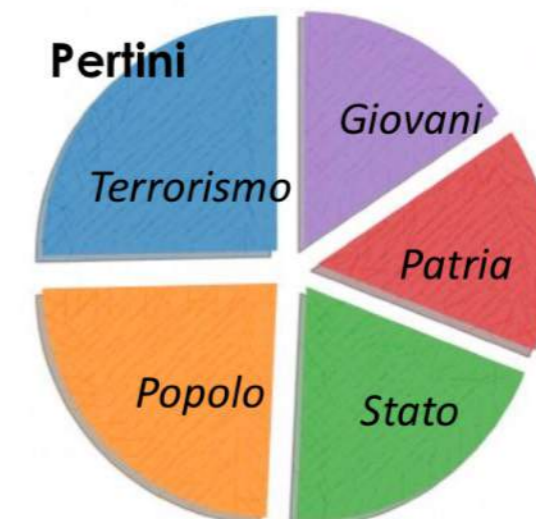
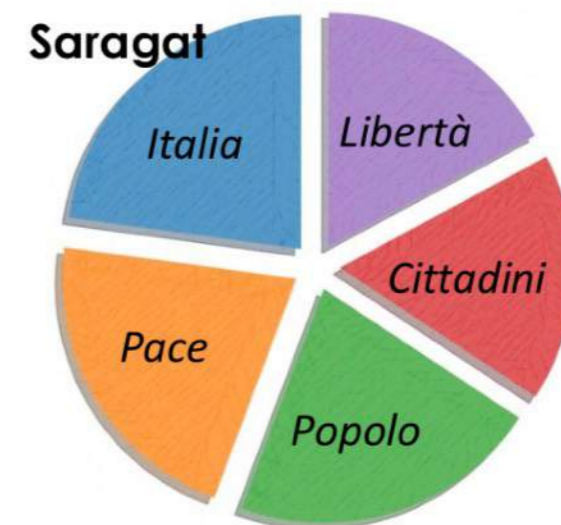
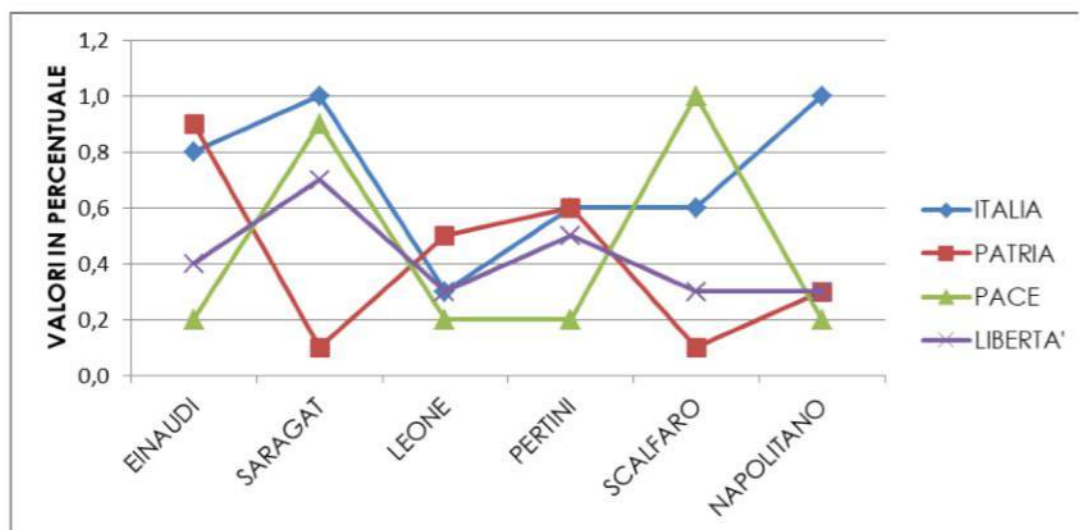
LE PAROLE PIU' AMATE DAI PRESIDENTI

Quali sono le parole più amate dai 6 Presidenti scelti? La Fig. 2 mostra i diagrammi a torta per le 5 parole maggiormente preferite nei discorsi pubblici da ciascun Presidente della Repubblica Italiana. Anche i grafici sono stati costruiti in linguaggio Python

LE PAROLE NEL TEMPO

Le parole chiave usate dai vari Presidenti a partire da Einaudi fino a Napolitano mostrano una differenza al passare del tempo? Il buonsenso dice di sì, la statistica lo conferma e lo quantifica mediante l'analisi dell'andamento dell'uso di quattro parole chiave (Italia, Patria, Pace, Libertà). I dati di Fig.3, in percentuale, sono estratti dai discorsi privati di «stop words» mentre il grafico è stato creato con Excel.

Fig. 3 Le 4 parole chiave nel tempo



Riscontro nella classe

- Una maggioranza di studenti ha mostrato un grosso entusiasmo per i vari aspetti del coding
- La attività laboratoriale (sia di fisica che di matematica) ha avuto un buon successo: gli studenti mostrano una buona attitudine al lavoro di gruppo
- Le attività interdisciplinari sono risultate interessanti alla classe
- Si è confermato il problema di "genere" relativo all'uso delle nuove tecnologie (anche se studentesse nella classe erano 6 su 22 e ora 4 su 20)

Bibliografia

- [11] C. Basile, D. Benedetto, E. Caglioti, M. Degli Esposti, An example of mathematical authorship attribution, *Journal of Mathematical Physics*, 49, 125211, 2008.
- [12] M. Degli Esposti, Riconoscere Gramsci con la matematica, articolo disponibile in rete qui http://www.treccani.it/magazine/lingua_italiana/speciali/attributione/Degli_Esposti.html
- [13] A.A.V.V., Drawing Elena Ferrante's profile, *Workshop Proceedings Padova*, 7 September 2017