Tools and Environments

Neural Networks for Machine Learning Applications

Spring 2023

Sakari Lukkarinen

Metropolia University of Applied Sciences

Contents

- Tools – Jupyter Notebook
- Environments
  - Google Colaboratory (Cloud service)
  - Kaggle online community (Cloud service)
  - Anaconda Distribution (local)
- More about Kaggle
- Jupyter Notebook
  - with Anaconda Distribution on own computer
- Case 0. Learning basics
- Summary

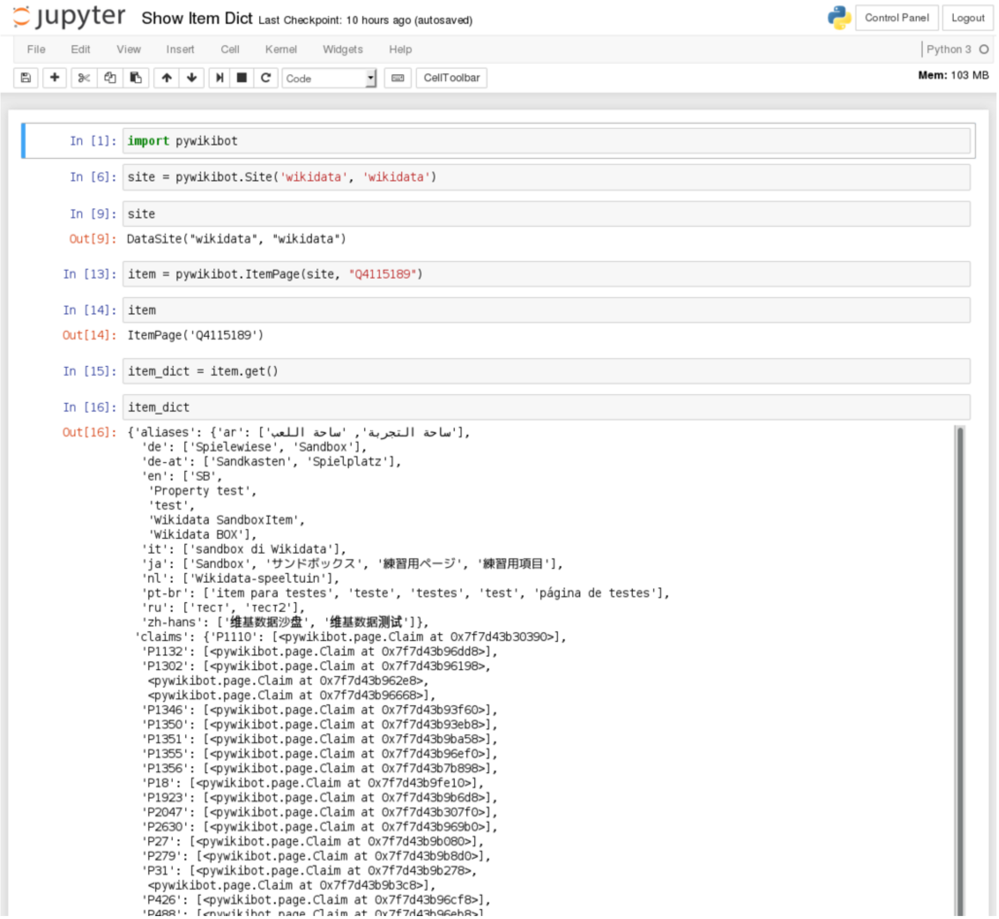
Tools – Jupyter Notebook

Jupyter Notebook

Jupyter Notebook is a web-based interactive computational environment for creating notebook documents. It is built using several open-source libraries.

Notebooks can be shared with others including Github, Kaggle, Google Colab, email, etc.

Jupyter Notebooks are used by millions of users worldwide.



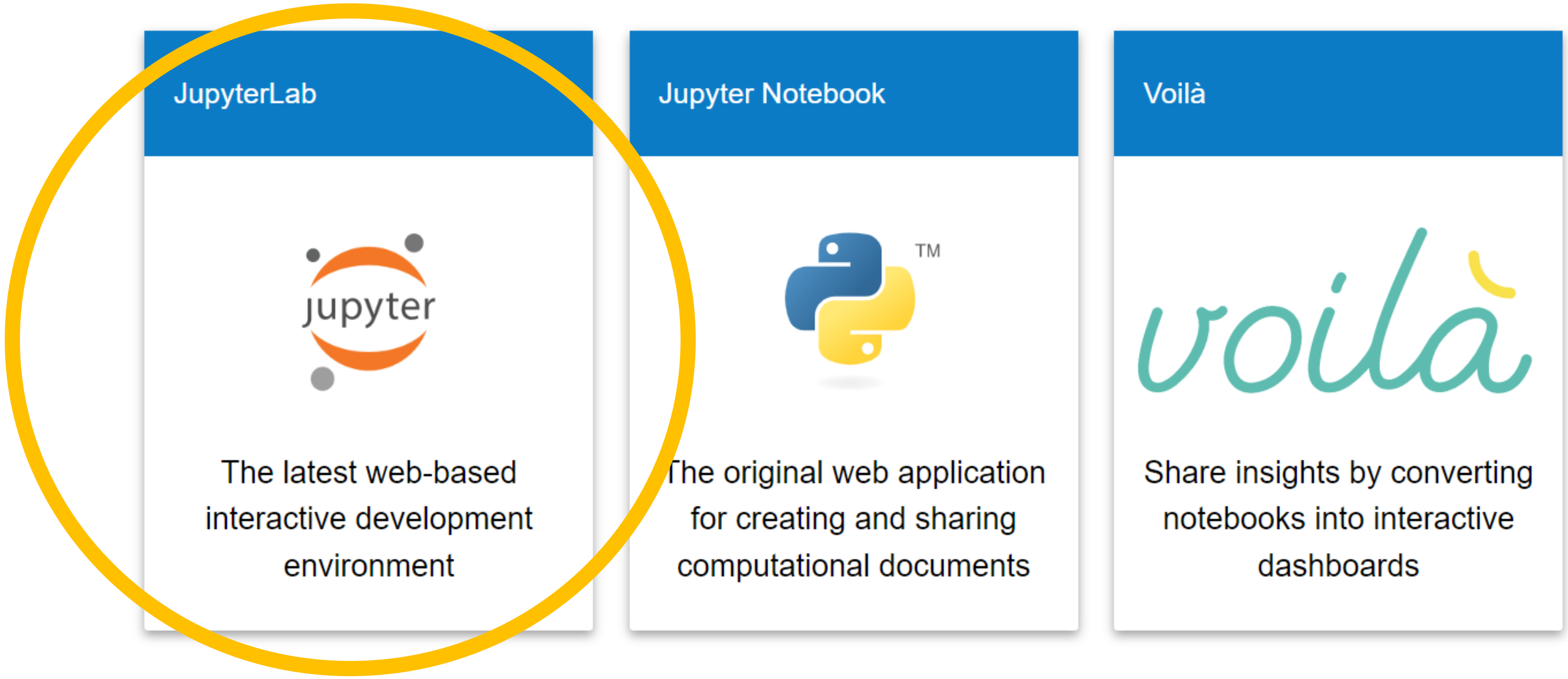
```
In [1]: import pywikibot

In [6]: site = pywikibot.Site('wikidata', 'wikidata')

In [9]: site
Out[9]: DataSite("wikidata", "wikidata")

In [13]: item = pywikibot.ItemPage(site, "Q4115189")

In [14]: item
Out[14]: ItemPage('Q4115189')

In [15]: item_dict = item.get()

In [16]: item_dict
Out[16]: {'aliases': {'ar': ['ساحة اللعب' ,'ساحة التجربة'],
           'de': ['Spielewiese', 'Sandbox'],
           'de-at': ['Sandkasten', 'Spielplatz'],
           'en': ['SB',
            'Property test',
            'test',
            'Wikidata SandboxItem',
            'Wikidata BOX'],
           'it': ['sandbox di Wikidata'],
           'ja': ['Sandbox', 'サンドボックス', '練習用ページ', '練習用項目'],
           'nl': ['Wikidata-speeltuin'],
           'pt-br': ['item para testes', 'teste', 'testes', 'test', 'página de testes'],
           'ru': ['тест', 'тест2'],
           'zh-hans': ['维基数据沙盘', '维基数据测试']},
          'claims': {'P1110': [<pywikibot.page.Claim at 0x7f7d43b30390>],
           'P1132': [<pywikibot.page.Claim at 0x7f7d43b96dd8>],
           'P1302': [<pywikibot.page.Claim at 0x7f7d43b96198>,
            <pywikibot.page.Claim at 0x7f7d43b962e8>,
            <pywikibot.page.Claim at 0x7f7d43b96668>],
           'P1346': [<pywikibot.page.Claim at 0x7f7d43b93f60>],
           'P1350': [<pywikibot.page.Claim at 0x7f7d43b93eb8>],
           'P1351': [<pywikibot.page.Claim at 0x7f7d43b9ba58>],
           'P1355': [<pywikibot.page.Claim at 0x7f7d43b96ef0>],
           'P1356': [<pywikibot.page.Claim at 0x7f7d43b7b898>],
           'P18': [<pywikibot.page.Claim at 0x7f7d43b9fe10>],
           'P1923': [<pywikibot.page.Claim at 0x7f7d43b9b6d8>],
           'P2047': [<pywikibot.page.Claim at 0x7f7d43b307f0>],
           'P2630': [<pywikibot.page.Claim at 0x7f7d43b969b0>],
           'P27': [<pywikibot.page.Claim at 0x7f7d43b9b080>],
           'P279': [<pywikibot.page.Claim at 0x7f7d43b9b8d0>],
           'P31': [<pywikibot.page.Claim at 0x7f7d43b9b278>,
            <pywikibot.page.Claim at 0x7f7d43b9b3c8>],
           'P426': [<pywikibot.page.Claim at 0x7f7d43b96cf8>],
           'P488': [<pywikibot.page.Claim at 0x7f7d43b96eb8>]
```

[Project Jupyter | Home](#)

[Project Jupyter - Wikipedia](#)

Try Jupyter

You can try Jupyter without installing anything.
Try the **JupyterLab** in the Project Jupyter.



JupyterLab

The latest web-based interactive development environment

Jupyter Notebook

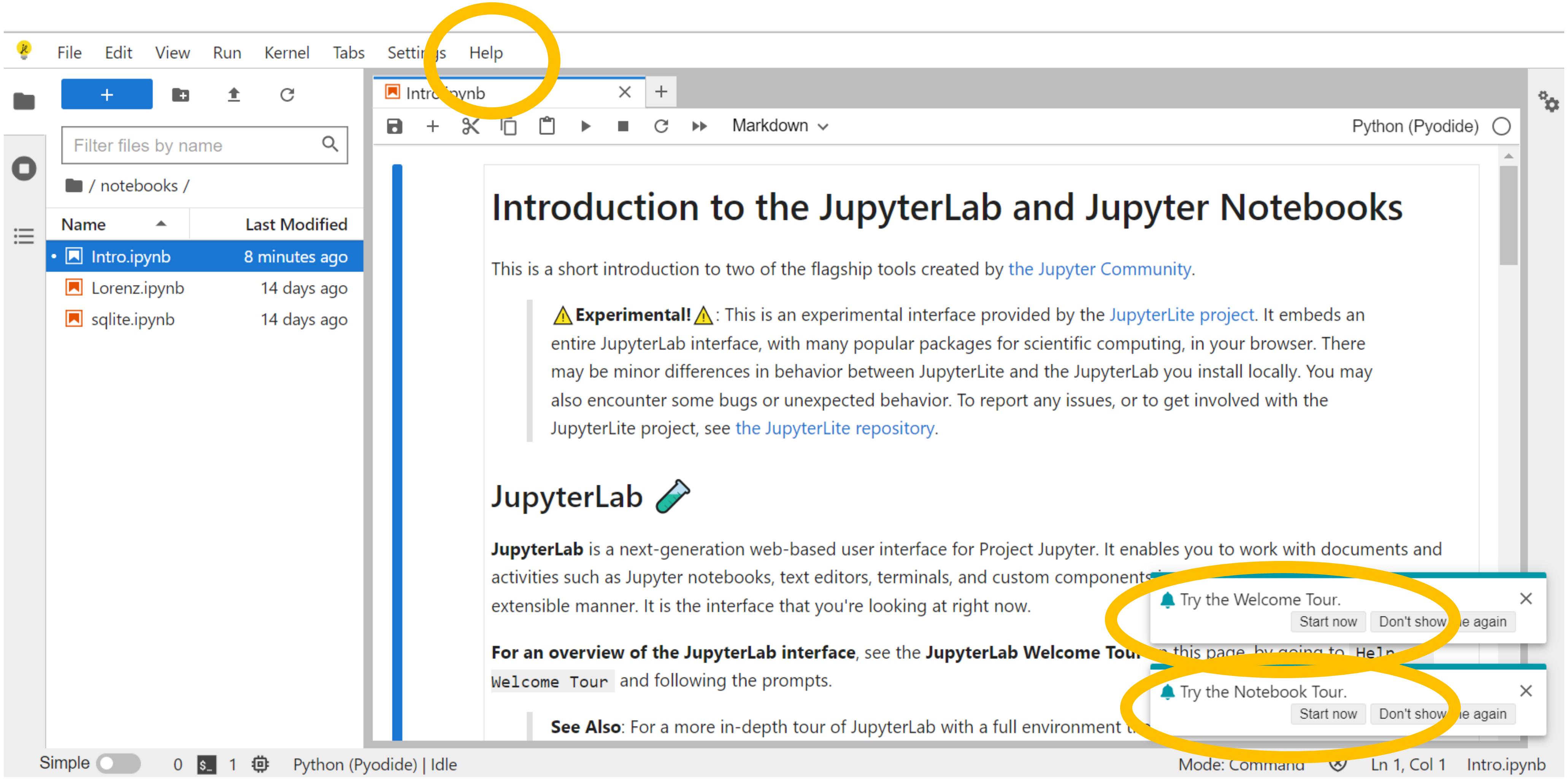
The original web application for creating and sharing computational documents

Voilà

Share insights by converting notebooks into interactive dashboards

Project Jupyter | Try Jupyter

Notebook and Welcome Tours



Name	Last Modified
Intro.ipynb	8 minutes ago
Lorenz.ipynb	14 days ago
sqlite.ipynb	14 days ago

Introduction to the JupyterLab and Jupyter Notebooks

This is a short introduction to two of the flagship tools created by the Jupyter Community.

> ⚠ **Experimental!** ⚠: This is an experimental interface provided by the JupyterLite project. It embeds an entire JupyterLab interface, with many popular packages for scientific computing, in your browser. There may be minor differences in behavior between JupyterLite and the JupyterLab you install locally. You may also encounter some bugs or unexpected behavior. To report any issues, or to get involved with the JupyterLite project, see the JupyterLite repository.

JupyterLab 🧪

JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components extensible manner. It is the interface that you're looking at right now.

For an overview of the JupyterLab interface, see the **JupyterLab Welcome Tou** this page, by going to `Help` `Welcome Tour` and following the prompts.

> **See Also**: For a more in-depth tour of JupyterLab with a full environment

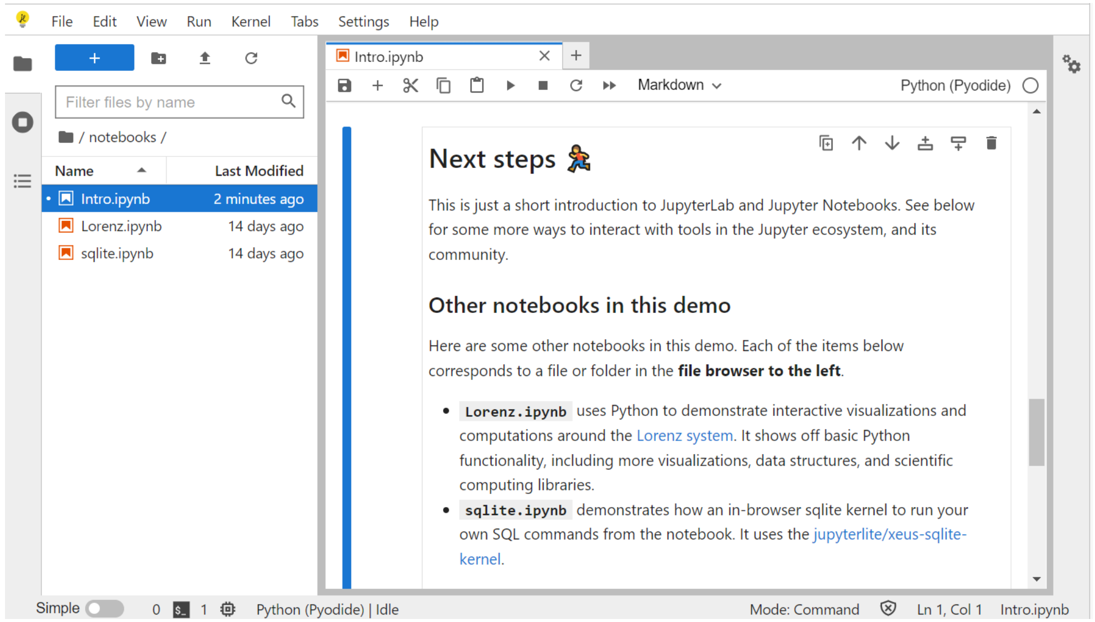
Try the Welcome Tour.
Start now | Don't show me again

Try the Notebook Tour.
Start now | Don't show me again

Intro (jupyter.org)

Practice

- Try JupyterLab.
- Run Notebook Tour and Welcome Tour.
- Scroll up and down the Intro Notebook.
- Get familiar with
  - Main work area
  - Toolbars
  - Menus
  - Status bar, and
  - Command palette



Name	Last Modified
Intro.ipynb	2 minutes ago
Lorenz.ipynb	14 days ago
sqlite.ipynb	14 days ago

Next steps

This is just a short introduction to JupyterLab and Jupyter Notebooks. See below for some more ways to interact with tools in the Jupyter ecosystem, and its community.

Other notebooks in this demo

Here are some other notebooks in this demo. Each of the items below corresponds to a file or folder in the **file browser to the left**.

- `Lorenz.ipynb` uses Python to demonstrate interactive visualizations and computations around the Lorenz system. It shows off basic Python functionality, including more visualizations, data structures, and scientific computing libraries.
- `sqlite.ipynb` demonstrates how an in-browser sqlite kernel to run your own SQL commands from the notebook. It uses the jupyterlite/xeus-sqlite-kernel.

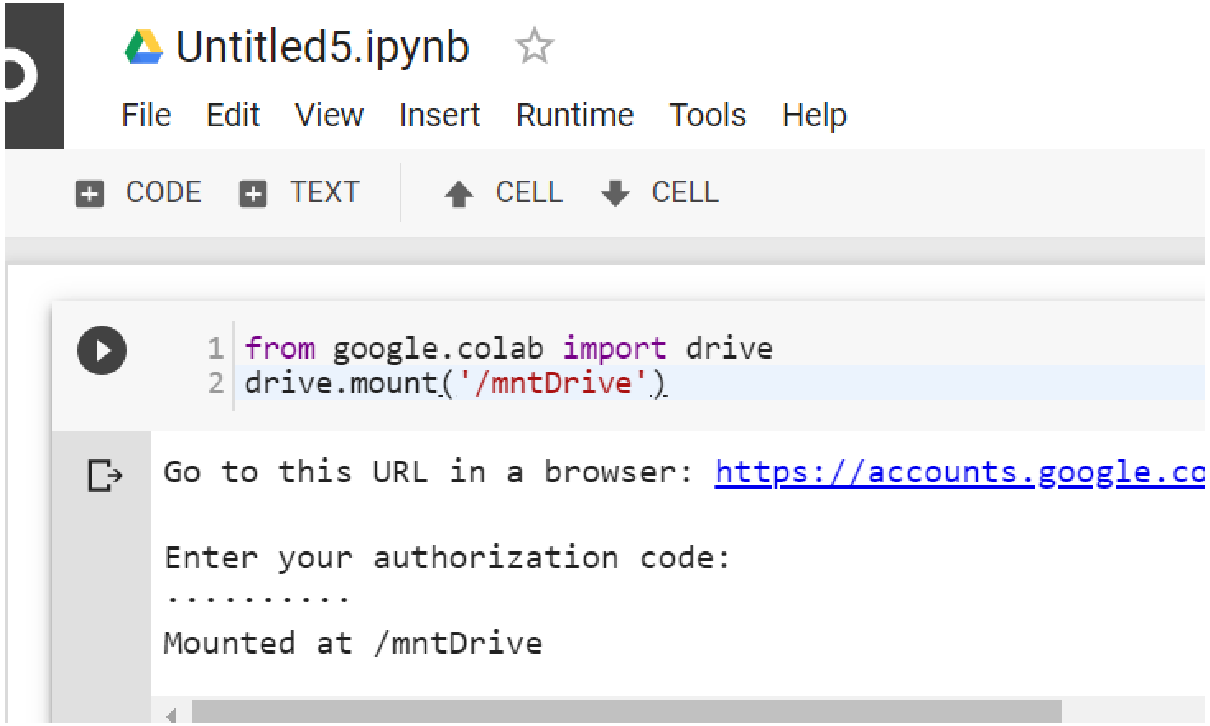
Environments

Google Colab, Kaggle and Anaconda Distribution

Environments

Major cloud computing providers have adopted the Jupyter Notebook as a frontend. Examples include:

- **Google's Colaboratory**
- **Kaggle** online community
- Google Cloud
- Amazon Sagemaker
- Microsoft's Azure



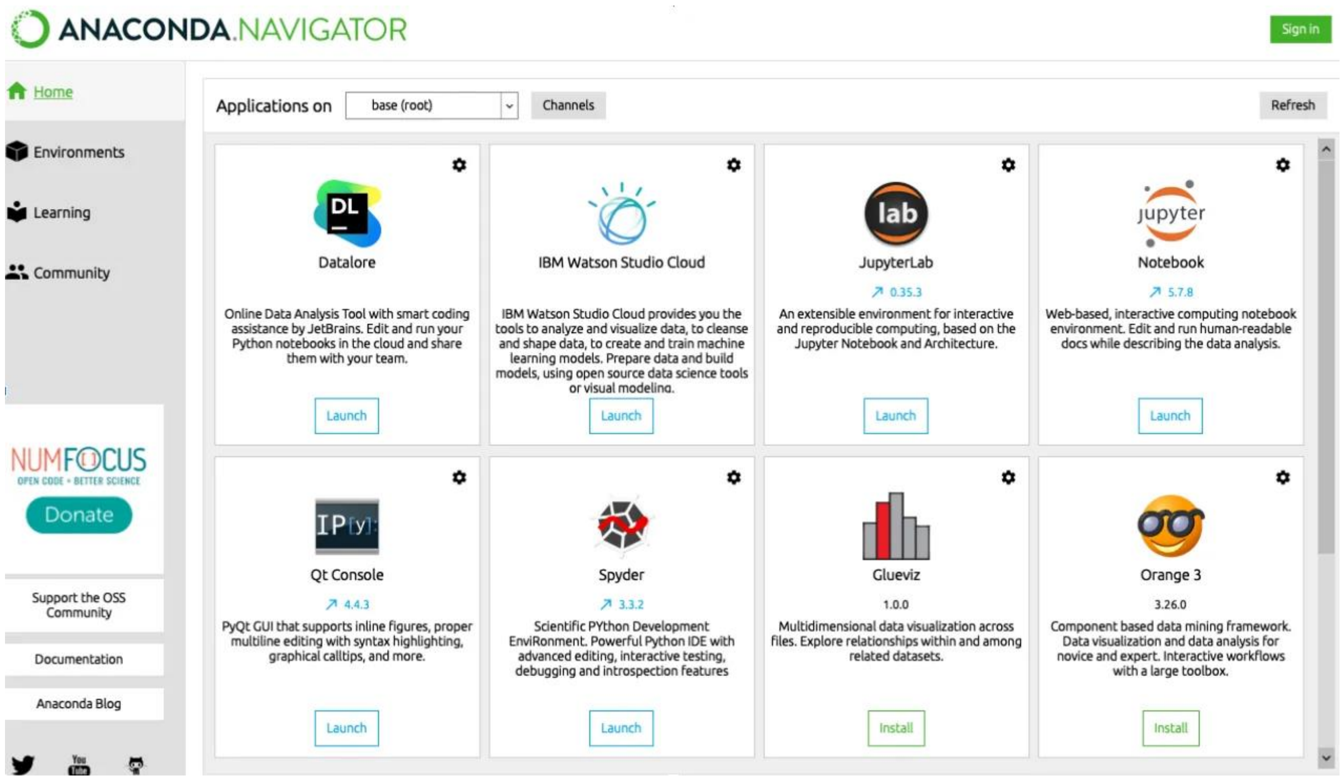
```python
from google.colab import drive
drive.mount('/mntDrive')
```

```
Go to this URL in a browser: https://accounts.google.co

Enter your authorization code:
··········
Mounted at /mntDrive
```

Environments

You can also install the environment locally and develop Notebooks on your laptop. The most popular local environments are:

- **Anaconda distribution**
- Visual Studio Code
- JetBrain PyCharm



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.

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.

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

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

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

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

Glueviz: Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Environments – this course

During this course we recommend to use the following environments:

- Google's colaboratory (Cloud)

  + : Easy to use, no installation required

  - : Requires some extra code for using the case datasets

- Kaggle (Cloud)

  + : No installation required, all case datasets available

  - : Slightly more complex to use than Google colab

- Anaconda Distribution (Local)

  + : Full control of the environment, no latencies in running the code

  - : Needs to download and extract the datasets, usually no GPU/TPU acceleration available

Google Colaboratory

Using Metropolia’s Google Drive

G Suite Marketplace

搜索“apps”

Colaboratory

colab-team

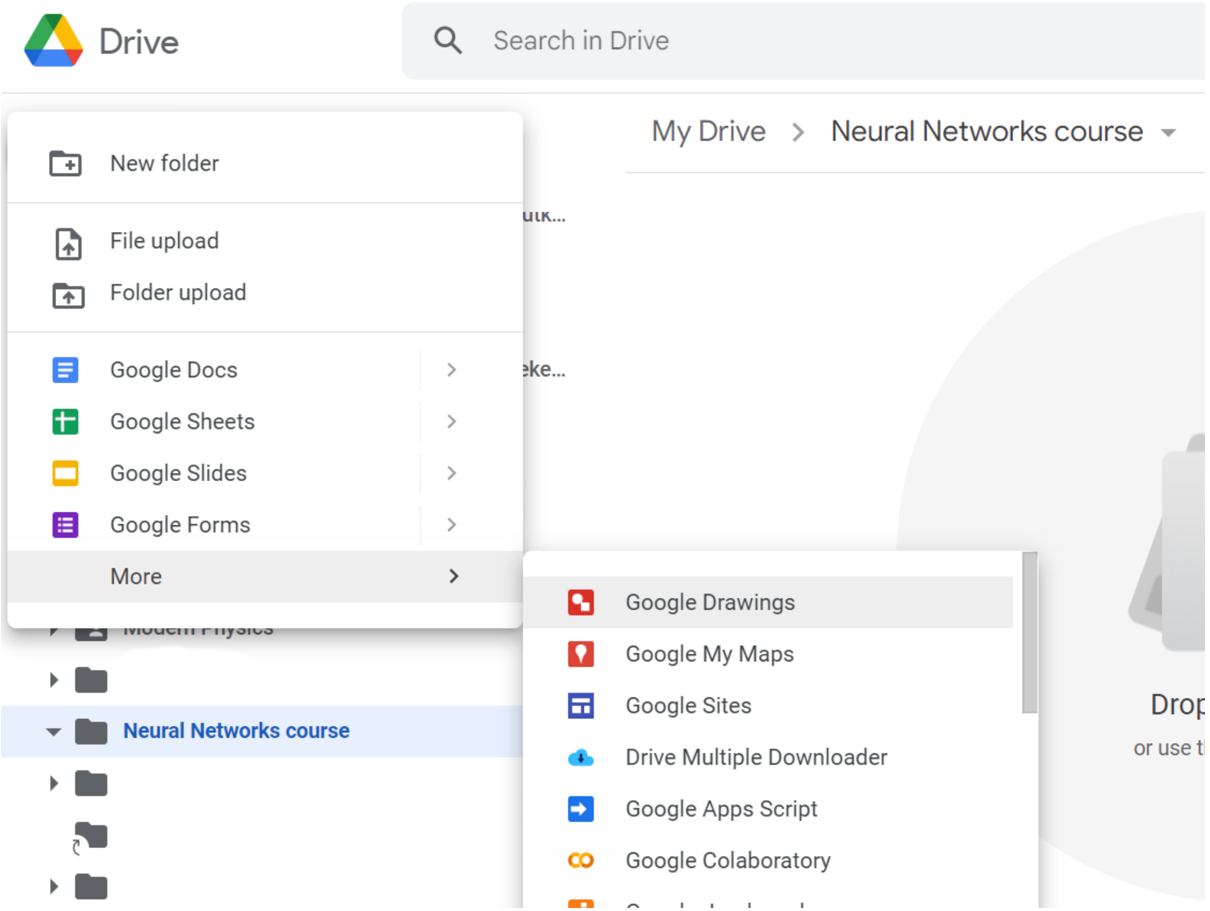
云端硬盘插件

Google Colaboratory已与Go

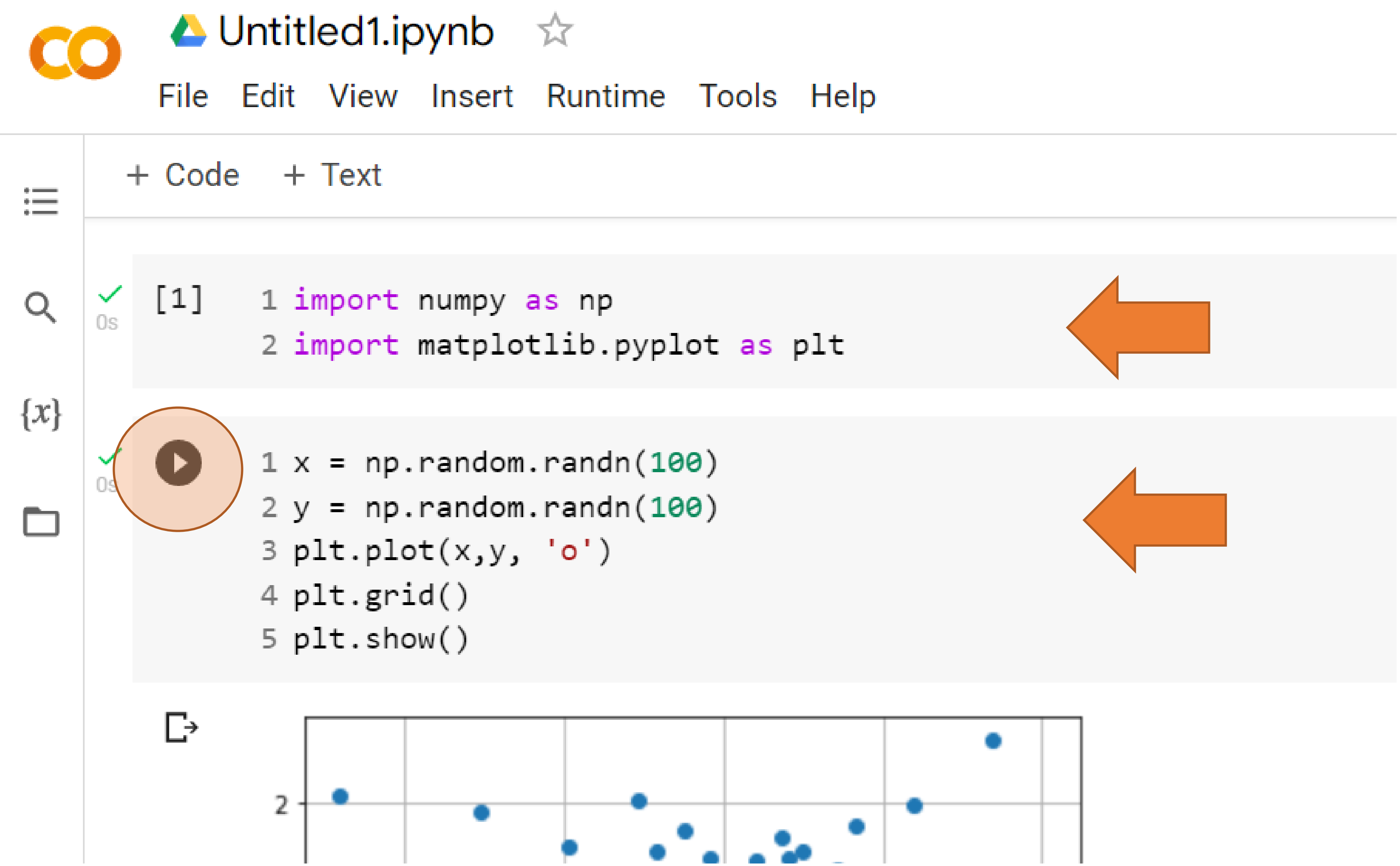
☑ 将Google Colaboratory设为它能

Google collaboratory with Metropolia’s account

1. Open Google Drive with Metropolia’s account
   - See: Cloud services - IT Services – Metropolia
2. Create a folder **Neural Networks course**
3. Select **New > More > Google Colaboratory**



Practise - Write Code and Run

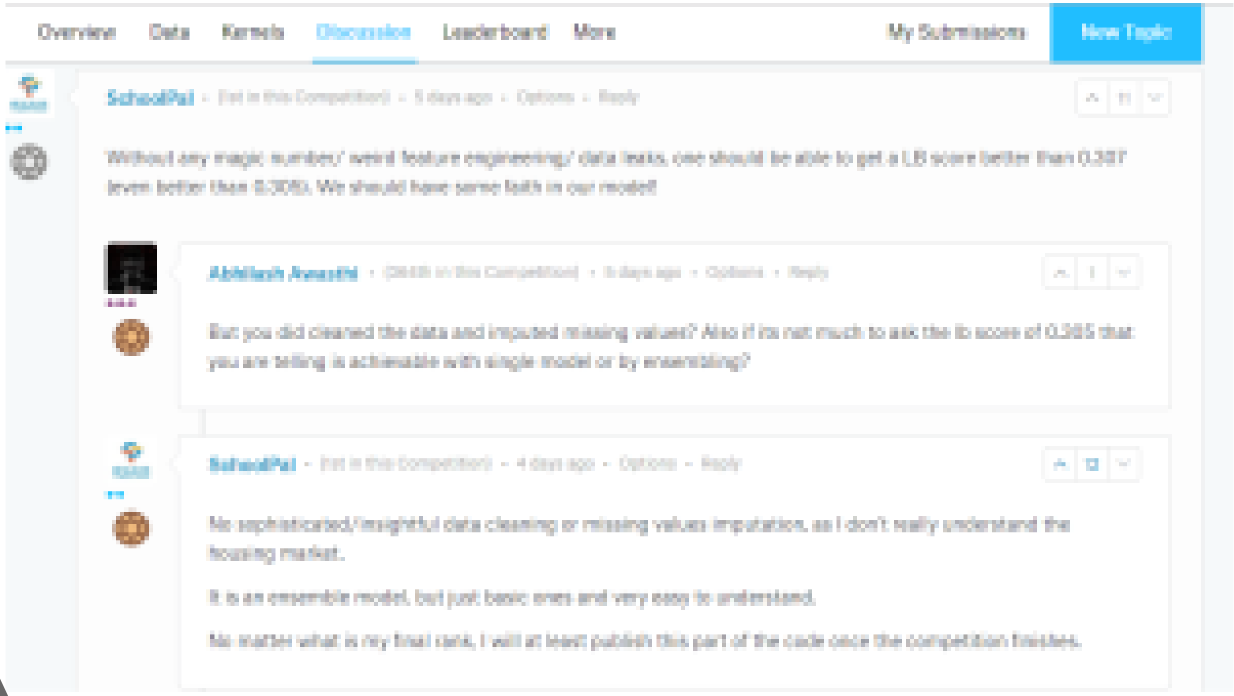


```python
import numpy as np
import matplotlib.pyplot as plt
```

```python
x = np.random.randn(100)
y = np.random.randn(100)
plt.plot(x,y, 'o')
plt.grid()
plt.show()
```

Using Metropolia’s Google account

Kaggle



SchoolPal · (1st in this Competition) · 5 days ago · Options · Reply

Without any magic number/ weird feature engineering/ data leaks, one should be able to get a LB score better than 0.307 (even better than 0.306). We should have some faith in our model!

Abhilash Awasthi · (2648th in this Competition) · 5 days ago · Options · Reply

But you did cleaned the data and imputed missing values? Also if its not much to ask the lb score of 0.305 that you are telling is achievable with single model or by ensembling?

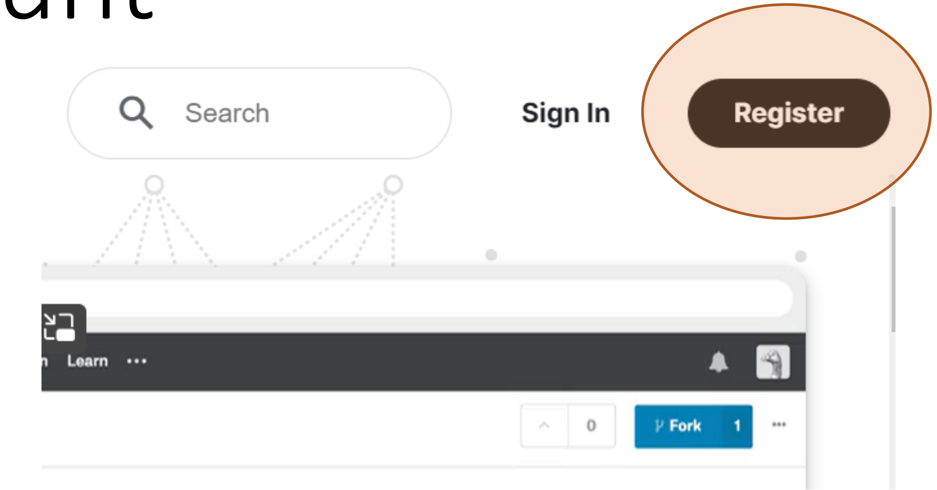
SchoolPal · (1st in this Competition) · 4 days ago · Options · Reply

No sophisticated/insightful data cleaning or missing values imputation, as I don't really understand the housing market.

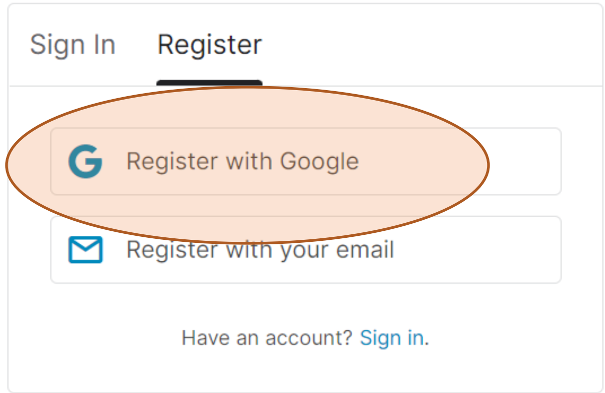
It is an ensemble model, but just basic ones and very easy to understand.

No matter what is my final rank, I will at least publish this part of the code once the competition finishes.

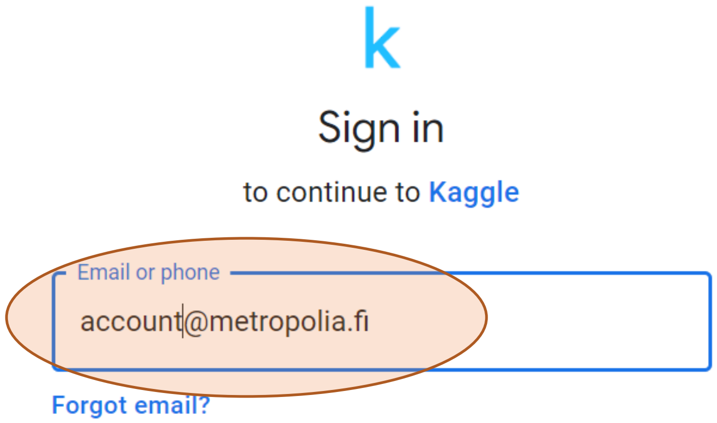
Kaggle with Metropolia's account



1. Open www.kaggle.com
2. Click **Register**.
3. **Register with Google**.
4. Give email: account@metropolia.fi
5. Enter your Username and Password in Metropolia's login.

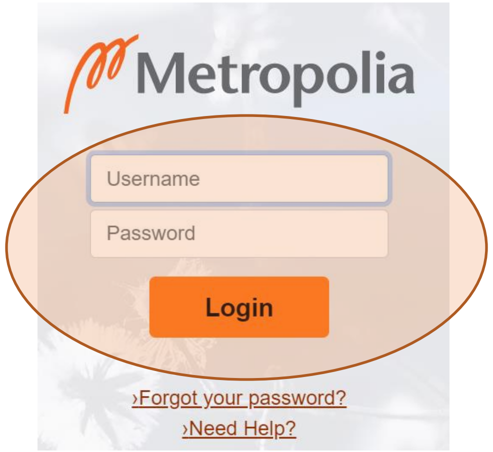


Have an account? Sign in.



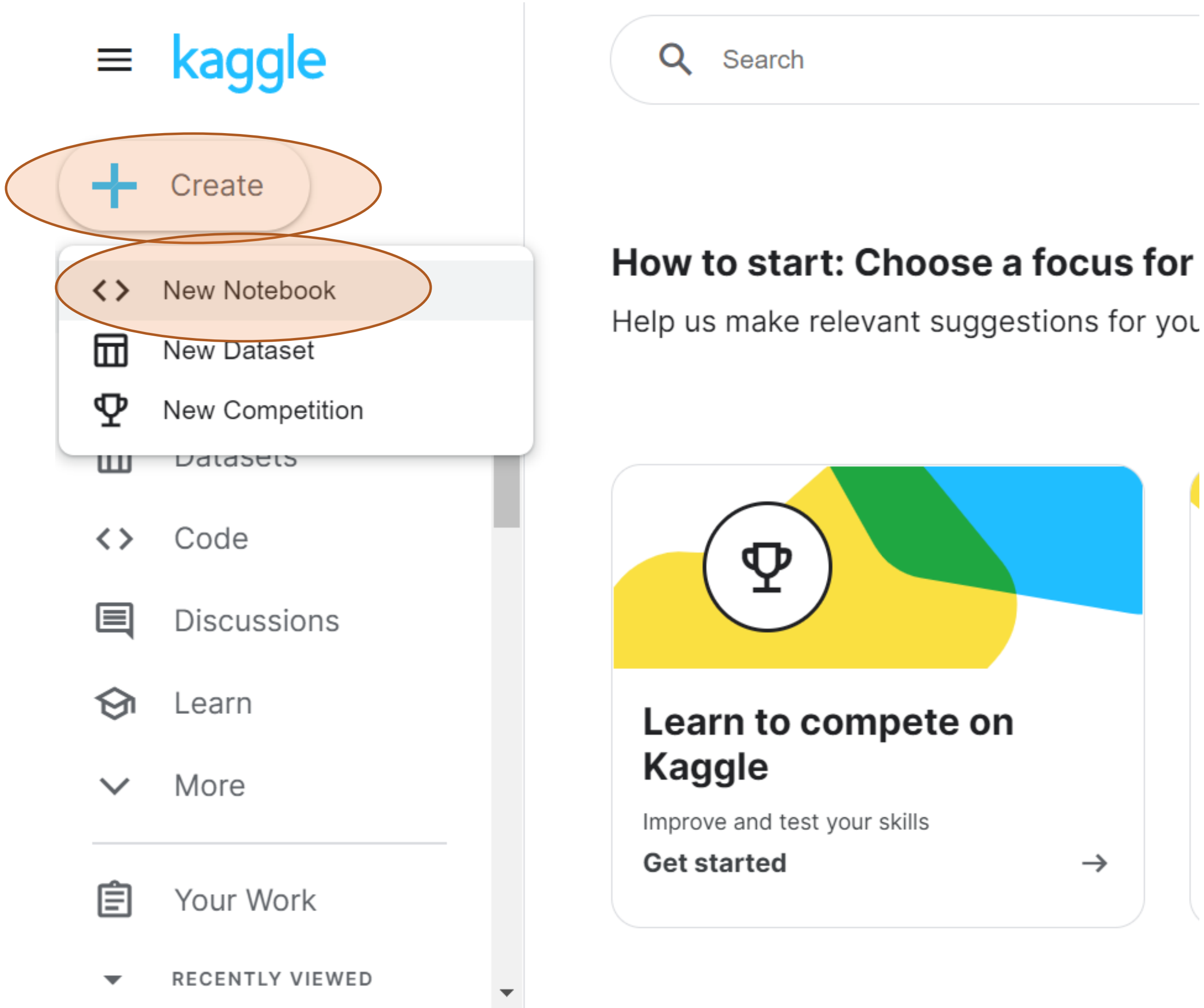
to continue to Kaggle

Email or phone: account@metropolia.fi

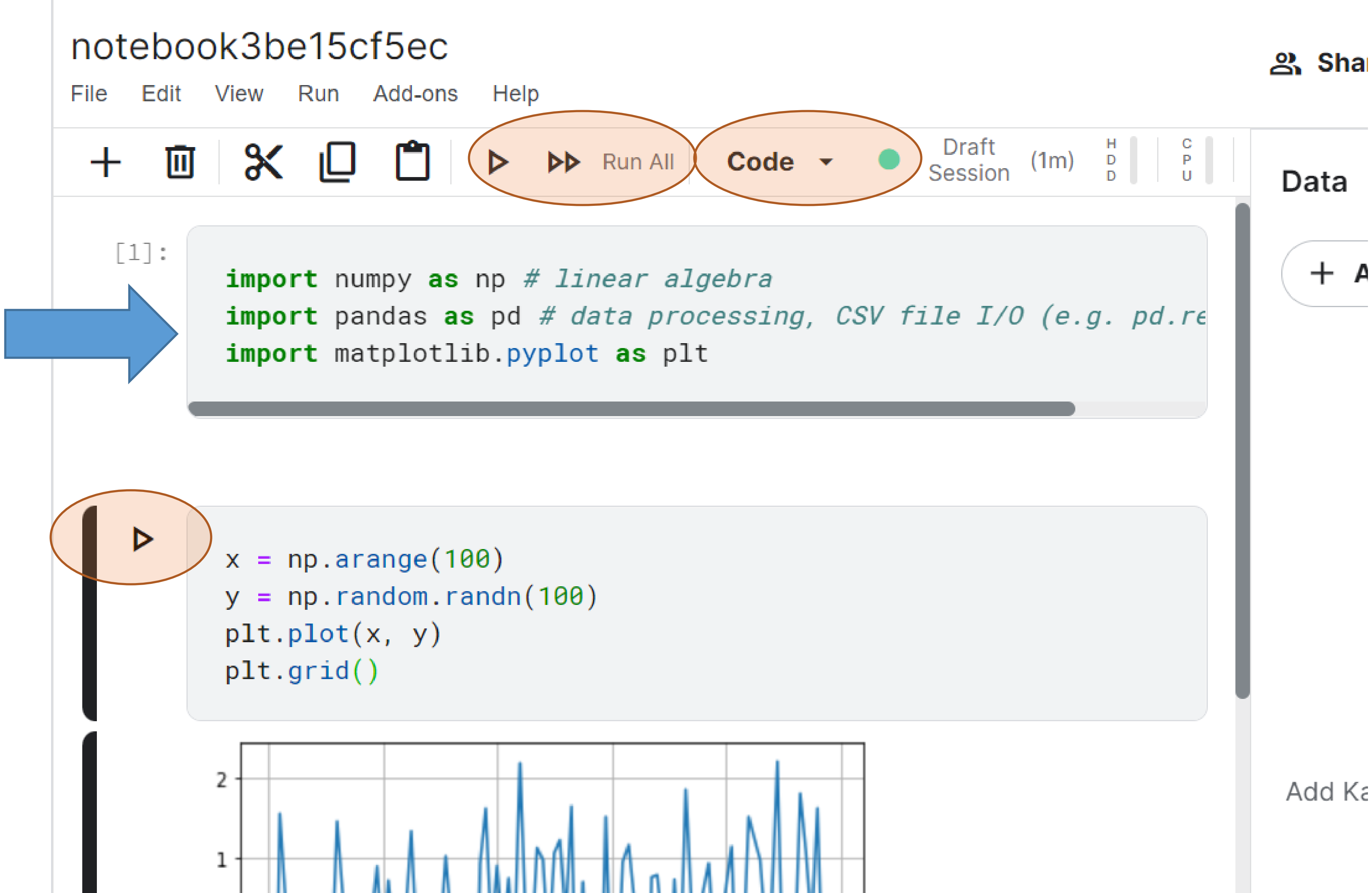


Practise - Create New Notebook

1. Click Create.
2. Select < > New Notebook.


kaggle
Search
Create
New Notebook
New Dataset
New Competition
Datasets
Code
Discussions
Learn
More
Your Work
RECENTLY VIEWED
How to start: Choose a focus for
Help us make relevant suggestions for you
Learn to compete on Kaggle
Improve and test your skills
Get started

Practise – Write Code and Run



```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.re
import matplotlib.pyplot as plt
```

```
x = np.arange(100)
y = np.random.randn(100)
plt.plot(x, y)
plt.grid()
```

Anaconda Distribution

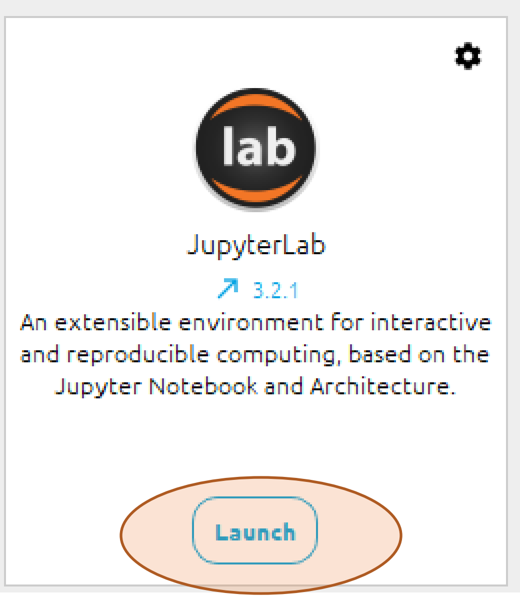
On your own laptop

Install Anaconda Distribution

1. Open Anaconda | Anaconda Distribution.
2. Select Download.
3. Save and Run the Installation package.

   NOTE: It is recommended that you install for **Just Me**.

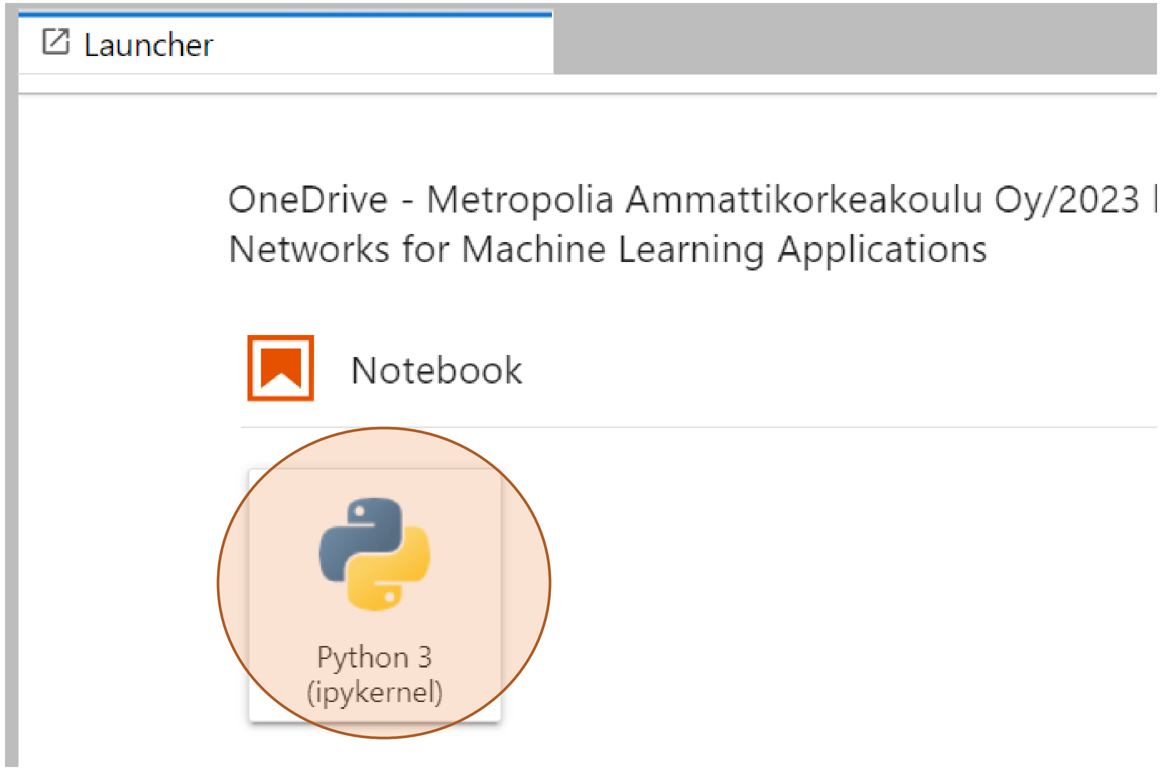
   See: Installation — Anaconda documentation
4. Verify your installation
   I. Windows: Click **Start**, search for **Anaconda Navigator**, and click open.
   II. In Anaconda Navigator, select **JupyterLab**, and **Launch**



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

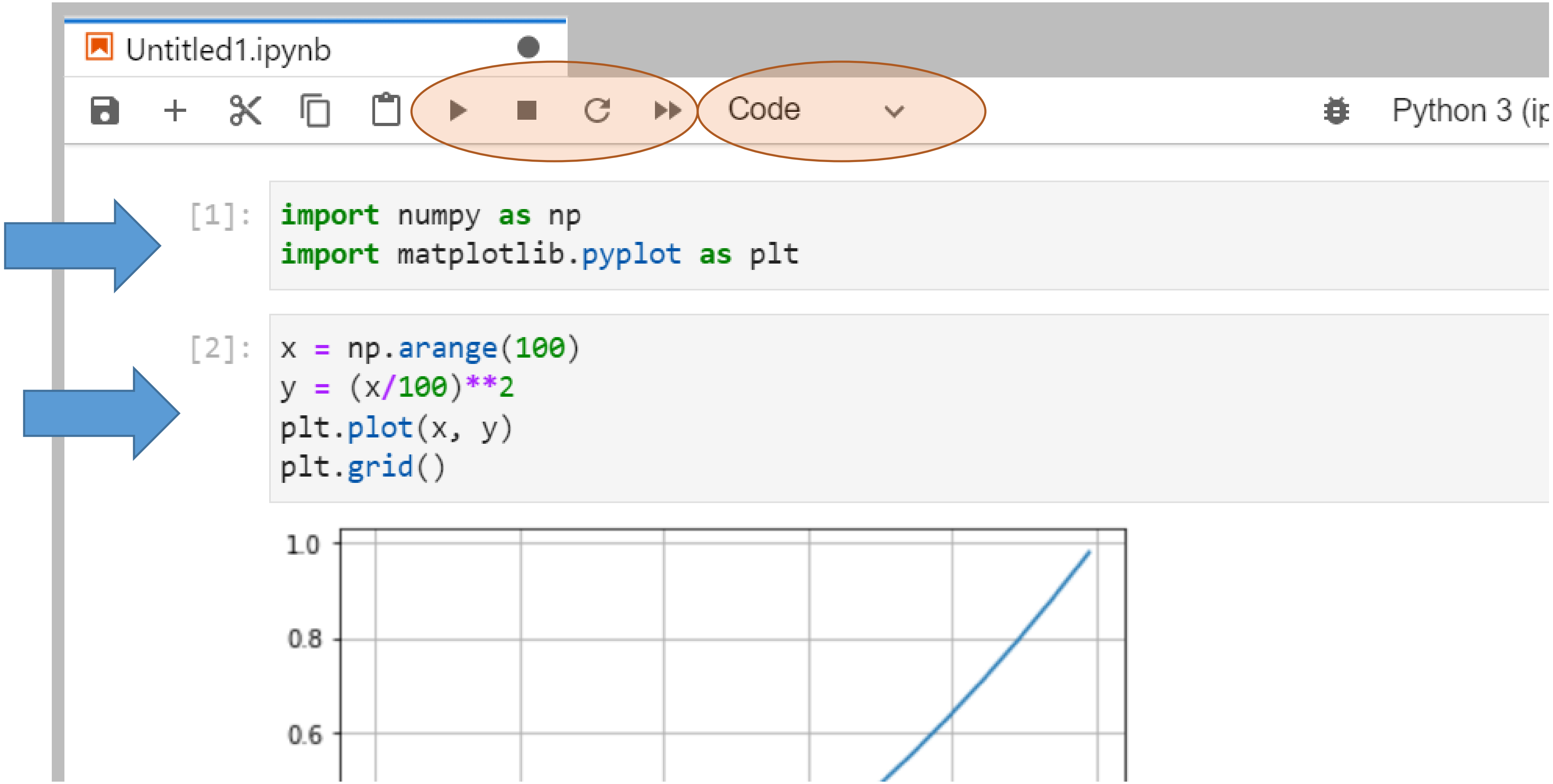
Practise – Create New Notebook

1. Create a local folder for your Notebooks
   - For example: Documents > Neural Networks course
2. Open your local folder in JupyterLab
3. Click Notebook > Python 3 (ipykernel) in Launcher.



OneDrive - Metropolia Ammattikorkeakoulu Oy/2023 I
Networks for Machine Learning Applications

Practise – Write Code and Run

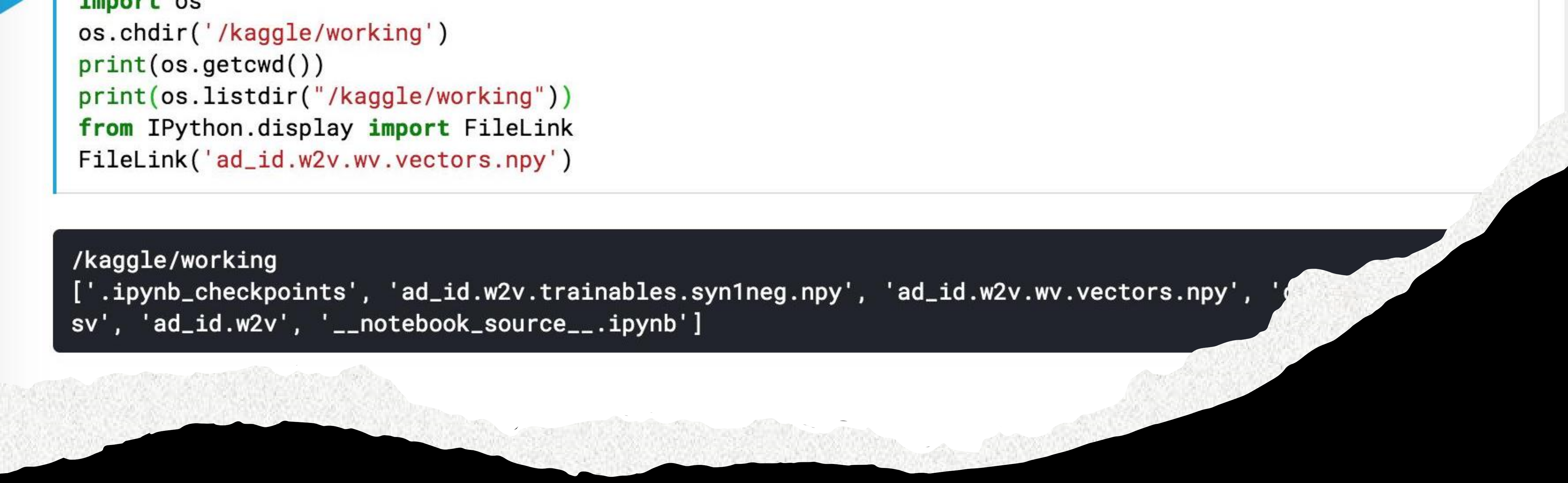


```
[1]: import numpy as np
     import matplotlib.pyplot as plt
```

```
[2]: x = np.arange(100)
     y = (x/100)**2
     plt.plot(x, y)
     plt.grid()
```

Learn more

- Intro to Programming Tutorials (kaggle.com)
  - Python Tutorials (kaggle.com)
  - Data Visualization Tutorials (kaggle.com)
  - Pandas Tutorials (kaggle.com)
  - Intro to Machine Learning Tutorials (kaggle.com)
- Intro to Deep Learning Tutorials (kaggle.com)
- Python Tutorial (w3schools.com)



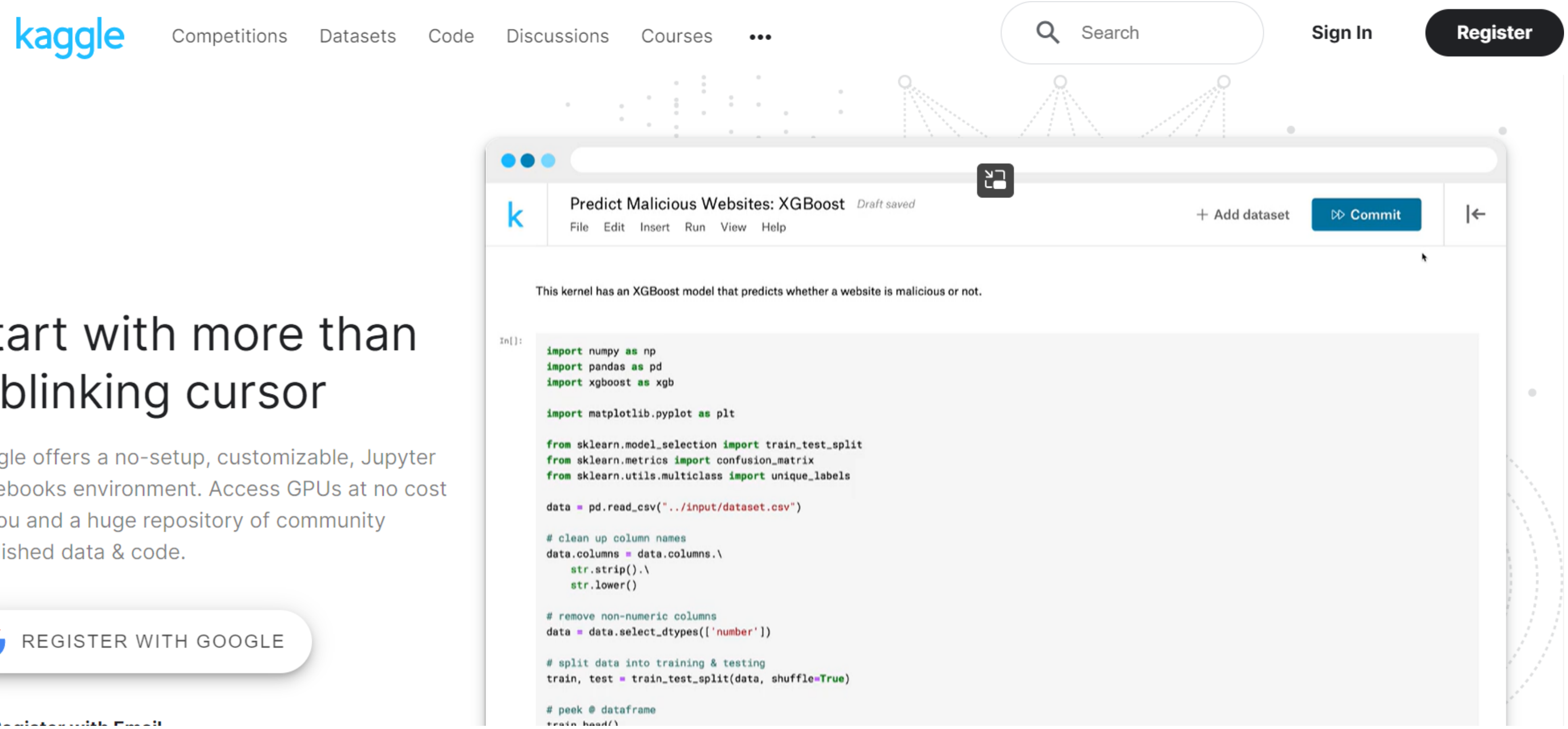
```
import os
os.chdir('/kaggle/working')
print(os.getcwd())
print(os.listdir("/kaggle/working"))
from IPython.display import FileLink
FileLink('ad_id.w2v.wv.vectors.npy')
```

```
/kaggle/working
['.ipynb_checkpoints', 'ad_id.w2v.trainables.syn1neg.npy', 'ad_id.w2v.wv.vectors.npy', 'c
sv', 'ad_id.w2v', '__notebook_source__.ipynb']
```

More about Kaggle

Kaggle, Notebooks, and Python

Kaggle – online community of data scientists and machine learning pracititones



Start with more than a blinking cursor

Kaggle offers a no-setup, customizable, Jupyter Notebooks environment. Access GPUs at no cost to you and a huge repository of community published data & code.

This kernel has an XGBoost model that predicts whether a website is malicious or not.

```
import numpy as np
import pandas as pd
import xgboost as xgb

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels

data = pd.read_csv("../input/dataset.csv")

# clean up column names
data.columns = data.columns.\
    str.strip().\
    str.lower()

# remove non-numeric columns
data = data.select_dtypes(['number'])

# split data into training & testing
train, test = train_test_split(data, shuffle=True)

# peek @ dataframe
train.head()
```

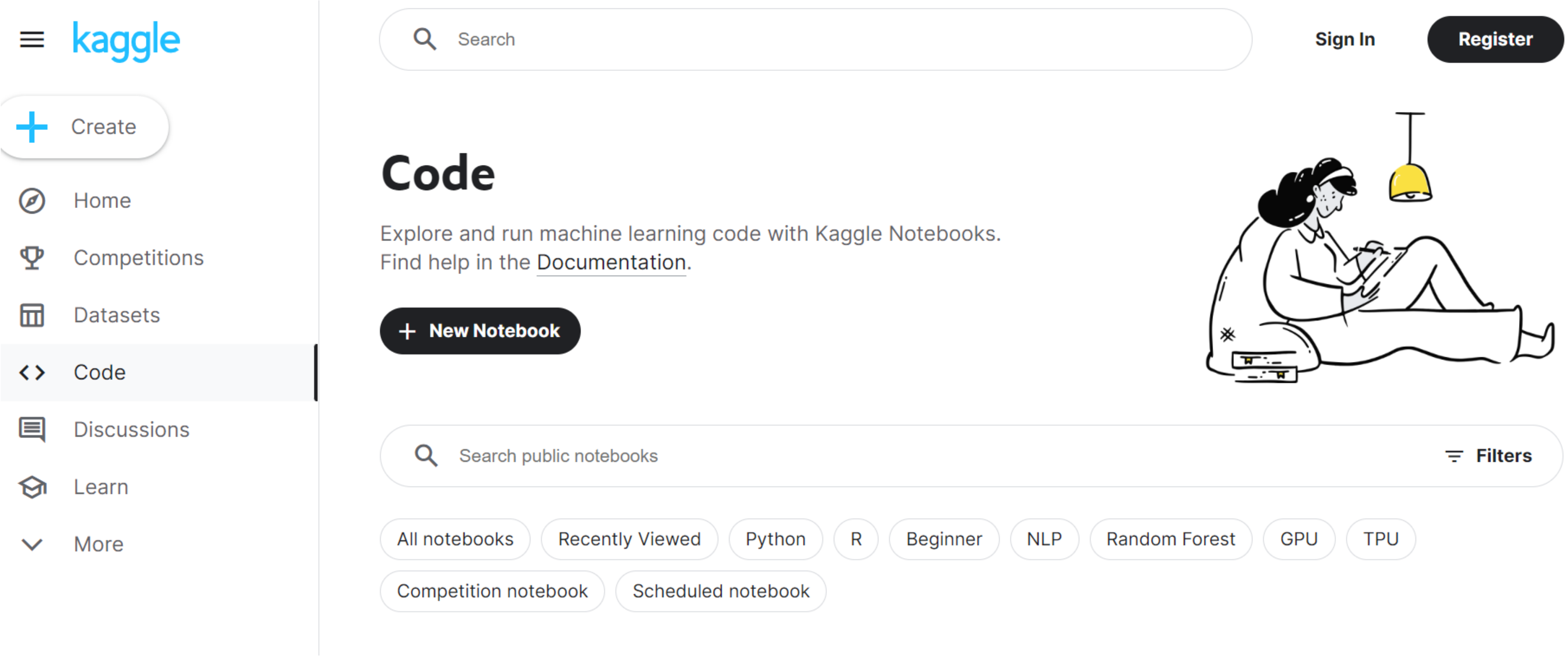
www.kaggle.com

Kaggle

- First launched in 2010
- Now a subsidiary of Google LLC (acquired in March 2017)
- Over 8 million users from 190+ countries
- Hundreds of machine-learning competitions were run on Kaggle since the company was founded
- Allows users
    - To find and publish data sets
    - Explore and build models in a web-based data-science environment
    - Work with other data scientists and machine learning engineers
    - Enter competitions to solve data science challenges

Kaggle - Wikipedia

Kaggle Notebooks



Code

Explore and run machine learning code with Kaggle Notebooks.
Find help in the Documentation.

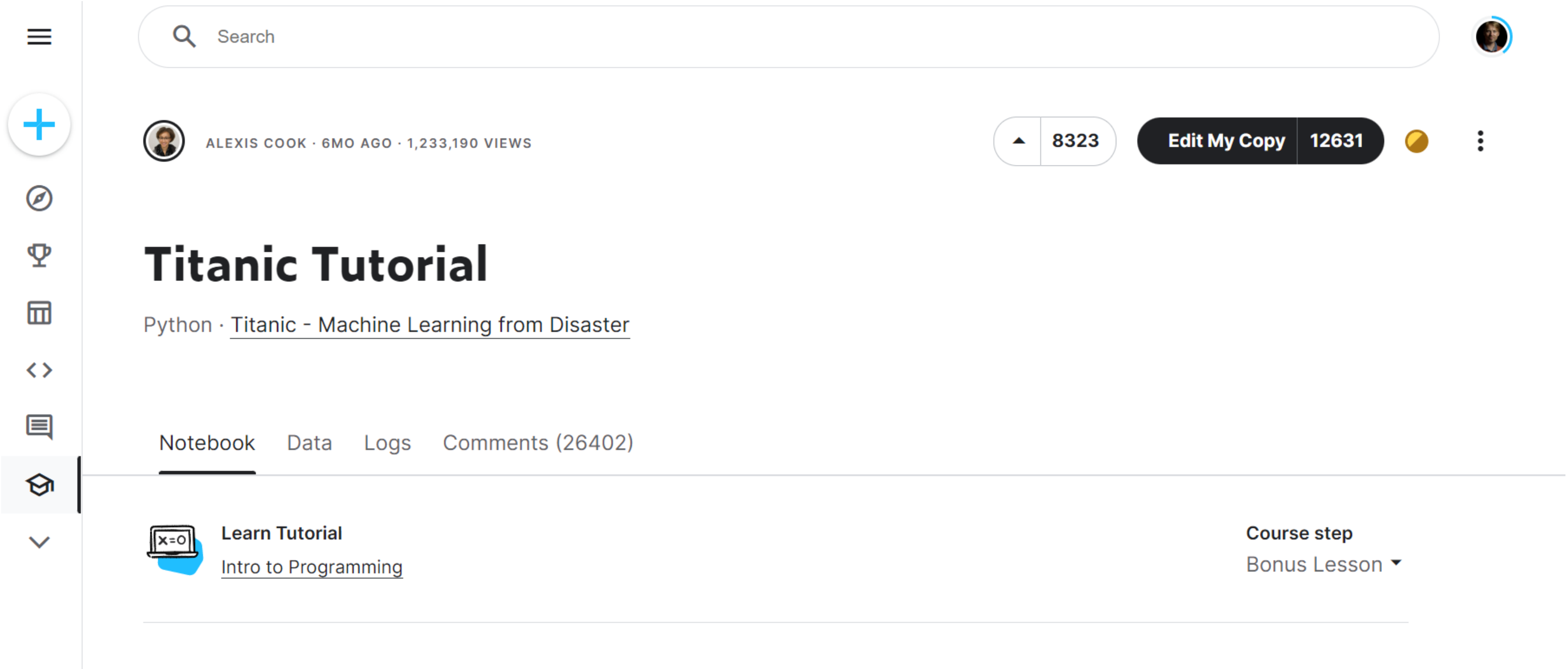
Run Data Science & Machine Learning Code Online | Kaggle

Kaggle Notebooks

- Notebooks consists of a sequence of cells.
- Each cell is formatted in either
  - Markdown (for writing text) or
  - in a programming language of your choice (for writing code)
- To start a notebook, click on ”Create Notebook”, and select ”Notebook”.
- Notebooks may be written in either R or Python.

Notebooks Documentation | Kaggle

Example – Titanic tutorial



Search

ALEXIS COOK · 6MO AGO · 1,233,190 VIEWS

Titanic Tutorial

Python · Titanic - Machine Learning from Disaster

Learn Tutorial
Intro to Programming

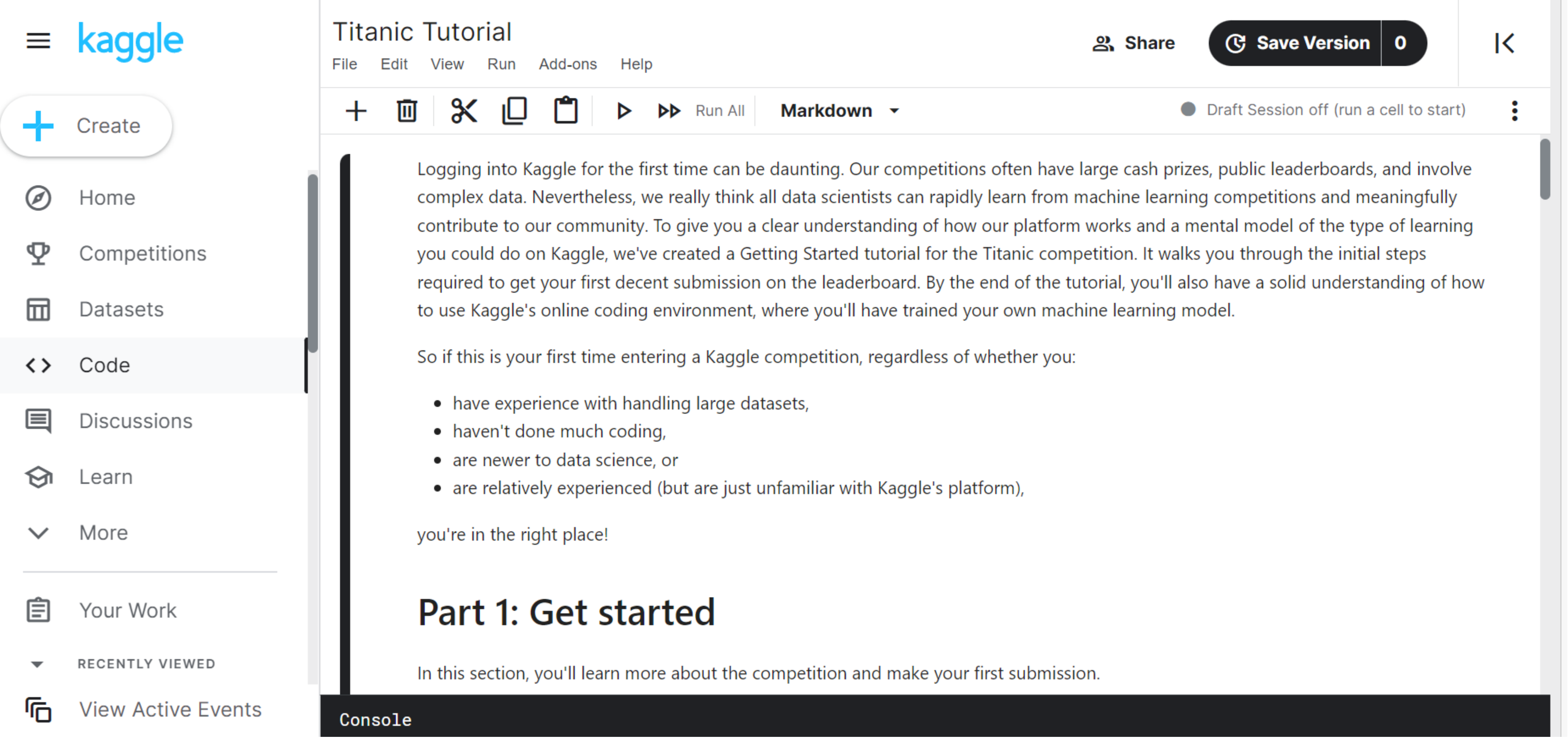
Course step
Bonus Lesson

Titanic Tutorial | Kaggle

Make a copy of a Notebook at Kaggle

1. Click: Edit My Copy.
    - Titanic Tutorial | Kaggle
2. Scroll down the Notebook and study how the Notebook is constructed
3. Study the Data & Settings sidebar on the left.

Example – Edit My Copy



Logging into Kaggle for the first time can be daunting. Our competitions often have large cash prizes, public leaderboards, and involve complex data. Nevertheless, we really think all data scientists can rapidly learn from machine learning competitions and meaningfully contribute to our community. To give you a clear understanding of how our platform works and a mental model of the type of learning you could do on Kaggle, we've created a Getting Started tutorial for the Titanic competition. It walks you through the initial steps required to get your first decent submission on the leaderboard. By the end of the tutorial, you'll also have a solid understanding of how to use Kaggle's online coding environment, where you'll have trained your own machine learning model.

So if this is your first time entering a Kaggle competition, regardless of whether you:

- have experience with handling large datasets,
- haven't done much coding,
- are newer to data science, or
- are relatively experienced (but are just unfamiliar with Kaggle's platform),

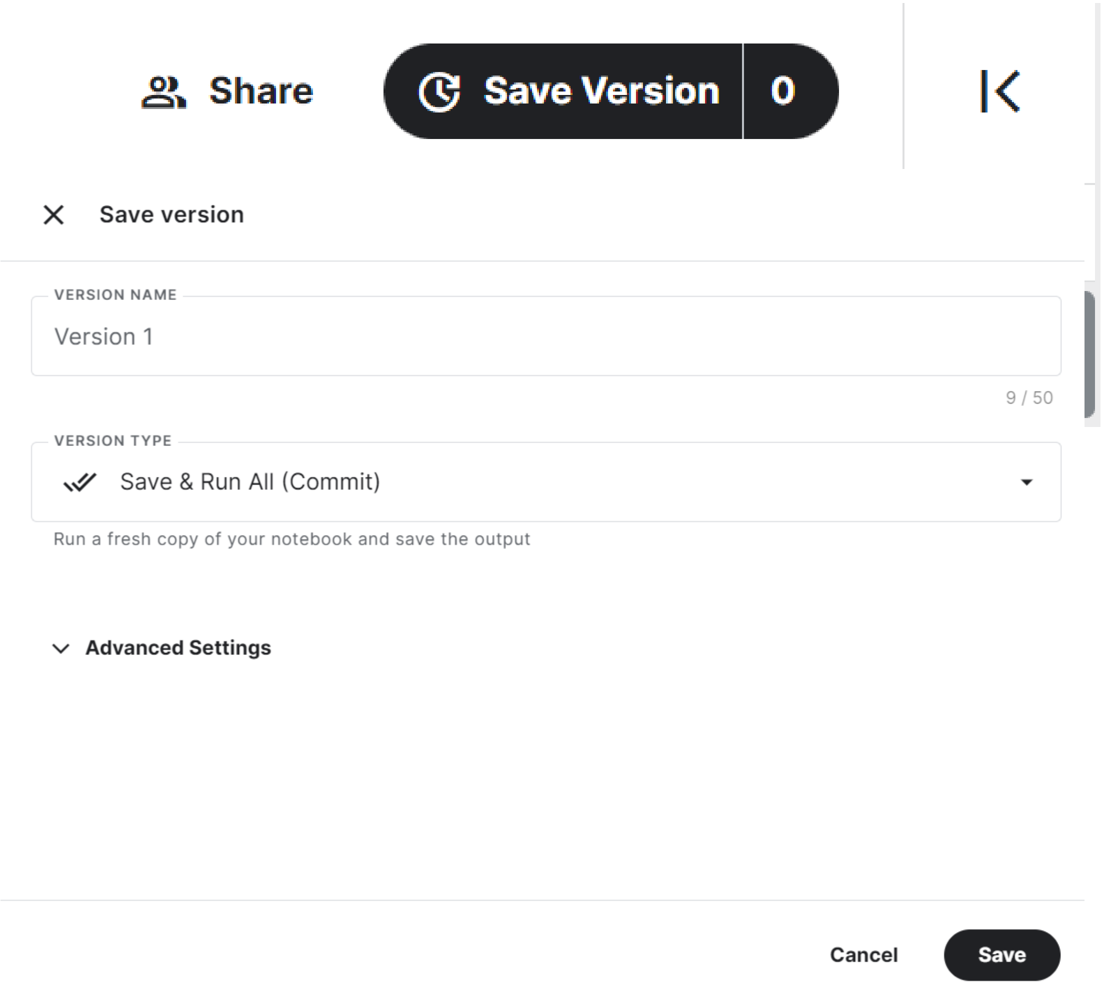
you're in the right place!

Part 1: Get started

In this section, you'll learn more about the competition and make your first submission.

Save version

- Click Save.
  - A new version of the Notebook is saved and all cells are run (Commit).
  - If there are errors in your Notebook, they are shown after the process is done.
  - If everything went ok, new version of the Notebook is saved and all the outputs are shown to you.



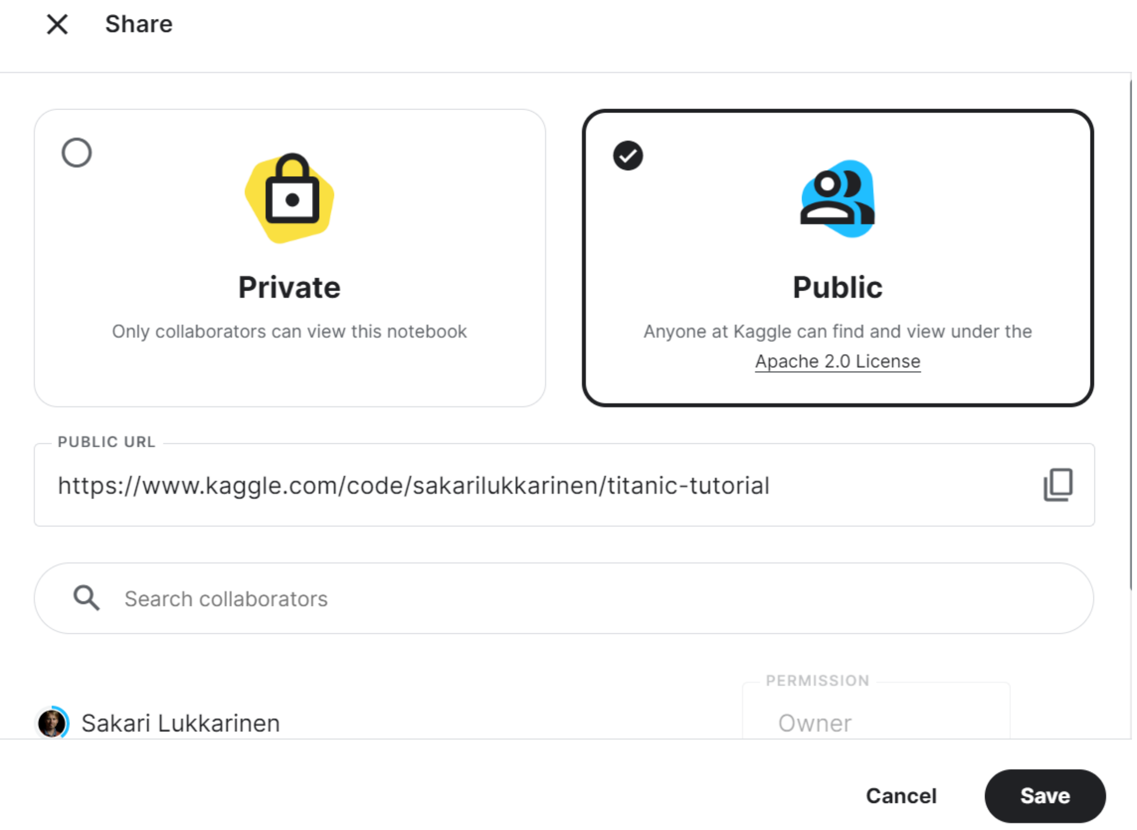
VERSION NAME
Version 1
9 / 50

VERSION TYPE
Save & Run All (Commit)
Run a fresh copy of your notebook and save the output

Share (and Publish)

Click **Share**.

- Select **Public** to share anyone at Kaggle to find and view.
  - Public URL is shown.
- Select **Private** to share only collaborators to find and view.
  - Search collaborators.
  - Change permissions for collaborators (View/Edit)
  - Allow comments: on/off



○ **Private**
Only collaborators can view this notebook

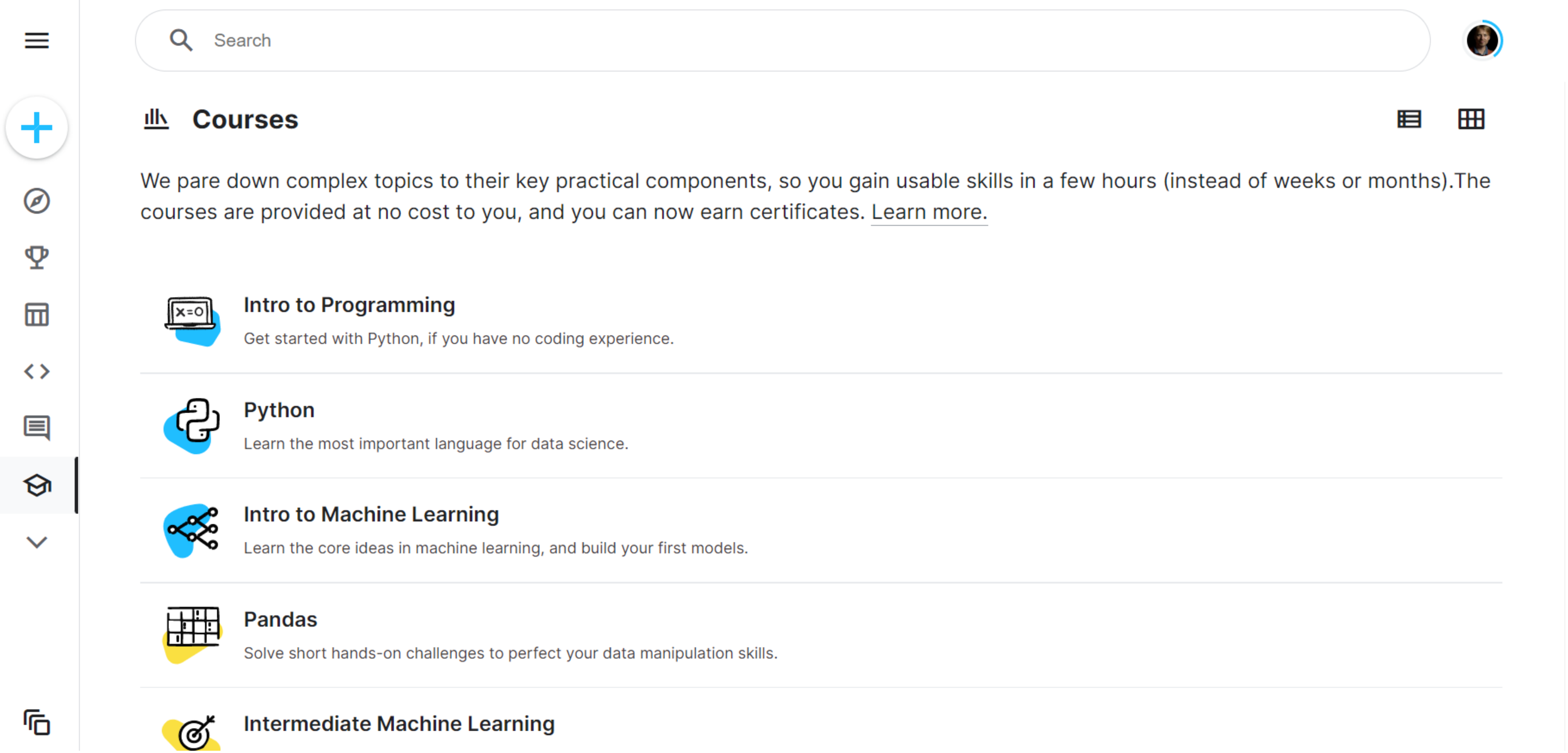
✓ **Public**
Anyone at Kaggle can find and view under the Apache 2.0 License

PUBLIC URL: https://www.kaggle.com/code/sakarilukkarinen/titanic-tutorial

Search collaborators

Sakari Lukkarinen — PERMISSION: Owner

Learn Python & more



Courses

We pare down complex topics to their key practical components, so you gain usable skills in a few hours (instead of weeks or months).The courses are provided at no cost to you, and you can now earn certificates. Learn more.

Intro to Programming
Get started with Python, if you have no coding experience.

Python
Learn the most important language for data science.

Intro to Machine Learning
Learn the core ideas in machine learning, and build your first models.

Pandas
Solve short hands-on challenges to perfect your data manipulation skills.

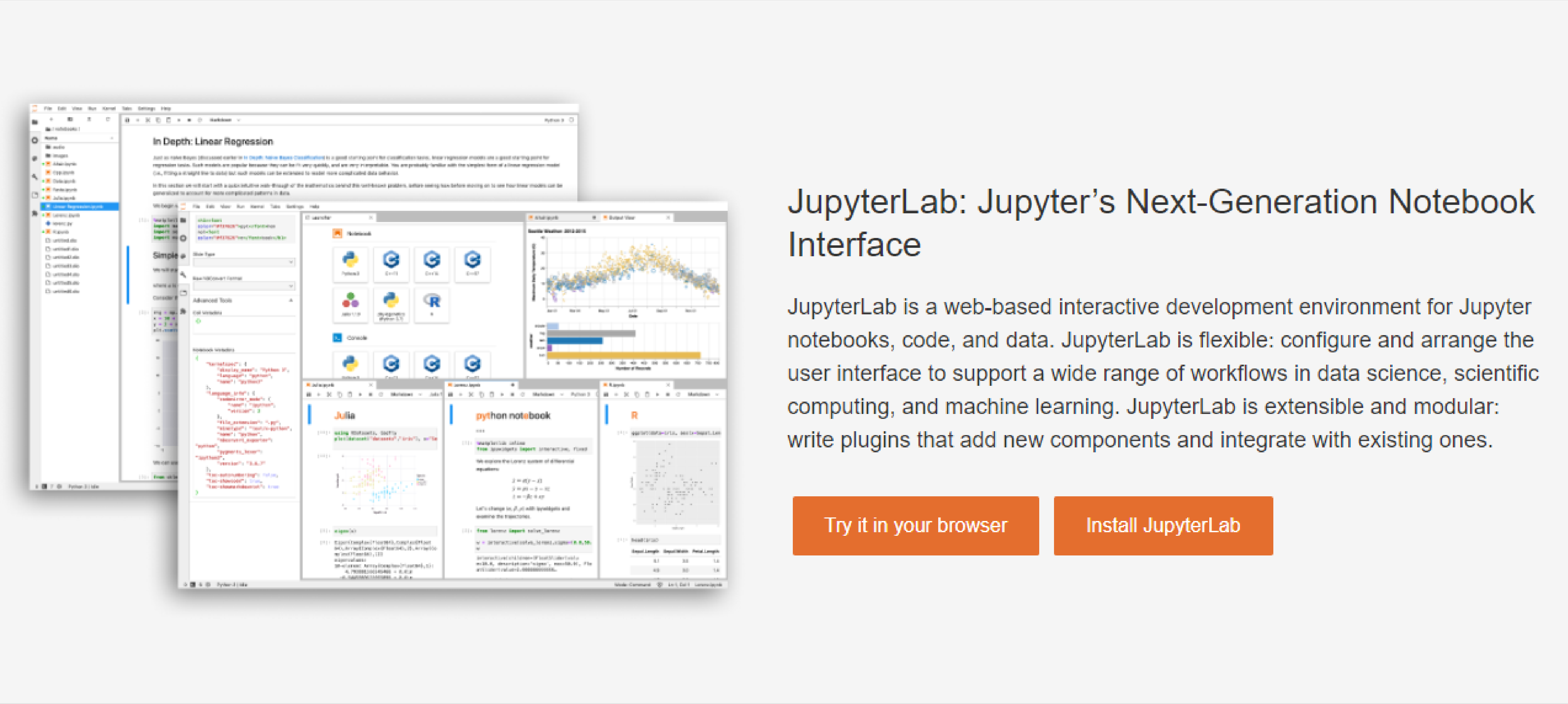
Intermediate Machine Learning

Learn Python, Data Viz, Pandas & More | Tutorials | Kaggle

Jupyter Notebook

with Anaconda Distribution on own computer

JupyterLab

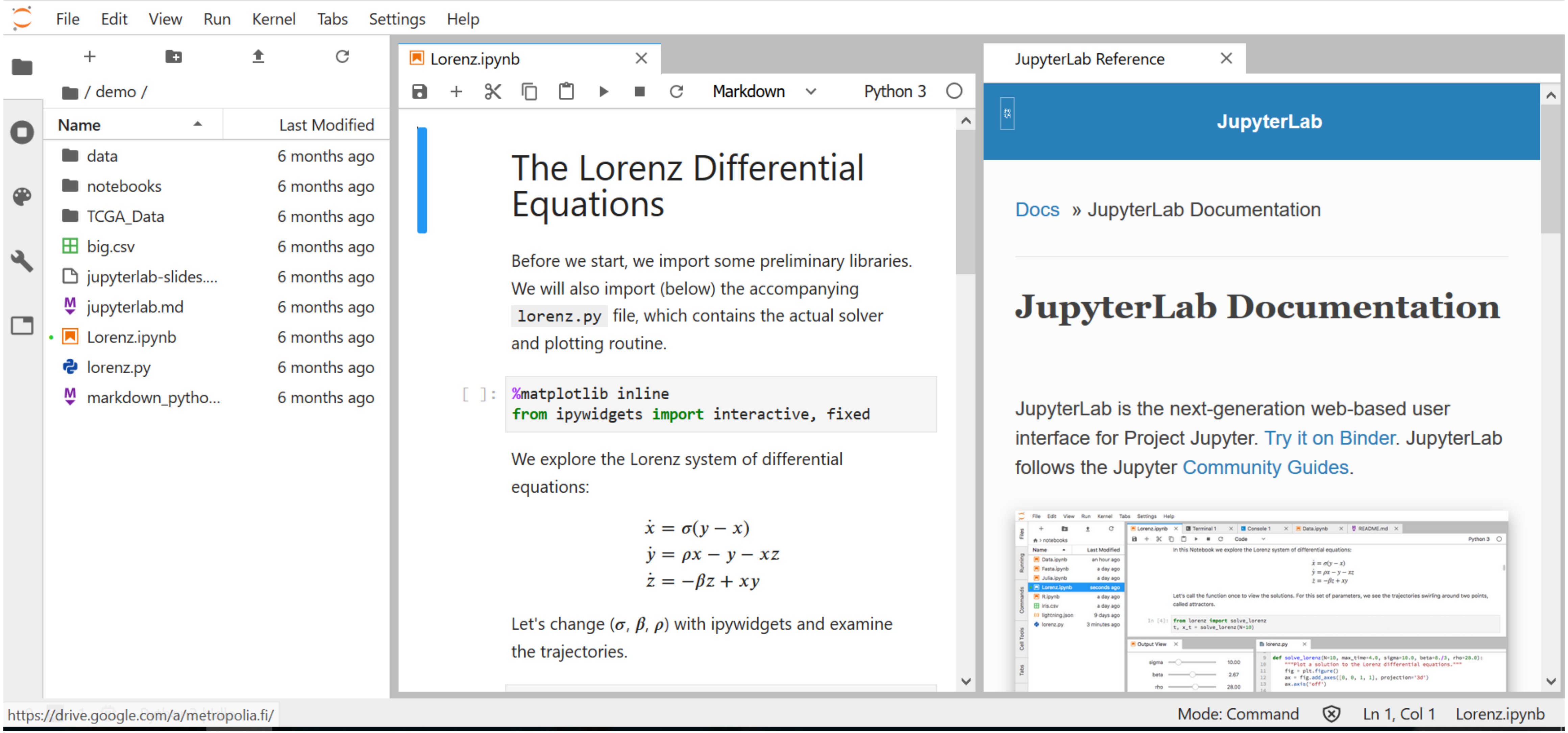


JupyterLab: Jupyter’s Next-Generation Notebook Interface

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

https://jupyter.org/

JupyterLab Interface



Name	Last Modified
data	6 months ago
notebooks	6 months ago
TCGA_Data	6 months ago
big.csv	6 months ago
jupyterlab-slides....	6 months ago
jupyterlab.md	6 months ago
Lorenz.ipynb	6 months ago
lorenz.py	6 months ago
markdown_pytho...	6 months ago

The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
[ ]: %matplotlib inline
     from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

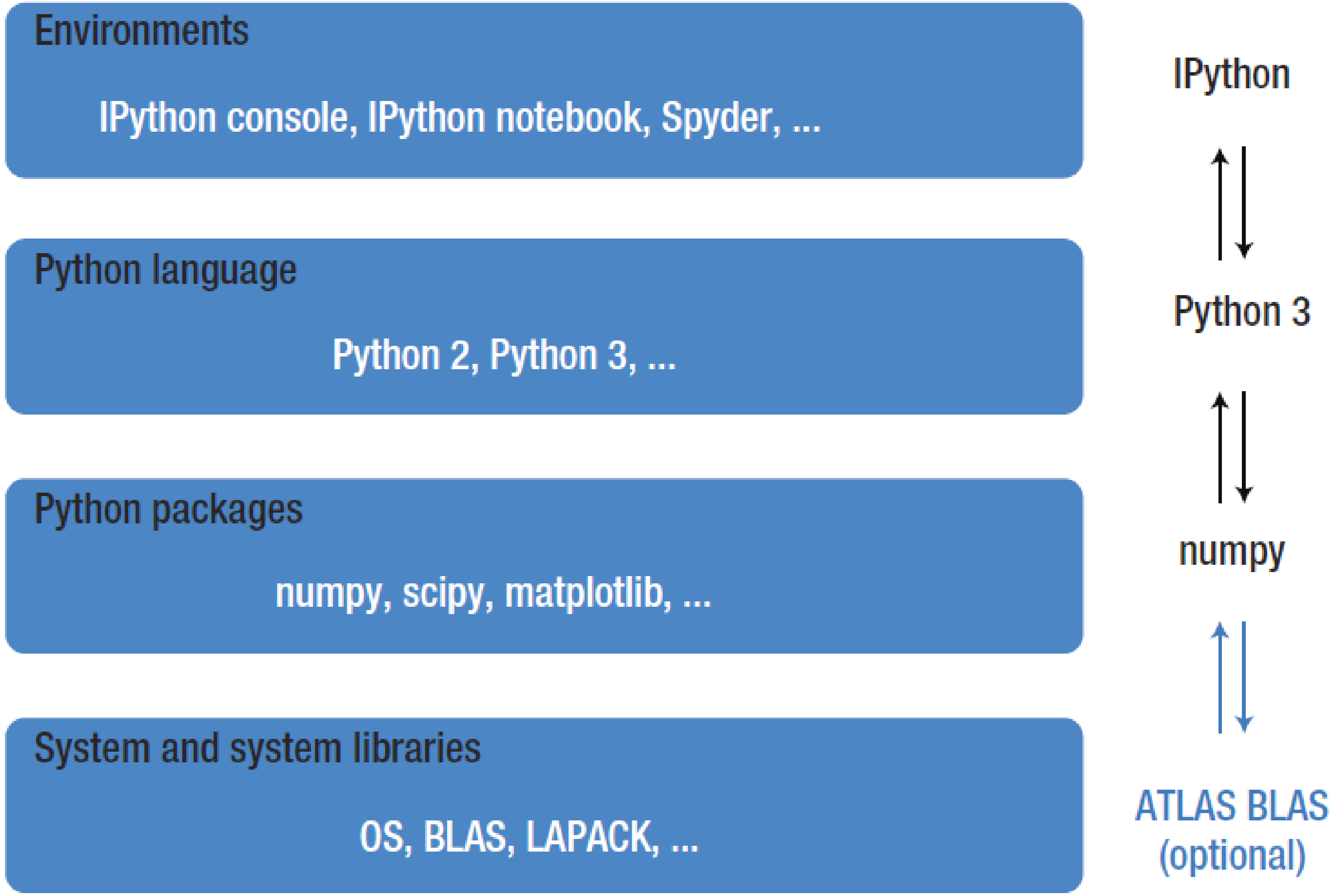
JupyterLab Documentation

JupyterLab is the next-generation web-based user interface for Project Jupyter. Try it on Binder. JupyterLab follows the Jupyter Community Guides.

Behind the scene – typical Web-application

Front End:
- for example JupyterLab, or
- Kaggle Notebook in Browser

Back End:
- Your own Laptop as a server, or
- Cloud server like Kaggle, Google Colab, AWS, Azure, …



Environments

IPython console, IPython notebook, Spyder, ...

Python language

Python 2, Python 3, ...

Python packages

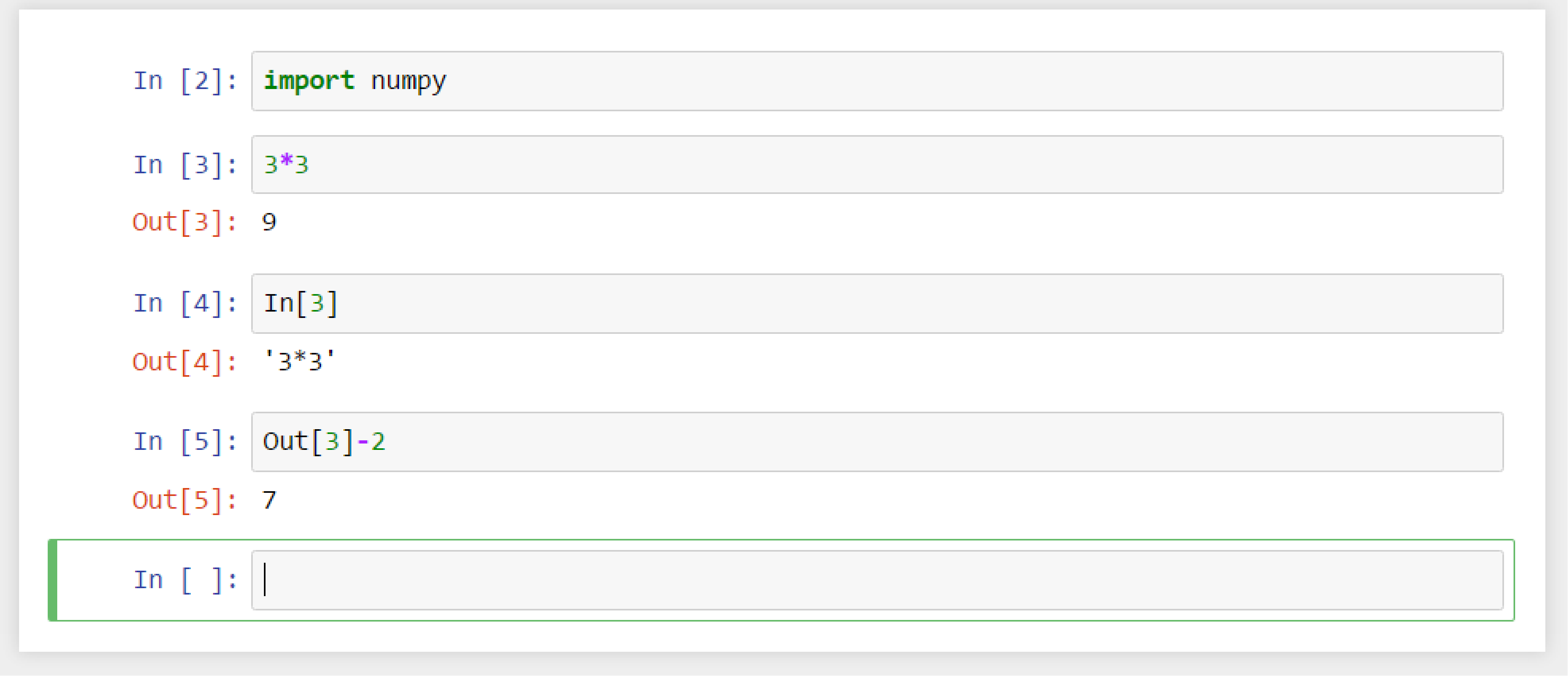
numpy, scipy, matplotlib, ...

System and system libraries

OS, BLAS, LAPACK, ...

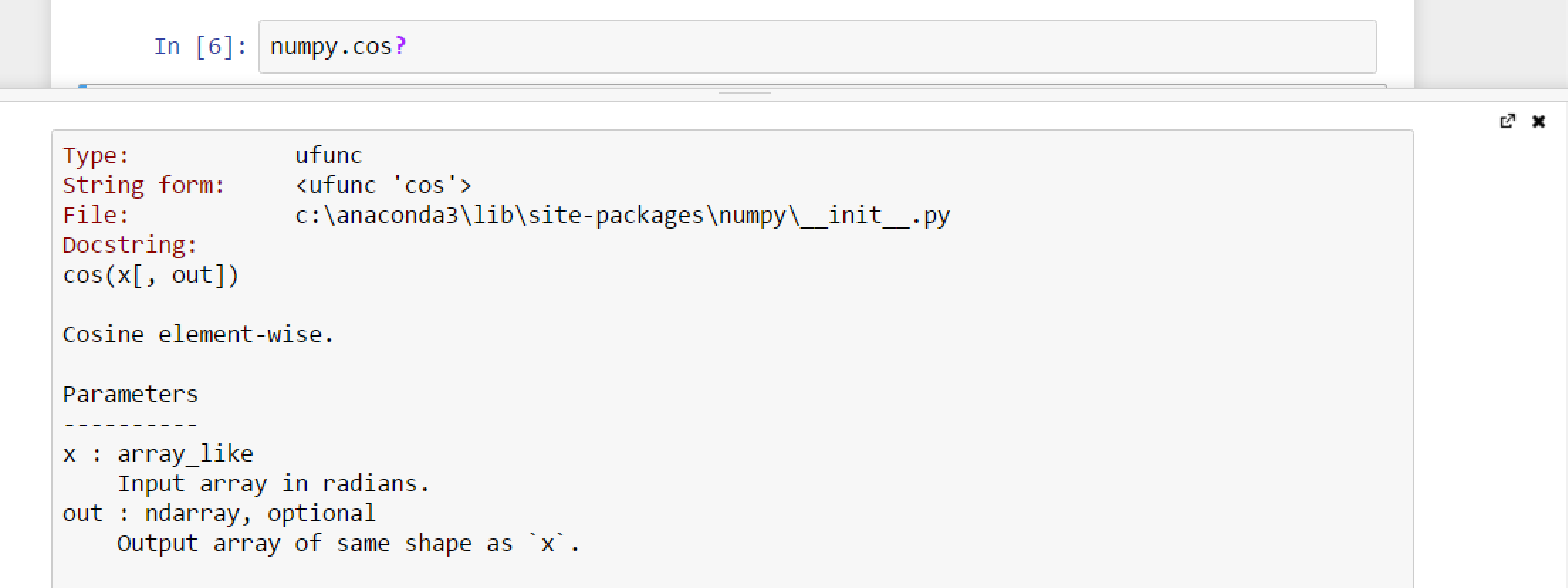
Figure 1-2. *An overview of the components and layers in the scientific computing environment for Python, from a user's perspective, from top to bottom. Users typically only interact with the top three layers, but the bottom layer constitutes a very important part of the software stack. An example of specific software components from each layer in the stack is shown in the right part of the figure*

Notebook: Input and output cells



```
In [2]: import numpy

In [3]: 3*3

Out[3]: 9

In [4]: In[3]

Out[4]: '3*3'

In [5]: Out[3]-2

Out[5]: 7

In [ ]: 
```

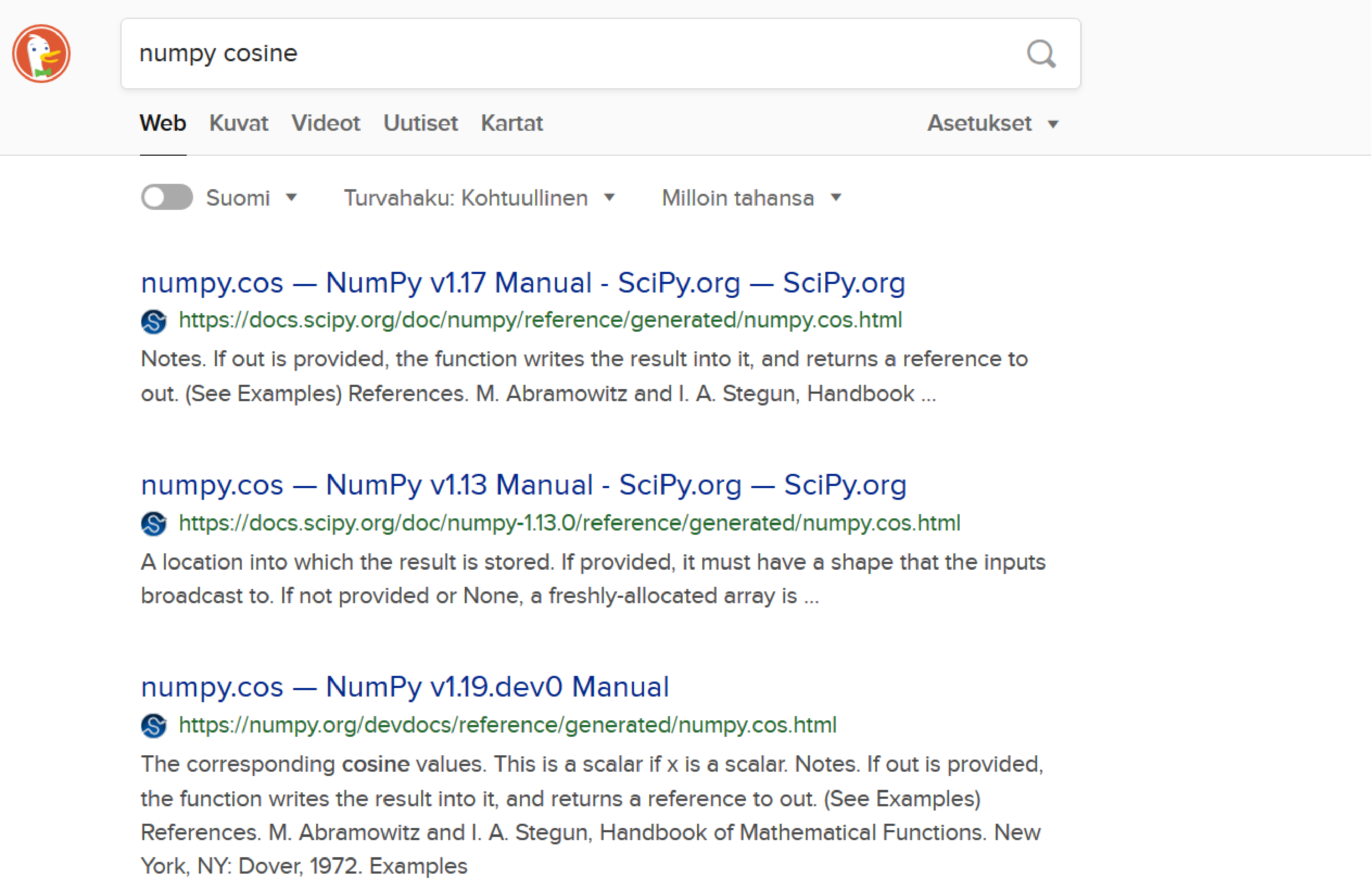
Documentation



```
In [6]: numpy.cos?
```

```
Type:            ufunc
String form:     <ufunc 'cos'>
File:            c:\anaconda3\lib\site-packages\numpy\__init__.py
Docstring:
cos(x[, out])

Cosine element-wise.

Parameters
----------
x : array_like
    Input array in radians.
out : ndarray, optional
    Output array of same shape as `x`.
```

Tip: Fastest way to find information is to use Google with proper keywords, try for example: numpy cosine



numpy cosine

Suomi ▾ Turvahaku: Kohtuullinen ▾ Milloin tahansa ▾

numpy.cos — NumPy v1.17 Manual - SciPy.org — SciPy.org

https://docs.scipy.org/doc/numpy/reference/generated/numpy.cos.html

Notes. If out is provided, the function writes the result into it, and returns a reference to out. (See Examples) References. M. Abramowitz and I. A. Stegun, Handbook ...

numpy.cos — NumPy v1.13 Manual - SciPy.org — SciPy.org

https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.cos.html

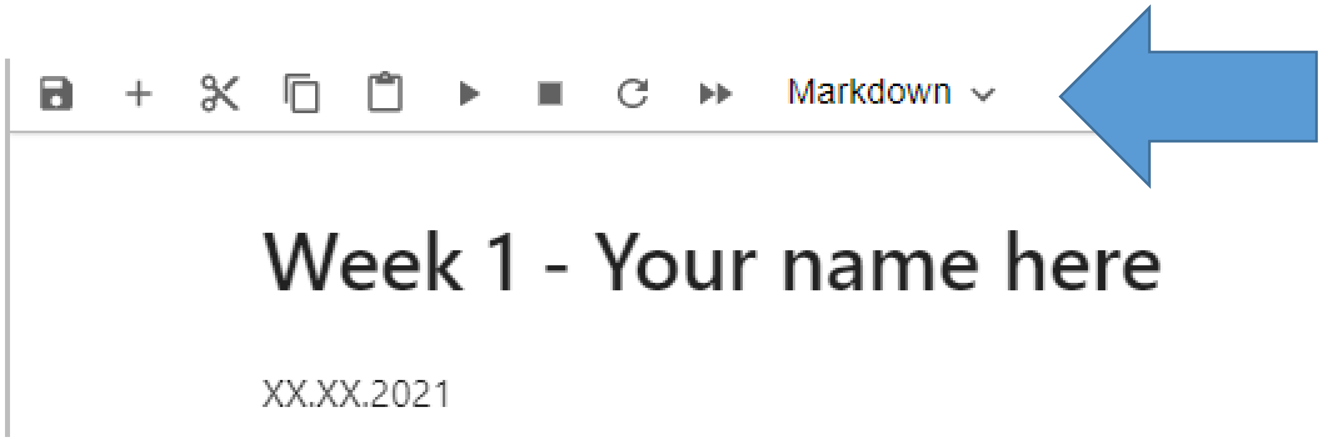
A location into which the result is stored. If provided, it must have a shape that the inputs broadcast to. If not provided or None, a freshly-allocated array is ...

numpy.cos — NumPy v1.19.dev0 Manual

https://numpy.org/devdocs/reference/generated/numpy.cos.html

The corresponding **cosine** values. This is a scalar if x is a scalar. Notes. If out is provided, the function writes the result into it, and returns a reference to out. (See Examples) References. M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions. New York, NY: Dover, 1972. Examples

Cell types



Week 1 - Your name here

XX.XX.2021

Code

Any **Python code**.

Press **Shift+Enter** to send the code to the kernel. The results are sent back to the browser.

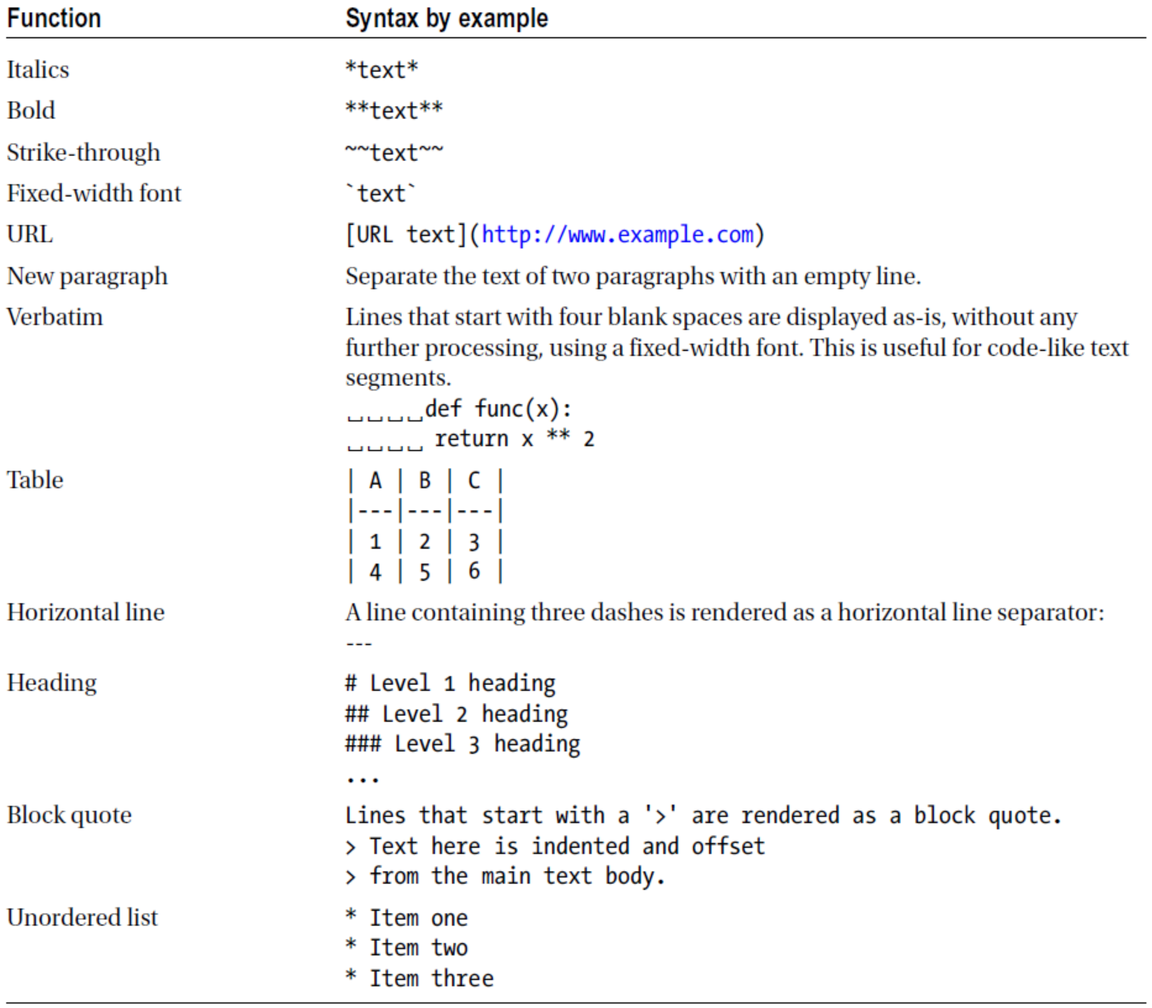
Markdown (=Documentation)

Contains marked-up plain text, which is interpreted using **Markdown Language and HTML** (and Latex).

Raw

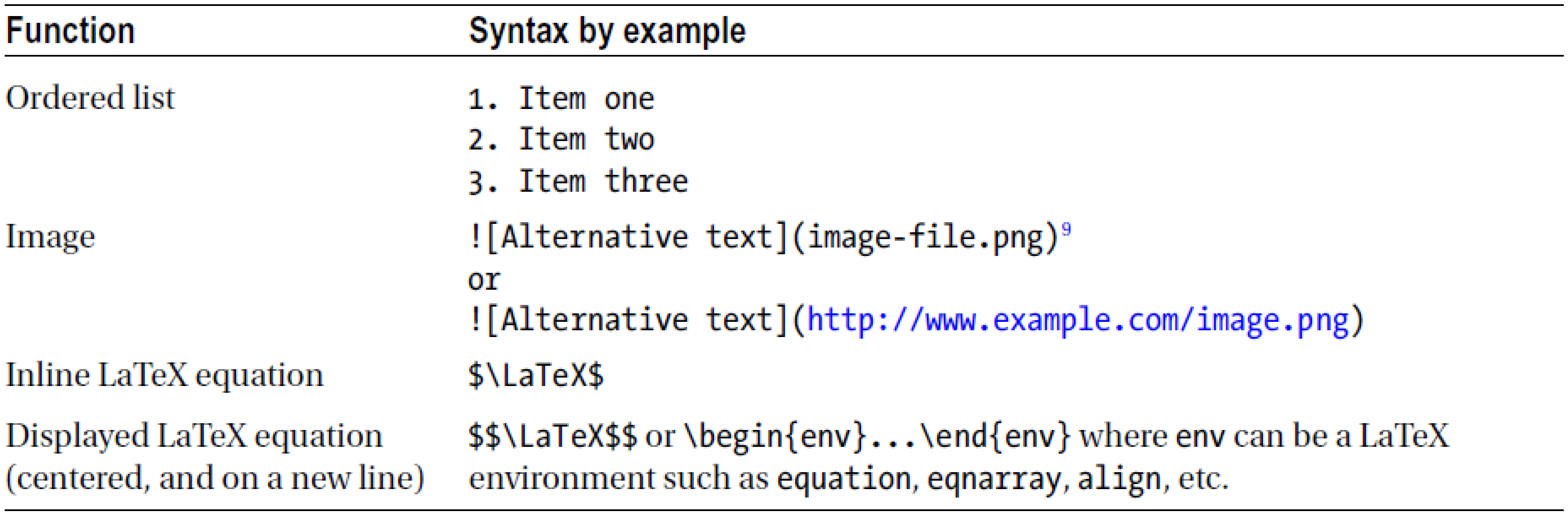
A raw text cell, displayed w.ithout any interpretation

Markdown cells
- your documentation



<table>
<tr><th>Function</th><th>Syntax by example</th></tr>
<tr><td>Italics</td><td>*text*</td></tr>
<tr><td>Bold</td><td>**text**</td></tr>
<tr><td>Strike-through</td><td>~~text~~</td></tr>
<tr><td>Fixed-width font</td><td>`text`</td></tr>
<tr><td>URL</td><td>[URL text](http://www.example.com)</td></tr>
<tr><td>New paragraph</td><td>Separate the text of two paragraphs with an empty line.</td></tr>
<tr><td>Verbatim</td><td>Lines that start with four blank spaces are displayed as-is, without any further processing, using a fixed-width font. This is useful for code-like text segments.
␣␣␣␣def func(x):
␣␣␣␣ return x ** 2</td></tr>
<tr><td>Table</td><td>| A	B	C
1	2	3
4	5	6 |</td></tr>
<tr><td>Horizontal line</td><td>A line containing three dashes is rendered as a horizontal line separator:
---</td></tr>
<tr><td>Heading</td><td># Level 1 heading
Level 2 heading
Level 3 heading
...</td></tr>
<tr><td>Block quote</td><td>Lines that start with a '>' are rendered as a block quote.
> Text here is indented and offset
> from the main text body.</td></tr>
<tr><td>Unordered list</td><td>* Item one
* Item two
* Item three</td></tr>
</table>

Markdown cells (continued)



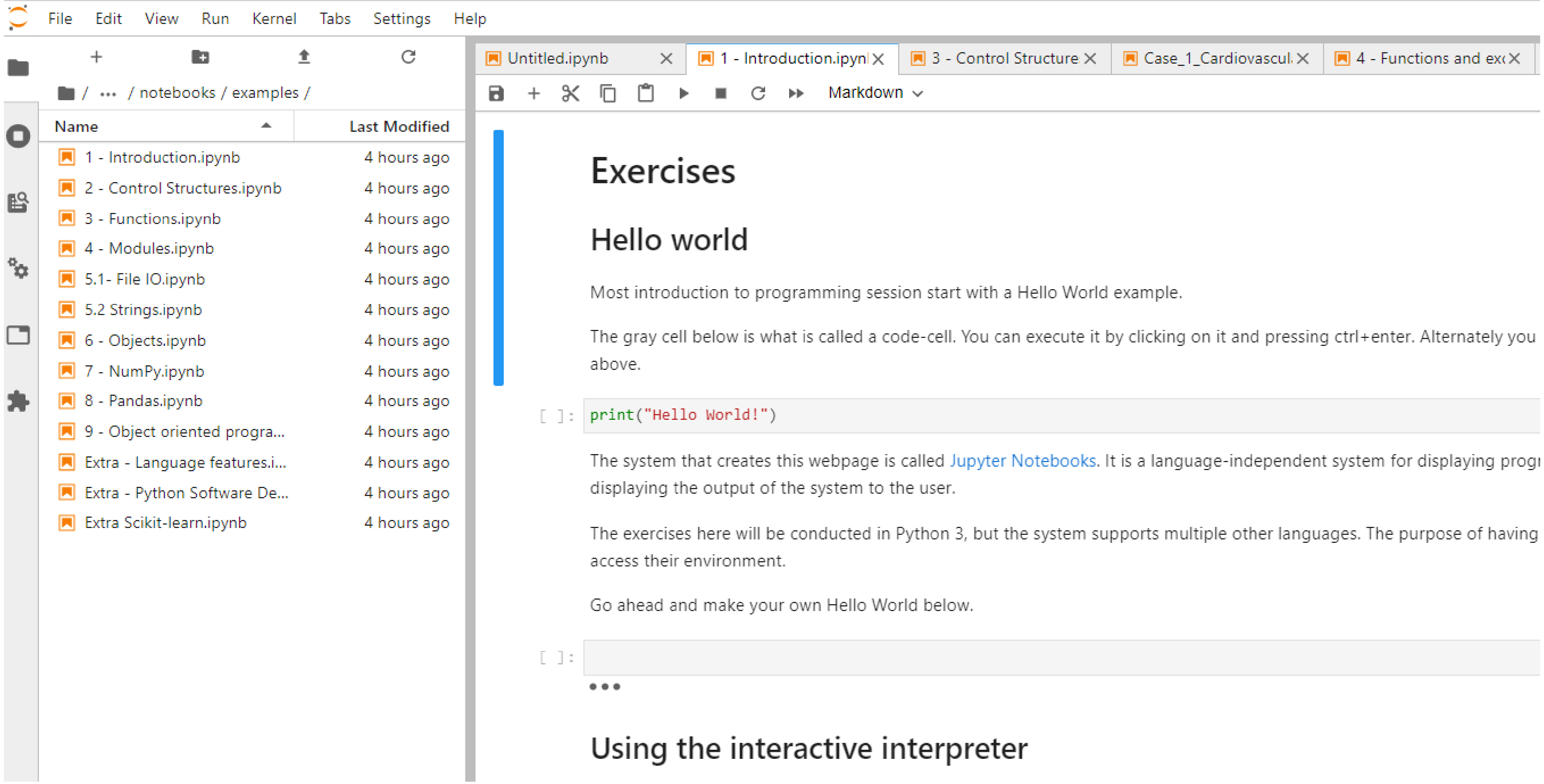
Function	Syntax by example

Ordered list	`1. Item one`
	`2. Item two`
	`3. Item three`

Image	`![Alternative text](image-file.png)`[9]
	`or`
	`![Alternative text](http://www.example.com/image.png)`

Inline LaTeX equation	`$\LaTeX$`
Displayed LaTeX equation (centered, and on a new line)	`$$\LaTeX$$` or `\begin{env}...\end{env}` where `env` can be a LaTeX environment such as `equation`, `eqnarray`, `align`, etc.

File explorer - JupyterLab



Name	Last Modified
1 - Introduction.ipynb	4 hours ago
2 - Control Structures.ipynb	4 hours ago
3 - Functions.ipynb	4 hours ago
4 - Modules.ipynb	4 hours ago
5.1- File IO.ipynb	4 hours ago
5.2 Strings.ipynb	4 hours ago
6 - Objects.ipynb	4 hours ago
7 - NumPy.ipynb	4 hours ago
8 - Pandas.ipynb	4 hours ago
9 - Object oriented progra...	4 hours ago
Extra - Language features.i...	4 hours ago
Extra - Python Software De...	4 hours ago
Extra Scikit-learn.ipynb	4 hours ago

Exercises

Hello world

Most introduction to programming session start with a Hello World example.

The gray cell below is what is called a code-cell. You can execute it by clicking on it and pressing ctrl+enter. Alternately you above.

```
[ ]: print("Hello World!")
```

The system that creates this webpage is called Jupyter Notebooks. It is a language-independent system for displaying progı displaying the output of the system to the user.

The exercises here will be conducted in Python 3, but the system supports multiple other languages. The purpose of having access their environment.

Go ahead and make your own Hello World below.

```
[ ]:
```

•••

Using the interactive interpreter

Case 0. Learning basics

Case 0. Learning basics - Aims

The aims of this assignment are to learn to:

1. Use Kaggle/Jupyter Notebooks
2. Use basic tensorflow neural networks functions
3. Document the results clearly and in an easily readable format using Notebook’s Markdown cells.

Case 0. Learning basics - Task

Your task is to find **the simplest neural network model** that can learn **in the least number of epochs** to classify the black and white handwritten digits**.**

By trial and error, find

- The simplest neural network model (= least number of hidden neurons),
- The least number of epochs, and
- To achieve 0.970 of accuracy with test set.

Document the Notebook

- remove the instructions and
- replace them with your own notes.

Case 0. Learning basics - Return

Save a hyperlink to your Notebook in OMA assignments.

To create a hyperlink:

- In Kaggle
  - File > Share > Public, or
  - File > Share > Private and Search collaborators
- In Colab
  - Share > General Access: Metropolia Ammattikorkeakoulu, Copy link
- In Github
  - Settings > Danger Zone > Change repository visibility, or
  - Settings > General > Collaborators

Summary

During this course we use:

- Python programming language
- Jupyter Notebooks to document the code
- The following environments
  - Google Colaboratory Notebooks
  - Kaggle Notebooks, or
  - Anaconda Distribution and JupyterLab Notebooks

First assignment (Case 0) is about learning the basics and how to use Notebooks.