Modern Data Analytics

Martial Luyts

Catholic University of Leuven, Belgium

`martial.luyts@kuleuven.be`

KU LEUVEN

LEUVEN STATISTICS
RESEARCH CENTRE

Contents

Part 0:

Introduction

0.1 Goal & organisation of the course

In the course of MDA, you will

- Acquire the necessary practical skill set and theoretical knowledge to deal with a wide variety of data science tasks.
- Be exposed to a solid set of tools to successfully approach machine learning (ML) problems.
- Focus on modeling knowledge, as well as knowledge related to setting up the required IT infrastructure to execute ML projects in practice.

Organisation:

- **Traditional on-campus lecture format** in which a lecture introduces the material to be studied by you.
- After class, recording of the lecture will become available on Toledo.
- Each lecture is related to a specific theme of the course (see Section 0.2).
- Throughout the course, examples of implementations will be given using the **Python programming language** or the **AWS cloud environment**.
- To successfully complete the course, you will need to be able to execute Python code in both Jupyter Notebook and Integrated Development Environment (IDE). Version of Python that will be used is 3.9.13.

0.2 Scheme

Date	Content	Teacher
	Part 1: Python Fundamentals for Data Scientists	
11/02	Introduction to Python and Object Oriented Programming	Martial Luyts
18/02	Data Science in Python: Packages	Ruben Kerkhofs
25/02	Data Science Algorithms in Python	Gregory van Kruijsdijk
04/03	Machine Learning Pipelines in Scikit Learn	Gregory van Kruijsdijk
	Part 2: Dashboarding in Python	
11/03	Dashboarding in Python: Shiny for Python	Martial Luyts
18/03	Q&A session 1	
	Part 3: IT infrastructure for Data Science and Analytics	
25/03	Key Components of an Analytics Infrastructure	Ruben Kerkhofs
01/04	Code Collaboration and Versioning	Gregory van Kruijsdijk
22/04	Containerisation using Docker	Martial Luyts
29/04	Cloud Computing 1	Martial Luyts
06/05	Cloud Computing 2	Martial Luyts
13/05	Data Processing Analytics, Scalability, and Performance	Ruben Kerkhofs
20/05	Q&A session 2	

0.3 Evaluations & deadlines

The evaluation of MDA is based on four components:

1. A **group report** which consists of a description of your group project. This group project is submitted on a per-group basis meaning that there is only a single submission per group. (20%)

2. A **group project** in which you work together with your group to solve a data science project which will be presented during an **oral examination** (in group). (50%)

3. Your **individual questions** asked to you during the group project presentation. Each student should expect to receive questions on both the group project content as well as any of the course material presented during the lectures. (20%)

4. **Peer evaluation** where your performance is assessed by your fellow group members. (10%)

The following deadlines needs to be respected:

- 01/03: Group composition
- 24/03: Project proposal
- 27/05: Project submission

!!Important: These deadlines needs to be respected. Late submission will not be allowed, and results automatically in an NA (Not Attended) score on the total grade.

!!Important: Presentations of the group project will be planned outside the exam period. Details about the date will be follow soon on Toledo!

0.4 Questions?

Do you have any questions?

- Ask them in class and/or during the Q&A sessions
- Ask them in the Discussion board on Toledo

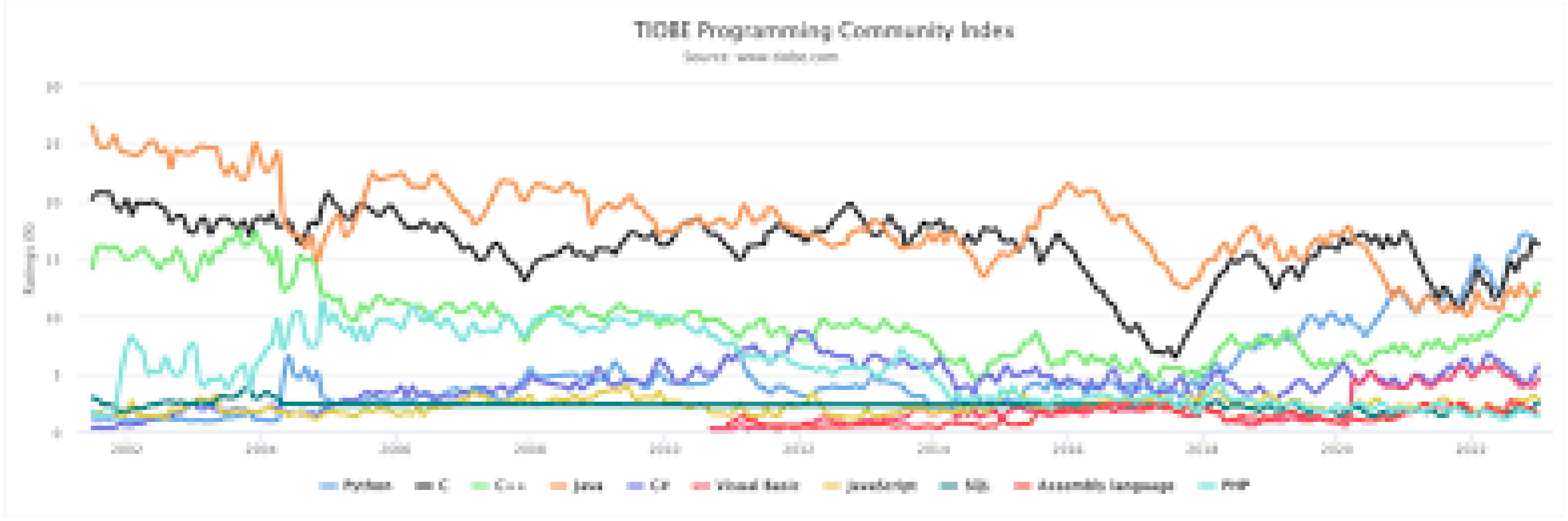
!!Important: Questions related to the course will not be answered by mail! The lecturers of the course are happy to provide you with all necessary info during the official moments and directories.

Part 1:

Python Fundamentals for Data Scientists

1.1 Python

- Nowadays, Python is considered as one of the most popular programming languages among data scientists, and beyond.



- Generally used when data analysis tasks need to be integrated with web apps or if statistics code needs to be incorporated into a production database.

- While R is mainly used for traditional statistical computing, Python is a good tool for the analysis of extended data science tasks like NLP, computer vision, etc.

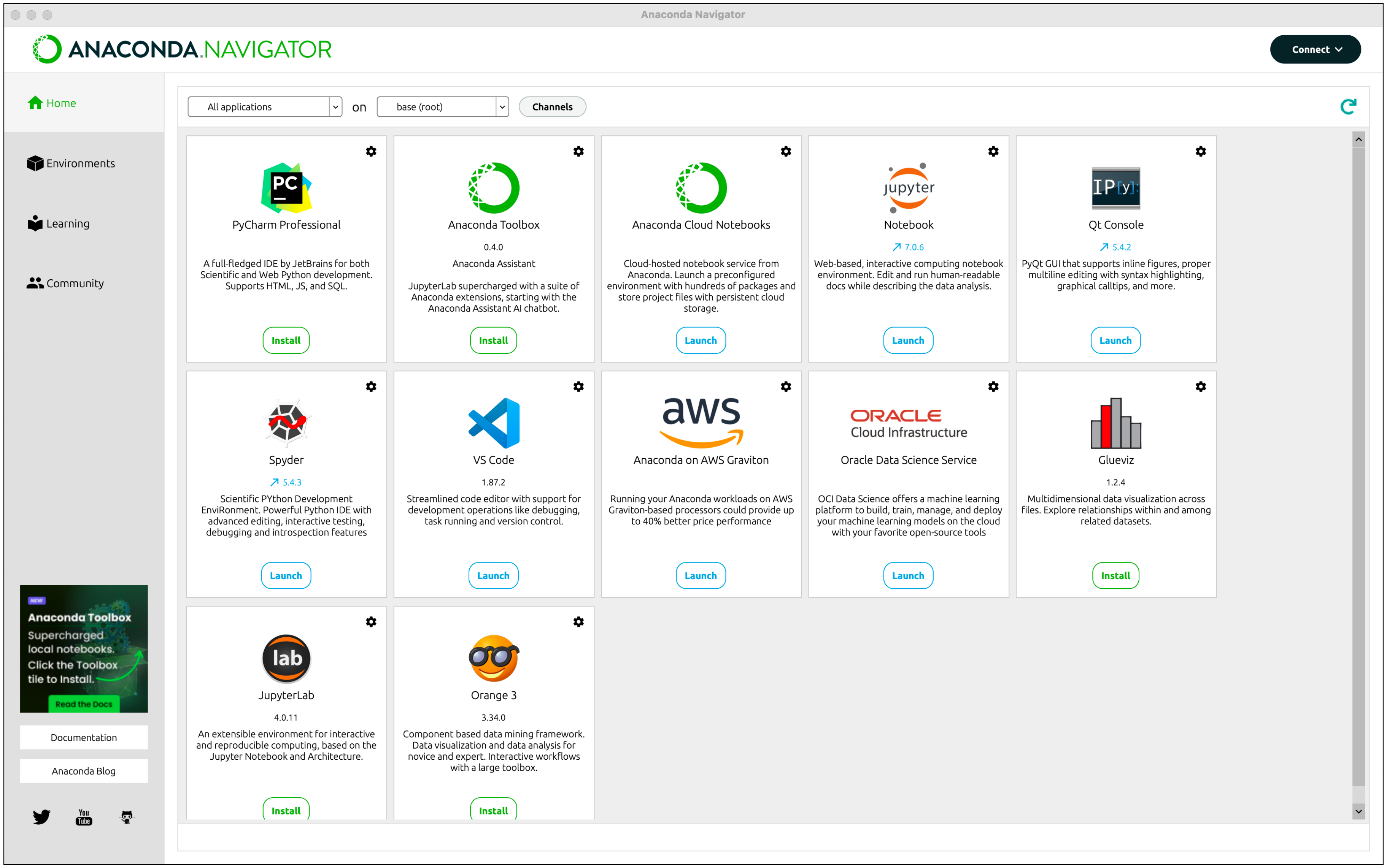
- Download & start: **Anaconda**

  - Installation: www.anaconda.com/products/individual

  - Instead of going to the Python.org website to download the required version of Python and then installling the different packages needed, Anaconda, i.e., a package manager, is a quicker introduction

  - Anaconda offers the most useful packages for mathematics, science and engineering

  - In particular, over 300 packages are automatically installed with Anaconda



PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

Install

Anaconda Toolbox

0.4.0

Anaconda Assistant

JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.

Install

Anaconda Cloud Notebooks

Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.

Launch

Notebook

7.0.6

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

Launch

Qt Console

5.4.2

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

Launch

Spyder

5.4.3

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

Launch

VS Code

1.87.2

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch

Anaconda on AWS Graviton

Running your Anaconda workloads on AWS Graviton-based processors could provide up to 40% better price performance

Launch

Oracle Data Science Service

OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools

Launch

Glueviz

1.2.4

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

Install

JupyterLab

4.0.11

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

Install

Orange 3

3.34.0

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

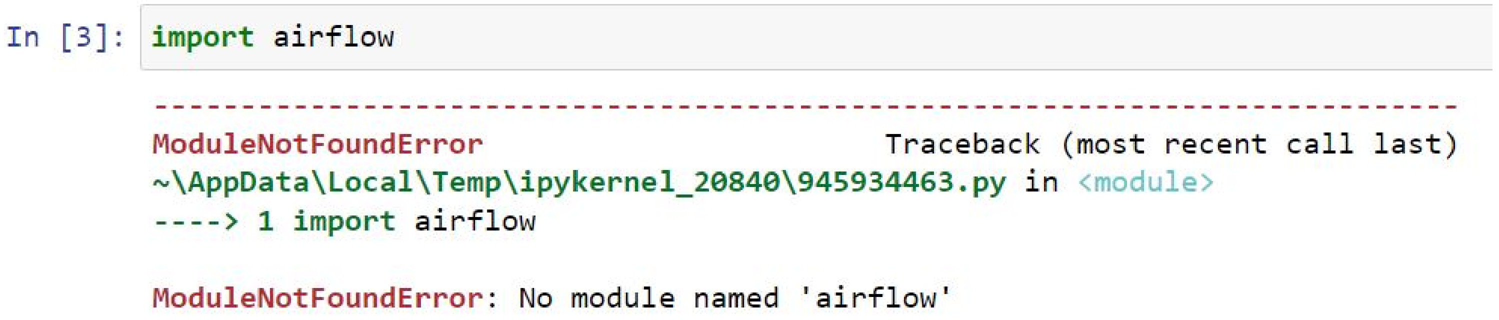
Install

NEW Anaconda Toolbox Supercharged local notebooks. Click the Toolbox tile to install. Read the Docs

Documentation

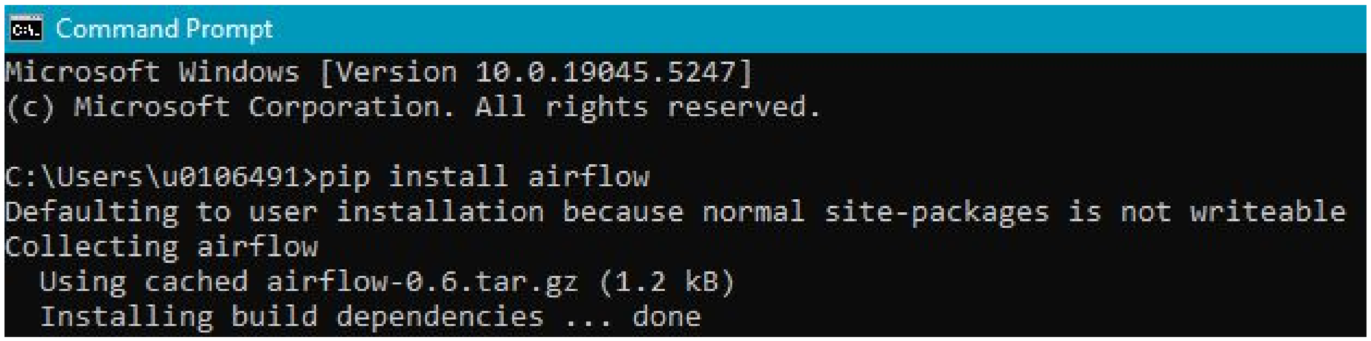
Anaconda Blog

- It can happen that some packages are not installed yet.



```
In [3]: import airflow

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20840\945934463.py in <module>
----> 1 import airflow

ModuleNotFoundError: No module named 'airflow'
```

- **Solution:** Use the "pip install" command in the command window in order to fetch a particular package and install it on your computer:



```
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\u0106491>pip install airflow
Defaulting to user installation because normal site-packages is not writeable
Collecting airflow
  Using cached airflow-0.6.tar.gz (1.2 kB)
  Installing build dependencies ... done
```

→ c:\Users\u0106491\Documents>pip install airflow

- **Important packages:**
  - **SciPy and Numpy** for numerical computing
  - **Pandas** for easy data processing, cleaning, etc.
  - **Matplotlib** to produce graphs
  - **Scikit-Learn** for machine learning
  - **Statsmodels** for statistical models and unit tests
  - **PIL** for basic image importing, manipulation, and exporting

- **MoviePy** is to videos what Pillow is to images. It provides a range of functionality for common tasks associated with importing, modifying and exporting video files
- **Requests** if you application sends any data over HTTP
- **Plotly & Shiny** to produce interactive plots & dashboards
- **PyTorch:** for mathematical operations over tensors. Particular useful for deep learning programming
- **Transformers** for the use of state-of-the-art pre-trained machine learning models. Particular useful in the area of NLP and computer vision

1.2 Writing Code

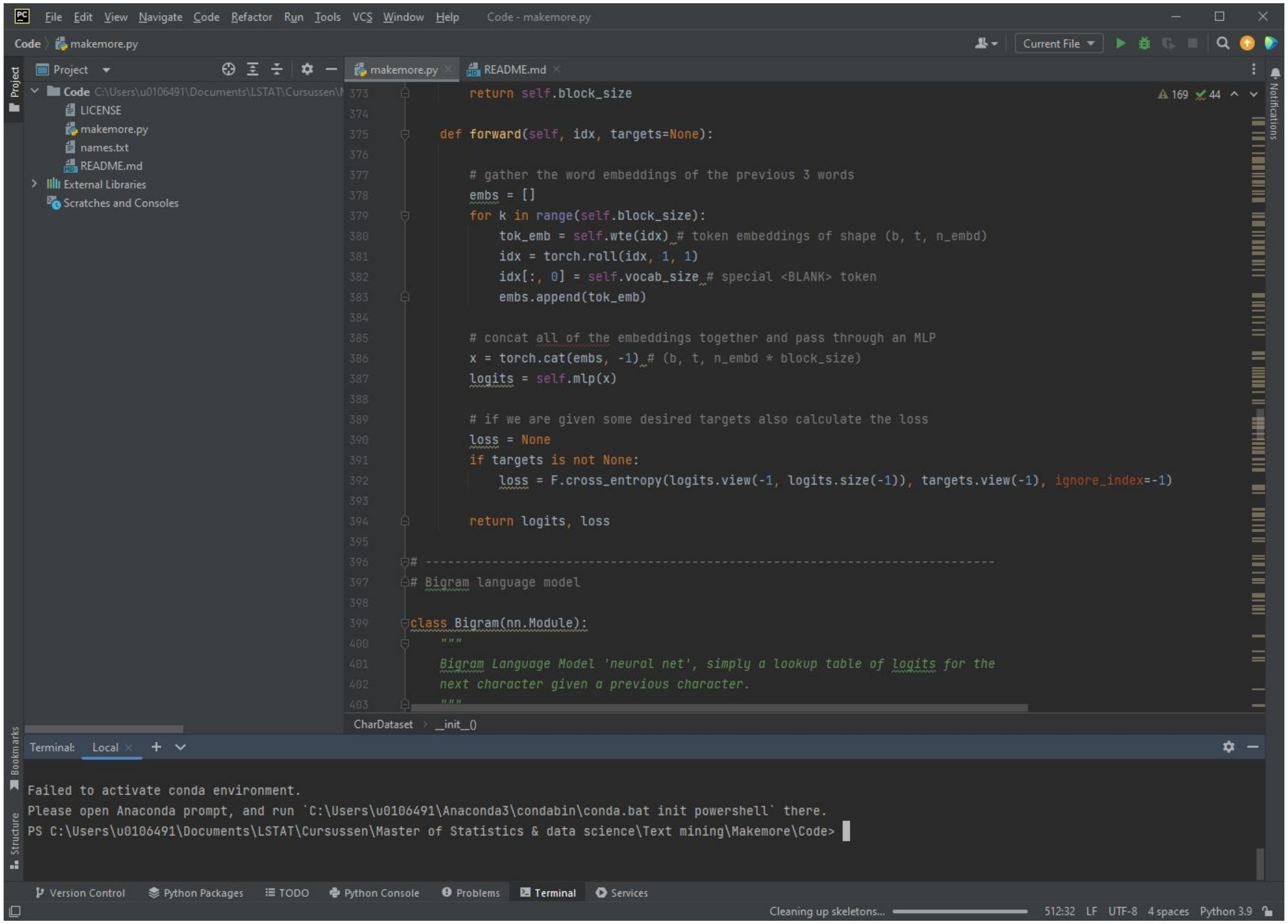
- When developing and executing code in Python, **IDEs (Integrated Development Environment's)** and **notebooks** play crucial roles.

- An **IDE** is a program dedicated to software development, integrating several tools specifically designed for software development:
  - An editor designed to handle code
  - Build, execution and debugging tools
  - Source control

- In Python, for example, several IDEs are available:
  - Atom
  - Spyder
  - **PyCharm**
  - **Visual Studio (VS) Code**
- In what follows, we will discuss the latter two IDEs, i.e., **PyCharm** & **VS Code**

1. PyCharm

- Download: www.jetbrains.com/pycharm/
- One of the best and full-featured, dedicated IDE for Python
- Available in both paid (Professional) and free open-source (Community) editions, for Windows, Mac OS X, and Linux platforms
- Supports Python development directly



```python
        return self.block_size

    def forward(self, idx, targets=None):

        # gather the word embeddings of the previous 3 words
        embs = []
        for k in range(self.block_size):
            tok_emb = self.wte(idx) # token embeddings of shape (b, t, n_embd)
            idx = torch.roll(idx, 1, 1)
            idx[:, 0] = self.vocab_size # special <BLANK> token
            embs.append(tok_emb)

        # concat all of the embeddings together and pass through an MLP
        x = torch.cat(embs, -1) # (b, t, n_embd * block_size)
        logits = self.mlp(x)

        # if we are given some desired targets also calculate the loss
        loss = None
        if targets is not None:
            loss = F.cross_entropy(logits.view(-1, logits.size(-1)), targets.view(-1), ignore_index=-1)

        return logits, loss

# -----------------------------------------------------------------------------
# Bigram language model

class Bigram(nn.Module):
    """
    Bigram Language Model 'neural net', simply a lookup table of logits for the
    next character given a previous character.
    """
```

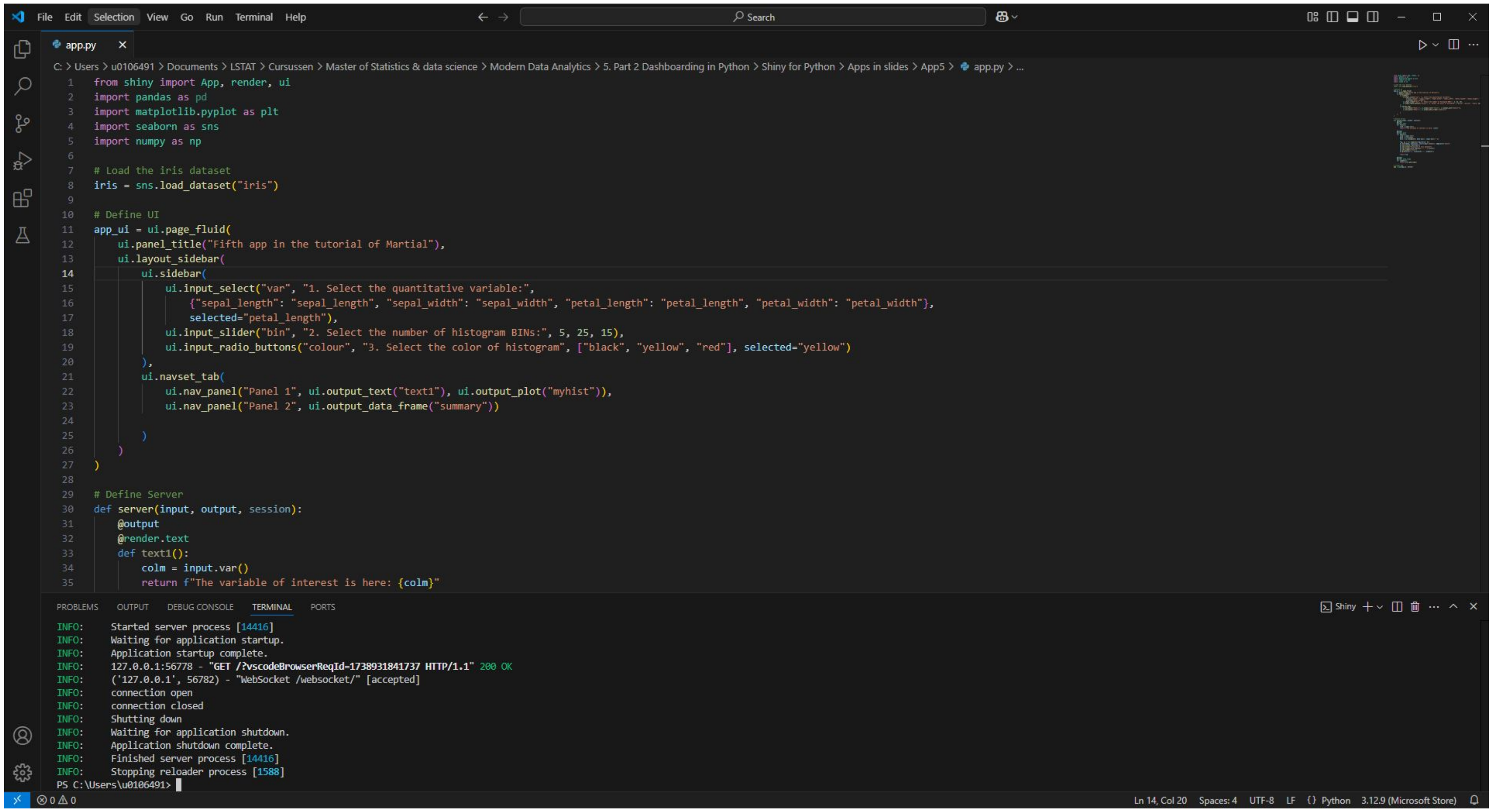
```
Failed to activate conda environment.
Please open Anaconda prompt, and run `C:\Users\u0106491\Anaconda3\condabin\conda.bat init powershell` there.
PS C:\Users\u0106491\Documents\LSTAT\Cursussen\Master of Statistics & data science\Text mining\Makemore\Code>
```

2. Visual Studio

- Download: code.visualstudio.com/download

- Main difference compared to PyCharm is the user interface:
  - PyCharm provides a comprehensive IDE with a more traditional layout, including various panels and tool windows
  - Visual studio offers a lightweight editor with a minimalistic interface that can be customized extensively with extensions and themes

- More details: https://bit.ly/3UBx58g



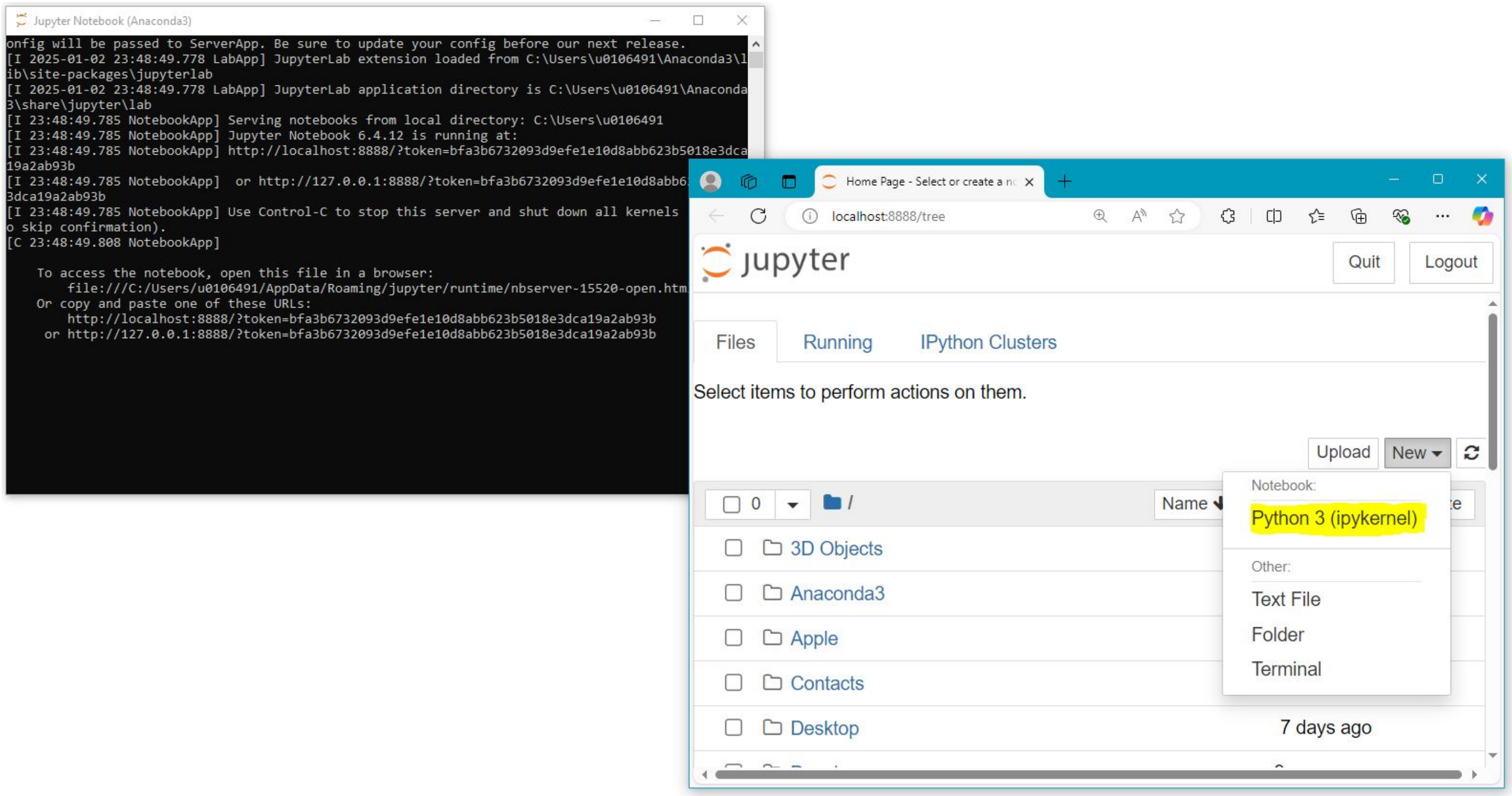
File Edit Selection View Go Run Terminal Help
Search
app.py
C: > Users > u0106491 > Documents > LSTAT > Cursussen > Master of Statistics & data science > Modern Data Analytics > 5. Part 2 Dashboarding in Python > Shiny for Python > Apps in slides > App5 > app.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Shiny
Ln 14, Col 20 Spaces: 4 UTF-8 LF {} Python 3.12.9 (Microsoft Store)

```python
from shiny import App, render, ui
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Load the iris dataset
iris = sns.load_dataset("iris")

# Define UI
app_ui = ui.page_fluid(
    ui.panel_title("Fifth app in the tutorial of Martial"),
    ui.layout_sidebar(
        ui.sidebar(
            ui.input_select("var", "1. Select the quantitative variable:",
                {"sepal_length": "sepal_length", "sepal_width": "sepal_width", "petal_length": "petal_length", "petal_width": "petal_width"},
                selected="petal_length"),
            ui.input_slider("bin", "2. Select the number of histogram BINs:", 5, 25, 15),
            ui.input_radio_buttons("colour", "3. Select the color of histogram", ["black", "yellow", "red"], selected="yellow")
        ),
        ui.navset_tab(
            ui.nav_panel("Panel 1", ui.output_text("text1"), ui.output_plot("myhist")),
            ui.nav_panel("Panel 2", ui.output_data_frame("summary"))

        )
    )
)

# Define Server
def server(input, output, session):
    @output
    @render.text
    def text1():
        colm = input.var()
        return f"The variable of interest is here: {colm}"
```

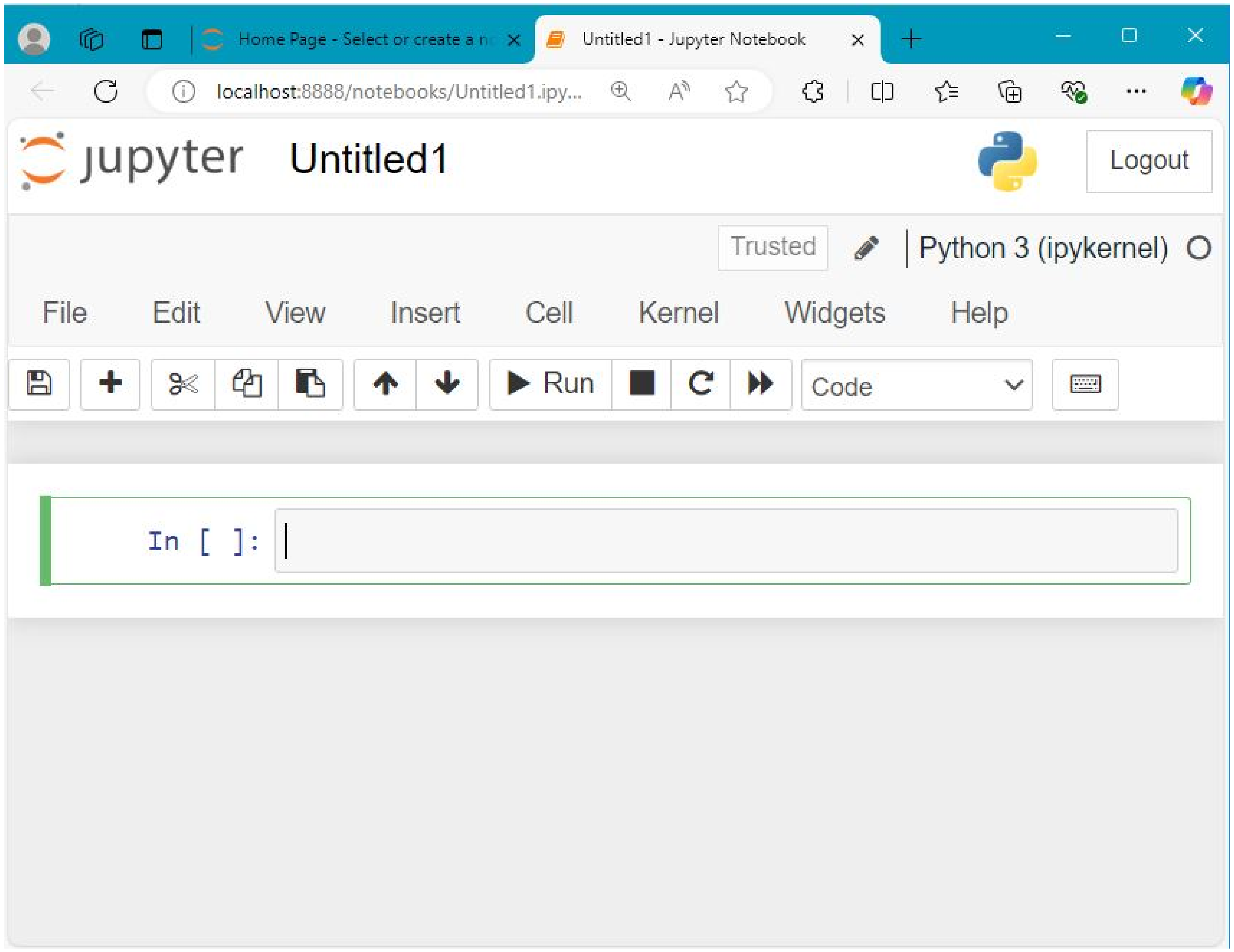
```
INFO:     Started server process [14416]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:56778 - "GET /?vscodeBrowserReqId=1738931841737 HTTP/1.1" 200 OK
INFO:     ('127.0.0.1', 56782) - "WebSocket /websocket/" [accepted]
INFO:     connection open
INFO:     connection closed
INFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [14416]
INFO:     Stopping reloader process [1588]
PS C:\Users\u0106491>
```

- Alternative to IDEs, **Jupyter notebooks** can also be considered
  - **Jupyter notebook** = Open-source web application that allows you to create and share documents that contain live code (in Python and other packages) with supporting equations, visualizations, and narrative text
  - Jupyter supports over 100 programming languages, including Python, R, Julia, and Scala
  - Can be shared with others through email, Github, etc.
- In this course, Jupyter notebooks (& VS Code) are used for executing Python code

- To run code in Jupyter notebook, the IPython kernel is used, but additional kernels may be installed

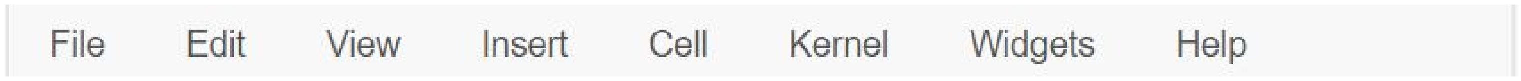


```
onfig will be passed to ServerApp. Be sure to update your config before our next release.
[I 2025-01-02 23:48:49.778 LabApp] JupyterLab extension loaded from C:\Users\u0106491\Anaconda3\l
ib\site-packages\jupyterlab
[I 2025-01-02 23:48:49.778 LabApp] JupyterLab application directory is C:\Users\u0106491\Anaconda
3\share\jupyter\lab
[I 23:48:49.785 NotebookApp] Serving notebooks from local directory: C:\Users\u0106491
[I 23:48:49.785 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 23:48:49.785 NotebookApp] http://localhost:8888/?token=bfa3b6732093d9efe1e10d8abb623b5018e3dca
19a2ab93b
[I 23:48:49.785 NotebookApp]  or http://127.0.0.1:8888/?token=bfa3b6732093d9efe1e10d8abb6
3dca19a2ab93b
[I 23:48:49.785 NotebookApp] Use Control-C to stop this server and shut down all kernels
o skip confirmation).
[C 23:48:49.808 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/u0106491/AppData/Roaming/jupyter/runtime/nbserver-15520-open.htm
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=bfa3b6732093d9efe1e10d8abb623b5018e3dca19a2ab93b
     or http://127.0.0.1:8888/?token=bfa3b6732093d9efe1e10d8abb623b5018e3dca19a2ab93b
```

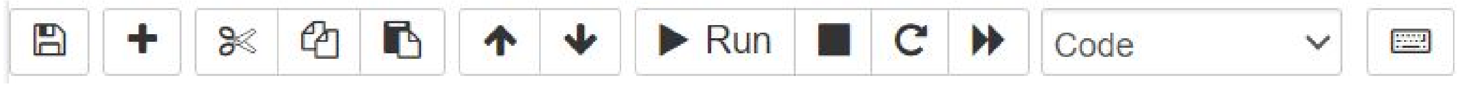

Home Page - Select or create a n
Untitled1 - Jupyter Notebook
localhost:8888/notebooks/Untitled1.ipy...
jupyter
Untitled1
Logout
Trusted
Python 3 (ipykernel)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [ ]:
```

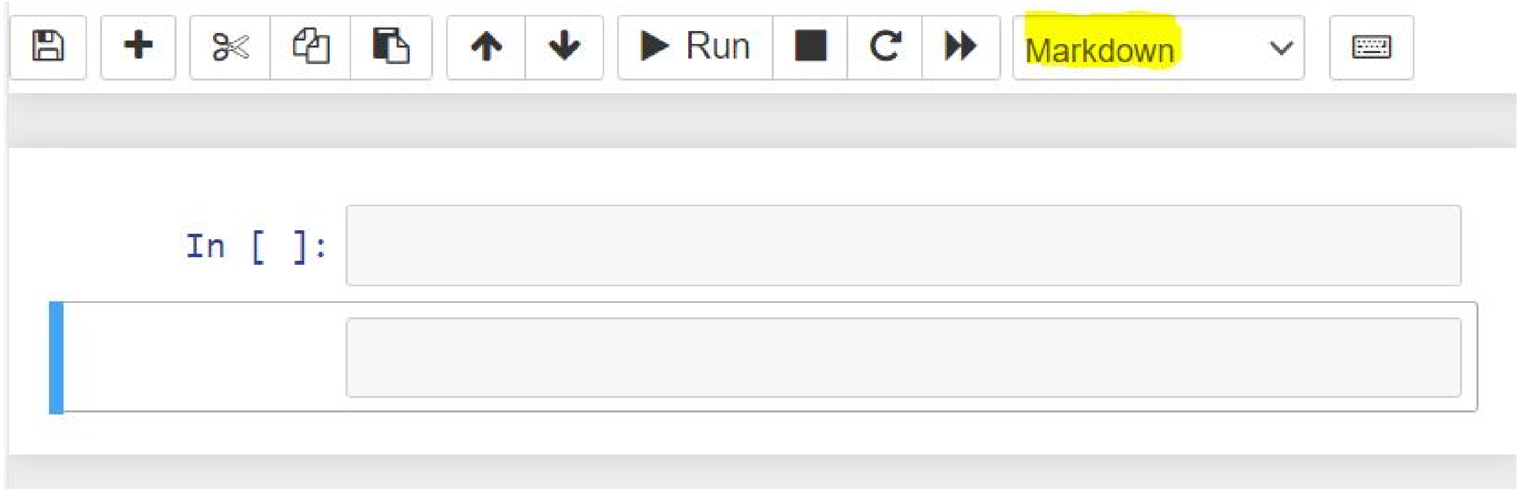
- Different components are available in a Jupyter notebook:
  - **Menu bar** presenting different options that may be used to manipulate the way the notebook functions



  - **Toolbar** giving a quick way of performing the most-used operations within the notebook, by clicking on an icon



  - **Cell**, either "markdown" (comments, titles, etc.) or "code" (Python scripts)



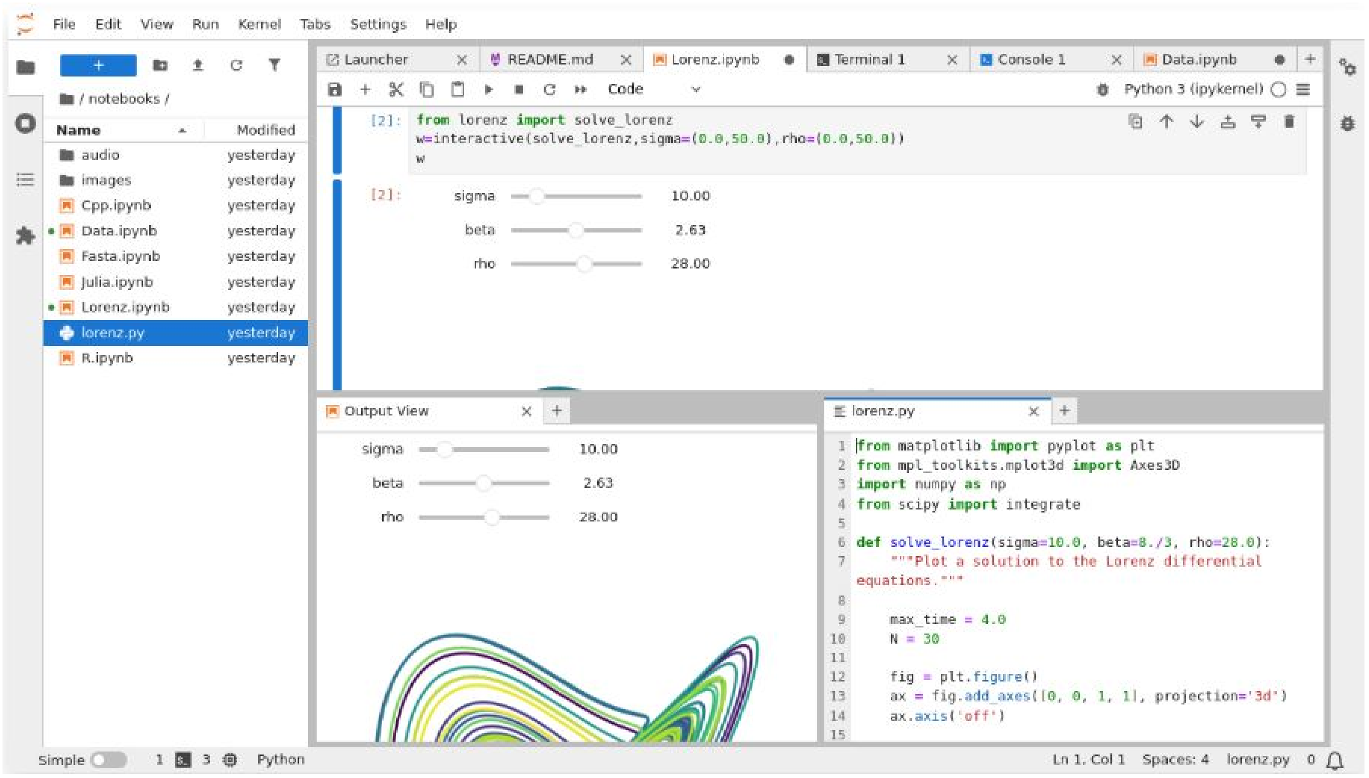
- All actions in the notebook can be performed with a mouse, but keyboard shortcuts are also available for the most common ones:
    - **Shift-Enter** to execute the current cell, show any output, and jump to the next cell below. If this is invoked on the last cell, a new cell will be created below. This is equivalent to clicking the Cell, Run menu item, or the Play button in the toolbar
    - **Ctrl-Enter** to execute the current cell, show the output and stay in the cell
- **Tutorials:**
    - jupyter-notebook.readthedocs.io/en/stable/notebook.html
    - realpython.com/jupyter-notebook-introduction/

- www.datacamp.com/community/tutorials/tutorial-jupyter-notebook
- jupyter-notebook-beginner-guide.readthedocs.io/en/latest/
- www.tutorialspoint.com/jupyter.index.htm

- In addition, extensions can be added to your notebook:
  - Info: towardsdatascience.com/jupyter-notebook-extensions-517fa69d2231
  - Run the following lines in your command line on your PC/Mac:

```
pip install jupyter_contrib_nbextensions && jupyter contrib nbextension install -user
```

- Alternatively, one could also consider **Jupyterlab**, i.e., the next-generation user interface including notebooks.



```
[2]: from lorenz import solve_lorenz
     w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
     w
```

```
[2]:      sigma   10.00
          beta     2.63
          rho     28.00
```

```
from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
from scipy import integrate

def solve_lorenz(sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential
equations."""

    max_time = 4.0
    N = 30

    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

```

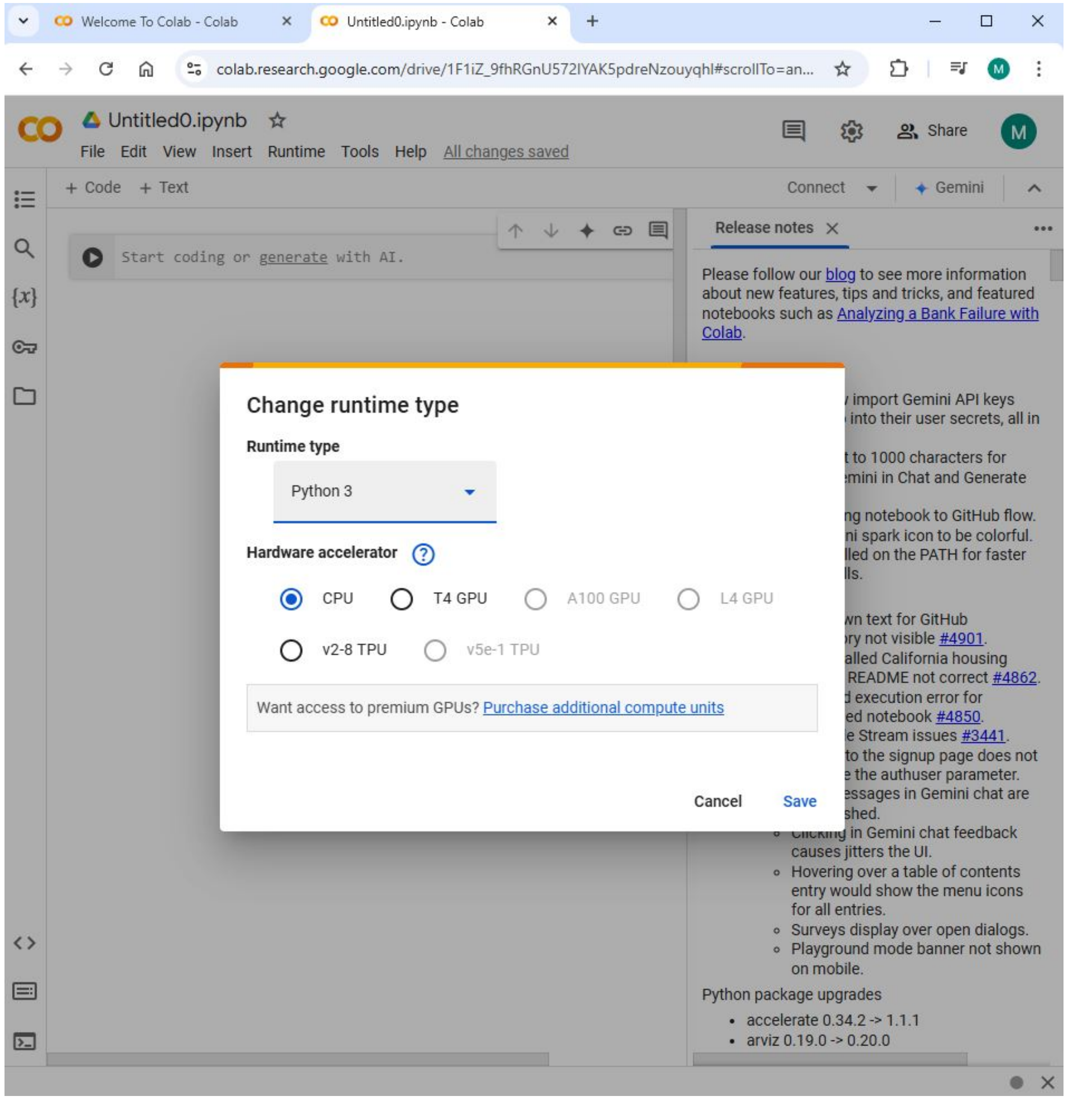
- It has a modular structure, where you can open several notebooks or files (e.g., HTML, Text, etc.) as tabs in the same window. Therefore, it offers more of an IDE-like experience

1.3 Google Colab

- While traditional Jupyter notebooks are versatile and interactive environments that allows users to integrate code, text, equations, and visualizations in a single document (notebook), it runs on local systems
  - **Constraint:** Limited computational power available
- **Question:** What if you have a computer that can't take the workload for, for example, training a machine learning model in Jupyter notebook locally?
- **Possible solution: Google Collaborate (Colab)**
  - **Google Colab** = A free cloud-based platform *provided by Google* that allows users to write and execute Python code collaboratively in a Jupyter notebook environment

- It offers free GPU and TPU access, which could be useful when significant computation power is required
- It is integrated with Google Drive (by mounting; see later), allowing users
  - To access large datasets stored in Google Drive without needing to upload them every time
  - To save their work directly to Google Drive for easy access and sharing
  - To load pre-trained models and other assets directly from Google Drive
- No setup is required, making it convenient for quick coding and collaboration

- Documentation: colab.research.google.com/notebooks/welcome.ipynb
- To get started
  - make sure you have a google account (gmail)
  - then go to this link: colab.research.google.com
- When a new notebook is created, Colab will create Untitled0.ipynb and saves it to your Google Drive in a folder named Colab Notebooks
- You can choice your runtime (CPU, GPU, TPU)


Welcome To Colab - Colab
Untitled0.ipynb - Colab
colab.research.google.com/drive/1F1iZ_9fhRGnU572lYAK5pdreNzouyqhl#scrollTo=an...
Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Share
+ Code + Text
Connect
Gemini
Start coding or generate with AI.
Release notes

Please follow our blog to see more information about new features, tips and tricks, and featured notebooks such as Analyzing a Bank Failure with Colab.

Change runtime type

Runtime type: Python 3

Hardware accelerator: [✓ CPU] T4 GPU A100 GPU L4 GPU v2-8 TPU v5e-1 TPU

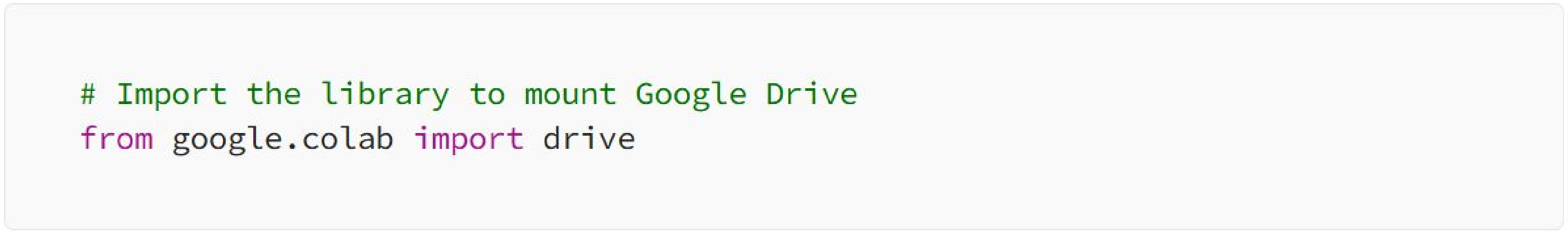
Want access to premium GPUs? Purchase additional compute units

- Clicking in Gemini chat feedback causes jitters the UI.
- Hovering over a table of contents entry would show the menu icons for all entries.
- Surveys display over open dialogs.
- Playground mode banner not shown on mobile.

Python package upgrades

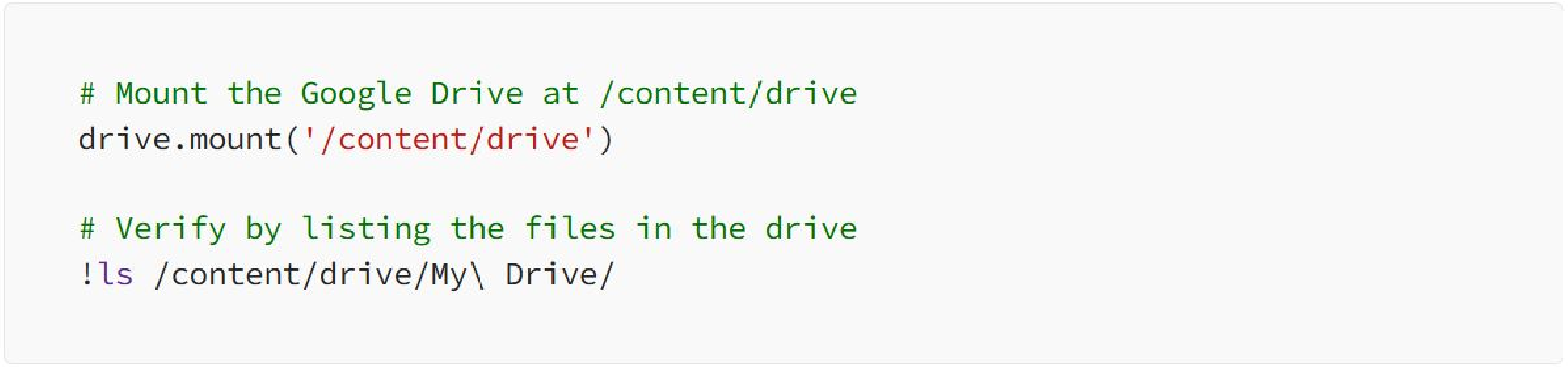
- accelerate 0.34.2 -> 1.1.1
- arviz 0.19.0 -> 0.20.0

- **Question:** How can a Google Drive be mounted in Google Colab?
- **Answer:**
  1. Import the "drive" module from the "google.colab" package. This module provides functions to interact with Google Drive.



```
# Import the library to mount Google Drive
from google.colab import drive
```

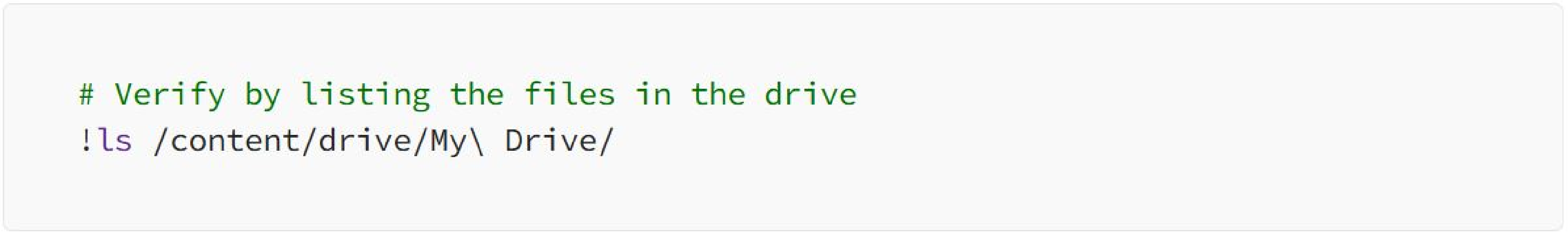
  2. Use the "mount" function to mount the Google Drive. The "mount" function requires the path where to mount the drive, which is usually '/content/drive'.



```
# Mount the Google Drive at /content/drive
drive.mount('/content/drive')

# Verify by listing the files in the drive
!ls /content/drive/My\ Drive/
```

3. When you run the above code, you will see a prompt with a link to obtain an authorization code. Follow these steps:
   1) Click on the URL provided in the output
   2) Select the Google account you want to use
   3) Allow Google Colab to access your Google Drive

4. The Google Drive files can now be accessed as if they were on your local file system. For example, to list the files in the Drive, the "ls" command can be used:

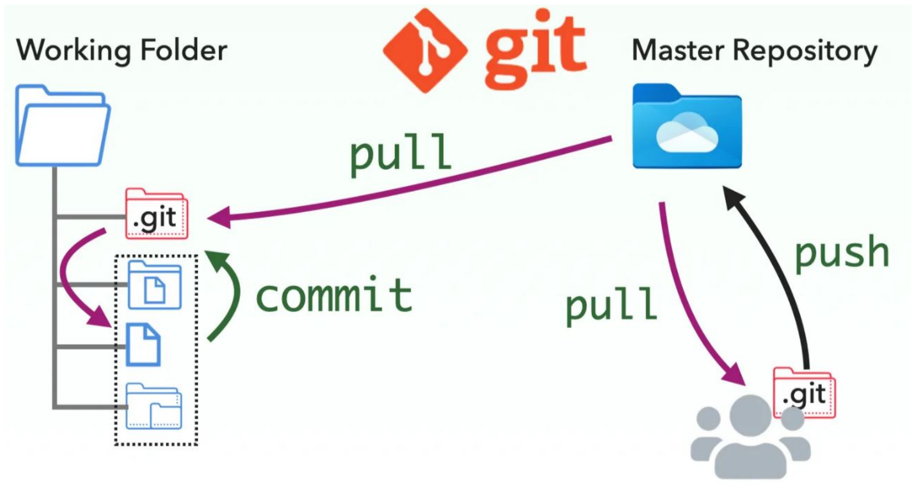


```
# Verify by listing the files in the drive
!ls /content/drive/My\ Drive/
```

1.4 Git, Github & Hugging Face

- Imagine you are working on a group project, and wishes
  - To split the task (one will work in a separate item)
  - To avoid sending files via email, etc.
  - To track changes
  - To access the code from another device
- **Question:** What kind of system is available that can ensure these demands?

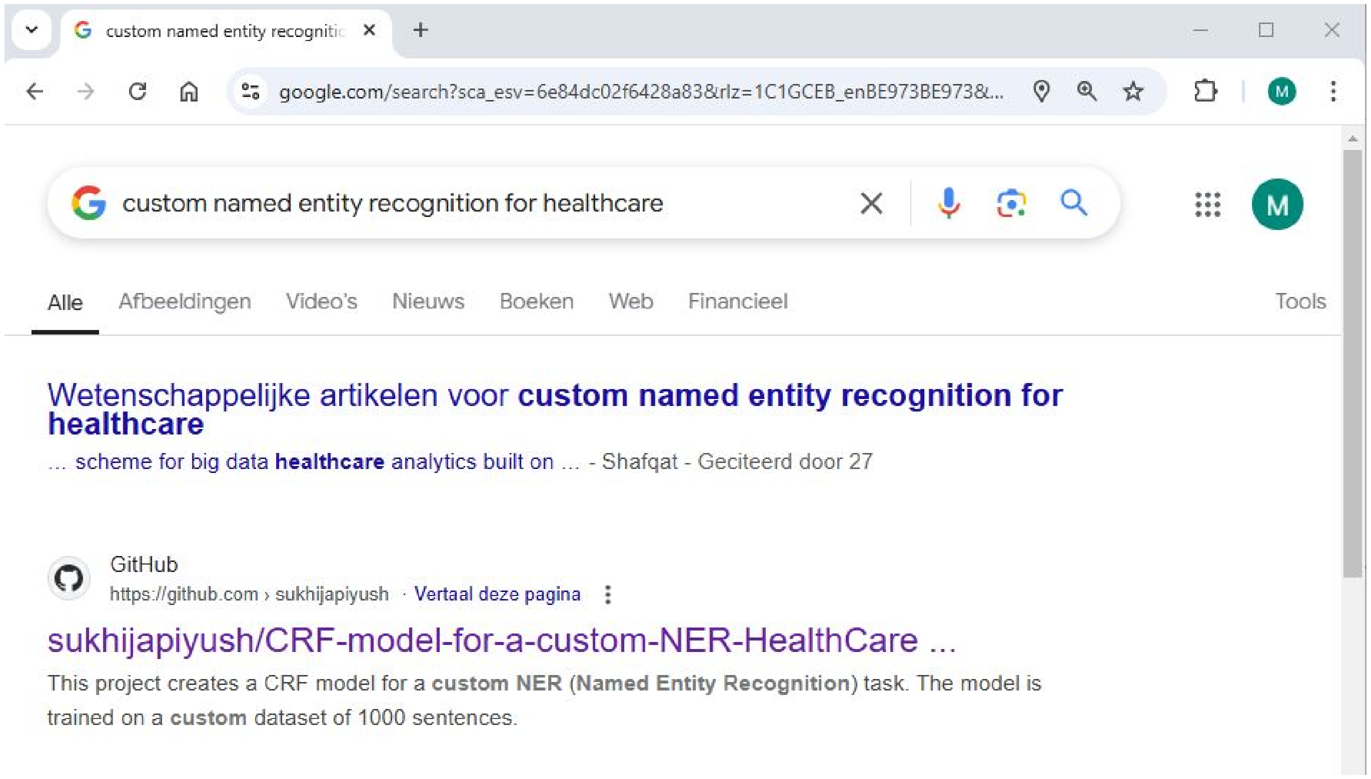
- **Solution: Git and Github**
  - **Git** was developed in 2005 by Linus Torvalds as open source software for tracking changes in a distributed version control system, i.e., a peer-to-peer type of version control where the complete codebase – including its full version history – is mirrored on every developer's computer.



  - Changes to files are tracked between computers, from one developer's workstation to another

- Compared to other version control systems, Git is
  - Responsive
  - Easy to use
  - Inexpensive
- In particular, **Git** distinguish itself from other version control systems by means of its **branching model**
  - **Branching** allows you to create independent local branches in your code
  - In other words, developers can create new ideas, set aside branches for production work, jump back to earlier branches, and easily delete merge, and recall branches at the click of a button

- To keep track of and share Git version control projects outside of your local computer/server, **GitHub**, i.e., a cloud-based database, can be used.
  - An individual's Git repositories can be remotely accessed by any authorized person, from any computer, anywhere in the world.


custom named entity recognitic
google.com/search?sca_esv=6e84dc02f6428a83&rlz=1C1GCEB_enBE973BE973&...
custom named entity recognition for healthcare
Alle
Afbeeldingen
Video's
Nieuws
Boeken
Web
Financieel
Tools

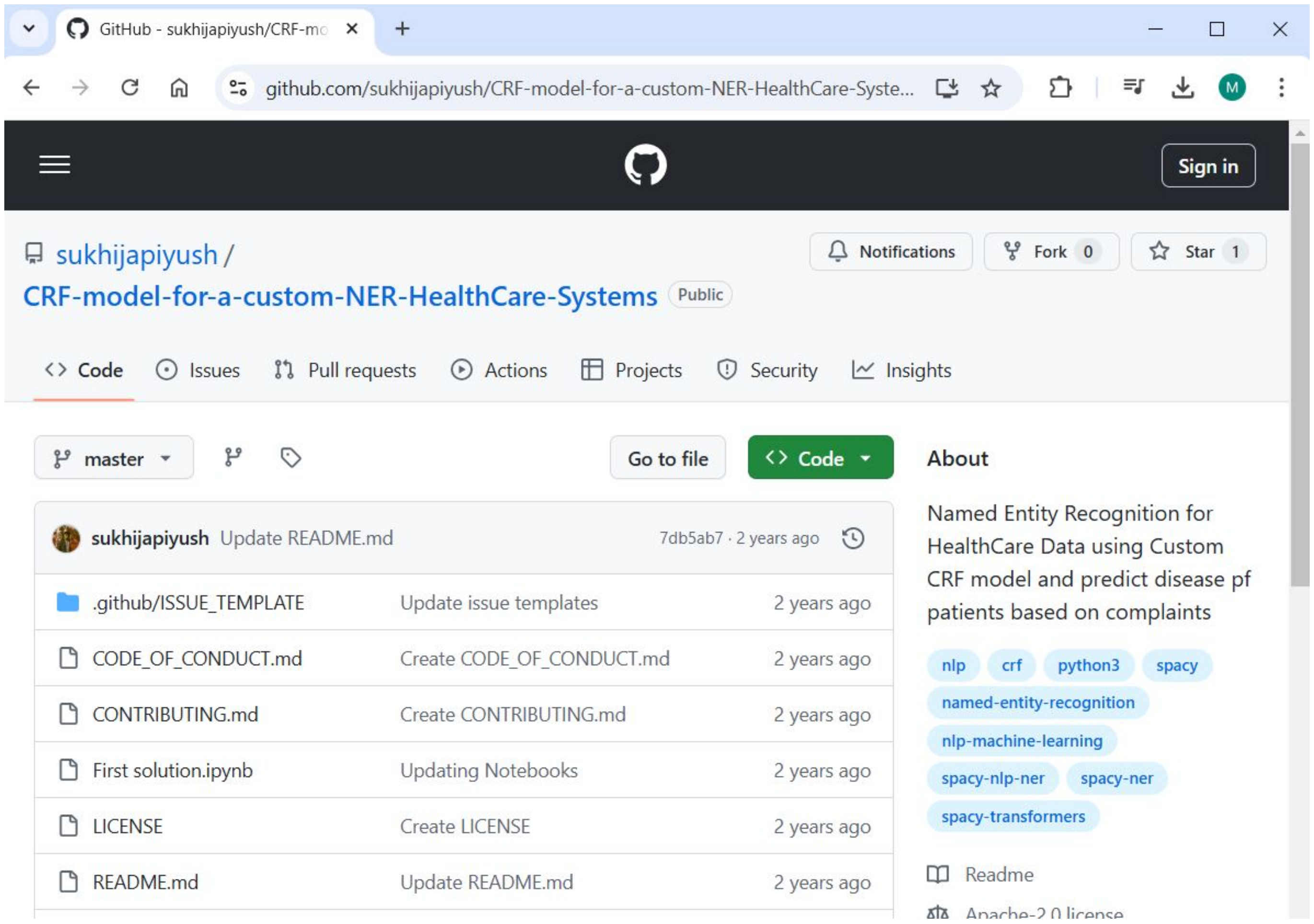
Wetenschappelijke artikelen voor **custom named entity recognition for healthcare**

... scheme for big data **healthcare** analytics built on ... - Shafqat - Geciteerd door 27

GitHub
https://github.com › sukhijapiyush · Vertaal deze pagina

sukhijapiyush/CRF-model-for-a-custom-NER-HealthCare ...

This project creates a CRF model for a **custom NER (Named Entity Recognition)** task. The model is trained on a **custom** dataset of 1000 sentences.



GitHub - sukhijapiyush/CRF-mo

github.com/sukhijapiyush/CRF-model-for-a-custom-NER-HealthCare-Syste...

sukhijapiyush / CRF-model-for-a-custom-NER-HealthCare-Systems Public

Notifications | Fork 0 | Star 1

Code | Issues | Pull requests | Actions | Projects | Security | Insights

master | Go to file | Code

sukhijapiyush Update README.md 7db5ab7 · 2 years ago

.github/ISSUE_TEMPLATE	Update issue templates	2 years ago
CODE_OF_CONDUCT.md	Create CODE_OF_CONDUCT.md	2 years ago
CONTRIBUTING.md	Create CONTRIBUTING.md	2 years ago
First solution.ipynb	Updating Notebooks	2 years ago
LICENSE	Create LICENSE	2 years ago
README.md	Update README.md	2 years ago

About

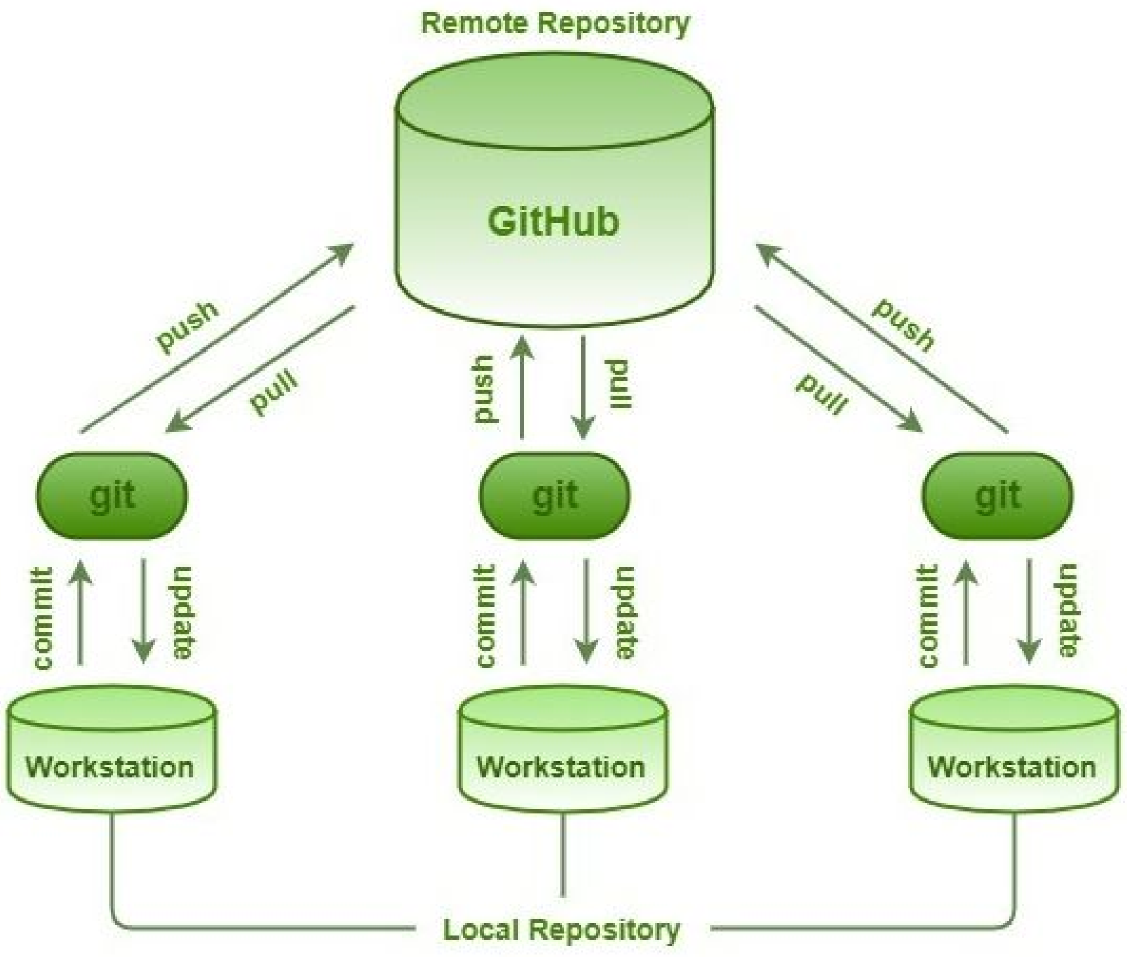
Named Entity Recognition for HealthCare Data using Custom CRF model and predict disease pf patients based on complaints

nlp crf python3 spacy named-entity-recognition nlp-machine-learning spacy-nlp-ner spacy-ner spacy-transformers

Readme

Apache-2.0 license

- Through Github, you can share your code with others, giving them the power to make revisions or edits on your various Git branches



    - As changes are introduced, new branches are created, allowing the team to continue to revise the code without overwriting each other's work. These branches are like copies, and changes made on them do not reflect in the main directories on other users' machine unless users choose to push/pull the changes to incorporate them

- It is an essential tool to know nowadays for data science/related jobs

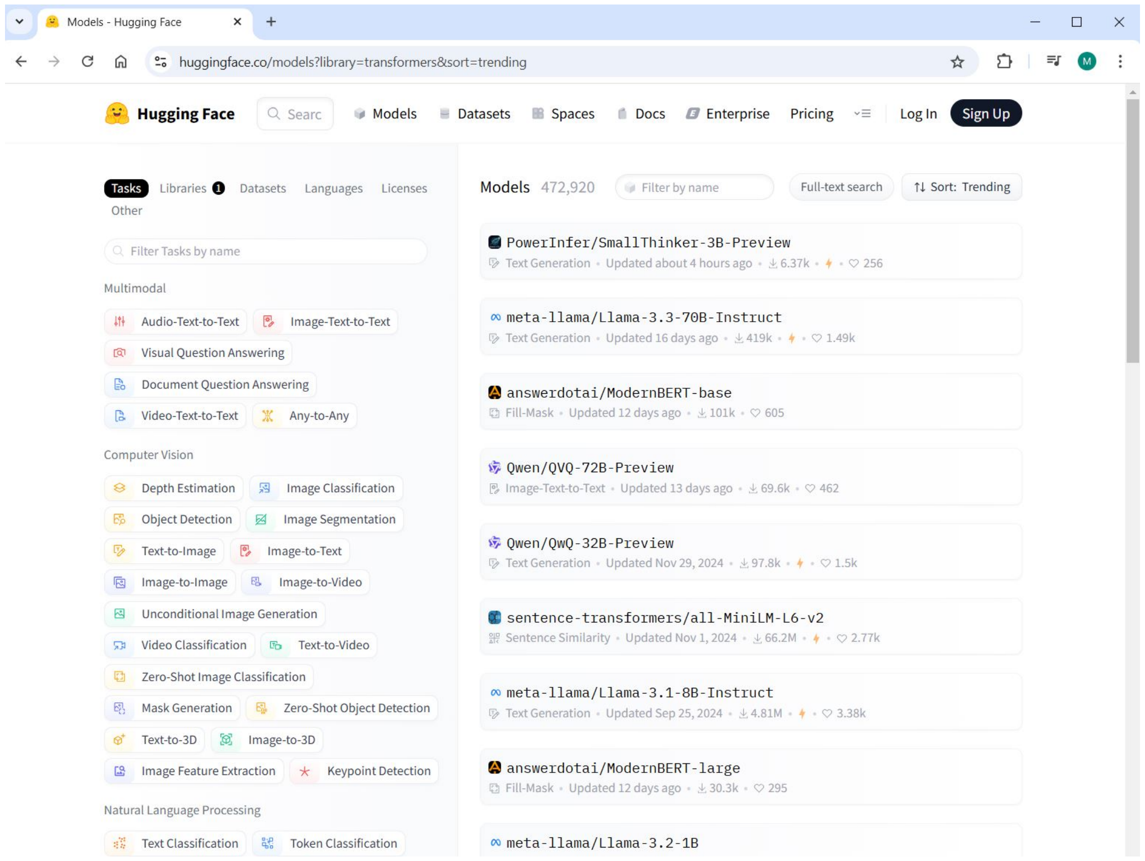
- Useful tutorial: youtube.com/watch?v=mj-qvsxPHpY

- In order to use a GitHub repository, use the following code:

```
> git clone https://github.com/zspatter/weather-forecast.git
```

  - This could be useful for considering models where the source code is available on GitHub, for example

- Another useful and popular community platform is **Hugging Face**
  - It is a powerhouse for natural language processing, computer vision and reinforcement learning, offering state-of-the-art open-source pre-trained (language) models like BERT, GPT, and many more



Models - Hugging Face

huggingface.co/models?library=transformers&sort=trending

Hugging Face | Searc | Models | Datasets | Spaces | Docs | Enterprise | Pricing | Log In | Sign Up

Tasks | Libraries 1 | Datasets | Languages | Licenses | Other

Filter Tasks by name

Multimodal

Audio-Text-to-Text | Image-Text-to-Text | Visual Question Answering | Document Question Answering | Video-Text-to-Text | Any-to-Any

Computer Vision

Depth Estimation | Image Classification | Object Detection | Image Segmentation | Text-to-Image | Image-to-Text | Image-to-Image | Image-to-Video | Unconditional Image Generation | Video Classification | Text-to-Video | Zero-Shot Image Classification | Mask Generation | Zero-Shot Object Detection | Text-to-3D | Image-to-3D | Image Feature Extraction | Keypoint Detection

Natural Language Processing

Text Classification | Token Classification

Models 472,920 | Filter by name | Full-text search | Sort: Trending

```
PowerInfer/SmallThinker-3B-Preview
```
Text Generation • Updated about 4 hours ago • 6.37k • • 256

```
meta-llama/Llama-3.3-70B-Instruct
```
Text Generation • Updated 16 days ago • 419k • • 1.49k

```
answerdotai/ModernBERT-base
```
Fill-Mask • Updated 12 days ago • 101k • 605

```
Qwen/QVQ-72B-Preview
```
Image-Text-to-Text • Updated 13 days ago • 69.6k • 462

```
Qwen/QwQ-32B-Preview
```
Text Generation • Updated Nov 29, 2024 • 97.8k • • 1.5k

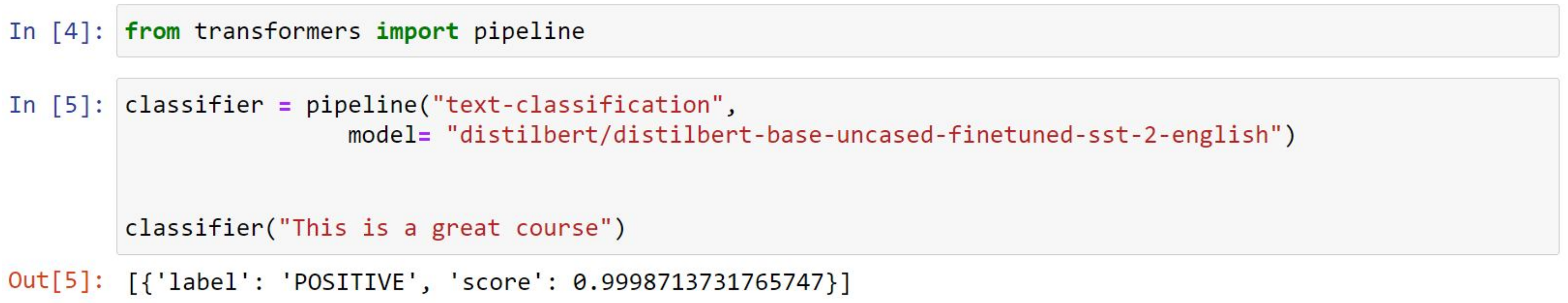
```
sentence-transformers/all-MiniLM-L6-v2
```
Sentence Similarity • Updated Nov 1, 2024 • 66.2M • • 2.77k

```
meta-llama/Llama-3.1-8B-Instruct
```
Text Generation • Updated Sep 25, 2024 • 4.81M • • 3.38k

```
answerdotai/ModernBERT-large
```
Fill-Mask • Updated 12 days ago • 30.3k • 295

```
meta-llama/Llama-3.2-1B
```

- It incorporates the state-of-the-art Transformers library, developed by Hugging Face, and can be incorporated in your Python code as follows:



```
In [4]: from transformers import pipeline

In [5]: classifier = pipeline("text-classification",
                              model= "distilbert/distilbert-base-uncased-finetuned-sst-2-english")


        classifier("This is a great course")

Out[5]: [{'label': 'POSITIVE', 'score': 0.9998713731765747}]
```

- In addition to pre-trained models, Hugging face also includes datasets, research papers & a community place for data scientists and others.

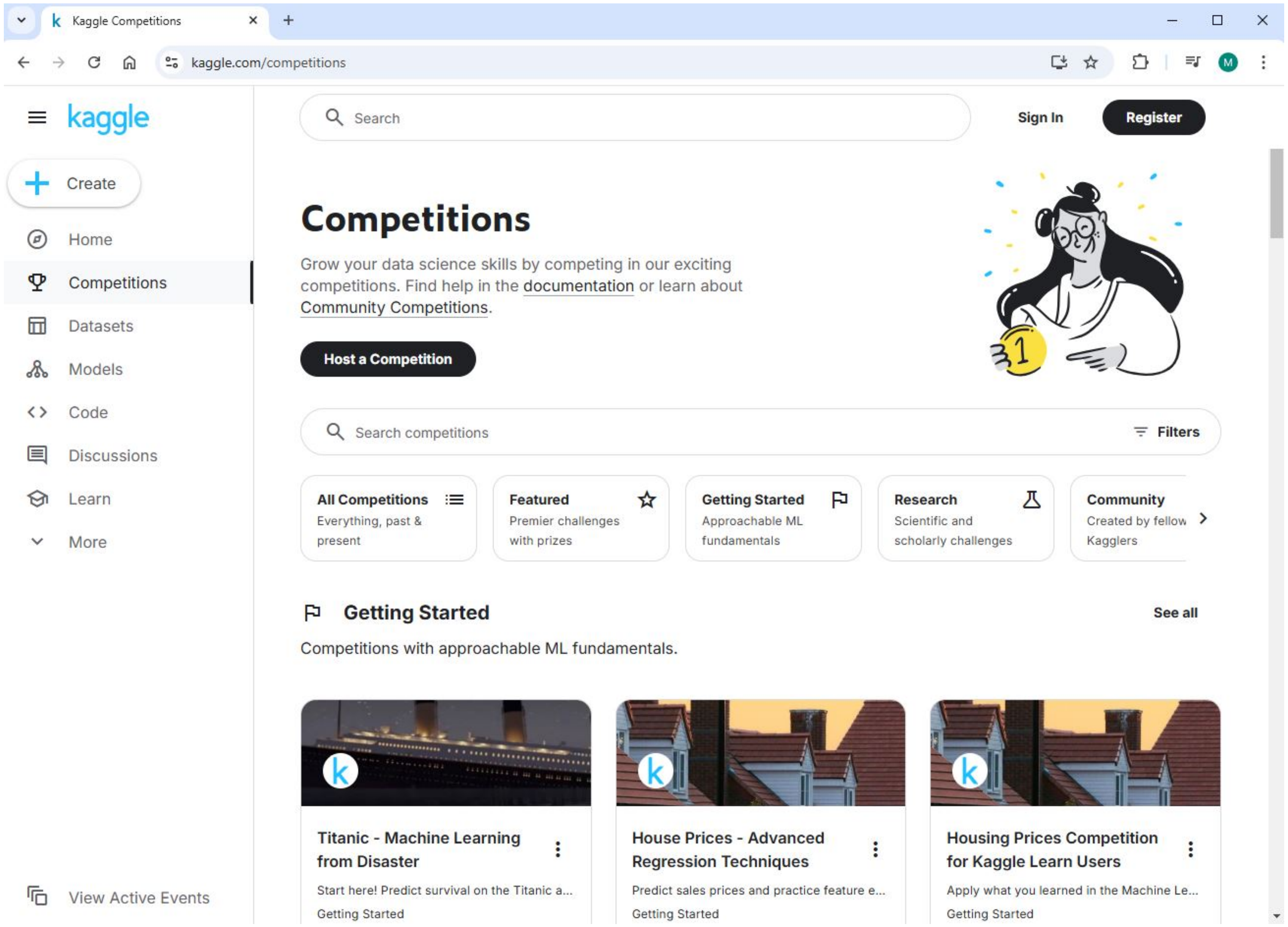
- In this course, the following GitHub repository is used for sharing Python code:

github.com/GregCollab/G0Z39A

1.5 Kaggle and UCI Machine Learning Repository

- **Question:** Does there exist other platforms, in addition to HuggingFace, that offer free datasets?

- **Answer: Kaggle and UCI Machine Learning Repository**
  - **Kaggle** is a data science competition platform and online community for data scientists and machine learning practitioners under Google.
  - **Kaggle** enables users to
    - find and publish datasets
    - explore and build models in a web-based data science environments
    - work with other data scientists
    - enter competitions to solve data science challenges



Competitions

Grow your data science skills by competing in our exciting competitions. Find help in the documentation or learn about Community Competitions.

All Competitions — Everything, past & present

Featured — Premier challenges with prizes

Getting Started — Approachable ML fundamentals

Research — Scientific and scholarly challenges

Community — Created by fellow Kagglers

Getting Started

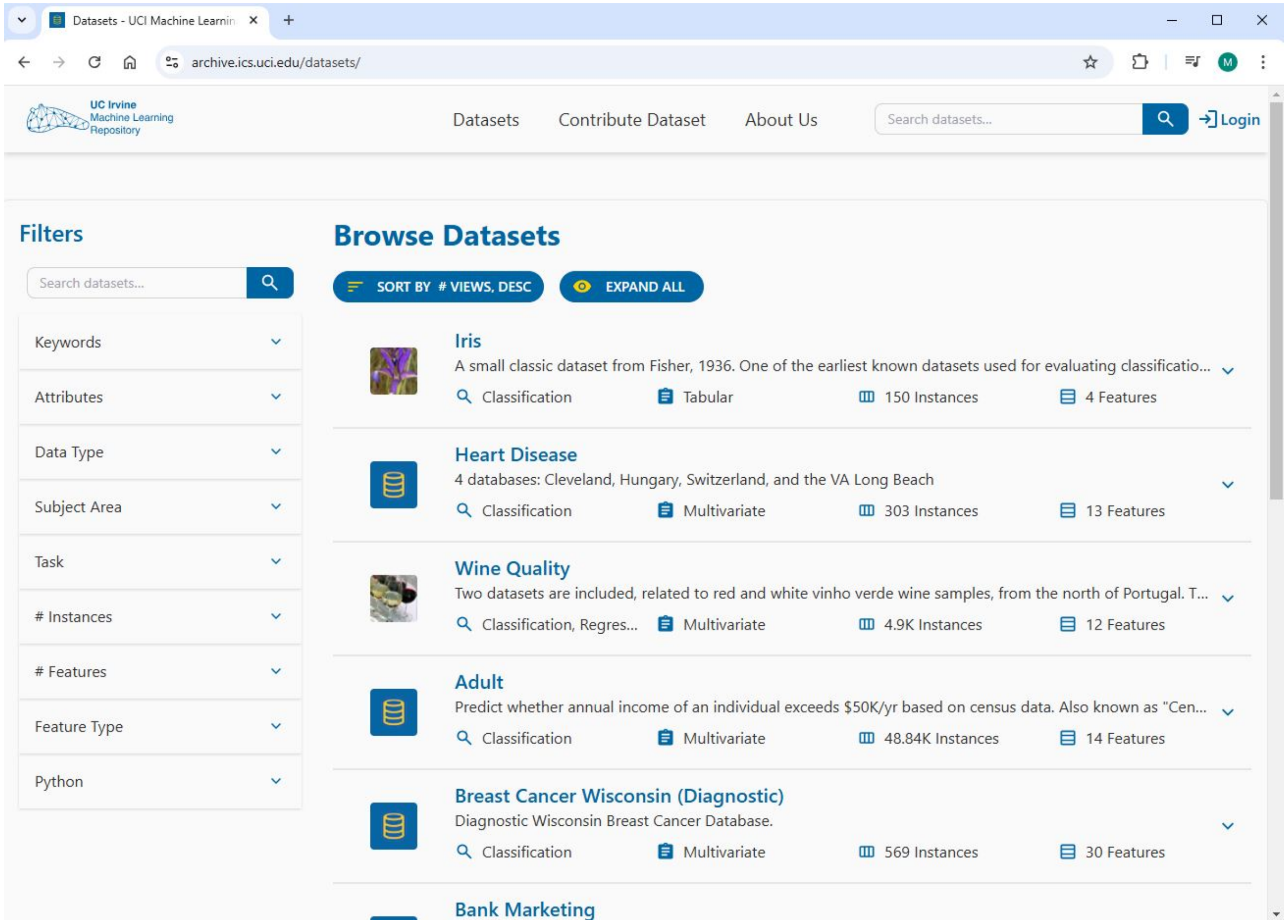
Competitions with approachable ML fundamentals.

Titanic - Machine Learning from Disaster — Start here! Predict survival on the Titanic a... — Getting Started

House Prices - Advanced Regression Techniques — Predict sales prices and practice feature e... — Getting Started

Housing Prices Competition for Kaggle Learn Users — Apply what you learned in the Machine Le... — Getting Started

- In addition, the **UC Irvine Machine Learning Repository** is available, offering datasets for *classification, regression, clustering and others*



Browse Datasets

Iris
A small classic dataset from Fisher, 1936. One of the earliest known datasets used for evaluating classificatio...
Classification | Tabular | 150 Instances | 4 Features

Heart Disease
4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach
Classification | Multivariate | 303 Instances | 13 Features

Wine Quality
Two datasets are included, related to red and white vinho verde wine samples, from the north of Portugal. T...
Classification, Regres... | Multivariate | 4.9K Instances | 12 Features

Adult
Predict whether annual income of an individual exceeds $50K/yr based on census data. Also known as "Cen...
Classification | Multivariate | 48.84K Instances | 14 Features

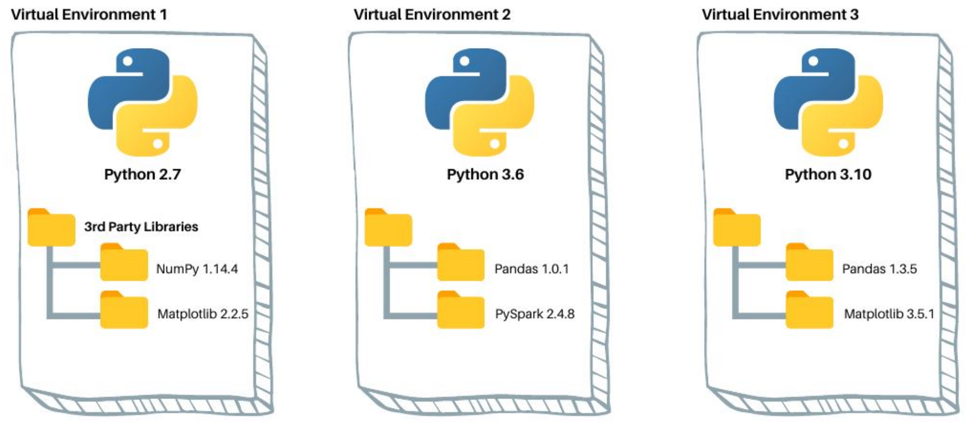
Breast Cancer Wisconsin (Diagnostic)
Diagnostic Wisconsin Breast Cancer Database.
Classification | Multivariate | 569 Instances | 30 Features

Bank Marketing

1.6 Virtual Environments

- Assume you are working on app A, using your system installed Python and you pip install packageX version 1.0 to you global Python library.
- After a while, you switch to project B on your local machine, and you install the same package packageX but version 2.0, which has some breaking changes between version 1.0 and 2.0.
- When going back to run appA, it can occur that your app does not run, and you get a lot of errors
- **Question:** How can you solve this issue?

- **Answer: Virtual Environments**
  - **Virtual environment (VI)** = Python environment s.t. the Python interpreter, libraries and scripts installed into it are isolated from those installed in other virtual environments, and (by default) any libraries installed in a "system" Python, i.e., one which is installed as part of your operating system.
  - As implication, a project in a defined VI becomes its own self contained application, independent of the system installed Python and its modules

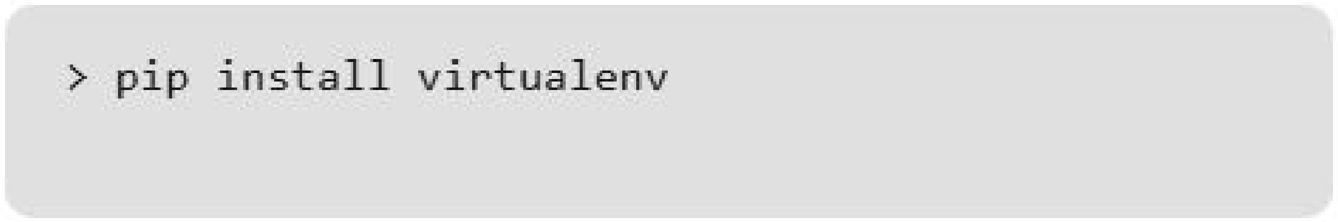


- A new VI has its own pip to install libraries, its own libraries folder, and its own Python interpreter for the Python version you used to active the environment.

- To set up a Python environment, the virtualenv package can be used.

  Steps:

  1. Install virtualenv in your command line:

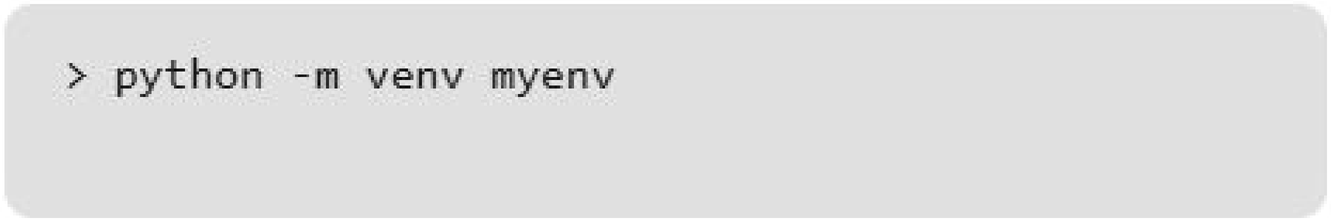


```
> pip install virtualenv
```

2. Create a new project folder, cd to the project folder in your terminal, and run the following command:

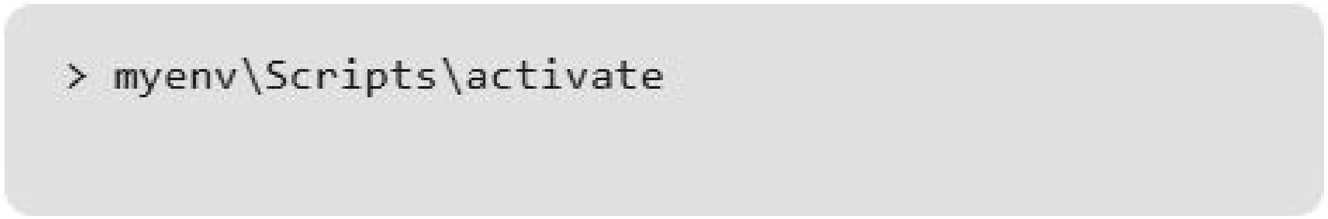
   > python<version> -m venv <virtual-environment-name>

   Example:



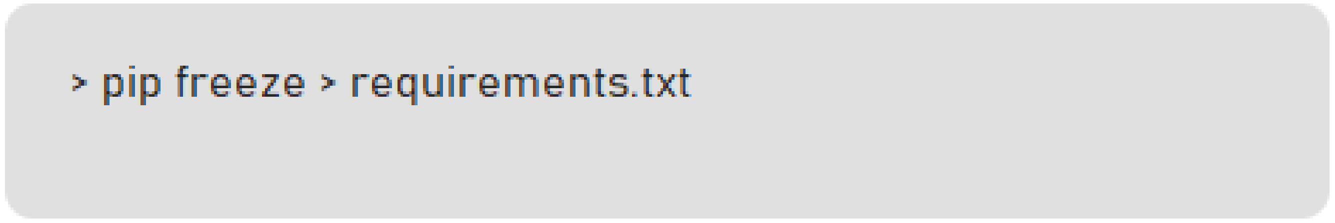
```
> python -m venv myenv
```

3. To activate the VI, use the following command:



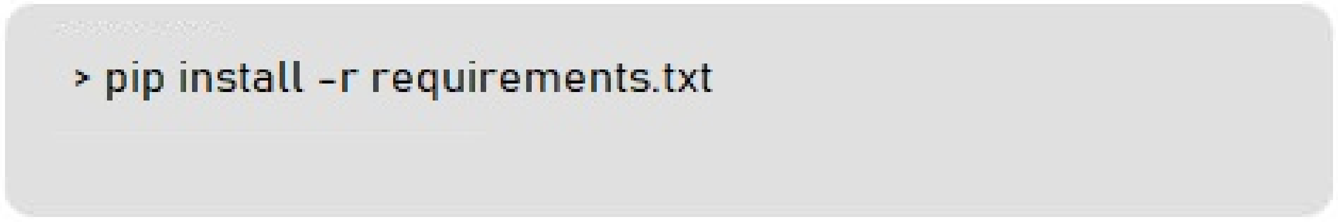
```
> myenv\Scripts\activate
```

4. To install all packages, it is best to generate a text file listing all your project dependencies:



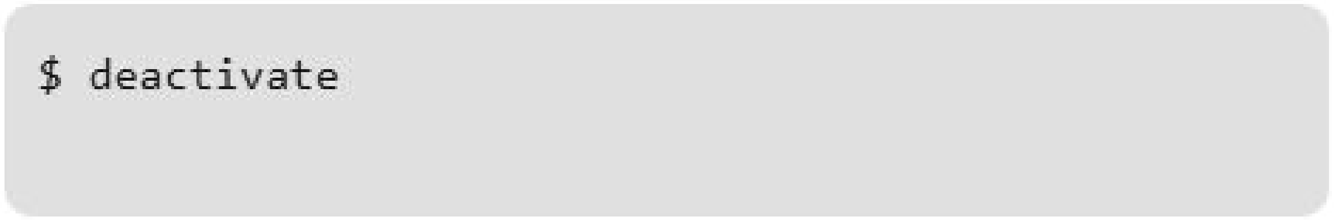
```
> pip freeze > requirements.txt
```

5. Instead of installing each dependency one by one, all dependencies listed in the requirements.txt file can be installed directly using the following command:



```
> pip install -r requirements.txt
```

6. To deactivate the virtual environment, use the following command:



```
$ deactivate
```

- **Documentation:**
  - bit.ly/3h4f8ty
  - bit.ly/3r5KUL4

1.7 Object Oriented Programming

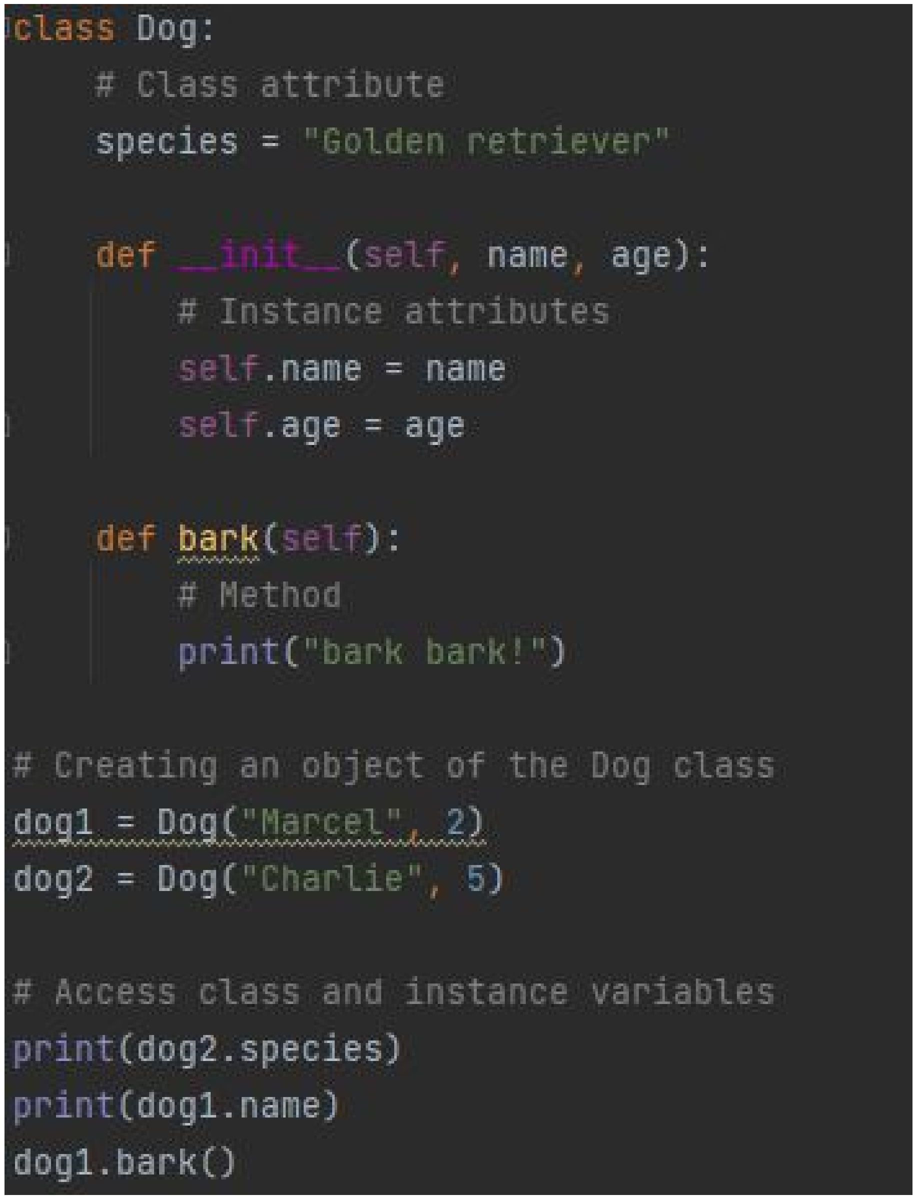
- Python is an **Object Oriented Programming (OOP) language**

- **OOP** is a style of programming used to build modular, maintainable, and scalable applications

- It is a way of organizing code that uses classes (representing concepts) and objects (instances of classes) to represent real-world entities and their behavior
  - Classes can be thought of as a 'blueprint' for objects. These can have their own attributes (characteristics they possess), and methods (actions they perform)

    Example: Consider the class "Dog"
    - Dogs usually have a name and age, i.e., instance attributes

    - Dogs can also bark, i.e., a method



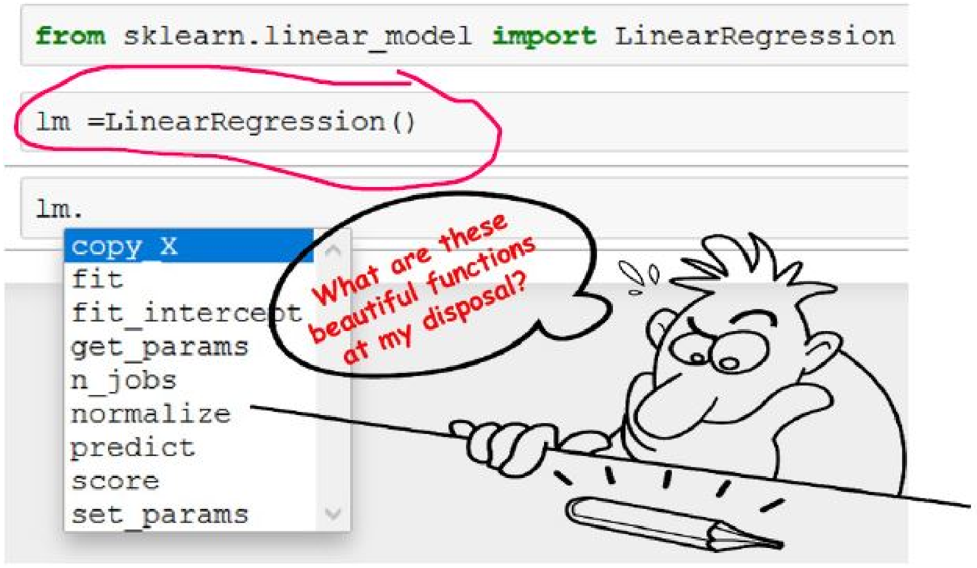
```
class Dog:
    # Class attribute
    species = "Golden retriever"

    def __init__(self, name, age):
        # Instance attributes
        self.name = name
        self.age = age

    def bark(self):
        # Method
        print("bark bark!")

# Creating an object of the Dog class
dog1 = Dog("Marcel", 2)
dog2 = Dog("Charlie", 5)

# Access class and instance variables
print(dog2.species)
print(dog1.name)
dog1.bark()
```

- **Remark 1:** All classes have a function called $\_\_init\_\_()$, which is always executed when the class is being initiated. It is used to assign values to object properties, or other operations that are necessary to do when the object is being created.

- **Remark 2:** $self$ parameter is a reference to the current instance of the class. It allows us to access the attributes and methods of the object.

- Key data science libraries such as pandas, numpy, and scikit-learn all heavily rely on OOP

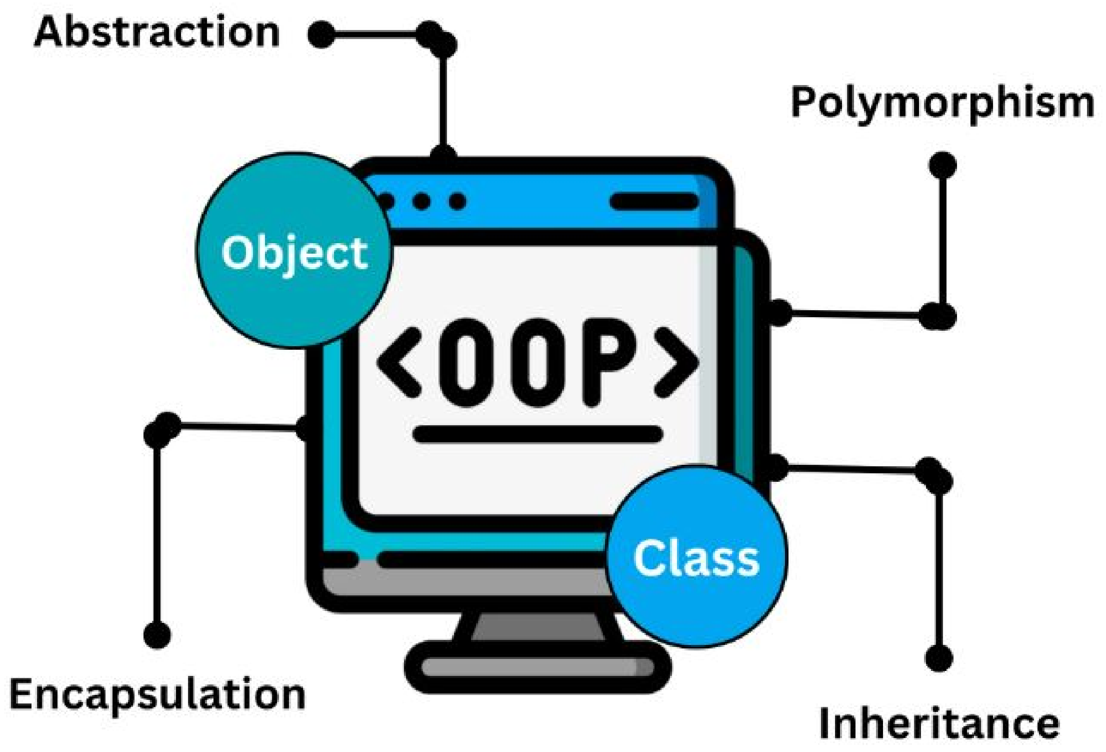
- **Question:** How can you see this?



```
from sklearn.linear_model import LinearRegression

lm =LinearRegression()

lm.
```

- **Answer:** In scikit-learn, for example, a regression model is as an instance of a class, and it has a fit() method to train your machine learning model or a predict() method for prediction

- OOP allows objects to interact with each other using four basic principles:



- These four principles enables objects to communicate and collaborate to create powerful applications

- **Tutorials:**
  - bit.ly/38AE7Rj
  - bit.ly/3ryOqOa
  - bit.ly/3poTrHg
- Let's get convinced with a simple example, i.e., build a linear regression model (as you always do) or using OOP

  $\rightarrow$ **Linear_regression.ipynb**