

IA e Machine Learning

Google Colaboratory e Dataset Iris

Corso: Gestione progetto e organizzazione d'impresa

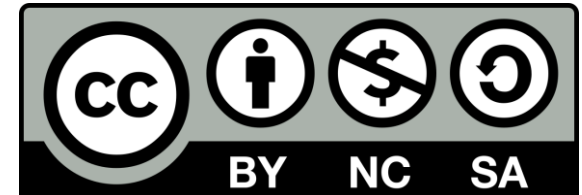
Docente: Leonardo Essam Dei Rossi

Licenze e crediti

Questo materiale è disponibile sul sito Web del docente per il corso di [Gestione progetto e organizzazione d'impresa](#) per le studentesse e gli studenti dell'anno scolastico 2025/2026.

Versione: 1.0.0 (A)
Ultima modifica: 05/05/2026 09:14

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)



Sezione #1

Google Colaboratory

Google Colaboratory

- **Google Colaboratory**, conosciuto semplicemente come **Colab**, è un ambiente di sviluppo basato su cloud offerto gratuitamente da Google:
 - <https://colab.research.google.com/>
- Permette di scrivere ed eseguire codice Python direttamente nel browser.



I vantaggi di Google Colaboratory

- Zero configurazioni: tutto l'ambiente è in cloud quindi non c'è bisogno di installare e configurare software aggiuntivo nel proprio PC;
- Accesso gratuito a risorse avanzate:
 - GPU per l'addestramento di modelli di IA;
 - TPU per migliorare ulteriormente l'addestramento.

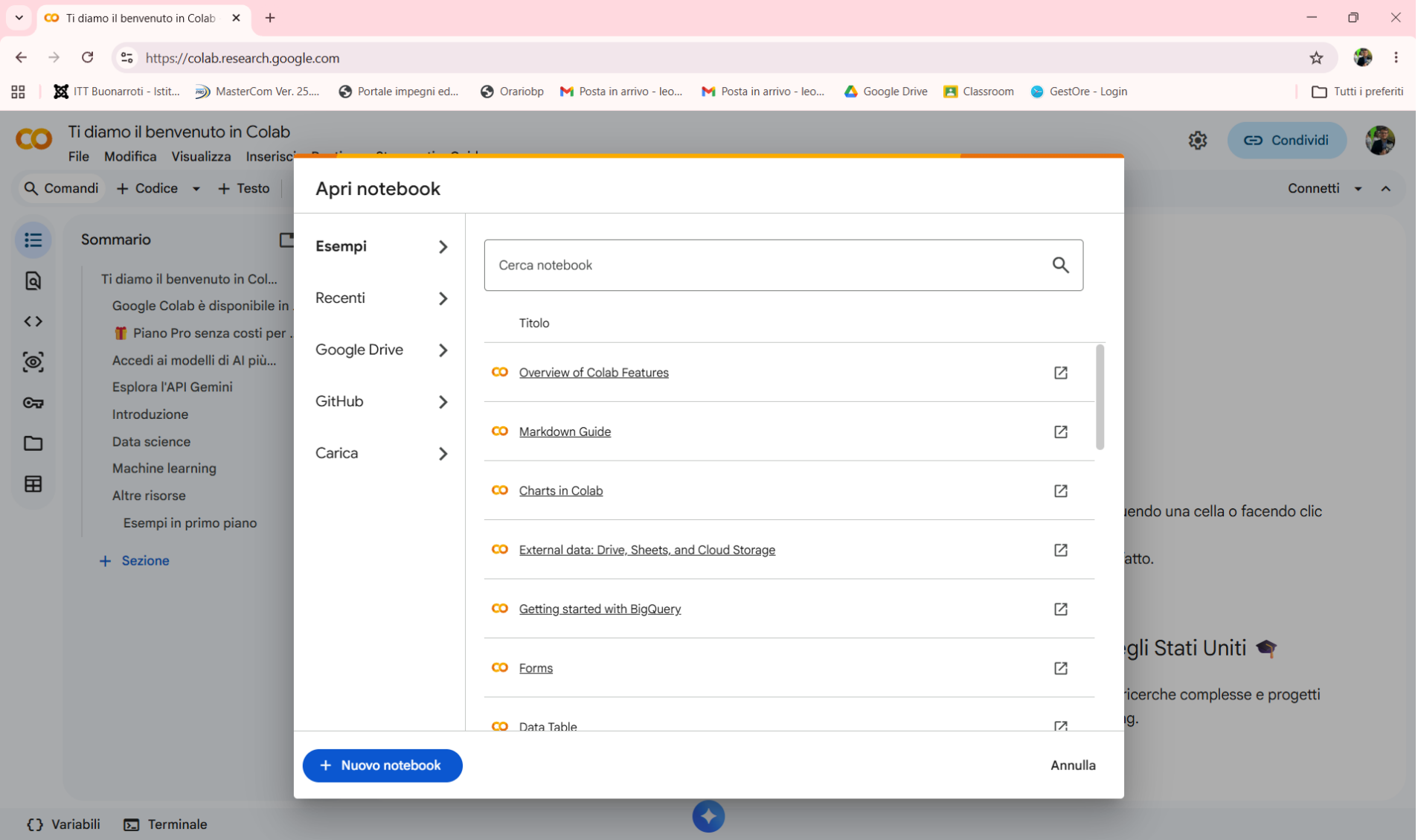
Q: *perché si usano le GPU nell'addestramento dei modelli di IA?*

Processing sequenziale e parallelo

Q: *perché si usano le GPU nell'addestramento dei modelli di IA?*

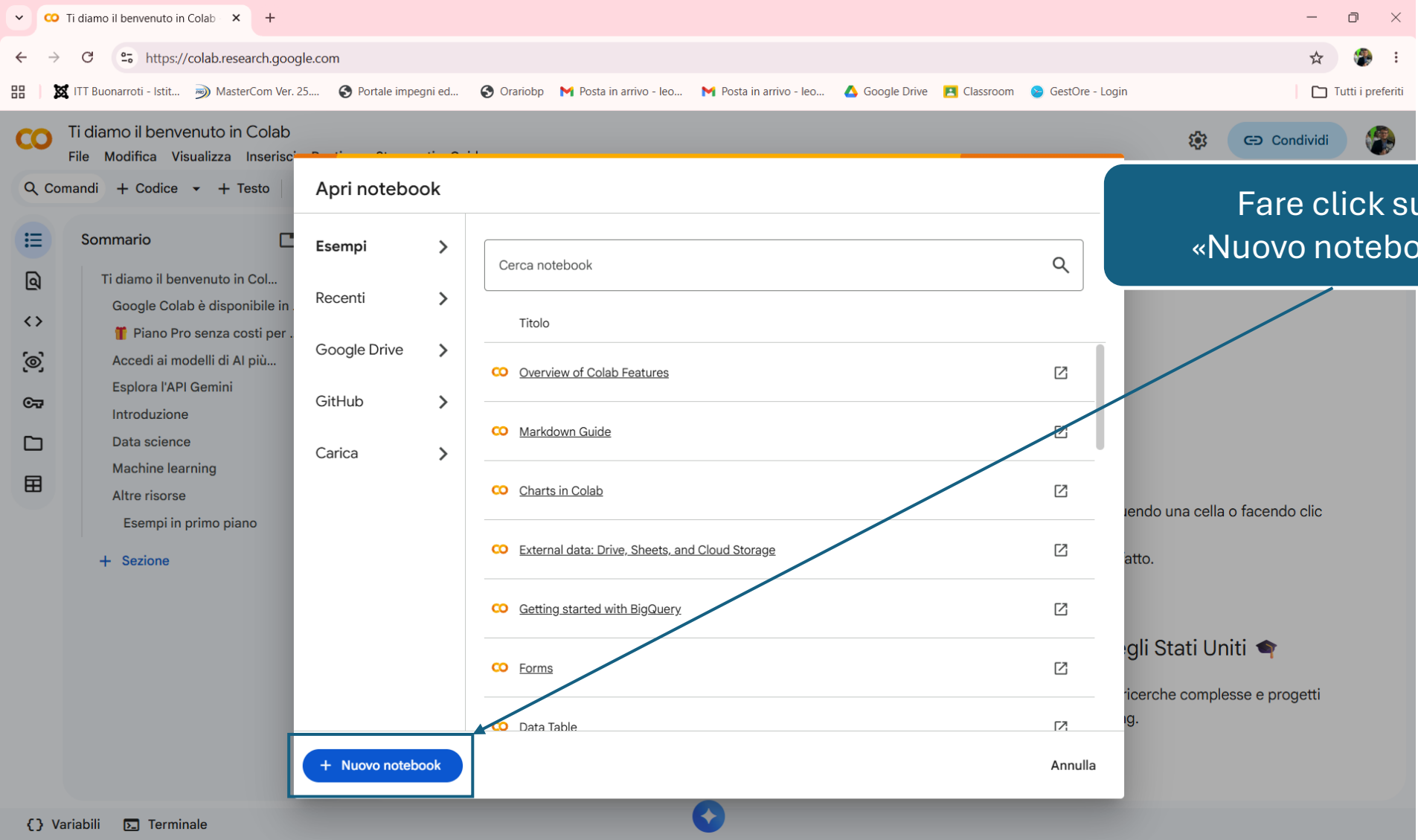
- Il motivo principale per cui vengono usate le GPU nell'addestramento dei modelli di IA è che le GPU elaborano le informazioni in modo parallelo:
 - A differenza della CPU che le elabora in modo sequenziale.
- Esempio: una GPU può elaborare parallelamente un milione di immagini molto più velocemente di quanto farebbe una CPU sequenzialmente (una alla volta!).

Primo progetto con Google Colaboratory



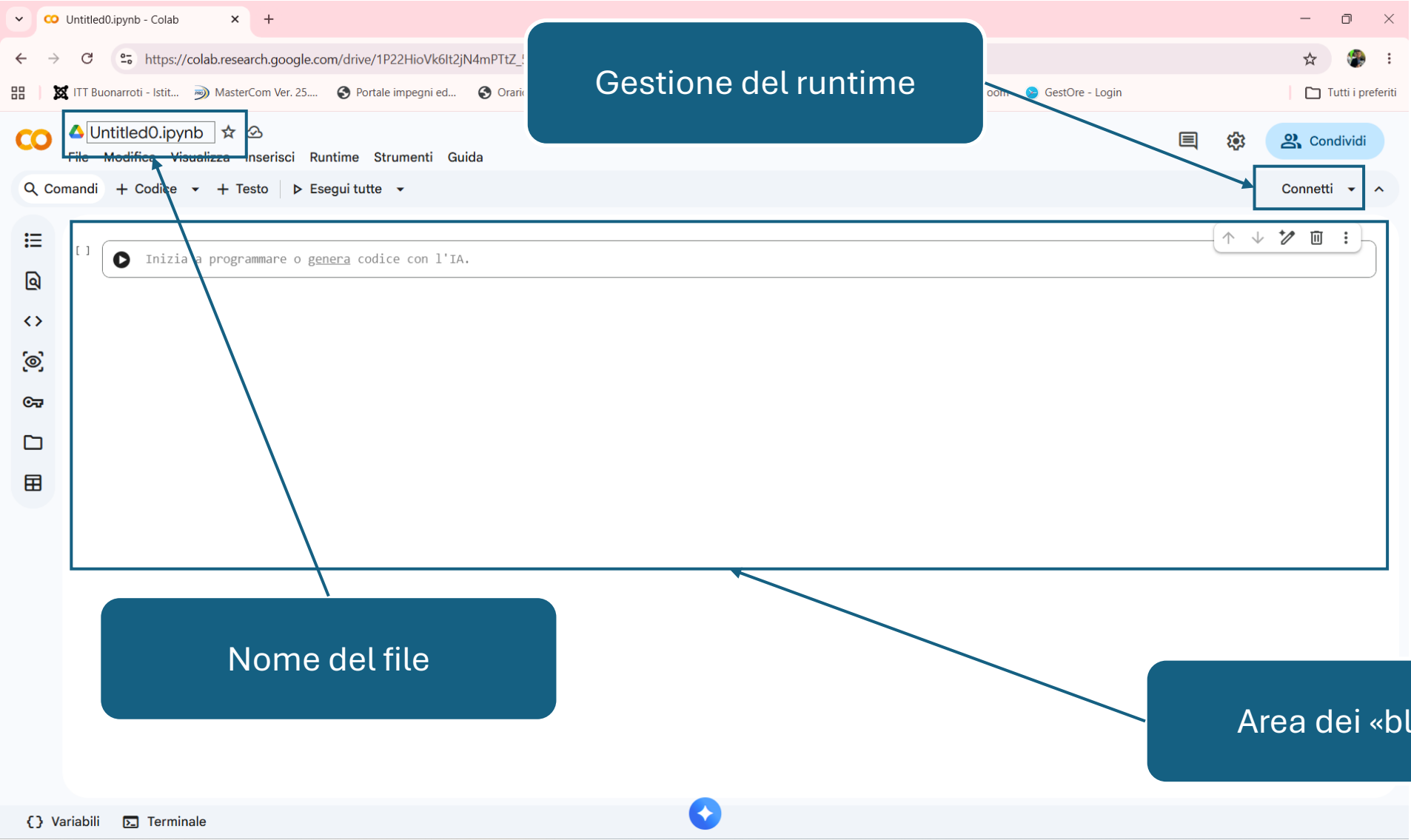
Schermata iniziale di Colab

Primo progetto con Google Colaboratory



Schermata iniziale di Colab

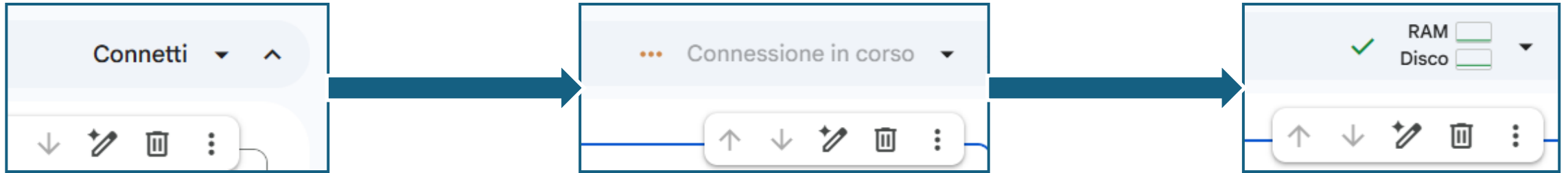
Primo progetto con Google Colaboratory



Schermata dell'editor di Colab

Primo progetto con Google Colaboratory

- Per avviare la macchina virtuale che eseguirà il codice, fare click su «Connetti»:



Primo progetto con Google Colaboratory

- Facendo click sulle risorse si apre la visualizzazione dei dettagli dell'istanza di Google Colab:
 - RAM disponibile;
 - Spazio su disco.

Nota bene

Per quello che faremo, il runtime di base è più che sufficiente.

Risorse

Non hai un abbonamento. [Ulteriori informazioni](#)
Al momento non hai unità di calcolo disponibili. Le risorse offerte senza costi non sono garantite. Acquista altre unità [qui](#).
Al tuo livello di utilizzo attuale, questo runtime potrebbe durare fino a 83 ore.

[Gestisci sessioni](#)

! Vuoi più memoria e spazio su disco?
[Esegui l'upgrade a Colab Pro](#)

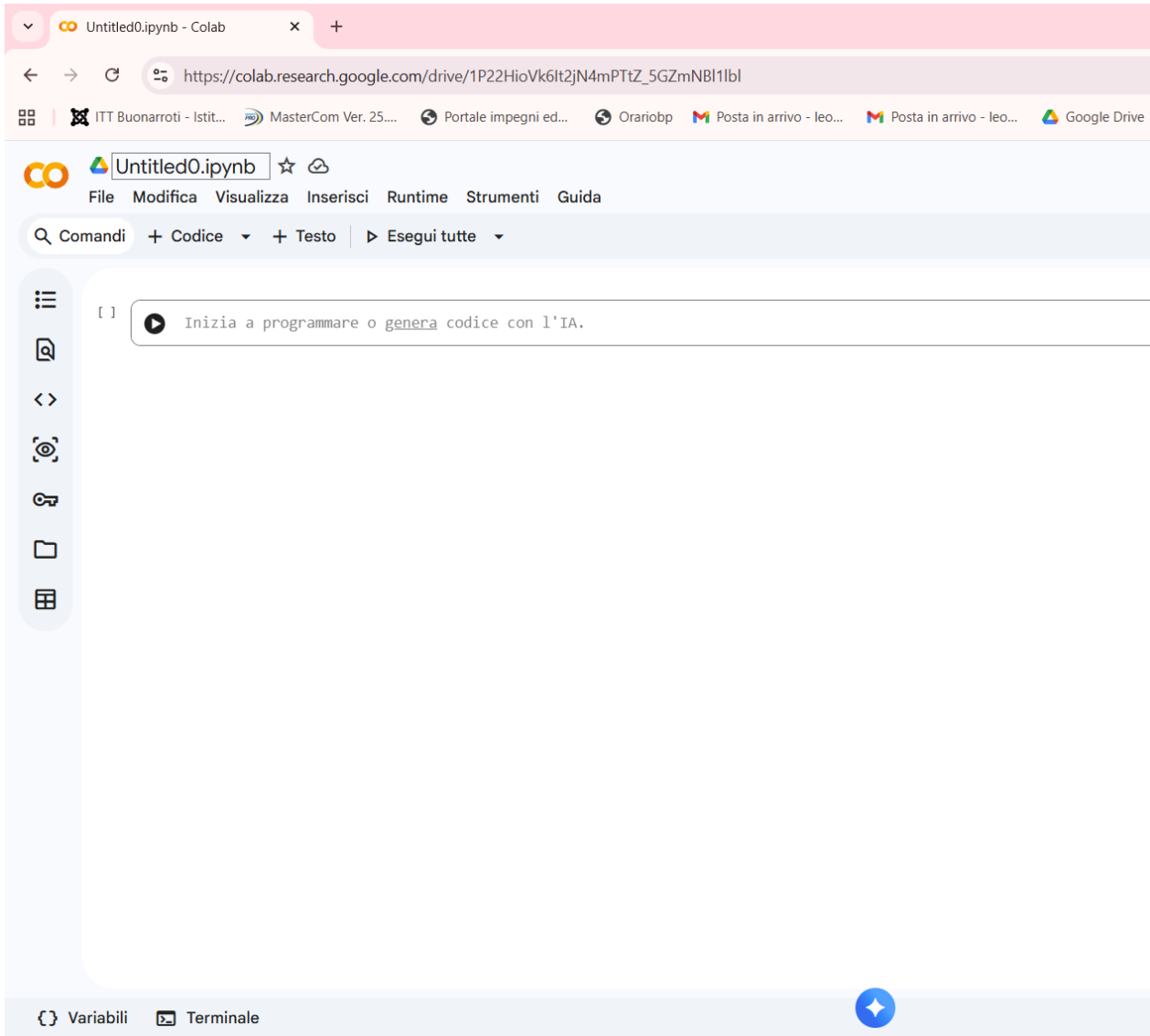
Backend Google Compute Engine Python 3
Visualizzazione delle risorse dal giorno 16:06 al giorno 16:09

RAM di sistema 1.0 / 12.7 GB	Disco 20.7 / 107.7 GB

[Cambia tipo di runtime](#)

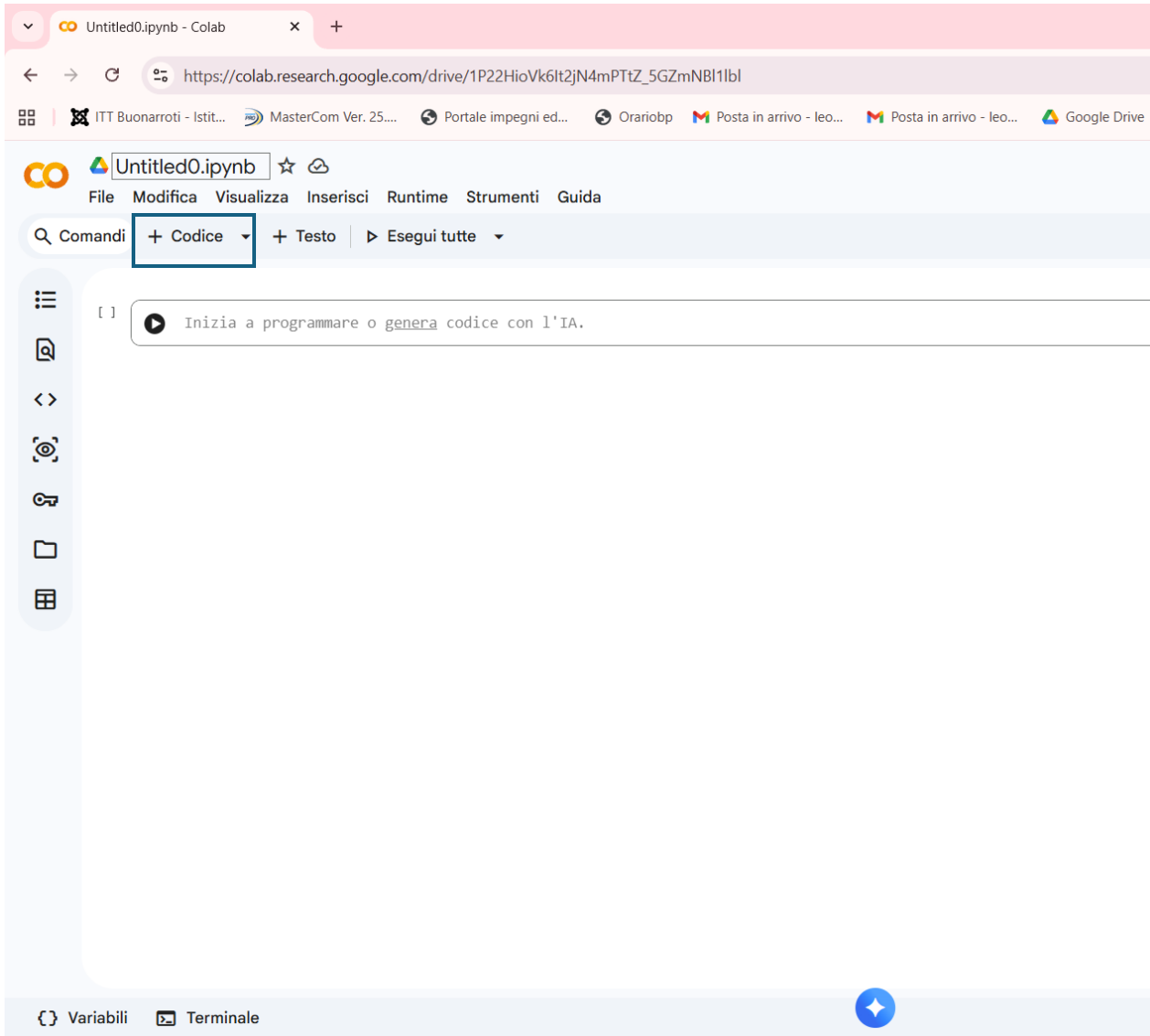
Python 3

Primo progetto con Google Colaboratory



- All'inizio, Colab predispone già un blocco di tipo «codice» dove si può da subito scrivere del codice Python;
- Per eseguirlo poi fare click sull'icona del «play» a sinistra del blocco stesso.

Primo progetto con Google Colaboratory



- Successivamente, aggiungeremo altri blocchi di tipo «codice» con il pulsante «+ Codice» in alto a sinistra.

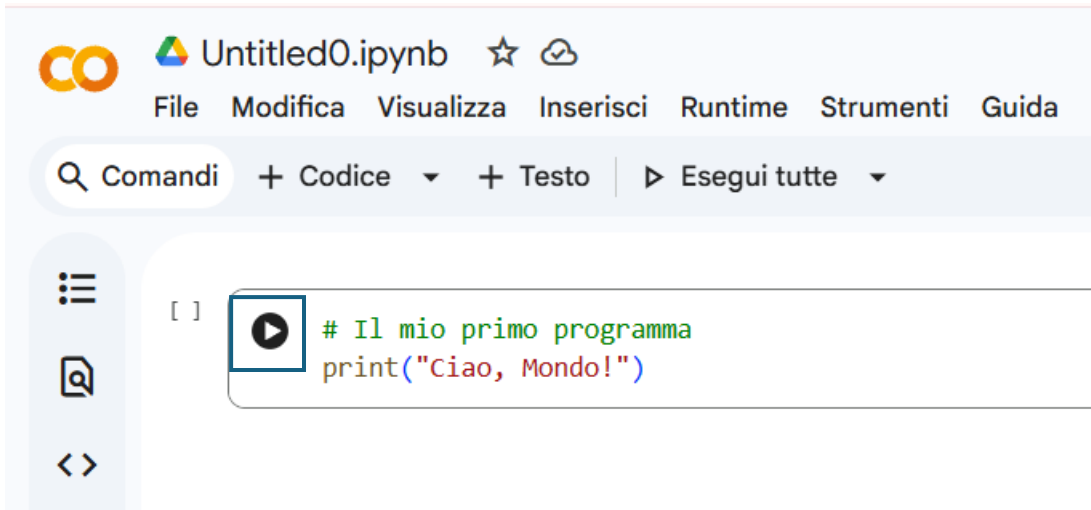
Primo progetto con Google Colaboratory

- Provate voi ad eseguire il programma «Ciao, Mondo!»:

Esempio: *programma «Ciao, Mondo!»*

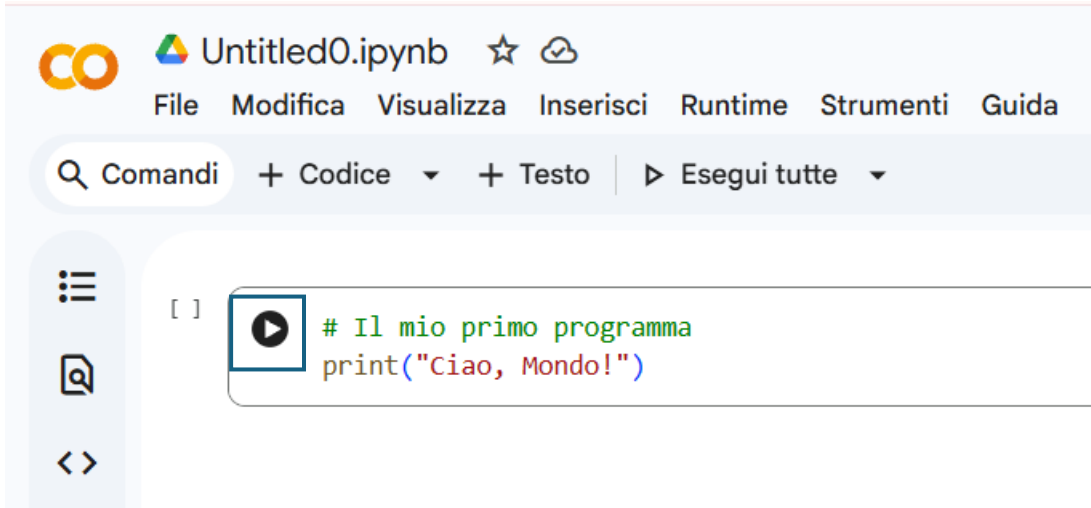
```
# Il mio primo programma!  
print("Ciao, Mondo!")
```

Primo progetto con Google Colaboratory

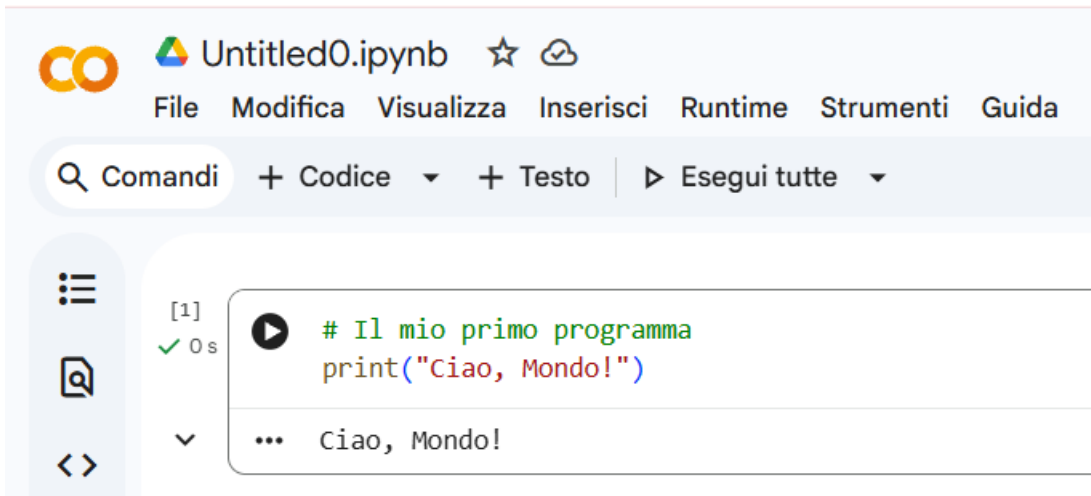


- Inserito il codice del programma fare click su «play».

Primo progetto con Google Colaboratory



- Inserito il codice del programma fare click su «play»;



- L'output dell'esecuzione del blocco viene visualizzato subito sotto.

Sezione #2

Il dataset Iris

Il dataset Iris

- Il dataset Iris è uno degli esempi più famosi come base di partenza del Machine Learning:
 - Un po' come «Hello World» in programmazione!
- È un dataset semplice composto da 150 campioni del fiore Iris.



Il dataset Iris

- Il dataset al suo interno contiene 150 campioni così divisi:
 - 50 di Iris setosa;
 - 50 di Iris versicolor;
 - 50 di Iris virginica.

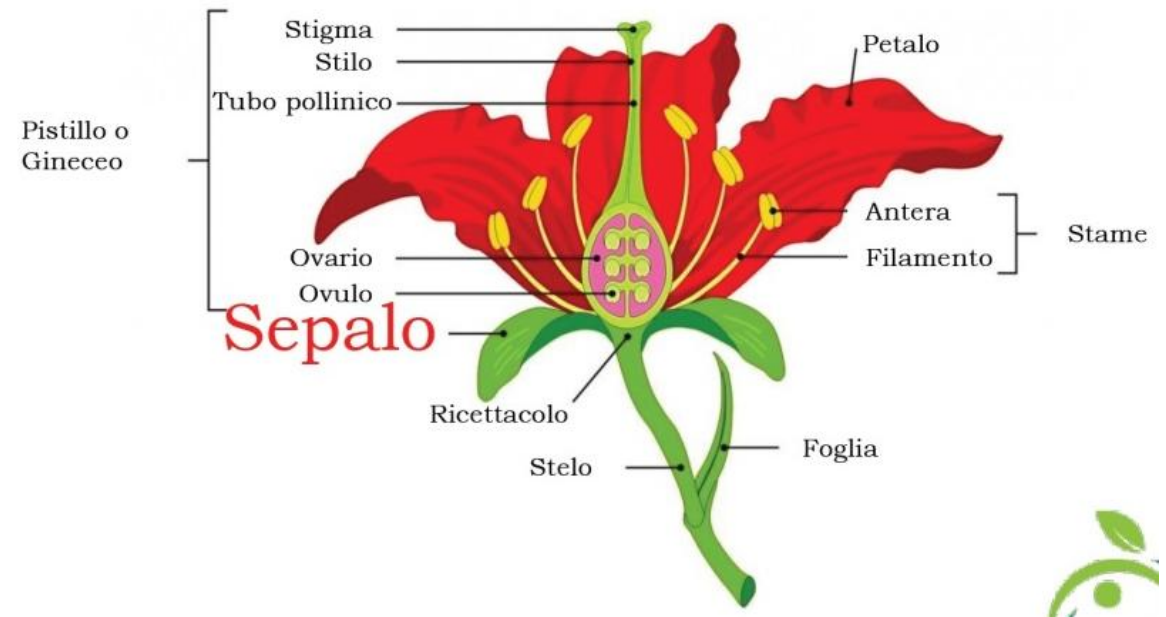


Formalizzare il problema

- Il dataset Iris solitamente si usa nel seguente modo:
 - Date le dimensioni del sepalò e del petalo, il modello confronta le misure tra quelle già presenti nel dataset e restituisce qual è la variante del fiore che più probabilmente abbiamo misurato.

Esercizio:

Formalizzare il problema che questo modello deve risolvere.



Formalizzare il problema

Nomi delle specie

Misure del fiore

Perché nel dataset la specie è identificata con un numero intero (0, 1, 2)

```
[7]
✓ 0s
▶ # Let's predict a new flower!
  new_flower = [[3.1, 5.5, 4.4, 0.2]] # measurements in cm

  prediction = model.predict(new_flower)
  species_names = ['Setosa', 'Versicolor', 'Virginica']

  print(f"🌸 For a flower with measurements: {new_flower[0]}")
  print(f"   Predicted species: {species_names[prediction[0]]}")

▼ ... 🌸 For a flower with measurements: [3.1, 5.5, 4.4, 0.2]
   Predicted species: Versicolor
```

Previsione del modello