

Data Science And Machine Learning

1

(Otherwise known as ML1, DS1, or MLDS1, or ...)



Founded by Harvard and UCLA faculty, a true online alternative
to the world's top institutions for AI & ML.

What is Data Science¹

Its everything you need to do to analyse data.

And thats a ...lot...

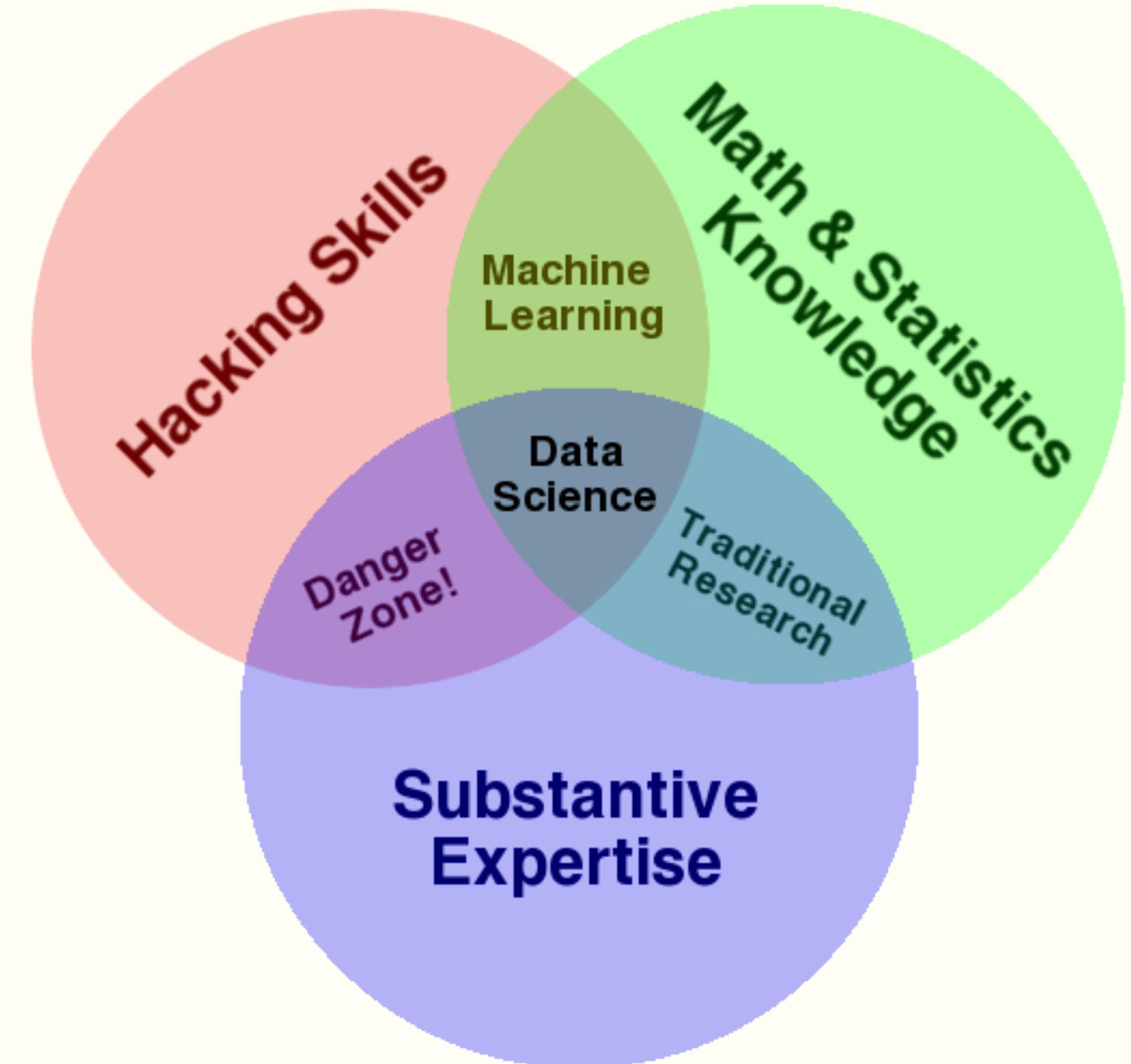
Its not just statistics or machine learning.

Its not just business analytics and large data.

Its not just subject expertise, or science.

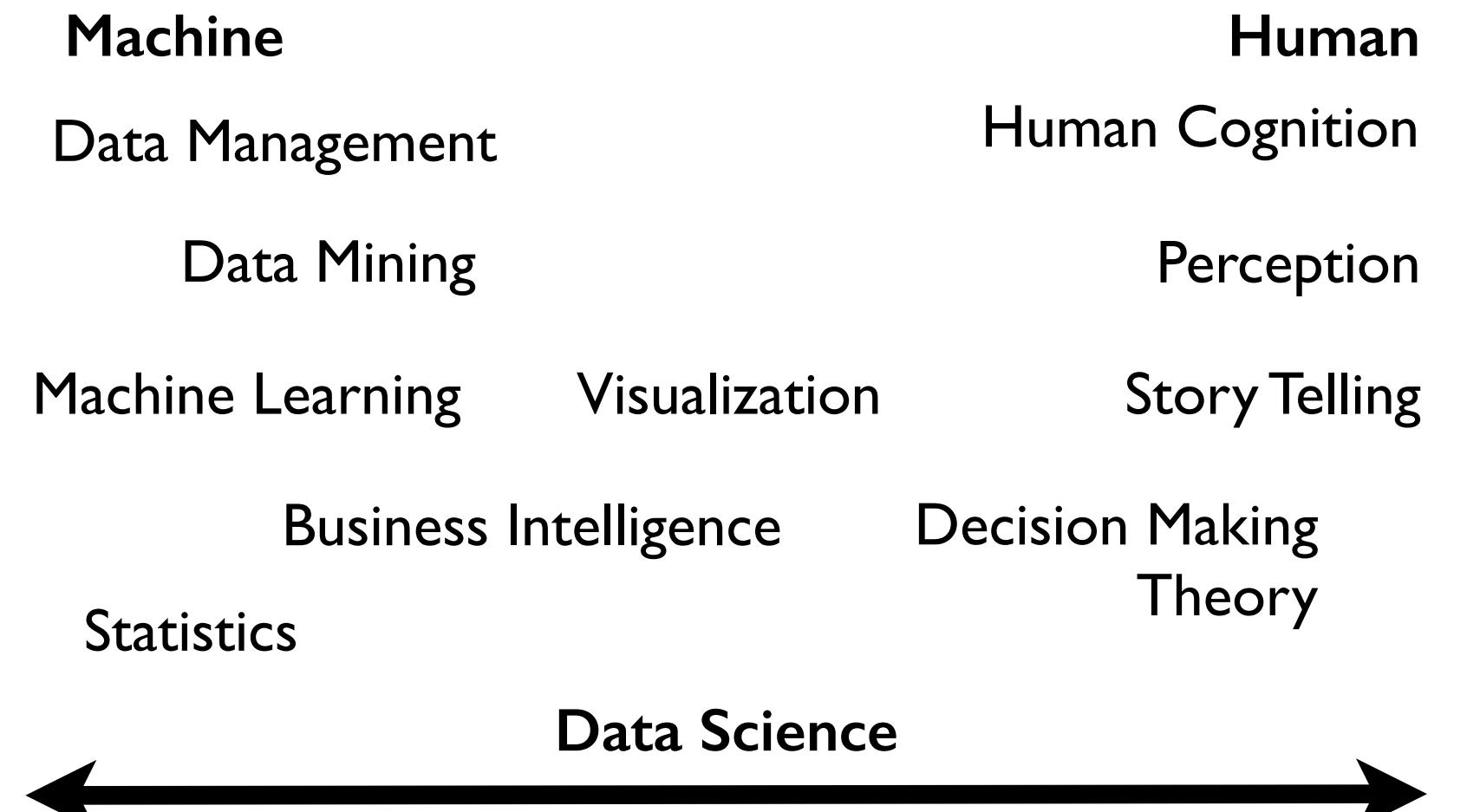
Its not just databases and data engineering.

Its ALL of these.



¹ Diagram by Drew Conway

Things going into Data Science²



Inspired by Daniel Keim, "Visual Analytics: Definition, Process, and Challenges"

In this (and the subsequent ML-2) course we'll study:

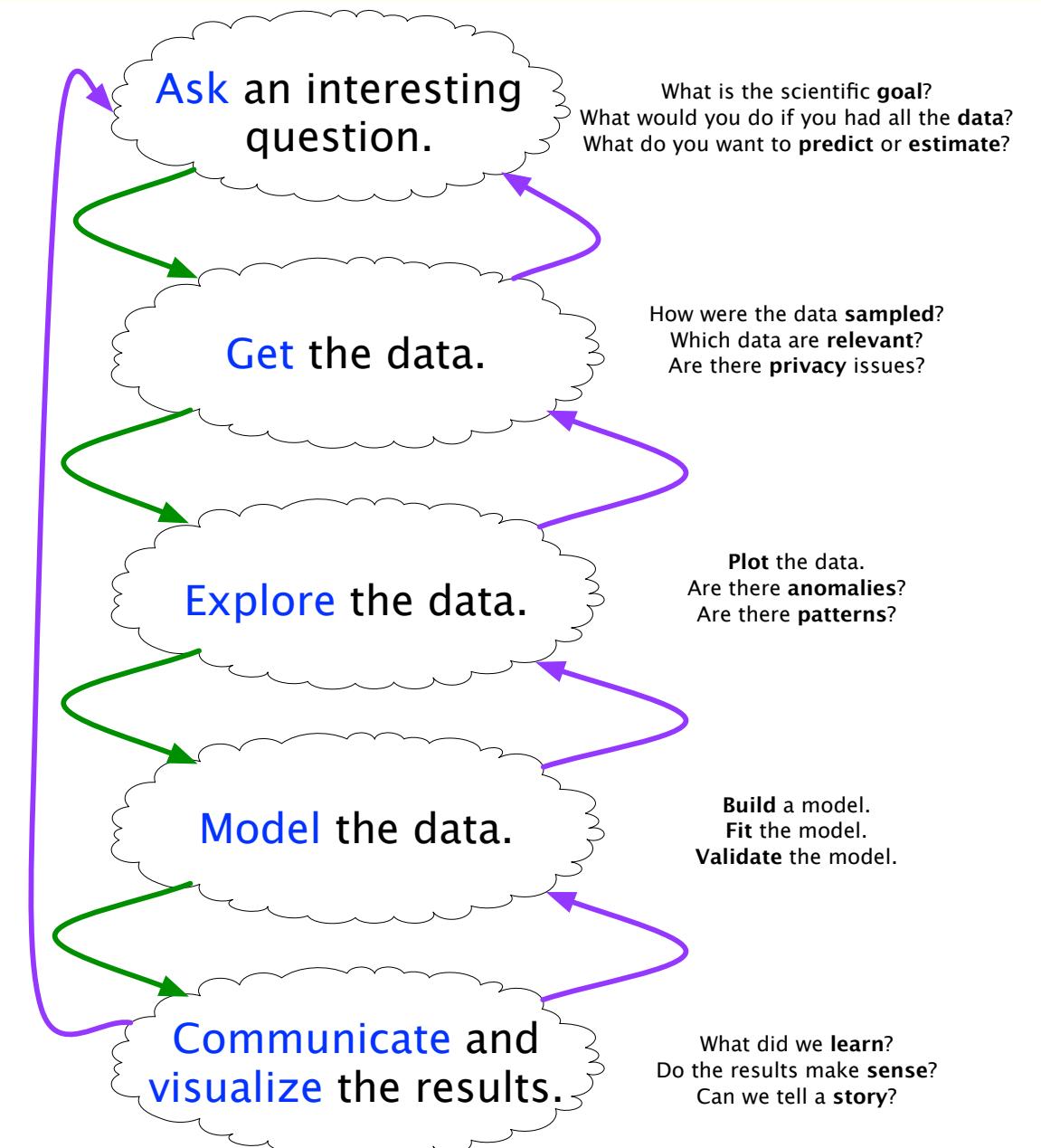
- Data Management
- Machine Learning
- Statistics
- Visualization
- Decision Making Theory
- Story Telling

²From cs109, 2015 edition

Data Science Process Ladder³

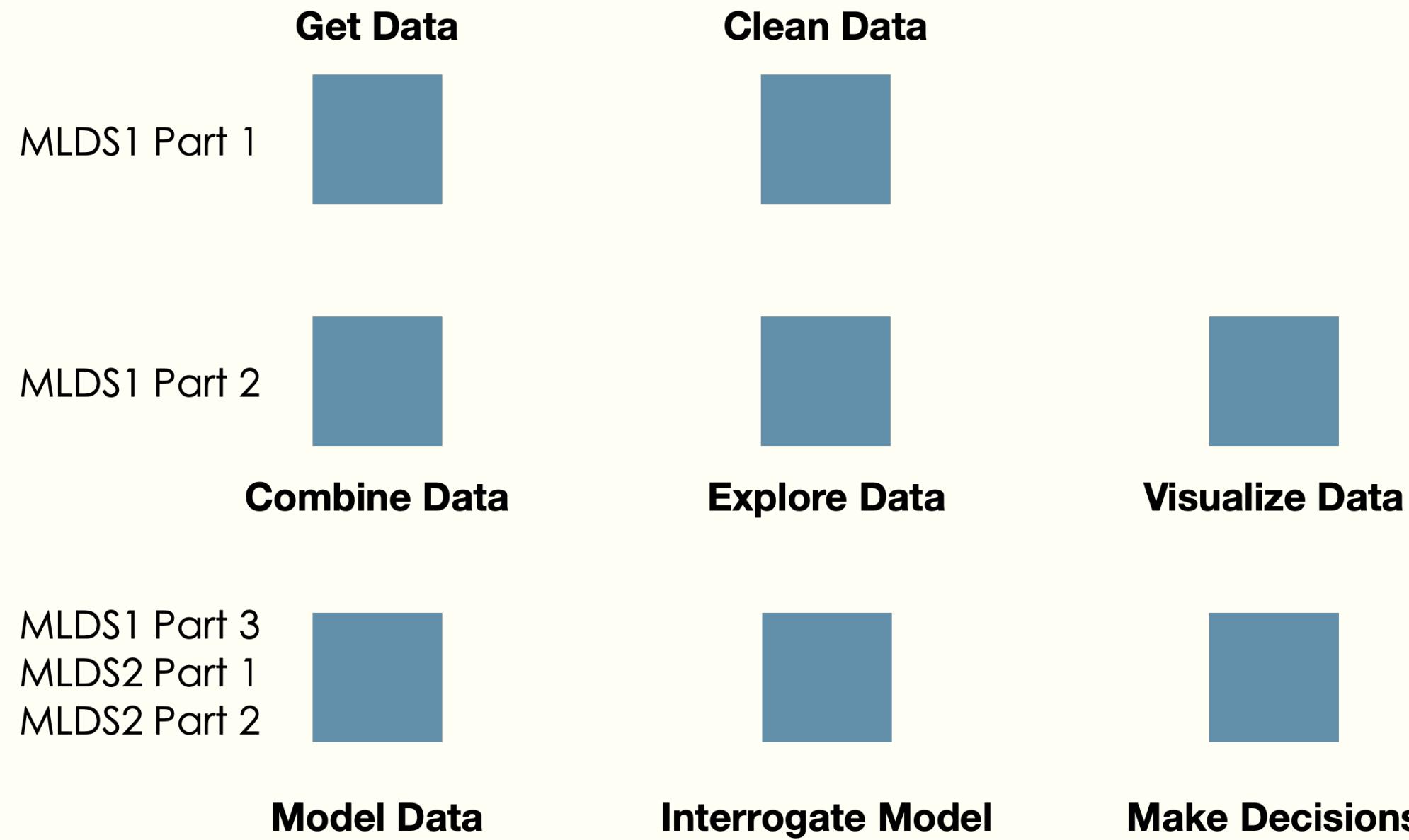
- You first ask a question, or have a dataset that needs sleuthing.
- You need to get data, or even additional data
- Now you must clean the data, deal with missing data, combine the data
- Next you want to explore, summarize, and visualize the data, understand its patterns and properties, its correlations
- After that you want to model the data, perhaps running multiple models
- Then you want to compare and interrogate these models, once again exploring and visualizing
- Finally you want to communicate the results and take decisions based on them

NOT ALWAYS LINEAR: You move up and down this ladder



³ From cs109, 2015 edition

The ML-X series: ML-1 and ML-2



What?

	Session	Topics Covered
Week 1	1	Introduction to Data, Pandas, SQL
	2	Types of Data: html/http/text/graph
Week 2	3	Cleaning of data: Parsing, cleaning with pandas, transforms
	4	Exploratory Data Analysis, Matplotlib,seaborn
Week 3	5	Data Visualisation: principles and communication
	6	End to end model with pipes, LR metrics
Week 4	7	Collaborative filtering, similarities
	8	Recommendation engines

Data Science NEEDED Skills

- programming data: python, perhaps R
- programming apps: web framework, html, perhaps javascript, SQL/postgres
- programming ops: git+github, unix (git-bash on windows)
- statistics: basic statistics, regression, probability and distributions
- machine learning: logistic regression, trees, recommendations, clustering

Increasingly, you are expected to be familiar with

- devops: running stuff in the cloud (AWS, GCP, Azure), reproducible experiments and environments with docker
- deep learning: at-least how to use pre-trained models.

Our Philosophy for ML-X

We're going to throw you in at the deep end of the swimming pool.

Unlike Ai-X, where you work mostly on Ed, here you will switch between discourse for forums, Anaconda Python on your own machine, Colab for GPU accelerated notebooks, github for code from our lessons, the terminal for unix commands, and our platform for slides/videos/notebooks integrated.

You will pick up some of the skills above on the fly, and some we will be explicit about teaching you by scheduling sessions, and in lab.

We want you to emerge a polyglot capable of multiple things, and more importantly with the attitude and confidence that you can pick them up quickly.

Who?

RAHUL



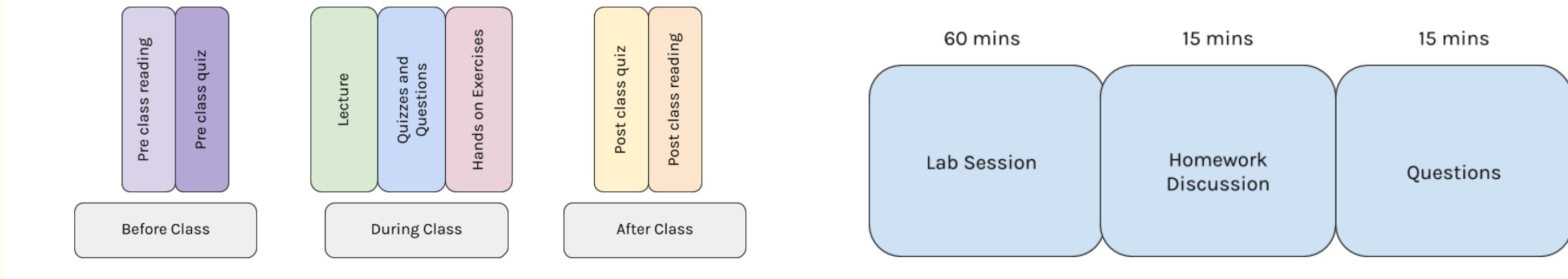
ANUSHA



ARYA



How?



And anytime on our forums.

Schedule

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
Week 1	25	26	27	28	29	30	31	January
	Welcome Session	Session 1	Office Hour (with Professor)		Session 2	Lab 1 HW 1 Released		
Week 2	1	2	3	4	5	6	7	February
	Office Hour (with Teaching fellows)	Session 3	Office Hour (with Professor)		Session 4	Lab 2 HW 1 Due HW 2 Released		
Week 3	8	9	10	11	12	13	14	February
	Office Hour (with Teaching fellows)	Session 5	Office Hour (with Professor)		Session 6	Lab 3 HW 2 Due HW 3 Released		
Week 4	15	16	17	18	19	20	21	February
	Office Hour (with Teaching fellows)	Session 7 Project Released	Office Hour (with Professor)		Session 8	Lab 4 HW 3 Due		
Week 5	22	23	24	25	26	27	28	March
	Office Hour (with Teaching fellows)		Office Hour (with Professor)				Project Due	
Week 6	1	2	3	4	5	6	7	March
			Open House for Project					

When?

Lecture Sessions:

- Tuesday Series: 9:00 PM - 11:00 PM IST
- Friday Series: 9:00 PM - 11:00 PM IST

Lab Sessions:

- Saturday Series: 7:30 PM - 8:30 PM IST

Office hours:

- Mondays: 9:00 PM - 10:00 PM IST
- Wednesdays: 9:00 PM - 10:00 PM(IST Professor office hours)

And anytime on our forums.

Grading Breakdown

- Quiz/Exercises: 29%
- Participation: 12%
- Homework: 39%
- Project: 20%

There are three homework assignments in this course.

- Homework 1 will be released after Lecture 2,
- Homework 2 after Lecture 4, and
- Homework 3 after Lecture 6.

Participation Breakdown

The participation is overall 12%.

- Lectures - 3%, you get the complete score if you attend 75% of the lectures, and also actively participate by asking or answering questions.
- Labs - 3%, you get the complete score if you attend 75% of the labs, and also actively participate in by asking or answering questions.
- Forum Discussion - 5%, complete score if you answer fellow student questions and/or your questions and answers are endorsed by a teaching staff
- Office hours - 1%

Deadlines

The deadline for the homework is before the next even lecture i.e. duration of one week.

For example, consider the first homework is released on 30th January 2021, 21:00 hours IST. The deadline for the submission is 6th February 2021, 17:00 hours IST.

There will be no extension given for the submission unless there are exceptional circumstances.

The quizzes that you work on during the main class sessions and the pre-class quiz can be attempted multiple times. The **post class quiz however can be submitted only once**.

The exercises that you worked on during the main sessions can be attempted multiple times. You may not finish the exercise in the time allotted in class. You can attempt/continue the exercises after class but you MUST submit them by 5 PM IST on the day of the next lecture session with the Professor.

Lets Get Started!

Our [welcome.univ.ai](#) has all this information in more detail.

1. Download and install anaconda on your machine if you have not already. If you are doing this now, lets do the rest while it is downloading.
2. Have you created a Univ.AI account? You must have to sign up for this course. Go to [Our Platform](#). You should find our course (ML-1) as one of your courses. Explicitly, its [here](#).
3. Log onto discourse. This will require a Univ.AI account. Go to [discourse.univ.ai](#). Our class's forum is [here](#)
4. Do you have a github account? If not, go to [github](#) and create one. Then link your github-id from your Univ.AI profile [here](#).
5. Log onto edstem. You should have received an invitation. The specific link for this class is: <https://edstem.org/us/courses/4024/>

The platform: course.univ.ai

Our lectures are split up into chapters. Each chapter is accessible on the platform, and typically involves some slides and supporting notes, rendered on the platform from a jupyter notebook.

The chapters are combined together in a coherent navigation scheme, roughly reflecting the organization of the lectures.

You will notice that there is easy ability to open a discussion on any chapter and see the results of the discussion right in the UI. This is of-course in addition to any other questions you might have on the lecture.

A screenshot of a web application interface. At the top right, there is a sidebar with a 'Open discussions' button, a 'HTTP And API' section containing links to 'Introduction', 'Slides', 'Notebook', and 'Summary', and a 'HTML And Scraping' section with a 'DOWNLOAD LINKS' button. The main content area features a large, bold, black 'HTTP' title. Below the title is a dark grey code editor window containing Python code:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

The code editor has a small 'Univ.AI' logo in the top left corner. There are also some small, faint icons in the bottom right corner of the editor window.

Studying using the platform

The screenshot shows a presentation slide with the following content:

HTTP Status Codes
(from <http://www.garshol.priv.no/download/text/http-tut.htm>)

- 200 OK: Means that the server did whatever the client wanted it to, and all is well.
- 201 Created: The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field.
- 400: Bad request The request sent by the client didn't have the correct syntax.
- 401: Unauthorized Means that the client is not allowed to access the resource. This may change if the client retries with an authorization header.
- 403: Forbidden The client is not allowed to access the resource and authorization will not help.
- 404: Not found Seen this one before? :) It means that the server has not heard of the resource and has no further clues as to what the client should do about it. In other words: dead link.
- 500: Internal server error Something went wrong inside the server.
- 501: Not implemented The request method is not supported by the server.

The HTTP protocol

Our notebooks also talk to a local web server on our machines:
<http://localhost:8888/Documents/ml-1/>
BLA.ipynb#something

- protocol is http, hostname is localhost, port is 8888
- url is /Documents/ml-1/BLA.ipynb
- url fragment is "#something"

Request is sent to localhost on port 8888. It says:
Request:
GET /Documents/ml-1/BLA.ipynb HTTP/1.0
Univ.AI

Using `requests`

You might need to retrieve some data from the Internet. Python has many built-in libraries that were developed over the years to do exactly that (e.g. urllib, urllib2, urllib3).

However, these libraries are very low-level and somewhat hard to use. They become especially cumbersome when you need to issue POST requests or authenticate against a web service.

Luckily, as with most tasks in Python, someone has developed a library that simplifies these tasks. Get acquainted to `requests` as soon as possible, since you will probably need it in the future.

```
import requests
```

Now that the `requests` library was imported into our namespace, we can use the functions offered by it.

In this case we'll use the appropriately named `get` function to issue a `GET` request. This is equivalent to typing a URL into your browser and hitting enter.

We typically assign our notes as post lecture reading. The platform allows you to go over these notes. As can be seen in the image, the slides are synchronised with the notes so you can learn from the notes in the context of the lecture.

The slides can also be perused in full screen mode right from the platform.

We have developed (alpha software) the ability to run code right in the platform, and the ability to add personal notes in the platform itself. We'll probably turn these features on at some point to see if you find them useful.

Discussions are on Discourse

Log onto discourse by using your Univ.AI id. You will see an icon representing you in the upper right corner. You can access your profile from there.

The screenshot shows the Univ.AI forum homepage. At the top, there are navigation links: 'all categories' (with a dropdown arrow), 'Categories' (highlighted in yellow), 'Latest', 'New (4)', 'Unread (18)', and 'Top'. Below these are two tabs: 'Category' and 'Topics'. Under 'Category', there is a link to 'Course: Machine Learning and Data Scie...'. Under 'Topics', there is a link to 'Welcome to Discourse'. On the right side of the page, there is a search bar, a user icon with a notification count of 2, and a 'New Topic' button.

The first *category* there is the forum for our course.

Our class's forum is <https://discourse.univ.ai/c/course-machine-learning-and-data-science-1/24>

We have created some threads. Please Introduce yourself in the introduction thread.

The screenshot shows a list of topics in the 'Course: Machine Learning and Data Science-1 C1' category. The topics are: 'Session Recordings' (general), 'How to use Ed and Discourse?' (general), 'Github, Anaconda Navigator and colab support!', 'Introduce yourself!' (introduction), and 'About the Course: Machine Learning and Data Science-1 C1 category'. Each topic has a small profile picture, a reply count (0 or 1), a view count (2 or 9), and a timestamp (22h or 1d). At the top of the list, there are buttons for 'Edit', '+ New Topic', and a bell icon.

Links to the video recordings are provided in the Session Recordings thread.

Some useful features in discourse.

When you create a new topic, you can tag it. Please do so, it enables a faster response from us. We will create tags and threads for homework and labs.

Session Recordings •
general
How to use Ed and Disc general
general x2
Github, Anaconda Navig introduction x1
Introduce yourself! introduction
+ Create a new Topic
Type title, or paste a link here
Course: Machine Learning and Data Sci... optional tags
homework
finding_partner
+ Create Topic cancel

Introduce yourself! edit

Course: Machine Learning and Data Science-1 C1 introduction

Type here. Use Markdown, BBCODE, or HTML to format. Drag or paste images.

Here you can see that posts in the Introduction thread are tagged **introduction**.

You can send private messages to other users. This is great to raise private questions with the teaching staff, like when u want a clarification on a question in a homework but a public answer would be inappropriate as it would leak further details about your implementation to others.

rahuldave edit

Rahul Dave

Summary Activity Notifications

New Message

Inbox

Click on your icon, on then on the envelope icon on the popup. At the page that presents, click on New Message to send a message to one or more specific users.

Github

We'll be using github for you to access lesson material on your own computers, and for Homework and Lab work, and Homework submissions.

There is a github organization for this class. Our repositories are private in there: you may NOT share them outside of this class. We have asked for your github ids so we can add them to the organization.

The organization is [univai-ml1-c1](#). New repositories will show up there.

The screenshot shows the GitHub organization page for 'univai-ml1-c1'. The organization has 5 repositories, 7 people, 4 teams, and no projects. The repositories listed are:

- HTMLAndScraping** (Private): Jupyter Notebook, 0 forks, 0 stars, 0 issues, 0 pull requests, Updated 2 days ago
- HTTPAndAPI** (Private): Jupyter Notebook, 0 forks, 0 stars, 0 issues, 0 pull requests, Updated 2 days ago
- DataAndBases** (Private): Jupyter Notebook, 0 forks, 0 stars, 0 issues, 0 pull requests, Updated 2 days ago
- TabularDataModel** (Private): 0 forks, 0 stars, 0 issues, 0 pull requests, Updated 2 days ago
- ml1-c1** (Private): 0 forks, 0 stars, 0 issues, 0 pull requests, Updated 2 days ago

On the right side, there are sections for 'Top languages' (Jupyter) and 'People' (a list of users).

How to work with Github

The screenshot shows a GitHub repository page for 'univai-ml1-c1/HTMLAndScraping'. The repository is private, has 2 pull requests, 0 stars, and 0 forks. The 'Code' tab is selected. A commit from 'rahuldave' titled 'initial' was made 2 days ago. The commit message is 'Initial Commit'. The repository contains files: data, images, .gitignore, README.md, SUMMARY.md, distribute_html_and_s..., html_and_scraping.ipynb, and slides.pdf. The 'About' section notes that no description, website, or topics were provided. It includes links to Readme, Releases (no releases published), and Packages (no packages published).

File	Content	Last Commit
data	initial	2 days ago
images	initial	2 days ago
.gitignore	Initial Commit	2 days ago
README.md	initial	2 days ago
SUMMARY.md	initial	2 days ago
distribute_html_and_s...	initial	2 days ago
html_and_scraping.ipynb	initial	2 days ago
slides.pdf	initial	2 days ago

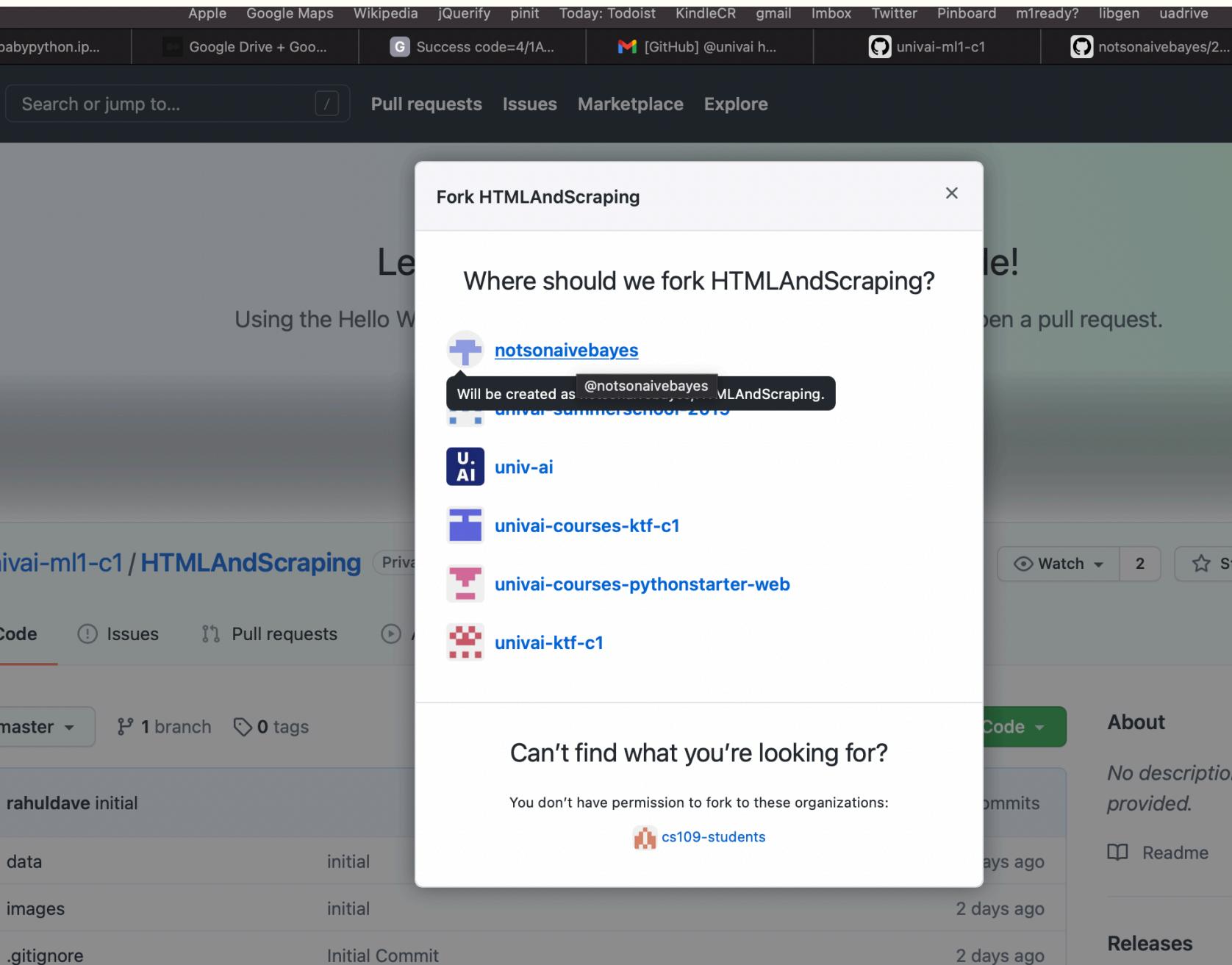
The idea behind providing you your own copy of our code, slides, etc is that you can edit the code, and make notes on the slides, and do other such fun things.

Thus you must bring our code into your own namespace on github.

Our lectures are made up of chapters. Each chapter is a repository on github, and comes with its own notes and slides. The notes are in jupyter notebooks and map to the notes on the platform. You can run and play with these notes on your own computer.

The way to do this is something called **forking**. On any repository, you will find a button on the upper right that says Fork. Click this, and you can bring the repository into your own namespace.

Forking process



<-- On forking you might get a popup.

Choose your account to fork into, and then you will have created a copy of the repository in your own namespace. So, univai-ml1-c1/REPONAME ->

A screenshot of a GitHub repository page for "notsonaivebayes / HTMLAndScraping". The page shows basic repository details: 1 branch, 0 tags, and 2 commits by rahuldave. The commits are for "data" and "images" files. The "About" section notes "No description, website, or topics provided." and has a "Readme" link. The "Releases" section is empty.

Now you can click the big green button and download your code using git (more on that later) or as a zip file.

Edstem

Our edstem is at <https://edstem.org/us/courses/4024/> .

The screenshot shows the Edstem platform interface. At the top, there is a purple header bar with the 'ed' logo and the course name 'univ.ai ML 1 – Lessons'. To the right of the header are several icons: a person icon, a bar chart icon, a gear icon, a house icon, a bell icon with a '20+' notification, and a user profile icon. Below the header, the word 'Lessons' is centered in a large, light gray box. At the bottom of the screen, there is a navigation bar with a search bar containing 'Search lessons', and buttons for 'Grid', 'List', 'New Module', and 'New Lesson'. The 'New Lesson' button is highlighted in blue. On the left side of the main content area, there is a section titled 'Sessions' with a person icon. Below this, there is a list of four sessions, each with a document icon and a status indicator (a small circle with a dot or a crossed-out circle):

- Course Information
- Session 0: Welcome
- Session 1: Introduction to Data, Pandas and SQL
- Session 2: Types of Data: html/http/text/graph

We are using EdStem for 2 purposes:

(1) Providing a Lecture Storyboard which gives you an idea of what is happening in each lecture. This Storyboard is also accessible at the main class github repo in our organization: [ml1-c1](#).

Storyboard

Before class

- Pre-class reading - 15 mins
- Pre-class Quiz - 15 mins

During Class

- Session
- In-class Quiz
- Session
- In-class exercise
- Session

After Class

- Post-class reading (30 mins)
- Post-class Quiz (15 mins)

(2) In-class and outside class quizzes are conducted on edstem, which makes it easy for us to grade the quizzes. Additionally, each class has a 1/2 hour long exercise done in breakout rooms..this is run on Ed as well.

Question 1

```
import pandas as pd
ds = pd.Series([2, 4, 6, 8, 10])
print("Pandas Series and type")
print(ds)
print("Convert Pandas Series to Python list")
print(_____)
```

To **convert** the Pandas series into a list what needs to go in the blank?

(ds)

(ds.tolist())

(ds.listto())

(type(ds.tolist()))

Exercises in Edstem.

Format: You will be provided with Description on the left side and the exercise on the right. Note that:

1. The exercises will be in fill in the blanks format, you will be asked to fill in ().
2. The exercises need to get marked after completion to check if the code filled by you is correct or not, hence do not forget to mark the exercise on completion.
3. In case you want to re - attempt the exercise and would like to clear your answers then the three dots on top are your friend. Click on that and then click on Reset to scaffold, this resets your challenge and allows you to reattempt and resubmit.
4. **You can attempt the exercises as many times as you like!!**

The screenshot shows a challenge titled "Convolutional Filters". The interface includes a navigation bar with "Challenge", "Submissions", "Solution (hidden)", "Edit Slide", and a three-dot menu. The main area has tabs for "Description", "Code", "Text", and "Commands". The "Code" tab is active, showing Python code for convolution. A red box highlights the three-dot menu icon in the top right corner of the slide header. Another red box highlights the code block containing the convolution function and its test. A third red box highlights the "Mark" button in the bottom right corner of the slide footer.

Description:

Convolutional Filters

The aim of this exercise is to understand convolution and convolutional filtering. We will perform convolution using different kernels for:

1. Blurring
2. Sharpening
3. Vertical Edge Detection
4. Horizontal Edge Detection
5. Edge Detection

The above are just a few examples of effects achievable by convolving kernels and images.

Instructions:

- Define a convolution function to convolute two 2D matrices by performing pixel-wise multiplication and addition.
- Load, normalize and resize the images.
- **Blurring:**
 - Perform blurring using the given 3x3 kernel.
 - Change the kernel size (say 2x2 or 5x5) and observe the output.
- **Sharpening:**

```
[ ] for y in range(image.shape[1]):  
    output[x,y]=((kernel * image_padded[x:x+kernel.shape[0], y:y+kernel.shape[1]]).sum(2))  
    return output  
  
[ ] ### edTest(test_cnn1) ###  
#Perform convolution of any two 2D matrices and observe the output  
  
#Define two 3x3 matrices for convolution  
img=np.array([[1,2,3],[4,5,6],[7,8,9]])  
  
kernel=np.array([[1,0,-1],[2,0,-2],[1,0,-1]])  
#Call the convolution function previously defined  
conv_img = convolve(img,kernel)  
print(conv_img)
```

Now that we understood what convolution entails, lets try to understand how convolution is used for:

1. Blurring
2. Sharpening
3. Vertical Edge Detection
4. Horizontal Edge Detection
5. Edge Detection by combining Horizontal and Vertical Edge Detection

Load the images

/home/Ex1_Scaffold.ipynb

Notebook saved

✓ Mark

Quizzes in Edstem

Question 1

Which of the following is NOT a valid use-case of Transfer Learning in neural networks:

- Layer substitution** - Replace current neural network layers with *most significant* layers of pre-trained model
- Model Architecture**- Use the pre-trained model architecture only, and initialize all weights
- Feature extractor** - Using the entire pre-trained network as a fixed feature extractor for the new dataset
- Partial training** - *Freeze* initial layers of the pre-trained model, while retraining the higher layers

Submit

Question 2

A ConvNet, trained on [ImageNet](#) which contains 1.2 million images with 1000 categories) is an excellent *pre-trained model* for **any** image classification problem.

- True

Quizzes in Edstem will be of various kinds - few will be Multi choice questions, others might be of short answers etc. You will be provided instructions on how to answer each quiz question.

Once you have selected the answer you have to click on Submit

Please note: Quizzes that are in-class/pre-class/ can be attempted multiple times, but quizzes that are post class can be only attempted **once**. Hence answers those questions carefully!

Lab/Homework Submission

(1) Lab need to be done either locally on your systems or on google colab, hence all the labs will be provided with a github repo to download the exercise from.

(2) For you to submit your lab, you need to upload your solution on github, and **share the repository with us using Ed.** *Labs are NOT graded,* so this is just for you to have us look at your lab.

(3) You will do this by clicking on files, and choosing new notebook - creating a new notebook called *submission.ipynb* and adding a text cell with your github repository link and then click on **Submit**

The screenshot shows a Google Colab interface titled "Lab Exercise - MLP". At the top, there are navigation links: "Challenge", "Submissions", "Solution (hidden)", "Edit Slide", "Prev", and "Next". Below the title, there's a "Description" section with the text: "Lab Exercise - MLP" and "The aim of this lab is to get you familiar with Google Colab and tensorflow.keras. Go to the following [link](#) to access the colab notebook.". To the right of this text is a "Files" button, which is highlighted with a red box. In the bottom right corner of the interface, there is a "Submit" button with a blue checkmark, also highlighted with a red box.

ercise - MLP

★ Challenge ⌚ Submissions ✓ Solution (hidden) Edit Slide ... Prev Next

submission.ipynb

+ Code + Text ▶ Run All ■ Stop Python

Github Repository Link:

[]

/home/submission.ipynb Notebook saved

✓ Submit

The screenshot shows a Jupyter Notebook interface. At the top, there's a dark header bar with navigation links like 'Challenge', 'Submissions', 'Solution (hidden)', 'Edit Slide', and 'Prev/Next'. Below the header, the left sidebar has tabs for 'Outline' and 'Files'. In the main area, there's a file list with 'submission.ipynb' selected. The notebook editor shows a cell with the text 'Github Repository Link:' and a code cell placeholder. The status bar at the bottom indicates the file is saved and shows a 'Submit' button.

(4) For Homework submissions the process is the same as Labs

(5) The entire process for doing homework/lab is:

1. Get homework by clicking on the link on Ed or going directly to <https://github.com/uniavai-ml1-c1>
2. Go to Github to fork the Homework file
3. work on your machine or in colab and update your repository
4. When finished, share the Repository on Ed using instructions in the previous slide.

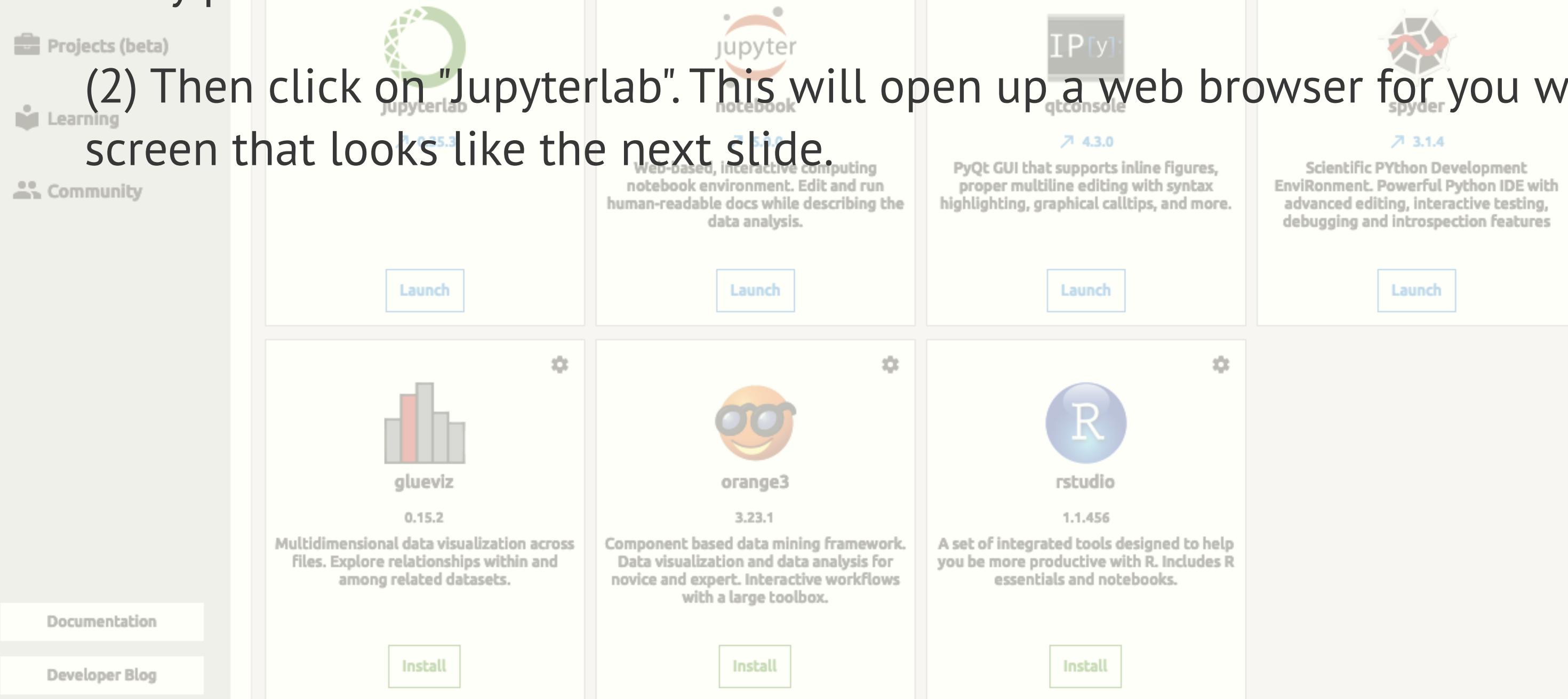
Anaconda: or how to access your code

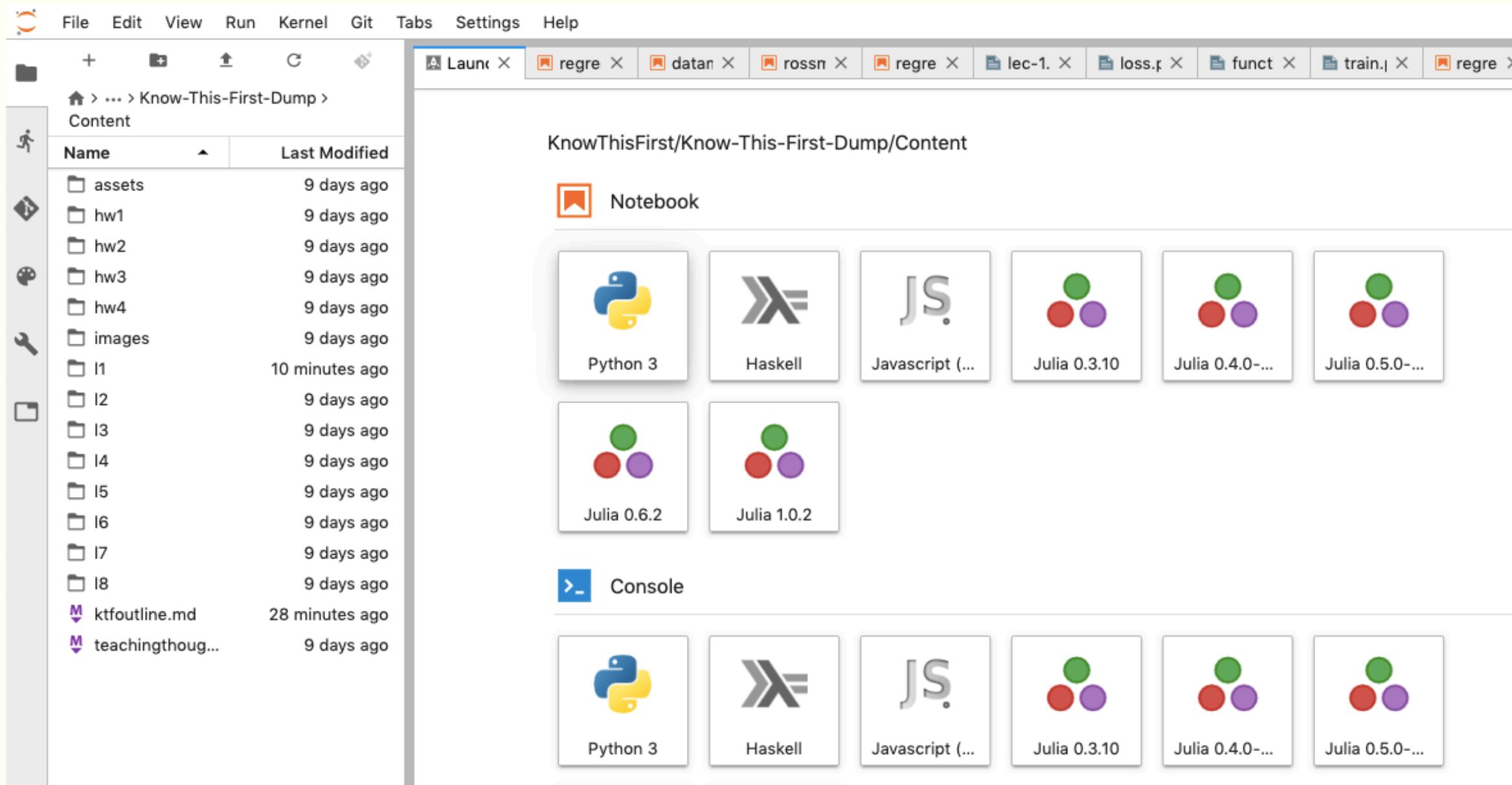
Install Anaconda on your local machine. Our [welcome.univ.ai](#) has links to an install video.

We will be using python 3. Make sure you atleast install 3.7. This is the case with the latest version of Anaconda.

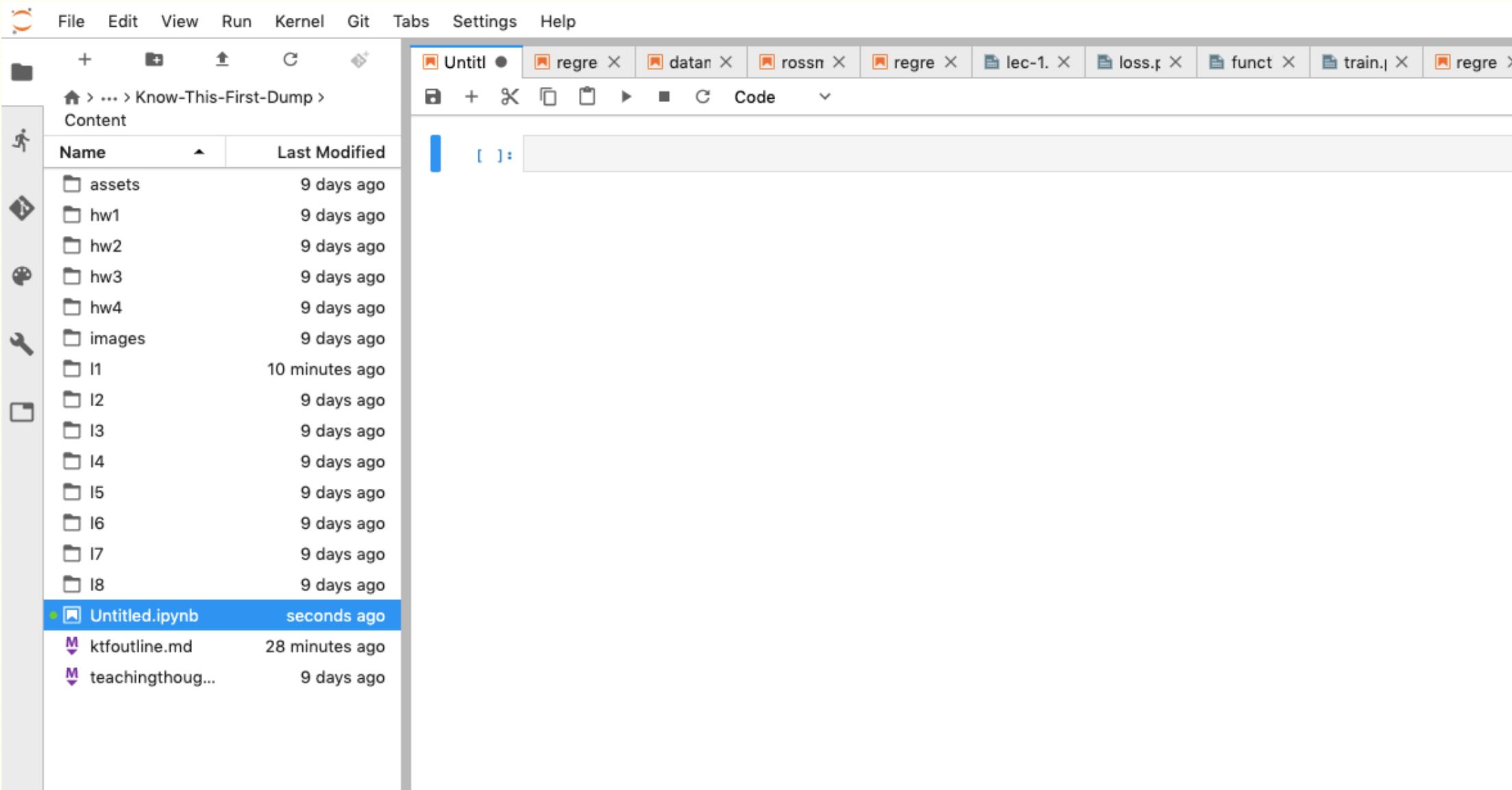
Furthermore, we'll be using Jupyterlab. The next slide tells you how to bring jupyterlab up.

(1) Bring up the "Anaconda Navigator" from your Start Menu, Panel or text-entry point.





This screen is called the **Launcher**. Click on "Python 3". This launches a "kernel" or python process, and connects a new document window, called a **Jupyter Notebook** to this process.



You can now type in text boxes in the Jupyter Notebook, called **cells** in this new window. The left side is a file manager and is likely showing your home folder. This notebook is called Untitled.ipynb.

Colab

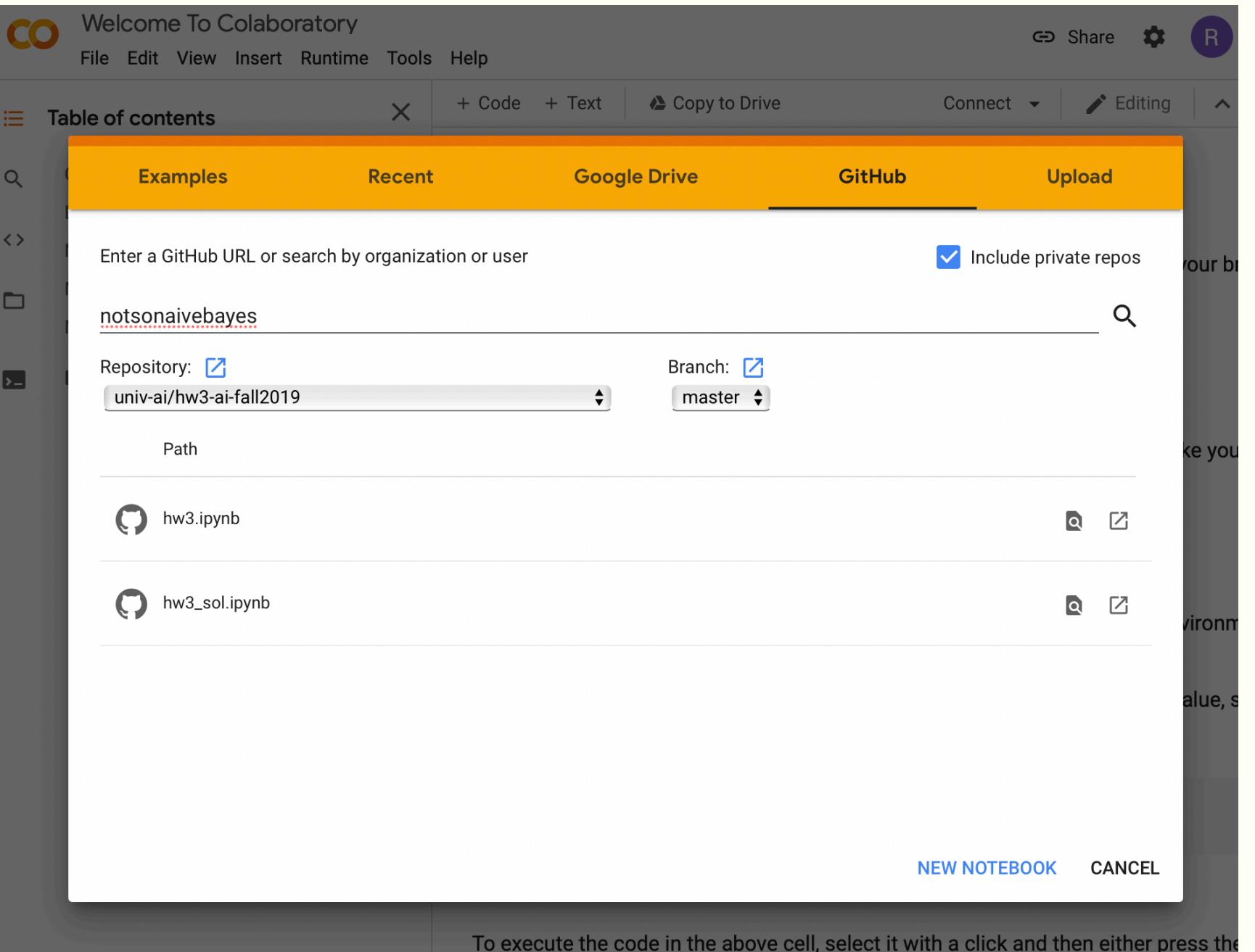
<https://colab.research.google.com> is a service provided by google that allows access to virtual machines with high speed gpu's, good for deep learning.

It is built off Jupyter Notebooks, but getting data in and out of it is a bit of a pain. We have more information on How to use Colab at the [Colab page on welcome.univ.ai](#)

The screenshot shows a Google Colab notebook interface. The title bar reads "Lab1-babypython.ipynb". The menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help. On the left, there's a sidebar with icons for Code, Text, and Copy to Drive. The main area has a section titled "Python and Friends" with the sub-section "This is a very quick run-through of some python syntax". Below this, a code cell contains the following Python code:

```
# The %... is an iPython thing, and is not part of the Python language.  
# In this case we're just telling the plotting library to draw things on  
# the notebook, instead of on a separate window.  
%matplotlib inline  
#this line above prepares IPython notebook for working with matplotlib  
  
# See all the "as ..." contracts? They're just aliasing the package names.  
# That way we can call methods like plt.plot() instead of matplotlib.pyplot.plot().  
  
import numpy as np # imports a fast numerical programming library  
import scipy as sp #imports stats functions, amongst other things  
import matplotlib as mpl # this actually imports matplotlib  
import matplotlib.cm as cm #allows us easy access to colormaps  
import matplotlib.pyplot as plt #sets up plotting under plt  
import pandas as pd #lets us handle data as dataframes  
#sets up pandas table display  
pd.set_option('display.width', 500)  
pd.set_option('display.max_columns', 100)  
pd.set_option('display.notebook_repr_html', True)  
import seaborn as sns #sets up styles and gives us more plotting options
```

Below this, another section titled "The Python Language" is shown with the sub-section "Lets talk about using Python as a calculator...".



Once you click "Open Notebook", you can connect colab to your private fork of our material, and run the notebook in colab. This entails giving google colab permissions to access your github repositories.

But you will have to figure how to get data into colab, and change the paths in the notebook to read data from elsewhere. The easiest (and persistent) way to do this is to upload the data from your machine to google drive. More on this later when we need colab

Logistics Summary

1. Lecture layout, logistics, exercises, and quizzes are in EdStem: <https://edstem.org/us/courses/4024/>
2. Lectures are divided into chapters and the slides and notes (jupyter notebooks) for these are in the platform. <http://course.univ.ai/ml1-c1>
3. The repositories for the slides and the notebooks are on github, and you can fork and download these repositories from there. <https://github.com/univai-ml1-c1>
4. Discussion are on a forum on our discourse: <https://discourse.univ.ai/c/course-machine-learning-and-data-science-1/24>
5. You will be working on your own machine using Anaconda Python, and on Colab.