Data Science/Machine Learning/Deep Learning Workshop

Nueva Ecija University of Science and Technology (NEUST)

Rodolfo C. Raga Jr., PhDCS

December 21-23, 2021

Day 1 : Basic Concepts, Intuitions, and Foundations

Topic 1: Introduction to Data Science, Machine Learning, and Deep Learning
- What is Data Science
- What is Machine Learning
- What is Deep Learning

Topic 2: Intro to Python, Jupyter Notebook, Google Colab, Scikit-learn, Tensorflow, and Keras
- Why use Python
- Jupyter Notebook vs. Google Colab
- Understanding sklearn, Tensorflow and Keras

Topic 3: Fundamentals of Data Analysis using sklearn
- Correlation and Distribution Analysis
- Data Preprocessing (Feature Selection, Data Normalization, Data Splitting)
- Building and Training of ML prediction models using sklearn

Topic 4: Fundamentals of Neural Networks
- Definition and NN Architecture
- Perceptron and Multi-layer Perceptron Architecture
- Building NN prediction models using sklearn

Outline

- Brief Description of Python
- What are Notebooks?
- Installing Jupyter Notebooks
- Getting Started with Google Colab
- Demo
    - Sklearn
    - Tensorflow
    - Keras

About Python

- Python is a high-level programming language
- Open source and community driven
- Standard distribution includes many modules
- Dynamic typed
- Source can be compiled or run just-in-time
- Python jibes pretty well with data analysis and it is touted as one of the most preferred language for data science.
- There are many IDEs designed to support Python development such as PyCharm, Spyder, Eclipse.
- For this workshop, we will be using a **notebook** interface.

What are notebooks?

- A *notebook* combines the functionality of:
  1. a word processor that handles formatted text
  2. a "shell" that executes statements in a programming language and includes output inline
  3. a rendering engine that renders and graphic outputs

- The Jupyter system notebook supports over 100 programming languages (called "kernels" in the Jupyter ecosystem) including Python, Java, R, Julia, Matlab, Octave, Scheme, Processing, Scala, and many more.
- Out of the box, Jupyter will only run the IPython kernel, but additional kernels may be installed. Language support continues to be added by the open source community and the best source for an up-to-date list is the wiki page maintained by the project: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels.

Installing Jupyter Notebook

- Simpliest approach to installing Python and Jupyter notebook is by using **Anaconda**.
- **Anaconda** is a distribution software which bundle many programming language in one package. Python and R, for example, come pre-loaded in Anaconda.
- For a guide on how to install Anaconda, you can refer to the tutorial provided in DataCamp (https://www.datacamp.com/community/tutorials/installing-anaconda-windows)

The Jupyter Notebook

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser.

The Jupyter Notebook App can be executed on a local desktop requiring no internet access or it can be installed on a remote server and accessed through the internet.

Best of all, as part of the open source Project Jupyter, they are completely free.

The Jupyter Notebook

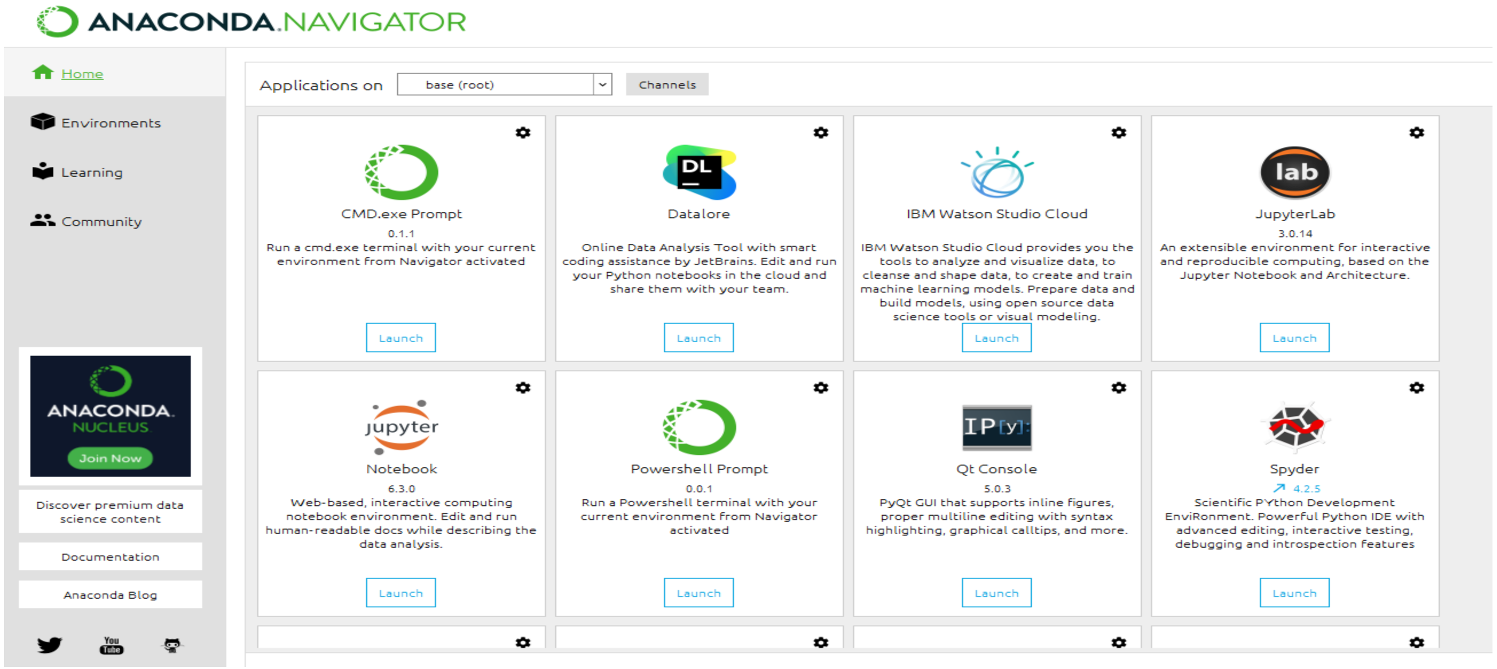
In addition to displaying/editing notebook documents, the Jupyter Notebook App provides

- a “Dashboard” showing local files and allowing to open notebook documents or shutting down their kernels.
- a Kernel engine that executes codes

Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Starting the Jupyter Notebook

The Jupyter Notebook App can be launched by clicking on the Launch button in the Jupyter Notebook icon in the
Anaconda Navigator interface.


ANACONDA.NAVIGATOR

Home
Environments
Learning
Community

Applications on base (root) Channels

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch

JupyterLab
3.0.14
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
6.3.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch

Qt Console
5.0.3
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

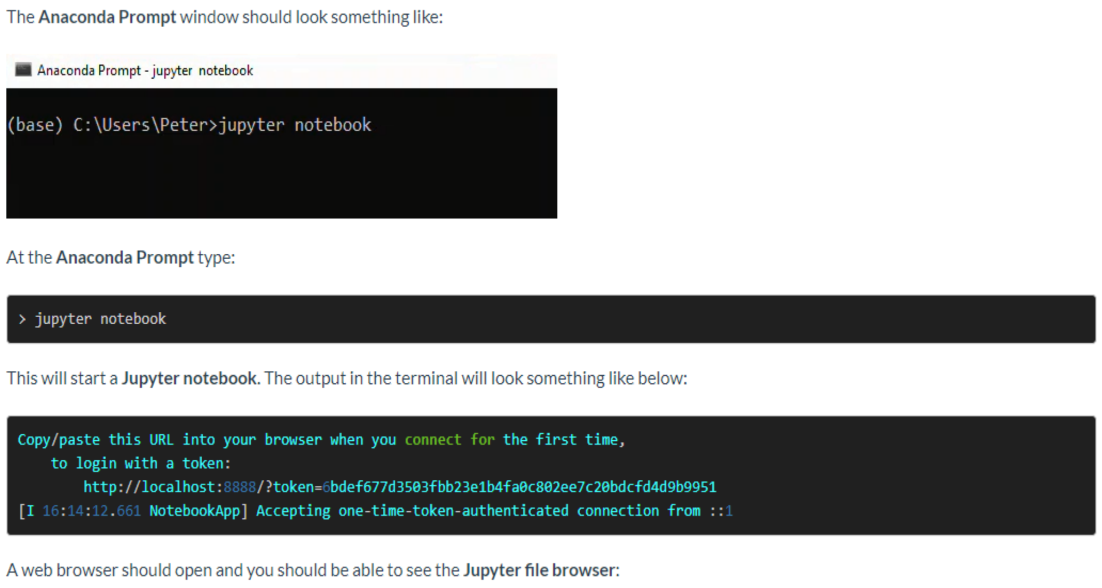
Spyder
4.2.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

ANACONDA NUCLEUS
Join Now
Discover premium data science content
Documentation
Anaconda Blog

This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open.

Starting the Jupyter Notebook

The Jupyter Notebook App can also be launched by going to the Windows start menu and selecting **[Anaconda Prompt]** under **[Anaconda3]** and typing the command "jupyter notebook" on the prompt:



The **Anaconda Prompt** window should look something like:

Anaconda Prompt - jupyter notebook

```
(base) C:\Users\Peter>jupyter notebook
```

At the **Anaconda Prompt** type:

```
> jupyter notebook
```

This will start a **Jupyter notebook.** The output in the terminal will look something like below:

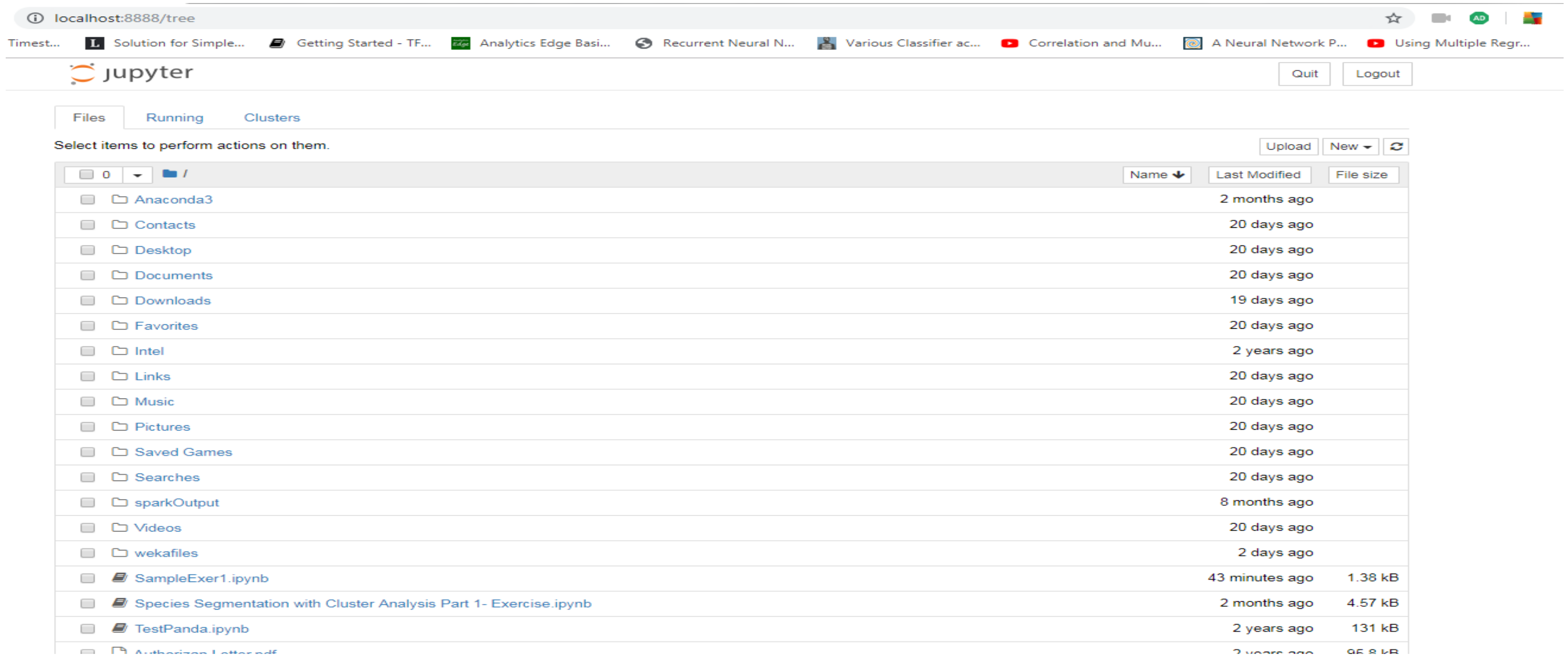
```
Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=6bdef677d3503fbb23e1b4fa0c802ee7c20bdcfd4d9b9951
[I 16:14:12.661 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

A web browser should open and you should be able to see the **Jupyter file browser**:

This will also launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open.

The Jupyter Dashboard

The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown).

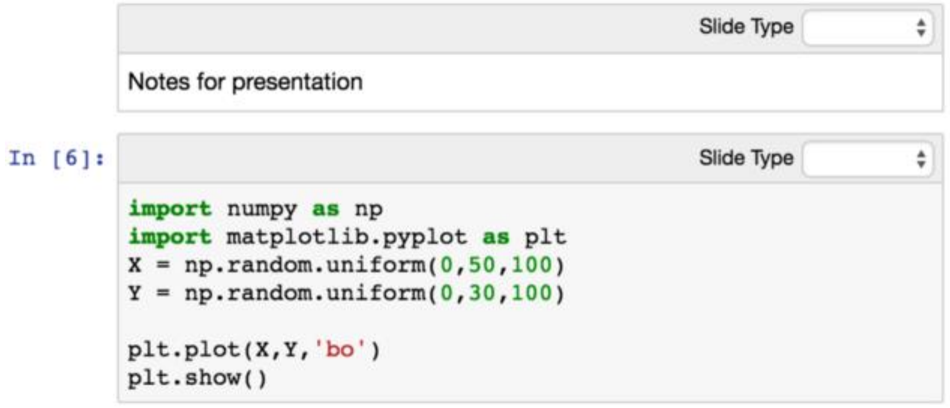

localhost:8888/tree
Timest... Solution for Simple... Getting Started - TF... Analytics Edge Basi... Recurrent Neural N... Various Classifier ac... Correlation and Mu... A Neural Network P... Using Multiple Regr...
jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size

	Name	Last Modified	File size
	Anaconda3	2 months ago	
	Contacts	20 days ago	
	Desktop	20 days ago	
	Documents	20 days ago	
	Downloads	19 days ago	
	Favorites	20 days ago	
	Intel	2 years ago	
	Links	20 days ago	
	Music	20 days ago	
	Pictures	20 days ago	
	Saved Games	20 days ago	
	Searches	20 days ago	
	sparkOutput	8 months ago	
	Videos	20 days ago	
	wekafiles	2 days ago	
	SampleExer1.ipynb	43 minutes ago	1.38 kB
	Species Segmentation with Cluster Analysis Part 1- Exercise.ipynb	2 months ago	4.57 kB
	TestPanda.ipynb	2 years ago	131 kB
	Authorizon Letter.pdf	2 years ago	95.8 kB

The Jupyter Cells

Code Cells
Contents in this cell are treated as statements in a programming language of current kernel. Default kernel is Python. So, we can write Python statements in a code cell. When such cell is run, its result is displayed in an output cell. The output may be text, image, matplotlib plots or HTML tables. Code cells have rich text capability.

Markdown Cells
These cells contain text formatted using markdown language. All kinds of formatting features are available like making text bold and italic, displaying ordered or unordered list, rendering tabular contents etc. Markdown cells are especially useful to provide documentation to the computational process of the notebook.



Notes for presentation

```
import numpy as np
import matplotlib.pyplot as plt
X = np.random.uniform(0,50,100)
Y = np.random.uniform(0,30,100)

plt.plot(X,Y,'bo')
plt.show()
```

Markdown Cells

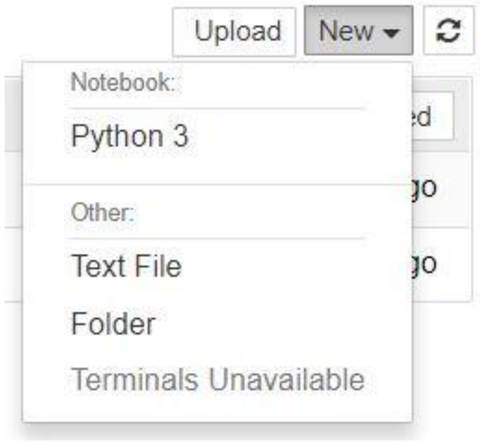
Code Cells

What is an ipynb File?

An .ipynb file is a text file that stores the contents of your notebook in a format called JSON.

Each cell and its contents, including image attachments that have been converted into strings of text, is listed in the .ipynb file along with some metadata.

To open a new ipynb file follow the image below and select Python 3 or any version avaialable that you want to use.



Some useful shortcuts in Jupyter Notebook

- Toggle between edit and command mode with Esc and Enter, respectively.
- Scroll up and down your cells with your Up and Down keys.
- Press A or B to insert a new cell above or below the active cell.
- M will transform the active cell to a Markdown cell.
- Y will set the active cell to a code cell.
- D + D (D twice) will delete the active cell.
- Z will undo cell deletion.
- Hold Shift and press Up or Down to select multiple cells at once.
  - With multple cells selected, Shift + M will merge your selection.
- Ctrl + Shift + -, in edit mode, will split the active cell at the cursor.
- You can also click and Shift + Click in the margin to the left of your cells to select them.

Naming a Notebook

A new notebook will have the default notebook file name Untitled.ipynb. You cannot rename a notebook while it is
running, so you’ve first got to shut it down. The easiest way to do this is to select “File > Close and Halt” from the
notebook menu. However, you can also shutdown the kernel either by going to “Kernel > Shutdown” from
within the notebook app or by selecting the notebook in the dashboard and clicking “Shutdown”.

Then, you can select your notebook and and click “Rename” in the dashboard controls.

Google colab

GETTING STARTED WITH GOOGLE COLAB

Dr. Rodolfo C. Raga Jr.

NEUST Workshop – Dec 21-23, 2021

The Colab Notebook

The Colab Notebook App is an online application that allows editing and running of notebook documents via a web browser.

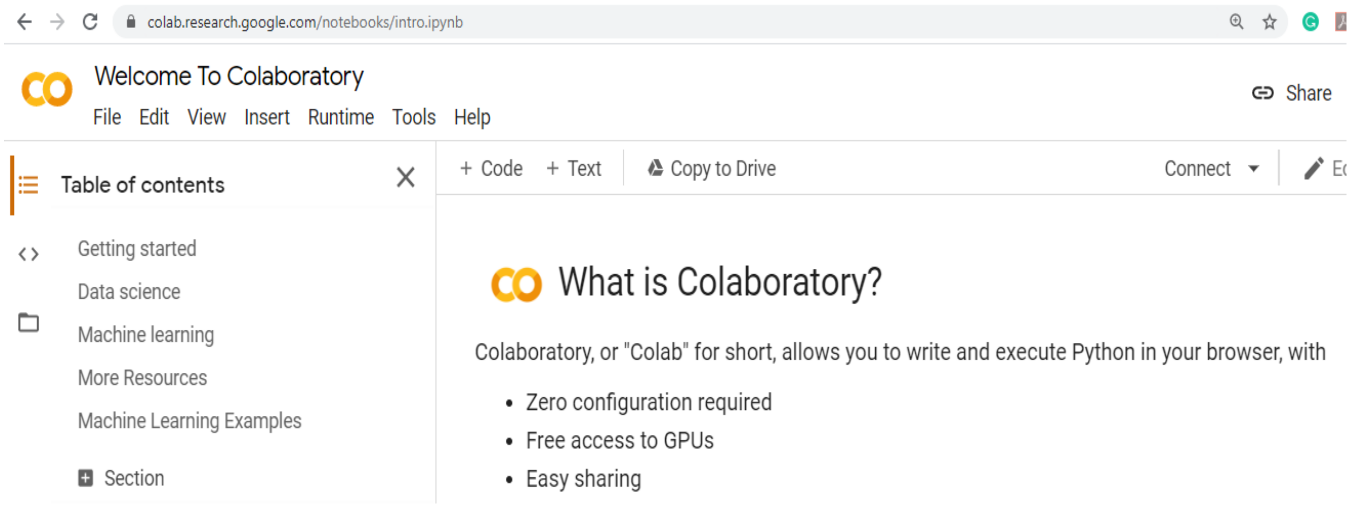
Colaboratory is a Google research project created to help disseminate machine learning education and research.

It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

The Colab Notebook

In addition to displaying/editing notebook documents, the Colab Notebook App provides

- develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV.
- The most important feature that distinguishes Colab from other free cloud services is; Colab provides GPU and is totally free.
- You will need to have a logged gmail account in order to access google colab



What is Colaboratory?

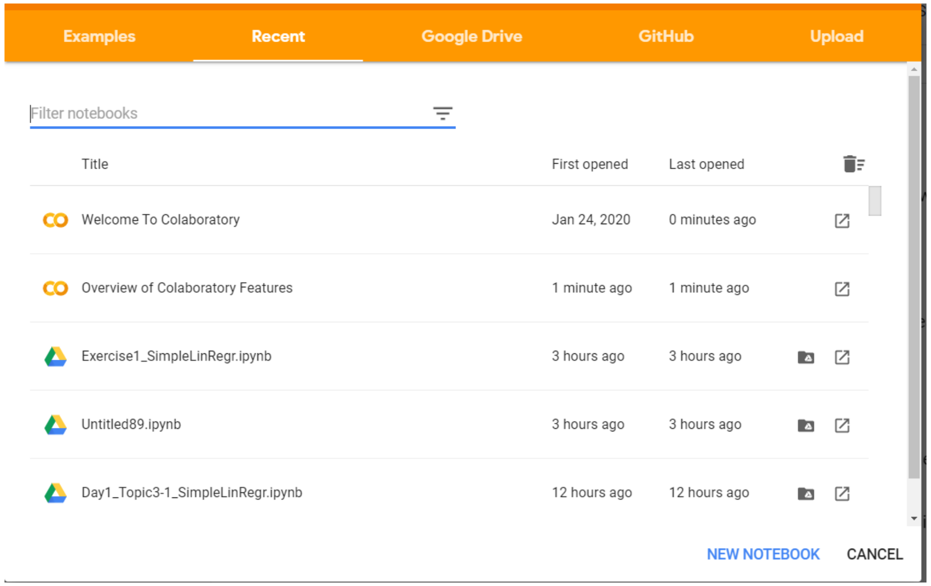
Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Starting the Colab Notebook

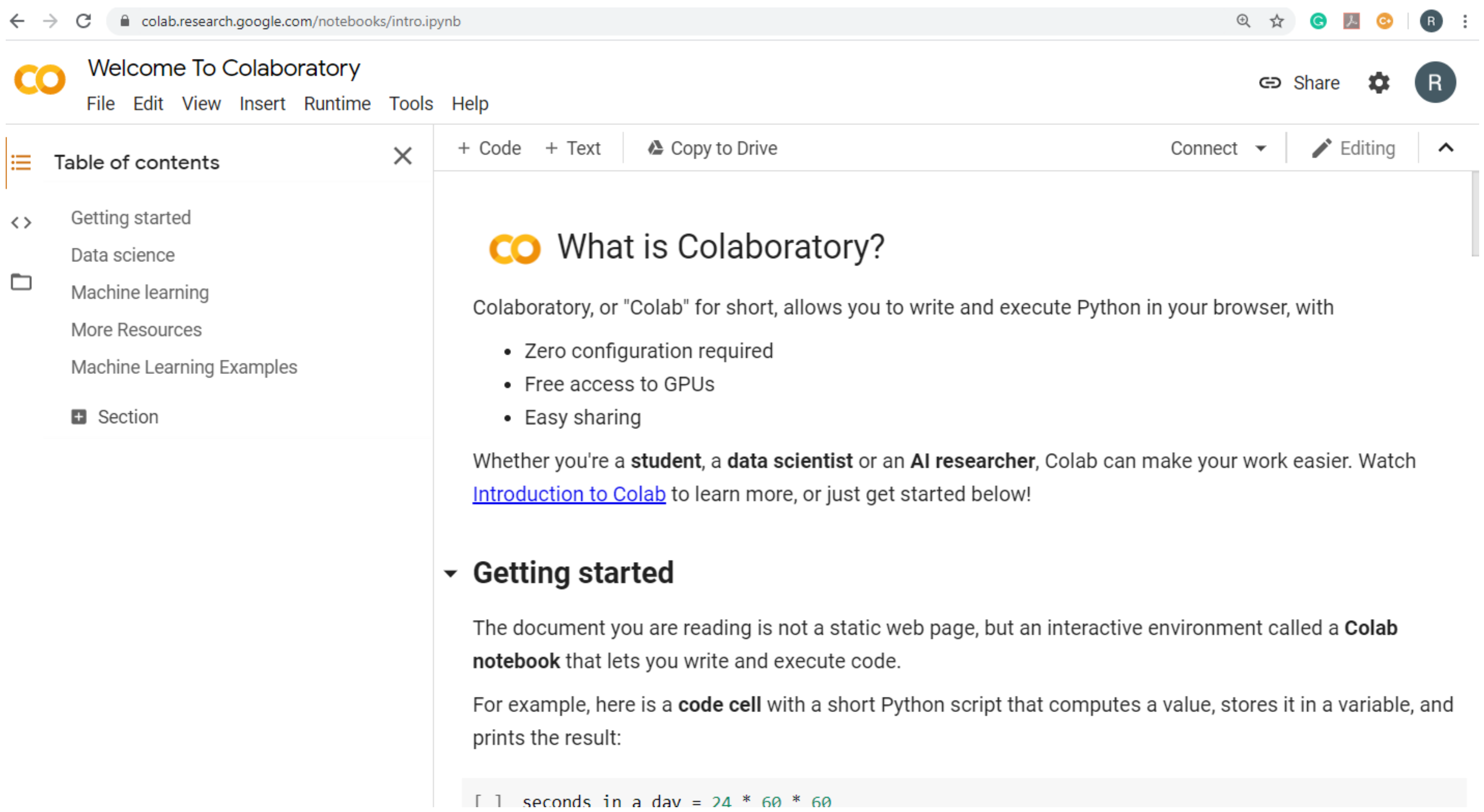
The Colab Notebook App can be launched by typing the URL : colab.research.google.com in your browser

This will launch a new browser window showing a sort of control panel that allows you to select which notebook to open ot upload an existing notebook file.



Title	First opened	Last opened
Welcome To Colaboratory	Jan 24, 2020	0 minutes ago
Overview of Colaboratory Features	1 minute ago	1 minute ago
Exercise1_SimpleLinRegr.ipynb	3 hours ago	3 hours ago
Untitled89.ipynb	3 hours ago	3 hours ago
Day1_Topic3-1_SimpleLinRegr.ipynb	12 hours ago	12 hours ago

The Colab Dashboard



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

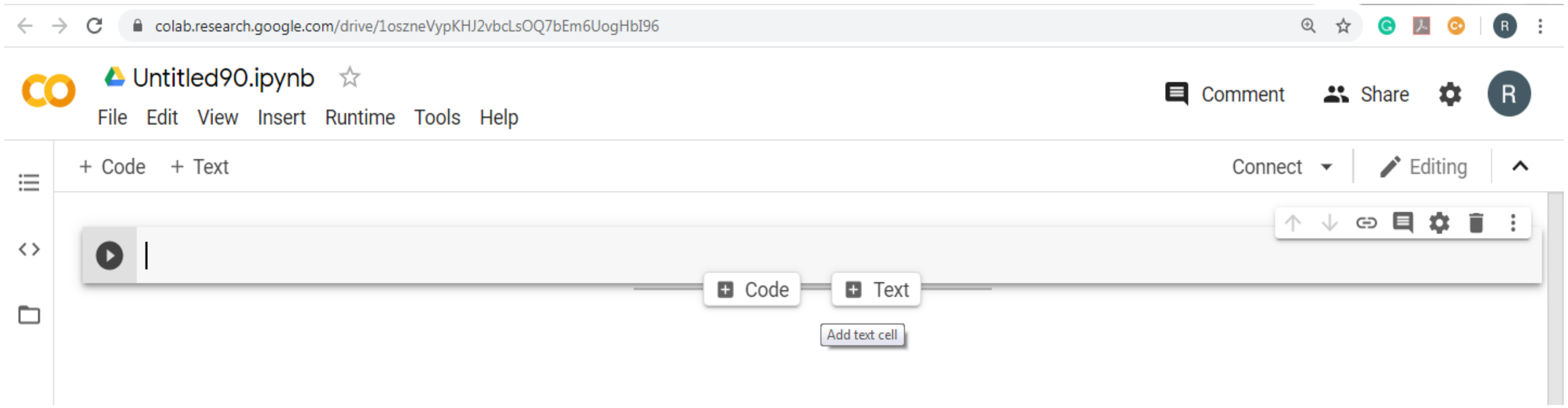
For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ]  seconds_in_a_day = 24 * 60 * 60
```

Dr. Rodolfo C. Raga Jr.
NEUST Workshop – Dec 21-23, 2021

The Colab Notebook Interface

Colab notebooks allow you to combine **executable code** and **rich text** in a single document, along with **images**, **HTML**, **LaTeX** and more. When you create your own Colab notebooks, they are stored in your Google Drive account.

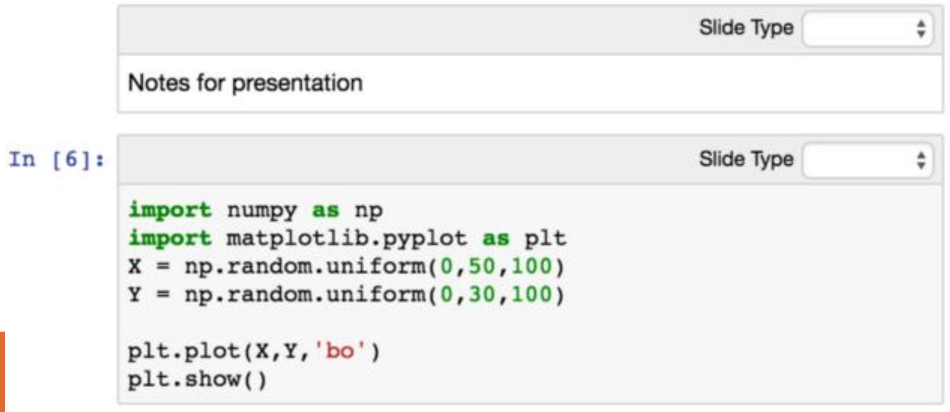


Dr. Rodolfo C. Raga Jr.
NEUST Workshop – Dec 21-23, 2021

The Colab Cells

Code Cells
Contents in this cell are treated as statements in a programming language of current kernel. Default kernel is Python. So, we can write Python statements in a code cell. When such cell is run, its result is displayed in an output cell. The output may be text, image, matplotlib plots or HTML tables. Code cells have rich text capability.

Markdown Cells
These cells contain text formatted using markdown language. All kinds of formatting features are available like making text bold and italic, displaying ordered or unordered list, rendering tabular contents etc. Markdown cells are especially useful to provide documentation to the computational process of the notebook.



Notes for presentation

In [6]:

```
import numpy as np
import matplotlib.pyplot as plt
X = np.random.uniform(0,50,100)
Y = np.random.uniform(0,30,100)

plt.plot(X,Y,'bo')
plt.show()
```

Markdown Cells

Code Cells

Dr. Rodolfo C. Raga Jr.
NEUST Workshop – Dec 21-23, 2021

Jupyter/Colab Notebook Demo

Dr. Rodolfo C. Raga Jr.

NEUST Workshop – Dec 21-23, 2021